

Adaptivitätssensitive Platzierung von Replikaten in Adaptiven Content Distribution Networks

Dissertation

zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

Dipl.-Inf. Sven Buchholz
geboren am 01.12.1975 in Frankfurt (Oder)

Gutachter:

Prof. Dr. Alexander Schill
Technische Universität Dresden

Prof. Dr. Klaus Meyer-Wegener
Friedrich-Alexander-Universität Erlangen-Nürnberg

Prof. Dr. Claudia Linnhoff-Popien
Ludwig-Maximilians-Universität München

Datum der Verteidigung:
8. Juli 2005

Dresden im Juli 2005

Die vorliegende Arbeit ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Vollständige oder teilweise Reproduktion bedarf der Zustimmung des Rechteinhabers. Ausgenommen davon ist die Vervielfältigung entsprechend den Bestimmungen des §14, Abs. 3 der Promotionsordnung der Fakultät Informatik der Technischen Universität Dresden in der Fassung vom 8. November 2001.

Viele der in der Arbeit erwähnten Gebrauchsnamen, Handelsnamen, Warenbezeichnungen sind auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen. Abbildungen, die mit freundlicher Genehmigung der jeweiligen Eigentümer aus fremden Quellen übernommen wurden, sind durch die jeweiligen Eigentümer urheberrechtlich geschützt. Die unberechtigte Benutzung ist nicht gestattet.

Kurzfassung

Adaptive Content Distribution Networks (A-CDNs) sind anwendungsübergreifende, verteilte Infrastrukturen, die auf Grundlage verteilter Replikation von Inhalten und Inhaltsadaption eine skalierbare Auslieferung von adaptierbaren multimedialen Inhalten an heterogene Clients ermöglichen. Die Platzierung der Replikate in den Surrogaten eines A-CDN wird durch den Platzierungsmechanismus des A-CDN gesteuert. Anders als in herkömmlichen CDNs, die keine Inhaltsadaption berücksichtigen, muss ein Platzierungsmechanismus in einem A-CDN nicht nur entscheiden, welches Inhaltsobjekt in welchem Surrogat repliziert werden soll, sondern darüber hinaus, in welcher Repräsentation bzw. in welchen Repräsentationen das Inhaltsobjekt zu replizieren ist.

Herkömmliche Platzierungsmechanismen sind nicht in der Lage, verschiedene Repräsentationen eines Inhaltsobjektes zu berücksichtigen. Beim Einsatz herkömmlicher Platzierungsmechanismen in A-CDNs können deshalb entweder nur statisch voradaptierte Repräsentationen oder ausschließlich generische Repräsentationen repliziert werden. Während bei der Replikation von statisch voradaptierten Repräsentationen die Wiederverwendbarkeit der Replikate eingeschränkt ist, führt die Replikation der generischen Repräsentationen zu erhöhten Kosten und Verzögerungen für die dynamische Adaption der Inhalte bei jeder Anfrage. Deshalb werden in der Arbeit adaptivitätssensitive Platzierungsmechanismen zur Platzierung von Replikaten in A-CDNs vorgeschlagen. Durch die Berücksichtigung der Adaptierbarkeit der Inhalte bei der Ermittlung einer Platzierung von Replikaten in den Surrogaten des A-CDNs können adaptivitätssensitive Platzierungsmechanismen sowohl generische und statisch voradaptierte als auch teilweise adaptierte Repräsentationen replizieren. Somit sind sie in der Lage statische und dynamische Inhaltsadaption flexibel miteinander zu kombinieren.

Das Ziel der vorliegenden Arbeit ist zu evaluieren, welche Vorteile sich durch die Berücksichtigung der Inhaltsadaption bei Platzierung von adaptierbaren Inhalten in A-CDNs realisieren lassen. Hierzu wird das Problem der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs als Optimierungsproblem formalisiert, Algorithmen zur Lösung des Optimierungsproblems vorgeschlagen und diese in einem Simulator implementiert. Das zugrunde liegende Simulationsmodell beschreibt ein im Internet verteiltes A-CDN, welches zur Auslieferung von JPEG-Bildern an heterogene mobile und stationäre Clients verwendet wird. Anhand dieses Simulationsmodells wird die Leistungsfähigkeit der adaptivitätssensitiven Platzierungsmechanismen evaluiert und mit der von herkömmlichen Platzierungsmechanismen verglichen.

Die Simulationen zeigen, dass der adaptivitätssensitive Ansatz in Abhängigkeit vom System- und Lastmodell sowie von der Speicherkapazität der Surrogate im A-CDN in vielen Fällen Vorteile gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen mit sich bringt. Wenn sich die Anfragelasten verschiedener Typen von Clients jedoch nur wenig oder gar nicht überlappen oder bei hinreichend großer Speicherkapazität der Surrogate hat der adaptivitätssensitive Ansatz keine signifikanten Vorteile gegenüber dem Einsatz eines herkömmlichen Platzierungsmechanismus.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Skalierbare Inhaltsauslieferung durch Replikation	4
1.1.1	Web-Caching	6
1.1.2	Serverfarmen	7
1.1.3	Spiegelserver	7
1.1.4	Content Distribution Networks	8
1.2	Inhaltsadaption	12
1.3	CDNs für adaptierbare Inhalte	15
1.4	Ziele der Arbeit	20
1.5	Inhaltsübersicht	21
2	Verwandte Forschungsarbeiten	23
2.1	Platzierung von Replikaten in traditionellen Content Distribution Networks . .	23
2.2	Caching adaptierbarer Multimediainhalte	28
2.3	Inhaltsadaption	31
2.3.1	Adaptionsziel	31
2.3.2	Adaptionsgegenstand	32
2.3.3	Zeitpunkt der Adaption	35
2.3.4	Ort der Adaption	35
2.4	Inhaltsadaption in CDNs	42
2.5	Fazit	44
3	Grundzüge und Modellierung von A-CDNs	47
3.1	Basiskonzepte und Realisierung	47
3.1.1	Basiskonzepte	47
3.1.2	Realisierung	49
3.2	Modellierung	57
3.2.1	Systemmodell	57
3.2.2	Lastmodell	68
3.2.3	Problemmodell	68
4	Algorithmen zur Komposition von Adaptionspfaden und zur Platzierung von Replikaten	73
4.1	Komposition optimaler Adaptionspfade	73
4.1.1	Abbildung des Problems	74
4.1.2	Algorithmen zur Ermittlung eines kürzesten Pfades	79
4.1.3	Heuristische Einschränkung des Lösungsraums	85
4.1.4	Optimale Adaptionspfade zur Beantwortung aller Anfragen	86
4.2	Adaptivitätssensitive Platzierungsalgorithmen	88
4.2.1	Komplexität des Platzierungsproblems	88
4.2.2	Erschöpfende Suche zur Bestimmung einer optimalen Lösung	90
4.2.3	Greedy-Heuristik zur Bestimmung einer Näherungslösung	91
4.2.4	Verteilte Realisierung der Greedy-Algorithmen	97

5	Simulationsszenario	101
5.1	Beschreibung der Objektmerkmale	101
5.2	Netzwerktopologie	102
5.3	Qualitätsmodell	106
5.3.1	Räumliche Auflösung	107
5.3.2	Subjektiv wahrgenommene Degeneration des Bildes	109
5.3.3	Antwortzeit	109
5.4	Adaptionoperationen	110
5.5	Objekte und Anfragen	113
5.5.1	Objekteigenschaften	113
5.5.2	Lastmodell	114
5.6	Preissystem	117
5.7	Implementierung	118
6	Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen	121
6.1	Bewertungsmetriken	121
6.2	A-CDN-Ansätze im Vergleich	122
6.3	Güte der Greedy-Heuristik	125
6.4	Robustheit gegenüber ungenauen Lastinformationen	126
6.5	Leistungsfähigkeit der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs im Vergleich zum Einsatz herkömmlicher Platzierungsmechanismen	127
6.6	Untersuchung der Einflussfaktoren auf die Leistungsfähigkeit der Verfahren	132
6.6.1	Überlappung der Anfragemerkmale verschiedener Clients	132
6.6.2	Verteilung der Objektpopularitäten	133
6.6.3	Nominelle Preise	134
6.6.4	Qualitätsmodell	136
6.6.5	Kosten und Verzögerungen infolge der Inhaltsadaption	137
6.6.6	Übertragungskosten und -verzögerungen im Internet	139
6.6.7	Räumliche Auflösung der primären Repräsentationen	140
6.6.8	Anfrageverteilung	142
6.6.9	Heterogenität der Clientpopulation	143
6.6.10	Systemgröße	144
6.6.11	Zahl der betrachteten Objekte	147
6.7	Quellen der Vorteile der adaptivitätssensitiven Platzierung	149
6.8	Adaptivitätssensitive Platzierung von Replikaten in hosting A-CDNs	152
6.9	Zusammenfassung der Ergebnisse	154
7	Abschließende Bemerkungen	157
7.1	Zusammenfassung	157
7.2	Ausblick	160
7.2.1	Modellierung und experimentelle Untersuchungen	160
7.2.2	A-CDNs in der Praxis	162
A	Korrektheit des Dijkstra-Algorithmus zur Ermittlung kürzester Pfade in einem Problemgraphen $\Gamma_{c,k}$	165
B	Monotonie der Nutzwertfunktion	169
C	Empirische Untersuchung und Modellierung der Eigenschaften von JPEG-Bildern im WWW	175
C.1	Analyse der Medieneigenschaften	176
C.2	Transcoding-Analyse	180
D	CPU-Kosten verschiedener Rechenzentren	185

E Verteilung der Anfragen auf die Clients	187
Abkürzungen	191
Symbole	195
Literatur	201

Abbildungsverzeichnis

1.1	Internetnutzer in Europa (nach [Rob02])	1
1.2	Beispiele mobiler Endgeräte	2
1.3	Mobilfunknetze (aus [Buc01])	3
1.4	Client-Server-Kommunikationsmodell	4
1.5	Ebenenmodell: Inhaltsobjekte und Replikate (nach [Len97])	5
1.6	Web-Caching	7
1.7	CDN-Referenzmodell	8
1.8	Adaptionsprozess, schematisch (aus [Spr04])	12
1.9	Ebenenmodell: Inhaltsobjekte und Repräsentationen	13
1.10	Verteilte Adaptionspfade	14
1.11	Inhaltsadaption in A-CDNs mit herkömmlichen Platzierungsmechanismen	15
1.12	Ebenenmodell: Inhaltsobjekte, Repräsentationen und Replikate	16
1.13	Ebenenmodelle für A-CDNs mit herkömmlichen Platzierungsmechanismen	17
1.14	Kombination statischer und dynamischer Inhaltsadaption in A-CDNs	18
1.15	Klassifikation von A-CDNs (relaying A-CDNs)	19
2.1	Systemmodell für Proxy-Caching	28
2.2	Hierarchische Kodierung von Videostreamen	29
2.3	Hierarchisches Caching	30
2.4	Schema zur Klassifikation von Ansätzen zur Inhaltsadaption	31
2.5	Modellierung von Adaptionsoperationen und Adaptionsstrategien	33
2.6	Effiziente Konvertierung von Medienobjekten	34
2.7	Ort der Adaption	36
2.8	Logische und physische Adaptionspfade	37
2.9	Beispiel zur Konstruktion eines logischen Adaptionspfades und einer XML-Typbeschreibung in Ninja [CMI00]	38
2.10	Graphmodell der Pfadsuche bei Active Pipes [CTW01]	39
2.11	Hierarchisches Modell der Pfadsuche bei Active Pipes [KRW01]	40
2.12	Modellierung nach [CSR98]	41
2.13	ICAP-Modell	42
2.14	Zusammenspiel von ADN und A-CDN	43
3.1	Konvertierung und Strukturadaption	48
3.2	Architektur von A-CDNs	49
3.3	Ort der Inhaltsadaption	51
3.4	Abstraktion von A-CDN-Rechenzentren	51
3.5	Anfrageweiterleitung in A-CDNs	54
3.6	Mobilität individueller Clients im formalen Modell	61
4.1	Problemgraph $\Gamma_{c,k}$, Beispiel 1 (keine Servicequalitätsmerkmale berücksichtigt)	75
4.2	Problemgraph $\Gamma_{c,k}$, Beispiel 2 (Servicequalitätsmerkmale berücksichtigt)	76
4.3	Entscheidungsgraph zur Auswahl eines geeigneten Pfadalgorithmus	84
4.4	Prinzip der Greedy-Heuristik	91

4.5	Hierarchischer Nachrichtenaustausch zwischen verteilten Platzierungsmanagern	99
5.1	Zuordnung von Clientknoten zu Zugangspunkten	105
5.2	Qualität von JPEG-Bildern in der Dimension der räumlichen Auflösung	107
5.3	Qualitätsfunktionen in den verschiedenen Qualitätsdimensionen	108
5.4	Adaptionsoperationen	110
5.5	Implementierung des Simulationsmodells	119
6.1	Vergleich der Möglichkeiten zur Bestimmung einer konkreten Repräsentation für konkrete Anfragen (anhand des mittleren Zuwachses an Gewinn pro Anfrage in Eurocent)	124
6.2	Mittlerer Zuwachs an Gewinn pro Anfrage (in Eurocent)	128
6.3	Relative Leistungsfähigkeit	130
6.4	Anteil der Anfragen, die durch Replikate in den Surrogaten des A-CDN beantwortet werden	131
6.5	Korrelation zwischen der Überlappung der Anfragemengen der verschiedenen Clientklassen (Parameter s) und der relativen Leistungsfähigkeit	133
6.6	Korrelation zwischen der Verteilung der Objektpopularitäten (Zipf-Parameter z) und der relativen Leistungsfähigkeit	134
6.7	Relative Leistungsfähigkeit in Abhängigkeit vom Preisniveau p	135
6.8	Abhängigkeit der relativen Leistungsfähigkeit vom Qualitätsmodell	136
6.9	Relative Leistungsfähigkeit bei unterschiedlichen räumlichen Auflösungen der primären Repräsentationen	141
6.10	Relative Leistungsfähigkeit in Abhängigkeit von der Systemgröße	145
6.11	Untergliederung des A-CDNs in Regionen	146
6.12	Relative Leistungsfähigkeit bei verschiedenen Zahlen von Inhaltsobjekten	148
6.13	Grenzwerte der Speicherkapazität $sc(h)$, bei denen der Mittelwert der relativen Leistungsfähigkeit $A_{G/VA/C K/V/C}$ bestimmte Schwellwerte überschreitet	148
6.14	Relative Leistungsfähigkeit $A_{G/VA/C G/A/C}$ für $sc(h) \rightarrow \infty$ in Abhängigkeit von der Zahl der Objekte $ K $	149
6.15	Quellen der Vorteile der G/VA/*-Ansätze gegenüber den K/V/*-Ansätzen	150
6.16	Absoluter Zuwachs an Gewinn pro Anfrage in einem A-CDN der Klasse K/V/* mit und ohne Einschränkung der Menge der unterstützten Repräsentationen (in Eurocent)	151
6.17	Quellen der Vorteile der G/VA/*-Ansätze gegenüber den G/A/*-Ansätzen	152
6.18	Anteil der Anfragen, die durch Replikate in den Surrogaten des hosting A-CDN beantwortet werden	153
6.19	Vergleich der relativen Leistungsfähigkeit adaptivitätssensitiver Platzierung in hosting und relaying A-CDNs	154
A.1	Illustration zum Beweis der Korrektheit des Dijkstra-Algorithmus	166
C.1	Korrelationsanalyse zur Modellierung der räumlichen Auflösung	178
C.2	Empirische Verteilungsfunktion und Modellbildung	178
C.3	Schema der Transcoding-Analyse	181

Tabellenverzeichnis

2.1	Heuristische Näherungsverfahren	25
2.2	Klassifikation der vorgestellten Heuristiken zur Platzierung von Replikaten . . .	26
4.1	Zeitkomplexität des Greedy-Algorithmus	96
5.1	Speicherkapazität pro Surrogat in den Simulationsexperimenten	103
5.2	Eigenschaften der AS-Topologien	103
5.3	Endgeräte und Netzwerkanbindungen	104
5.4	Toleranzschwelle del_{tol} in Abhängigkeit von der Netzwerkanbindung der Clients	109
5.5	Zielparameter der Adaptionoperationen	111
5.6	Ausführungszeiten der Teiloperationen bei der Adaption	112
5.7	Klasseneinteilung der Clients	116
5.8	Verteilung der Anfragen (Abschätzung eines möglichen Anfrageverhaltens) . . .	116
6.1	Güte der Greedy-Heuristik	125
6.2	Robustheit gegenüber ungenauen Lastinformationen	126
6.3	Mittlere untere Grenze der Speicherkapazität $sc(h)$, bei der $\Delta Profit_{\Sigma, \Lambda}$ maximal ist (inkl. 95%-Konfidenzintervall)	129
6.4	Mittlere untere Grenze der Speicherkapazität $sc(h)$, bei der beim alternativen Qualitätsmodell gemäß Formel 6.7 auf Seite 136 $\Delta Profit_{\Sigma, \Lambda}$ maximal ist (inkl. 95%-Konfidenzintervall)	137
6.5	Korrelation zwischen den Ausführungszeiten der Adaptionoperationen (Faktor tf) und der relativen Leistungsfähigkeit	138
6.6	Relative Leistungsfähigkeit für $tf \rightarrow 0$ (inkl. 95%-Konfidenzintervall)	138
6.7	Relative Leistungsfähigkeit in Abhängigkeit von den Übertragungsverzögerungen im Internet (für $sc(h) = 100$ KB, inkl. 95%-Konfidenzintervall)	139
6.8	Relative Leistungsfähigkeit bei alternativen Verteilungen der Anfragen auf die Clients ($\mathcal{V}_1 - \mathcal{V}_4$) und bei reduzierter Heterogenität der Clientpopulation (\mathcal{V}_5) .	143
C.1	Informationen in <code>access.log</code>	175
C.2	Empirische Verteilung des JPEG-Qualitätsfaktors und des Farbmodells	179
C.3	Zielparameter der Adaptionoperationen in der Transcoding-Analyse	180
C.4	Eigenschaften der Testplattform	180
C.5	Ergebnisse der Transcoding-Analyse	182
C.6	JPEG-Kodierungsparameter $p_{1k}, p_{2k}, p_{3k}, p_{4k}, p_{5k}, p_{6k}$ der analysierten Bilder .	183
D.1	CPU-Kosten verschiedener Rechenzentren	185
E.1	Alternative Anfrageverteilung \mathcal{V}_1 (mehr Anfragen aus dem Festnetz)	188
E.2	Alternative Anfrageverteilung \mathcal{V}_2 (mehr Anfragen von WLAN-Clients)	188
E.3	Alternative Anfrageverteilung \mathcal{V}_3 (mehr Anfragen von Clients in zellularen Mobilfunknetzen)	189

E.4	Alternative Anfrageverteilung \mathcal{V}_4 (gleichmäßige Verteilung über die verschiedenen Endgerätetypen)	189
E.5	Alternative Anfrageverteilung \mathcal{V}_5 (eingeschränkte Zahl von Endgerätetypen)	190

Algorithmenverzeichnis

4.1	Dijkstra-Algorithmus	80
4.2	Bellman-Ford-Algorithmus	82
4.3	DAG-SP-Algorithmus	83
4.4	Erschöpfende Suche	90
4.5	Greedy-Algorithmus, Variante 1	94
4.6	Greedy-Algorithmus, Var.2 (Wiederverwendung von Zwischenergebnissen) . . .	95

Kapitel 1

Einleitung

Seit Anfang der 1990er Jahre hat sich das Internet von einem Kommunikationsmedium für eine kleine Zahl von Nutzern, vornehmlich aus dem akademischen Umfeld, zu einer universellen Plattform für Informations- (Nachrichtendienste), Transaktions- (Shopping- und Bankanwendungen) und Unterhaltungsdienste (Video-on-Demand, Online-Spiele) für breite Schichten der Bevölkerung entwickelt. Diese Entwicklung schlägt sich in einer rasant wachsenden Zahl von Internetnutzern nieder (Abb. 1.1). Dabei nimmt das World Wide Web (WWW) eine zentrale Rolle ein. Allerdings wird der klassische WWW-Verkehr über das Hypertext Transfer Protocol (HTTP) [FGM⁺99] zunehmend durch multimediale Angebote, wie Video und Audio, ergänzt, die neue Anforderungen an die Übertragungsverfahren stellen, neue Technologien und Protokolle verwenden und sich durch deutlich höhere Datenvolumina und Datenraten auszeichnen.

Neben der Internetnutzung von Festnetzrechnern in Büros und Privathaushalten kommt dem mobilen Zugriff auf Informationen im Internet, z.B. über WAP, i-mode oder als klassischer WWW-Zugriff (HTTP) von Laptops und PDAs, eine steigende Bedeutung zu. Während zum Beispiel in Japan bereits im Februar 1999 29 Millionen zahlende Kunden über ihr Mobiltelefon den mobilen Informationsdienst i-mode des Anbieters NTT DoCoMo nutzten [oV01b], zeigen Europa und Nordamerika in diesem Bereich noch deutlichen Nachholbedarf. Bis 2006 wird aber auch in Westeuropa ein Anstieg der Nutzerzahlen von i-mode-Diensten auf 31,4 Millionen prognostiziert. Neben der Nutzung von Datendiensten in zellularen Mobilfunknetzen nimmt auch der Internetzugriff über drahtlose lokale Netze (WLANs) an sogenannten WLAN-Hotspots eine zunehmend wichtigere Rolle ein. Laut [Rob02] sollen bis zum Jahr 2006 rund 20 Millionen Europäer in Hotels, auf Flughäfen, in Restaurants, auf Messen oder auch in Einkaufszentren über WLAN-Zugangspunkte auf Internetdienste zugreifen. Der Einfluss der Einführung von Mobilfunknetzen der dritten Generation (UMTS) auf die Zahl von mobilen Internetnutzern ist bisher kaum abzuschätzen. Es scheint jedoch sicher, dass diese Technologien das Potential haben, ganz neuen Nutzergruppen das Internet zu erschließen [Rob02].

Während mobile Internetnutzer in der Vergangenheit vornehmlich textuelle Informationsdienste in Anspruch genommen haben, zeichnet sich zunehmend ein Trend in Richtung multimedialer Unterhaltungsdienste ab (z.B. die Dienste der Produktgruppe „Vodafone live!“ der

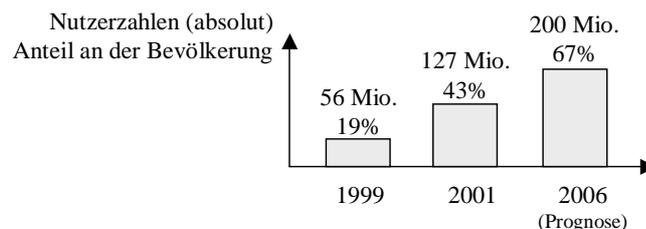


Abbildung 1.1: Internetnutzer in Europa (nach [Rob02])



(Quelle: Samsung Electronics)

(a) Samsung SGH E710



(Quelle: www.nokia.de)

(b) Nokia 9500 Communicator



(Quelle: Hewlett-Packard GmbH)

(c) HP Jornada 720



(Quelle: palmOne)

(d) Palm Tungsten E



(Quelle: IBM)

(e) IBM ThinkPad T Series



(Quelle: Fossil)

(f) Wrist Net FX3005

Abbildung 1.2: Beispiele mobiler Endgeräte

Vodafone D2 GmbH [oV04f]), der sich durch die Einführung neuer Netzwerktechnologien mit höheren Datenraten (z.B. UMTS) noch verstärken wird. Damit werden auch im mobilen Bereich multimediale Inhalte, wie Video und Audio mit deutlich höheren Datenvolumina als Textinformationen, an Bedeutung gewinnen.

Mobile Nutzer greifen mittels einer Vielzahl von verschiedenen Geräten, wie Mobiltelefonen mit WAP-, i-mode- oder WWW-Zugang (sogenannten Smart Phones, Abb. 1.2(a) und 1.2(b)), Personal Digital Assistants (PDAs) in verschiedenen Ausführungen (mit Tastatur oder Handschrift-Eingabe, Abb. 1.2(c) und 1.2(d)) oder Laptop-Computern (Abb. 1.2(e)), auf Informationen im Internet zu. Aufgrund ihrer begrenzten Abmessungen, aber auch bedingt durch die begrenzte Verfügbarkeit von elektrischer Energie unterscheiden sich die Eigenschaften mobiler Geräte teilweise erheblich von denen stationärer Geräte. Sie verfügen oft über kleinere Displays mit eingeschränkter Farbpalette, keine oder nur sehr kleine Tastaturen, Einschränkungen bezüglich der Tonausgabe und begrenzte Rechenkapazität. Letzteres führt unter anderem zu Einschränkungen bezüglich der unterstützten Medien- und Darstellungsformate. Zum Beispiel erfordert das Dekodieren von Videos in Echtzeit hinreichende Rechenkapazität auf dem Endgerät. Mobile Geräte unterscheiden sich in Bezug auf diese Eigenschaften nicht nur von stationären Geräten, sondern auch untereinander. Während die Leistungsfähigkeit von Laptop-Computern denen von Desktop-Computern oft sehr nahe kommt, liegt die von Smart Phones heute noch um Größenordnungen darunter.

Technologisch bedingte Limitationen spielen bei modernen Geräten jedoch immer weniger eine Rolle. Zum Beispiel die Fähigkeiten zur Darstellung von Farben bei Mobiltelefonen sind bei aktuellen Modellen durchaus mit denen von Desktop-Computern vergleichbar. Auch ist die Rechenkapazität von modernen Pocket PC PDAs, obwohl immer noch um eine Größenordnung geringer als bei aktuellen Desktop-Computern, bereits ausreichend, um Videos nach modernen Kompressionsstandards zu dekodieren. Viele Einschränkungen sind aber weniger technologisch

bedingt als durch die Anwendungsszenarien vorgegeben. So ist zum Beispiel nicht zu erwarten, dass die Größe der Displays von Mobiltelefonen auf die von Laptop-Computern oder stationären Rechnern steigt, weil dies die Gebrauchseigenschaften der mobilen Geräte einschränken würde. Ebenso wird eine akustische Ausgabe in vielen Anwendungsfällen nicht akzeptabel sein. Während sich im Hinblick auf verwendete Medien- und Darstellungsformate mit zunehmender Standardisierung eine gewisse Konvergenz zeigt, z.B. unterstützt die aktuelle Generation WAP 2.0 [oV02d] Protokolle und Darstellungsformate, die sich im WWW etabliert haben, wird die Diversifikation in Bezug auf die physischen Eigenschaften der Geräte eher noch zunehmen, wenn mobile Geräte neue Lebensbereiche erobern. Ein Indiz für diesen Trend ist das Erscheinen von Handgelenk-PDAs mit Netzanbindung, wie dem Wrist Net FX3005 des Uhrenherstellers Fossil in Zusammenarbeit mit MSN [oV05] (Abb. 1.2(f)).

Mobile Endgeräte sind über verschiedene Netzwerke, z.B. verschiedene Mobilfunknetze und drahtlose lokale Netze, mit dem Internet verbunden. Diese Netzwerke unterscheiden sich in ihren Übertragungseigenschaften, wie Datenrate, Verzögerung, Zuverlässigkeit, aber auch den Kommunikationskosten erheblich von Festnetzverbindungen und auch untereinander. Durch die Übertragungseigenschaften der Netzwerkanbindung eines Client werden die Dienste, die der Client in Anspruch nehmen kann, teilweise unmittelbar eingeschränkt. Zum Beispiel kann die maximale Datenrate eines Videostroms durch die verfügbare Datenrate des Netzwerkes begrenzt sein. Videostreams mit höherer Datenrate können nicht dargestellt werden. Aber selbst dann, wenn die Nutzbarkeit eines Dienstes nicht unmittelbar eingeschränkt wird, können Antwortzeitverzögerungen aufgrund einer langsamen Netzwerkanbindung die Qualität des Dienstes für den Nutzer inakzeptabel einschränken.

Die Wahl des verwendeten Netzwerks wird sowohl durch die Netzwerkschnittstellen des Endgerätes, als auch durch die Verfügbarkeit verschiedener Mobilfunknetze und drahtloser lokaler Netze an verschiedenen Orten bestimmt. Viele Endgeräte (z.B. Smart Phones) haben heute nur eine Netzwerkschnittstelle für ein Netzwerk. Allerdings zeichnet sich der Trend ab, dass gerade in leistungsfähigere Endgeräte verschiedene Netzwerkschnittstellen (z.B. GSM und WLAN) integriert werden. Auf diese Weise können Nutzer immer zum leistungsfähigsten Netz wechseln, welches verfügbar ist. Die Verfügbarkeit der verschiedenen Netzwerke wird jedoch durch ihre Reichweite bestimmt. Während Mobilfunknetze in Mitteleuropa nahezu flächendeckende Verfügbarkeit erreichen, werden WLAN-Hotspots auf bestimmte lokale Bereiche (z.B. Bahnhöfe, Flughäfen, Restaurants, öffentliche Plätze) beschränkt bleiben. Bei UMTS-Mobilfunknetzen der dritten Generation ist festzustellen, dass sogar innerhalb eines UMTS-Netzes je nach Ort unterschiedliche Übertragungseigenschaften geboten werden. Während in Microzellen Datenraten bis

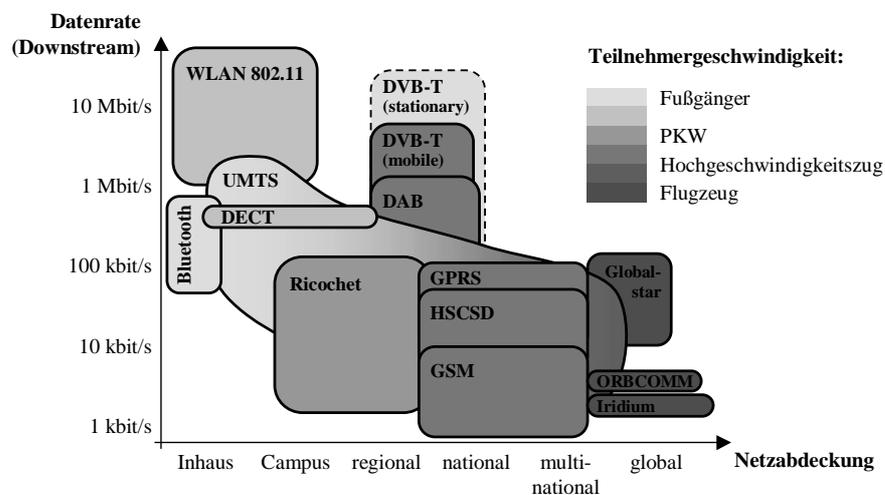


Abbildung 1.3: Mobilfunknetze (aus [Buc01])

zu 2Mbit/s erreicht werden, bleibt die großflächige Abdeckung in städtischen und Vorstadtbereichen auf 384 kbit/s beschränkt. Nutzer in ländlichen Bereichen müssen sich gar mit 144kbit/s begnügen [oV01a]. Da die Übertragungseigenschaften von Mobilfunknetzen grundsätzlich von physikalischen Parametern wie der Ausdehnung des Netzes oder der Geschwindigkeit der Teilnehmer bestimmt werden (vgl. Abb. 1.3), ist zu erwarten, dass die Heterogenität der Netzanbindungen mobiler Endgeräte bestehen bleibt und infolge einer stärkeren Differenzierung der Einsatzbereiche verschiedener Netze eher noch zunimmt.

Mobile Internetnutzung wird auch in Zukunft den Internetzugriff von stationären Geräten mit breitbandiger Netzwerkanbindung nicht ablösen, sondern lediglich ergänzen. Folglich werden in Zukunft mobile und stationäre Clients gleichzeitig auf Inhalte im Internet zugreifen. Inhaltsanbieter, die ihre Inhalte allen Clients zur Verfügung stellen möchten, müssen deshalb in der Lage sein, mit der Heterogenität einer gemischten Clientpopulation aus mobilen und stationären Clients umzugehen.

Um den Anforderungen einer steigenden Zahl mobiler und stationärer Internetnutzer mit stark heterogenen Endgeräten und Netzwerkanbindungen gerecht zu werden, werden in der vorliegenden Arbeit Adaptive Content Distribution Networks (A-CDN) vorgeschlagen. A-CDNs sind anwendungsübergreifende, verteilte Infrastrukturen, die auf Grundlage der verteilten Replikation von Inhalten sowie der Inhaltsadaptation eine skalierbare Auslieferung von adaptierbaren Inhalten an heterogene Clients ermöglichen.

1.1 Skalierbare Inhaltsauslieferung durch Replikation

Das WWW in seiner ursprünglichen Form als einfaches Client-Server-Modell skaliert nicht mit einer steigenden Zahl von Nutzern, die zunehmend auf datenvolumen- und datenratenintensive Inhaltsdienste (z.B. strombasierte Videodienste) zugreifen. Gemäß dem Client-Server-Kommunikationsmodell (Abb. 1.4) fordert ein Client ein Inhaltsobjekt an, indem er [FGM⁺99]

1. eine Verbindung zum Originalserver aufbaut, der das Inhaltsobjekt bereitstellt,
2. über diese Verbindung eine Anfrage nach dem Inhaltsobjekt an den Server sendet und
3. der Server als Antwort das angeforderte Objekt zum Client überträgt.

Bei populären Inhalten, die von einer Vielzahl von global verteilten Clients angefordert werden, führt dieses Verfahren zu einer Reihe von Problemen:

- **Hohe Serverlast.** Der Server muss jede Anfrage jedes Client separat beantworten. Es ist nicht möglich, eine Antwort an alle Clients zu schicken, die das Objekt angefordert haben.
- **Hohe Netzwerklast.** Angeforderte Inhaltsobjekte müssen bei jeder Anfrage vom Originalserver zum Client, ggf. über große Distanzen im Internet, übertragen werden.
- **Hohe Antwortzeiten.** Übertragungszeiten, Netzwerklatenzen und Wartezeiten in Routern führen zu einer Erhöhung der Antwortzeit, insbesondere bei Weitverkehrsverbindungen. Darüber hinaus kann eine Überlastung der Netzwerkverbindungen oder der Server zusätzlich zu einem drastischen Anstieg der Antwortzeit führen.

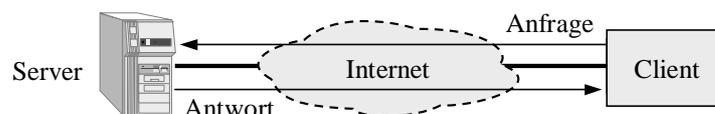


Abbildung 1.4: Client-Server-Kommunikationsmodell

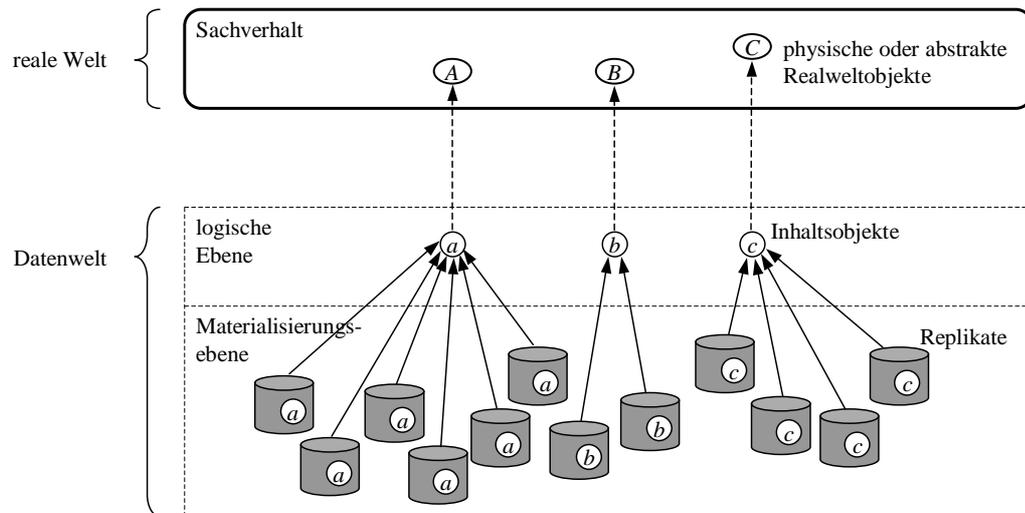


Abbildung 1.5: Ebenenmodell: Inhaltsobjekte und Replikate (nach [Len97])

Diese Probleme treten nicht erst mit einer zunehmenden Zahl mobiler Internetnutzer auf, sondern sie sind bereits seit langem bekannt. Bereits seit Mitte der 1990er Jahre sind verschiedene Verfahren entwickelt worden, diese Probleme zu adressieren: Web-Caching, Serverfarmen, Spiegelserver und Content Distribution Networks (CDNs).

Alle diese Verfahren beruhen auf dem gleichen Grundprinzip, der Replikation von Inhalten. Der Begriff der Replikation wird in der *Internet Web Replication and Caching Taxonomy* der Internet Engineering Task Force (IETF) [CMT01] folgendermaßen definiert:

„Erzeugen und Halten einer duplizierten Kopie einer Datenbank oder eines Dateisystems auf einem anderen Computer, typischerweise einem Server.“

In der Praxis wird Replikation oft mit geringerer Granularität als auf der Ebene ganzer Datenbanken oder Dateisysteme eingesetzt. Rabinovich und Spatschek [RS02] sprechen deshalb allgemeiner vom „Erzeugen und Halten eines Duplikates einer Ressource“. Lenz [Len97] konkretisiert den Begriff der „duplizierten Kopie“ bzw. des „Duplikates“ als „mehrere physische Ausprägungen“, die jeweils durch den „Rechnerknoten [...], an dem sie gespeichert“ sind, „gekennzeichnet“ sind. Da jedoch nicht materialisierte physische Ausprägungen (z.B. als Signal in einem Rechnernetz) im Allgemeinen nicht als Replikat verstanden werden, wird in dieser Arbeit eine präziserte Definition zugrunde gelegt:

Definition 1.1 (Replikation) *Replikation bezeichnet das Erzeugen und Halten mehrerer (nicht-volatil) materialisierter Ausprägungen eines logischen Inhaltsobjektes an verschiedenen Orten.*

Die duplizierten materialisierten Ausprägungen eines Inhaltsobjektes werden als Replikate bezeichnet. Zwischen logischen Inhaltsobjekten und Replikaten besteht eine 1:n-Beziehung. Dieser Zusammenhang wird durch das Ebenenmodell in Abbildung 1.5 verdeutlicht. Logische Inhaltsobjekte repräsentieren jeweils genau ein physisches oder abstraktes Objekt aus dem darzustellenden Sachverhalt der realen Welt. Dabei kann ein abstraktes Realweltobjekt, welches durch genau ein logisches Inhaltsobjekt repräsentiert wird, durchaus mehrere physische Realweltobjekte umfassen, z.B. kann eine Szenerie in einem Bild mehrere physische Objekte (Gegenstände, Personen usw.) einschließen. In diesem Fall ist die dargestellte Szenerie das abstrakte Realweltobjekt im Sinne von Abbildung 1.5. Die Beziehung zwischen der Szenerie und den dargestellten physischen Objekten ist für die Betrachtungen in diese Arbeit unerheblich.

Im Zusammenhang mit der Haltung mehrerer Replikate eines Datenobjektes ist der Begriff der Replikationskonsistenz relevant. Laut [Len97] ist Replikationskonsistenz dann ge-

währleistet, wenn alle Replikate eines Datenobjektes den selben Wert besitzen. Aus lokaler Sicht eines einzelnen Knotens ist Replikationskonsistenz gegeben, wenn die Replikate den aktuellen Wert des jeweiligen logischen Objektes besitzen. Im WWW wird im Allgemeinen von einem Primärkopie-Mechanismus zur Wahrung der Serialisierbarkeit (genauer *One-Copy-Serialisierbarkeit*, vgl. [Len97]) schreibender Zugriffe ausgegangen. Das heißt, alle schreibenden Zugriffe auf ein Inhaltsobjekt erfolgen auf einer ausgezeichneten Kopie, der Primärkopie auf dem Originalserver. Der aktuelle Wert des logischen Inhaltsobjektes wird als äquivalent zum Wert der Primärkopie betrachtet. Somit wird die Replikationskonsistenz durch die Wertäquivalenz der Replikate mit der Primärkopie repräsentiert. In heterogenen Systemen ist eine Definition der Wertäquivalenz als physische Äquivalenz (d.h. byteweise Übereinstimmung) nicht sinnvoll, da sich die verschiedenen Ausprägungen aufgrund ihrer Verteilung auf unterschiedliche Rechnerknoten hinsichtlich ihrer physischen Eigenschaften unterscheiden (z.B. Byteordnung, Kodierung, Darstellungsform). Anstelle dessen bietet sich eine abstrakte Definition der Äquivalenz an. Zwei Replikate eines Inhaltsobjektes werden als abstrakt äquivalent betrachtet, wenn sie die gleichen Änderungen des logischen Inhaltsobjektes reflektieren.

Strikte Replikationskonsistenz bedeutet somit, dass alle Replikate alle Änderungen des logischen Inhaltsobjektes reflektieren. Die Wahrung strikter Konsistenz ist in Systemen mit vielen, weit verteilten Replikaten sehr aufwendig und mit hohem Kommunikationsaufwand und Antwortzeitverzögerungen verbunden. Deshalb werden im WWW oft abgeschwächte Konsistenzmodelle realisiert. Bei abgeschwächter Konsistenz dürfen Replikate innerhalb gewisser Grenzen voneinander abweichen. Die Grenzen können dabei entweder zeit-, wert- oder reihenfolgebasiert definiert sein [SSPS04]. Bei zeitbasierter Konsistenz [TRAR99] muss sichergestellt werden, dass sich eine Änderung zum Zeitpunkt t spätestens zum Zeitpunkt $t + \Delta$ in allen Replikaten widerspiegelt. Aufgrund der Einfachheit und Datenunabhängigkeit zeitbasierter Konsistenzmodelle sind diese im Bereich des WWW sehr verbreitet. Das HTTP-Protokoll [FGM⁺99] bietet eine Unterstützung für zeitbasierte Konsistenzmodelle. Wertbasierte Konsistenzmodelle erfordern, dass sich der Wert zweier Replikate maximal um eine Distanz δ unterscheidet. Wertbasierte Konsistenzmodelle sind nur auf Inhaltsobjekte anwendbar, für die ein Wert und ein Distanzmaß für Werte definiert ist. Diese Datenabhängigkeit wertbasierter Konsistenzmodelle schränkt deren Anwendbarkeit in WWW-Szenarien ein. Reihenfolgebasierte Konsistenz erfordert, dass maximal N Änderungen der Primärkopie erfolgt sein dürfen, die sich nicht in den Replikaten widerspiegeln. Reihenfolgebasierte Konsistenz wurde in [KB94] für replizierte Datenbanken beschrieben. Im Kontext von Inhaltsdiensten im WWW spielen sie jedoch kaum eine Rolle.

Durch Replikation kann Lastverteilung zwischen verschiedenen Rechnerknoten erreicht werden. Die Vermeidung von Überlasten an Serverknoten kann bereits zu einer Reduzierung der Antwortzeiten und damit zu einer Verbesserung der Dienstqualität führen. Sind die Knoten, auf denen repliziert wird, räumlich im Internet verteilt, wie bei Web-Caching, Spiegelservern und CDNs, wird die räumliche Nähe der Inhalte zu den Clients erhöht und damit Netzlast und Antwortzeiten reduziert. Außerdem erhöht sich durch Replikation die Verfügbarkeit, da beim Ausfall eines Knotens auf ein anderes Replikat zurückgegriffen werden kann. Die Verfahren zur Verbesserung der Skalierbarkeit im WWW auf Basis von Replikation sollen im Folgenden kurz vorgestellt werden.

1.1.1 Web-Caching

Rabinovich und Spatschek [RS02] erklären Caching folgendermaßen: „Wenn man irgendeine Information nutzt, und man glaubt, man könnte sie in der nahen Zukunft erneut benötigen, dann speichert man eine Kopie der Information an irgendeinem leicht zugänglichen Ort.“ Für Web-Caching heißt das, dass Inhaltsobjekte, die bereits durch einen Client angefordert wurden, in einem Cache nahe des Client zwischengespeichert werden und für nachfolgende Anfragen die Kopie im Cache ausgeliefert werden kann. Der Zugriff auf den Originalserver entfällt (Abb. 1.6).

Der Cache kann sowohl als individueller Cache für jeden einzelnen Client ausgeführt sein, als auch als Proxy-Cache von einer Menge von Clients gemeinsam genutzt werden. Das Konzept des Proxy-Caches wurde erstmals in [LA94] vorgestellt. Dabei nimmt ein Stellvertreter, der

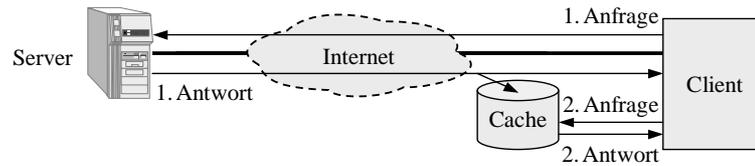


Abbildung 1.6: Web-Caching

den Cache beherbergt, die Anfragen der Clients entgegen und beantwortet sie entweder mit einer Kopie aus dem Cache oder übergibt sie an den Originalserver. Die Antwort wird dann vom Stellvertreter transparent an den Client weitergeleitet und eine Kopie zur Beantwortung späterer Anfragen im Cache gespeichert.

In [BCC⁺95] wird gezeigt, dass bei gemeinsamer Nutzung eines Caches durch die Clients in einem LAN eine größere Reduktion der Antwortzeitverzögerungen erreichbar ist (77% vs. 64%) und dabei der Speicherbedarf des Caches geringer ist als bei individuellen Caches. In [DMF97] und [WVS⁺99] wird argumentiert, dass bei einer Vergrößerung der Clientpopulation eines Proxies über LAN-Grenzen hinweg der Anteil der Anfragen, die durch eine Kopie im Cache beantwortet werden können (Hit-Rate), sogar noch weiter steigt. Laut [WVS⁺99] wächst die Hit-Rate logarithmisch mit der Größe der Clientpopulation. Durch Extrapolation des logarithmischen Zusammenhangs wird gezeigt, dass kein Zugewinn an Hit-Rate möglich ist, wenn mehr als ca. 2,4 Mio. Clients sich einen Cache teilen. Allerdings ist schon bei Populationen ab ca. 20 000 Clients der Zuwachs der Hit-Rate vernachlässigbar klein. Bei Populationen unter 2 500 Clients werden jedoch signifikante Zuwächse erreicht.

Da jedoch ein Proxy-Cache für sehr große Clientpopulationen nicht skalierbar realisiert werden kann und außerdem mit zunehmender Zahl von Clients die mittlere räumliche Nähe des Proxy-Cache zu den Clients abnimmt, wurden verschiedene Verfahren zum kooperativen Caching entwickelt. Dabei kooperieren mehrere lokale Proxy-Caches dergestalt, dass ein Proxy, wenn er ein angefordertes Inhaltsobjekt nicht in seinem Cache vorfindet, zuerst benachbarte Proxies nach dem Objekt fragt, bevor er es vom Originalserver anfordert. Ein prominentes Beispiel für kooperatives Caching stellt hierarchisches Caching dar [CDN⁺96]. Dabei sind die einzelnen kooperierenden Proxies hierarchisch organisiert. Aber auch Verfahren zum verteilten Caching mit gleichberechtigten Partner-Proxies wurden entwickelt. Für einen umfassenden Überblick über Verfahren zum kooperativen Caching sei auf [Wan99] oder [RS02] verwiesen.

1.1.2 Serverfarmen

Eine Serverfarm (vgl. [DCTR03]) besteht aus einer Gruppe von Servern, die an einem Standort betrieben werden und gegenüber den Clients wie ein einzelner besonders leistungsfähiger Server erscheinen. Anfragen der Clients werden durch einen intelligenten Switch (der Pakete auf Basis von Informationen aus der Transport- oder Anwendungsschicht vermittelt) transparent an einen Server der Serverfarm weitergeleitet. Auf diese Weise wird Lastverteilung zwischen den Servern erreicht und die Skalierbarkeit des Dienstes erhöht. Die Inhalte können dabei auf den verschiedenen Servern repliziert vorliegen, so dass parallele Anfragen nach den gleichen Inhalten von verschiedenen Servern beantwortet werden können. Während Serverfarmen gut geeignet sind, die Skalierbarkeit der Originalserver zu erhöhen, können sie bei Überlastsituationen im Netzwerk und netzwerkbedingten Verzögerungen nicht helfen.

1.1.3 Spiegelserver

Spiegelserver (vgl. [RS02]) sind Serverknoten, die den Inhalt eines Originalservers komplett oder teilweise replizieren. Spiegelserver werden durch einen Systemadministrator statisch konfiguriert. Die Auswahl eines Spiegelservers, von dem Inhalte heruntergeladen werden, erfolgt manuell durch den Nutzer, z.B. durch Auswahl aus einer Liste verfügbarer Spiegelserver. Sie ist nicht transparent.

1.1.4 Content Distribution Networks

Definition 1.2 (Content Distribution Network) *Ein Content Distribution Network (CDN) ist eine verteilte Infrastruktur zur effektiven Auslieferung von Inhalten an Clients, bestehend aus einer Menge von Serverknoten, sogenannten Surrogaten, an denen Inhaltsobjekte repliziert gespeichert werden, und weiteren unterstützenden Teilsystemen (nach [DCTR03, RS02, Pen03]).*

Typischerweise wird ein CDN durch einen unabhängigen CDN-Anbieter betrieben, der die Dienstleistung der Replikation und Auslieferung von Inhalten über das CDN an Inhaltsanbieter verkauft. Somit replizieren CDNs üblicherweise Inhalte verschiedener Inhaltsanbieter, die die Kunden des CDN-Anbieters sind. Als CDN-Anbieter treten sowohl große Internet Service Provider (ISPs), die Surrogate in ihrem eigenen Netzwerk platzieren (dementsprechend als *single-ISP* CDNs bezeichnet [RS02]), als auch unabhängige Anbieter auf, die vertragliche Regelungen mit verschiedenen ISPs haben, die es ihnen gestatten, Surrogate in deren Netzwerken zu platzieren (*multi-ISP* CDNs [RS02]). AT&T ist ein Beispiel für einen ISP, der ein single-ISP CDN betreibt. Das CDN von Akamai Technologies ist ein multi-ISP CDN.

Eine Anfrage nach einem Inhaltsobjekt, welches durch ein CDN repliziert wird, wird transparent für den Client an ein Surrogat des CDN weitergeleitet und vom CDN beantwortet. Die Platzierung von Replikaten in den Surrogaten wird je nach aktueller Anfragelast und aktuellem Systemzustand (Verkehrssituation im Internet, Auslastung und Verfügbarkeit der Surrogate) dynamisch angepasst. Ebenso kann die Weiterleitung von Anfragen an Surrogate der aktuellen Lastsituation angepasst werden.

Neben den Surrogaten, auf denen Inhalte repliziert werden, besteht ein CDN aus unterstützenden Teilsystemen, die für die Verteilung und Auslieferung der Inhalte, die Überwachung, Protokollierung und Steuerung der Funktionen des CDN und die Abrechnung gegenüber den Kunden benötigt werden (Abb. 1.7).

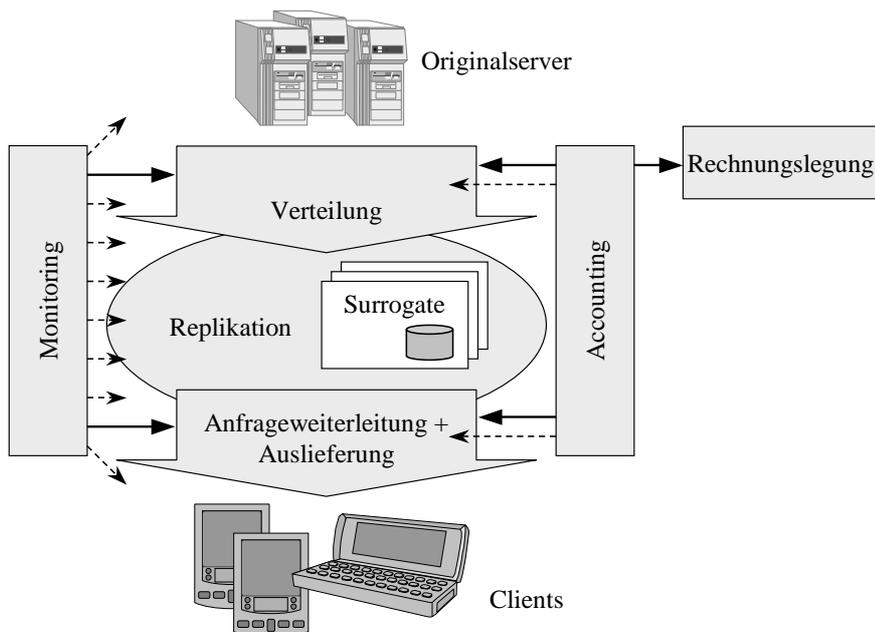


Abbildung 1.7: CDN-Referenzmodell

Definition 1.3 (Verteilung) *Unter Verteilung (Distribution) von Inhalten versteht man die Erzeugung von Replikaten durch Kopieren von Inhaltsobjekten auf die Surrogate (nach [DCTR03]).*

Das verantwortliche Teilsystem für die Verteilung ist das Verteilsystem (*Distribution System*). Die Verteilung kann sowohl im Voraus als auch bei Bedarf erfolgen. Im ersten Fall wird bereits vor der ersten Anfrage ein Replikat erzeugt (*Prefetching*). Bei der Verteilung bei Bedarf werden Inhaltsobjekte erst dann auf ein Surrogat kopiert, wenn dieses eine Anfrage nach dem Objekt erhält. Dieses Verfahren ist dem Caching sehr ähnlich. Allerdings wird in CDNs anders als bei Proxy-Caches die Verteilung der Replikate durch die gezielte Vermittlung von Anfragen gesteuert. Die Verteilung der Inhalte kann sowohl über das Internet als auch über ein privates Netz erfolgen. Verteilung über das Internet wird zum Beispiel von Akamai Technologies genutzt [Pen03]. Hierbei werden die Inhaltsobjekte über ein Overlay-Netzwerk im Internet verteilt. Verteilung über private Netzwerke hat den Vorteil, dass die Verkehrsbedingungen im Verteilsystem durch den CDN-Anbieter besser zu kontrollieren und vorherzusagen sind. CyberStar und Edgix zum Beispiel haben zu diesem Zweck Satelliten-Rundfunk verwendet [Pen03]. Über Satelliten-Rundfunk können Inhalte mit vorhersagbarer, guter Dienstqualität zu geringen Kosten weltweit verteilt werden.

Die Inhalte werden von den Inhaltsanbietern auf Originalservern bereitgestellt.

Definition 1.4 (Originalserver) *Ein Originalserver (Origin Server) ist ein Rechnerknoten, an dem ein Inhaltsobjekt dem Verteilsystem zur Verfügung gestellt wird. Der Originalserver hält die Primärkopie (auch: Authoritative Copy) des Inhaltsobjektes (nach [DCTR03]).*

Je nachdem, ob die Originalserver auch für Anfragen der Clients zur Verfügung stehen, wird zwischen *relaying* und *hosting* CDNs unterschieden [RS02]. In einem relaying CDN werden Anfragen, die nicht durch das CDN befriedigt werden können, an einen Originalserver weitergeleitet und von diesem beantwortet. Akamai's CDN ist ein typisches Beispiel für ein relaying CDN. In einem hosting CDN müssen alle Anfragen von einem Surrogat beantwortet werden. Die Originalserver der Inhaltsanbieter nehmen die Rolle von Staging-Servern wahr, auf denen der Inhaltsanbieter Inhalte und Aktualisierungen bereitstellt, die ausschließlich vom Verteilsystem des CDNs gelesen werden. Digital Island ist ein Beispiel eines CDN-Anbieters, welcher auch den Dienst eines hosting CDN anbietet.

Definition 1.5 (Client) *Ein Client ist die Instanz, die Anfragen nach Inhaltsobjekten sendet und die entsprechenden Antworten empfängt (nach [DCTR03]). Ein Client umfasst den Nutzer, die User-Agent-Software, mittels der der Nutzer eine Anfrage sendet und die Antwort empfängt, und das Endgerät, auf welchem die User-Agent-Software läuft.*

Definition 1.6 (Auslieferung) *Unter der Auslieferung (Delivery) von Inhalten versteht man die Aktivität, einem Client Inhalte in Form einer Antwort auf eine Anfrage zur Verfügung zu stellen (nach [DCTR03]).*

Der Begriff des Auslieferungssystems (*Delivery System*) als Teilsystem, welches für die Auslieferung der Inhalte verantwortlich ist, wird in der Literatur nicht geprägt. Bei herkömmlichen CDNs für WWW-Inhalte beschränkt sich die Funktionalität dieses Teilsystems auf die Übertragung der Inhalte als HTTP-Antwort über eine TCP-Verbindung. Bei fortgeschrittenen CDNs liegt jedoch ein signifikanter Teil der Funktionalität des CDN im Auslieferungssystem, z.B. Echtzeitübertragung (*Streaming*) von Medienströmen oder dynamische Inhaltsadaption. Deshalb ist die Definition des Begriffs des Auslieferungssystems naheliegend. Die Auslieferung von Inhalten ist eng mit der Anfrageweiterleitung verbunden.

Definition 1.7 (Anfrageweiterleitung) *Anfrageweiterleitung (Request Routing) ist die Aktivität, eine Anfrage eines Client an ein geeignetes Surrogat weiterzuleiten, von welchem das angeforderte Inhaltsobjekt an den Client ausgeliefert werden kann (nach [DCTR03]).*

Die Anfrageweiterleitung wird durch das Anfrageweiterleitungssystem (*Request Routing System*) realisiert. Durch die Anfrageweiterleitung wird gesteuert, wie eine Anfrage befriedigt wird. Somit plant die Anfrageweiterleitung eines CDN die Auslieferung der Inhalte. Die Regeln zur Planung der Auslieferung durch das Anfrageweiterleitungssystem werden als Weiterleitungsstrategie bezeichnet [SSPS04]. Neben der Planung der Auslieferung ist die transparente Weiterleitung der Anfragen an ein gemäß der Weiterleitungsstrategie geeignetes Surrogat Aufgabe des Anfrageweiterleitungssystems. Die Verfahren zur Realisierung dieser Aufgabe werden Weiterleitungsmechanismen genannt [SSPS04].

Zur Steuerung seiner Funktion benötigt ein CDN Informationen über die aktuelle Anfragelast und den Systemzustand. Diese werden durch Monitoring und Accounting ermittelt.

Definition 1.8 (Monitoring) *Monitoring ist die Aktivität, fortlaufend den aktuellen Systemzustand zu ermitteln und zur Steuerung der Funktionalität des CDN verfügbar zu machen.*

Monitoring wird durch das Monitoringsystem realisiert. Das Monitoringsystem wird in der Literatur [SSPS04] auch als *Metric Estimation Service* bezeichnet. Neben der Überwachung der ordnungsgemäßen Funktion und der Last der Surrogate ist das Netzwerkmonitoring und die Erstellung eines Modells des Internets eine wichtige Aufgabe des Monitoringsystems. Durch aktive oder passive Messungen oder durch die Auswertung von Routingtabellen werden Informationen über das Internet gesammelt und daraus ein Modell, eine Art Landkarte, des Internets erstellt. Deshalb wird der Vorgang des Monitoring und der Modellierung des Netzwerks auch als *Internet Map Generation* bezeichnet. Aufgrund der großen Zahl von Servern, Surrogaten, Routern und Clients müssen Netzwerkknoten dabei in geeigneter Weise aggregiert werden, um die Komplexität beherrschbar zu machen. Über die Monitoringsysteme in praktisch eingesetzten CDNs ist nur wenig öffentlich bekannt: Akamai's CDN zum Beispiel nutzt Routinginformationen von BGP-Routern und aktive Messungen durch sogenannte Agenten, die Clientanfragen emulieren [DMP⁺02]. Wie aus den einzelnen Routing-Einträgen und Messergebnissen ein umfassendes Modell des Internet mit beherrschbarer Komplexität erzeugt wird, gehört zum Betriebsgeheimnis von Akamai Technologies. In der wissenschaftlichen Literatur wurden ebenfalls Verfahren zum Monitoring und zur Modellierung des Internets vorgestellt, z.B. IDMaps an der University of Michigan [FJJ⁺01], Network Weather Service (NWS) von der UC San Diego [WSJ99] und das Global Network Positioning (GNP) der Carnegie Mellon University [NZ02]. Diese haben jedoch keine nennenswerte praktische Bedeutung erlangt.

Definition 1.9 (Accounting) *Accounting ist die Aktivität der Aufzeichnung aller Aktivitäten zur Verteilung und Auslieferung von Inhalten und Aufbereitung der aufgezeichneten Daten, insbesondere, wenn diese Informationen als Basis für den nachfolgenden Transfer von Geld, Gütern oder Verpflichtungen verwendet werden (nach [DCTR03]).*

Accounting-Informationen werden durch das Accountingsystem gesammelt und aufbereitet. Sie werden für die Rechnungslegung gegenüber den Kunden, die Abrechnung der Ressourcennutzung gegenüber den ISPs, die die Netzwerkanbindung der Surrogate sicherstellen, und gegenüber den Betreibern der Rechenzentren, in welchen die Surrogate betrieben werden, benötigt. Darüber hinaus können sie aber auch dem Verteilsystem, dem Anfrageweiterleitungssystem und dem Auslieferungssystem zur Verfügung gestellt werden, um die Verteilung der Replikat und die Auslieferung der Inhalte an die aktuelle Anfragelast anzupassen.

Auf Grundlage von Monitoring- und Accounting-Informationen kann ein CDN die Platzierung von Replikaten in den Surrogaten koordinieren und dynamisch der aktuellen Situation anpassen.

Definition 1.10 (Platzierung) *Eine Platzierung von Replikaten beschreibt, welche Replikate an welchen Rechnerknoten (nicht-volatil) materialisiert sind.*

Die Aktivität der Ermittlung einer Platzierung gemäß Definition 1.10 wird im Text ebenfalls mit dem Begriff Platzierung bezeichnet.

Definition 1.11 (Platzierungsmechanismen) *Die Verfahren, die die Platzierung der Replikate koordinieren, werden als Platzierungsmechanismen bezeichnet.*

Die Platzierungsmechanismen können sowohl durch das Verteilsystem als auch durch das Anfrageweiterleitungssystem des CDN umgesetzt werden. Ersteres ist mit der Verteilung von Inhalten im Voraus verbunden. Werden Replikate erst bei Bedarf erzeugt, ist es Aufgabe des Anfrageweiterleitungssystems, durch die gezielte Weiterleitung von Anfragen an bestimmte Surrogate die Platzierung der Replikate zu koordinieren.

Definition 1.12 (Platzierungsalgorithmus) *Die Berechnungsvorschrift zur Ermittlung einer Platzierung auf Grundlage von spezifischen Monitoring- und Accounting-Informationen, welche durch die Platzierungsmechanismen verwendet wird, ist der Platzierungsalgorithmus.*

In der wissenschaftlichen Literatur werden sowohl Algorithmen zur statischen als auch zur dynamischen Platzierung von Replikaten beschrieben. Mittels dynamischer Verfahren kann die Platzierung der Replikate kontinuierlich einem sich ändernden Systemzustand angepasst werden. Aufgrund der Systemgröße und der häufigen, fluktuierenden Änderungen relevanter Systemparameter sind dynamische Verfahren jedoch nicht in der Lage, eine globale Sicht auf das System zu berücksichtigen. Anstelle dessen werden einfache heuristische Verfahren auf Grundlage lokaler Systeminformationen verwendet. Diese werden teilweise durch ausgewählte, stark aggregierte globale Systeminformationen ergänzt [RA99].

Im Gegensatz zu dynamischen Verfahren werden in statischen Verfahren auch Algorithmen verwendet, die eine nahezu optimale Platzierung der Replikate ermitteln. Allerdings ist die Komplexität dieser Algorithmen deutlich höher und sie setzen eine globale Sicht auf das System voraus. Deshalb sind sie in der Praxis nur eingeschränkt anwendbar. Um der Dynamik im System gerecht zu werden, müssen statische Platzierungsalgorithmen periodisch eine neue, angepasste Platzierung ermitteln. Während der Intervalle zwischen den Anpassungen bleibt die Platzierung der Replikate stabil. Da sich das System aber während der Intervalle kontinuierlich ändert, die Generierung einer globalen Sicht auf den Systemzustand aufwendig ist und auch die Implementierung einer neuen Platzierung (d.h. die Neuverteilung der Replikate) ihre Zeit benötigt, arbeitet ein CDN, das einen Algorithmus zur statischen Platzierung verwendet, die meiste Zeit mit einer Platzierung, die für einen früheren Systemzustand und eine vorherige Lastsituation optimiert wurde. Trotz dieser Probleme sind statische Verfahren zur Platzierung von Replikaten in der wissenschaftlichen Literatur intensiv untersucht worden. Obwohl ihre praktische Anwendbarkeit fraglich ist, sind sie sehr gut für eine Abschätzung von Schranken der Leistungsfähigkeit von CDNs geeignet.

CDNs haben in den letzten Jahren stark an Bedeutung gewonnen. Während die ersten CDNs im Jahre 1998 auftauchten [VP03], nutzten bis Ende 2000 bereits 31% der populärsten Websites die Dienste eines CDN [KWZ01]. Im Jahr 2003 umfasste allein der Kundenstamm des Marktführers über 3600 Unternehmen [VP03]. Dieser Trend schlägt sich auch in der Geschäftsentwicklung der CDN-Anbieter nieder. Laut [oV02b] lag das Marktvolumen für CDN-Dienstleistungen im Jahr 2001 bei US-\$ 200 Mio., doppelt so hoch wie im Vorjahr. Für die weitere Entwicklung bis 2006 sagt [oV02c] ein Wachstum auf US-\$ 3 Mrd. voraus. Beherrscht wird der CDN-Markt heute durch Akamai Technologies Inc. mit einem Marktanteil von ca. 70% [VP03]. Akamai betreibt über 14 000 Surrogate in 1100 Netzwerken in mehr als 65 Ländern (Stand 2004) [oV04a]. In den vier Geschäftsquartalen bis Ende Juni 2004 hat Akamai im operativen Geschäft einen Umsatz von US-\$ 186 Mio. gemacht [oV04b, oV04c].

Kleinere Anbieter im CDN-Markt, wie AOL, Cable & Wireless, Digital Island, Mirror Image, und Technologieanbieter, wie Inktomi, Network Appliance und Cisco Systems, haben sich im Jahr 2000 zu zwei Allianzen, der *Content Bridge* und der *Content Alliance*, zusammengefunden, die das Ziel hatten, offene Standards zu etablieren, die eine Zusammenarbeit zwischen unabhängig administrierten CDNs ermöglichen [oV00b, oV00a]. Durch die Kooperation sollten kleinere CDN-Betreiber in die Lage versetzt werden, die Reichweite ihrer CDNs (d.h. die Zahl der erreichbaren Clients) zu erhöhen, um damit ihre Marktposition gegenüber den großen Anbietern zu verbessern.

Die Anstrengungen beider Allianzen mündeten in die Aktivitäten der Arbeitsgruppe für Content Distribution Internetworking (CDI) bei der IETF [JM01]. Die CDI-Arbeitsgruppe

stellte ihre Arbeit jedoch Mitte des Jahres 2003 aufgrund mangelnder Unterstützung durch die Mitglieder ein [oV03]. Bis zu diesem Zeitpunkt hatte sie einen Überblick über Szenarien für die Kooperation zwischen CDNs [RDG03], eine Taxonomie bekannter Mechanismen zur Anfragerweiterleitung [BCNS03] und ein abstraktes Modell für die Kooperation von CDNs [DCTR03] publiziert. Die weiteren Ziele der Gruppe wurden nicht erreicht.

1.2 Inhaltsadaption

Um der Heterogenität der Endgeräte und deren Netzwerkanbindungen gerecht zu werden, muss die Darstellung von Inhalten an die Anforderungen der Endgeräte und Netzwerke angepasst werden. Das bedeutet, dass verschiedene Repräsentationen der Objekte generiert werden, die den speziellen Eigenschaften und Fähigkeiten einzelner Geräte und den Übertragungseigenschaften der verwendeten Netzwerke entsprechen. Dieses Vorgehen wird als Inhaltsadaption bezeichnet.

Der Begriff der Inhaltsadaption leitet sich vom Begriff der Adaption ab. Adaption ist laut Langenscheidt Fremdwörterbuch Online [oV04e] folgendermaßen definiert:

„Adaption [...] = Adaptation, [...] 1. Anpassungsvermögen 2. BIOLOGIE, MEDIZIN Anpassung an Umweltbedingungen (Organe) 3. Bearbeitung, Umarbeitung eines Kunstwerks, Neubearbeitung in einer anderen Gattung“

Neben dem Anpassungsvermögen bezeichnet der Adaptionbegriff also den Prozess der Anpassung. Dieses Verständnis von Adaption liegt dem Begriff der Inhaltsadaption zugrunde.

Bei der Betrachtung von Adaption als Prozess der Anpassung sind nach [Spr04] der Adaptiongegenstand und das Adaptionziel zu berücksichtigen (Abb. 1.8). Der Adaptiongegenstand beschreibt, was adaptiert wird. Im Falle der Inhaltsadaption, die in dieser Arbeit betrachtet wird, ist der Adaptiongegenstand die Repräsentation der Inhalte. Darunter wird die Darstellungsform der Inhalte, welche durch Typ- und Medienqualitätsmerkmale beschrieben werden kann, verstanden. Werden durch Inhaltsadaption aus einer Repräsentation eines Inhaltsobjektes verschiedene andere Repräsentationen generiert, so beschreiben alle generierten Repräsentationen das selbe logische Inhaltsobjekt wie die Ausgangsrepräsentation (Abb. 1.9). Eine Ausgangsrepräsentation, aus der jede andere Repräsentation gewonnen werden kann, wird als generische Repräsentation bezeichnet.

Bei der Inhaltsadaption können Informationsverluste auftreten. Die verschiedenen Repräsentationen eines Inhaltsobjektes haben also nicht zwingend alle den gleichen Informationsgehalt, was sich in einer unterschiedlichen Qualität der Repräsentationen widerspiegelt. Um sinnentstellende Informationsverluste zu vermeiden, muss durch das System, welches die Inhaltsadaption ausführt, ein Mindestmaß an Qualität sichergestellt werden. Dazu werden Informationen über die Ausgabebedingungen (*Output Constraints*, vgl. [Hei04]) der Inhaltsobjekte benötigt, die durch die Inhaltsanbieter als Metadaten zur Verfügung gestellt werden müssen.

Das Adaptionziel drückt aus, woran adaptiert wird, das heißt, wodurch die Eigenschaften des Adaptionsergebnisses vorgegeben werden. In der Literatur wird Adaption an verschiedene

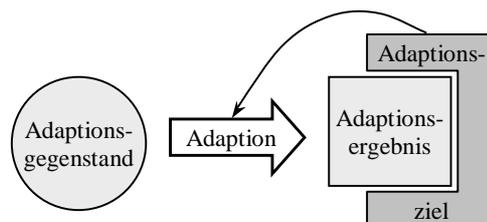


Abbildung 1.8: Adaptionprozess, schematisch (aus [Spr04])

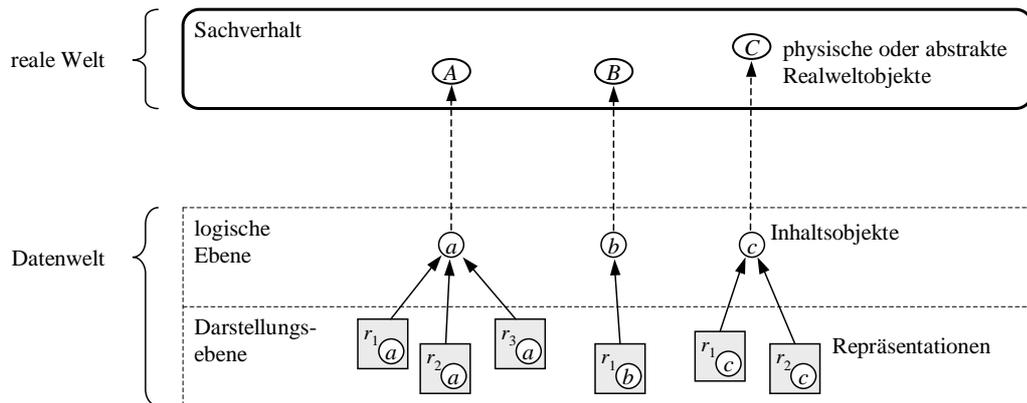


Abbildung 1.9: Ebenenmodell: Inhaltsobjekte und Repräsentationen

Adaptionsziele, wie die Nutzer, die aktuelle Situation und die technische Infrastruktur, beschrieben (vgl. Abs. 2.3). In dieser Arbeit soll schwerpunktmäßig die Adaption an die technische Infrastruktur, das heißt, an die Eigenschaften und Fähigkeiten der Endgeräte und Übertragungseigenschaften der Netzwerke, betrachtet werden, welche beim Internetzugang mit mobilen Geräten von zentraler Bedeutung ist. Adaption an die Nutzer und Adaption an die aktuelle Situation können in diesem Anwendungsszenario aber ebenso relevant sein. Diese Formen der Adaption beschränken sich jedoch oft nicht auf eine Adaption der Repräsentation, wie in dieser Arbeit betrachtet, sondern erfordern eine Anpassung der anwendungsspezifischen Inhalte. Adaption an den Nutzer erfordert darüber hinaus eine aufwendige Nutzermodellierung. Deshalb sind solche Adaptionen weniger für anwendungsübergreifende Infrastrukturen, wie A-CDNs, geeignet. Adaption anwendungsspezifischer Inhalte muss in die Anwendung integriert werden. Sie kann entweder direkt auf dem Originalserver oder aber verteilt durch Nutzung von Application Delivery Networks (ADNs) implementiert werden. Der Einsatz von ADNs für verteilte anwendungsspezifische Adaption ist orthogonal zum Einsatz von A-CDNs zur anwendungsübergreifenden Inhaltsadaption zu betrachten (siehe Abs. 2.4).

Entsprechend obiger Ausführungen wird in dieser Arbeit folgende Definition des Begriffs der Inhaltsadaption zugrunde gelegt:

Definition 1.13 (Inhaltsadaption) *Inhaltsadaption bezeichnet den Vorgang der Anpassung der Repräsentation eines Inhaltsobjektes an die technische Infrastruktur.*

Bei der Adaption von Inhalten an die Eigenschaften mobiler Endgeräte wird durch das Adaptionsziel nicht immer ein konkretes Adaptionsergebnis vorgegeben. Oft gibt es mehrere Repräsentationen eines Inhaltsobjektes, welche den Anforderungen eines Client entsprechen. Zum Beispiel kann ein PDA mit GPRS-Anbindung nur Videoströme mit einer eingeschränkten räumlichen Auflösung und mit einer begrenzten Datenrate empfangen, Videos mit geringerer Auflösung und Datenrate können aber ebenso wiedergegeben werden. Allerdings ist die Qualität der Wiedergabe bei einer geringeren Auflösung oder höheren Kompression schlechter. Es sind also im Allgemeinen unterschiedliche Repräsentationen mit unterschiedlicher Qualität darstellbar. Welche Repräsentation für den Client optimal ist, kann von weiteren Einflussfaktoren abhängen, z.B. von der Verfügbarkeit verschiedener Repräsentationen oder den Kosten und Verzögerungen bei der Adaption. Ein Client sollte deshalb in einer Anfrage nach einem adaptierbaren Inhalt nur das gewünschte logische Inhaltsobjekt spezifizieren, nicht jedoch die konkrete Repräsentation. Zusätzlich stellt er noch eine Beschreibung seiner Eigenschaften und Fähigkeiten zur Verfügung. Die konkrete Repräsentation, die schließlich ausgeliefert wird, wird auf Grundlage dieser Informationen unter Einbeziehung von Kenntnissen über die Verfügbarkeit von verschiedenen Repräsentationen sowie über Kosten und Verzögerungen für die Adaption und Auslieferung der Inhalte durch das System ermittelt. Anfragen dieser Art werden im Fol-

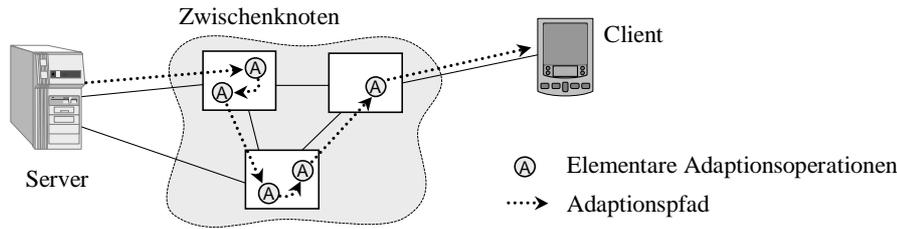


Abbildung 1.10: Verteilte Adaptionspfade

genden als generische Anfragen bezeichnet. Würde hingegen der Client in seiner Anfrage eine konkrete Repräsentation des Inhaltsobjektes spezifizieren, soll dies als konkrete Anfrage bezeichnet werden.

Inhaltsadaption kann sowohl statisch als auch dynamisch ausgeführt werden. Statische Inhaltsadaption bedeutet, dass bereits vor der Anforderung durch einen Client (oft bereits zum Zeitpunkt der Inhaltserstellung) verschiedene Repräsentationen der Inhalte erzeugt werden. Bei einer Anfrage eines Client zur Ausführungszeit wird dann nur noch eine voradaptierte Repräsentation ausgewählt und an den Client ausgeliefert. Bei dynamischer Inhaltsadaption hingegen werden die adaptierten Repräsentationen erst zum Zeitpunkt der Anforderung durch den Client erzeugt. Im Gegensatz zur statischen Adaption bringt der dynamische Ansatz Antwortzeitverzögerungen und Adaptionskosten (insbes. CPU-Last) zur Laufzeit mit sich, da Adaptionsoperationen rechenintensiv sein können. Das gilt insbesondere für sehr populäre Inhalte, auf die von einer großen Zahl von Nutzern konkurrierend zugegriffen wird. Statische Adaption ist hingegen mit einem erhöhten Aufwand für Verwaltung und Wartung der verschiedenen Repräsentationen verbunden. Außerdem besteht ein erhöhter Bedarf an Speicherplatz, um die verschiedenen Repräsentationen der Inhaltsobjekte zu speichern. Aufgrund der dargestellten Probleme der verschiedenen Ansätze kann es sinnvoll sein, beide Ansätze zu kombinieren [LL02]: Ausgewählte, häufig benötigte Repräsentationen werden vorab erzeugt und stehen ohne zusätzliche Adaptionskosten und Verzögerungen bereit. Weitere Repräsentationen, die aufgrund eingeschränkter Speicherkapazität oder eines hohen Aktualisierungsaufwands nicht vorgehalten werden können, werden dynamisch beim Zugriff erzeugt.

Server, Clients oder auch Zwischenknoten kommen als Ausführungsort für Inhaltsadaption in Frage. Inhaltsadaption auf den Clients spielt bei mobilen Internetszenarien jedoch eine untergeordnete Rolle, da sie keine Adaption der angeforderten Inhalte an die Eigenschaften der Netzwerkverbindung erlaubt und darüber hinaus die begrenzten Ressourcen mobiler Geräte die Möglichkeiten zur Adaption auf dem Client einschränken. Bei serverbasierter Adaption muss jeder Server über die Fähigkeiten und Kapazitäten, Inhaltsadaption auszuführen, verfügen. Dafür erlaubt dieser Ansatz eine enge Integration der Adaption und der Anwendung, was anwendungsspezifische Adaptionen ermöglicht. Wird die Inhaltsadaption auf Zwischenknoten (z.B. Proxies [FGCB98]) ausgeführt, können Inhalte verschiedener Server mittels der selben Adaptionsmechanismen angepasst werden. Außerdem wird auf diese Weise die Aufgabe der Inhaltsadaption von der Rolle des Inhaltsanbieters getrennt.

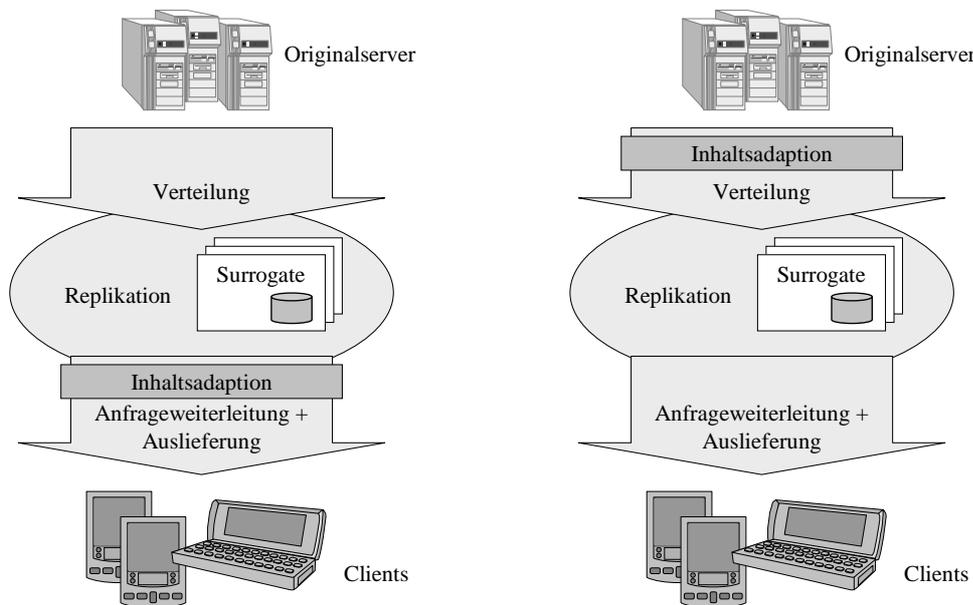
Inhaltsadaption kann auch auf verschiedene Knoten verteilt stattfinden. So können mehrere Zwischenknoten untereinander oder mit dem Server kooperieren, um Adaptionsfunktionalität zu erbringen (Abb. 1.10). Diesem Ansatz liegt die Annahme zugrunde, dass Inhaltsadaption oft mehrere elementare Adaptionsoperationen umfasst, die schrittweise nacheinander oder teilweise auch parallel ausgeführt werden, um komplexe Adaptionen zu ermöglichen. Werden die einzelnen Adaptionsoperationen ausschließlich sequentiell ausgeführt, spricht man von Adaptionspfaden [GWB⁺01, YRP99, CTW01], andernfalls von Adaptionsgraphen [Mar03, Spr04]. Elementare Adaptionsoperationen in einem Adaptionspfad oder -graphen können sowohl verteilt an verschiedenen Knoten (verteilte Adaptionspfade), als auch lokal an einem Knoten ausgeführt werden .

1.3 CDNs für adaptierbare Inhalte

Soll ein CDN in einer Umgebung mit einer heterogenen Clientpopulation eingesetzt werden, in der Inhalte an die Eigenschaften und Fähigkeiten der Endgeräte und Netzwerkanbindungen der Clients adaptiert werden müssen, muss das CDN adaptierte Inhalte ausliefern. Ein CDN, welches adaptierte Inhalte ausliefert, wird in dieser Arbeit als Adaptives Content Distribution Network (A-CDN) bezeichnet.

Inhaltsadaption in A-CDNs kann dynamisch bei der Auslieferung erfolgen, oder es werden bereits statische voradaptierte Repräsentationen der Inhalte in den Surrogaten repliziert. Im ersten Fall ist die Inhaltsadaption eine Funktionalität des Auslieferungssystems des A-CDN (Abb. 1.11(a)). Gegebenenfalls können auch Adaptionproxies von Drittanbietern die Aufgabe der Inhaltsadaption bei der Auslieferung wahrnehmen. Werden Inhalte ausschließlich bei der Auslieferung adaptiert, wird in den Surrogaten des A-CDN jeweils nur eine Repräsentation jedes Objektes repliziert, und zwar die generische Repräsentation des Originalreplikates im Originalserver. Zwischen logischen Inhaltsobjekten und Repräsentationen besteht bezüglich der Replikate eine 1:1-Beziehung (Abb. 1.13(a) auf Seite 17). Somit können logische Ebene und Darstellungsebene aus der Sicht der Platzierungsmechanismen als äquivalent betrachtet werden. Wird nicht zwischen der logischen Ebene und der Darstellungsebene unterschieden, können herkömmliche Platzierungsmechanismen eingesetzt werden, die auf dem Ebenenmodell in Abbildung 1.5 auf Seite 5 basieren. Aufgrund der 1:1-Beziehung zwischen logischen Inhaltsobjekten und Repräsentationen sind herkömmliche Platzierungsmechanismen in der Lage generische Anfragen zu berücksichtigen. Aus Sicht der Platzierungsmechanismen kann eine generische Anfrage als konkrete Anfrage nach der generischen Repräsentation aufgefasst werden.

Der dargestellte Ansatz erlaubt CDNs mit herkömmlichen Platzierungsmechanismen, Inhalte in adaptierter Form an Clients mit heterogenen Eigenschaften und Fähigkeiten auszuliefern. Inhaltsadaption bei der Auslieferung der Inhalte ist jedoch mit den typischen Problemen dynamischer Inhaltsadaption verbunden: Kosten und Verzögerungen infolge der Adaption bei jeder Anfrage.



(a) dynamische Inhaltsadaption bei der Auslieferung (b) statische Inhaltsadaption bei der Verteilung

Abbildung 1.11: Inhaltsadaption in A-CDNs mit herkömmlichen Platzierungsmechanismen

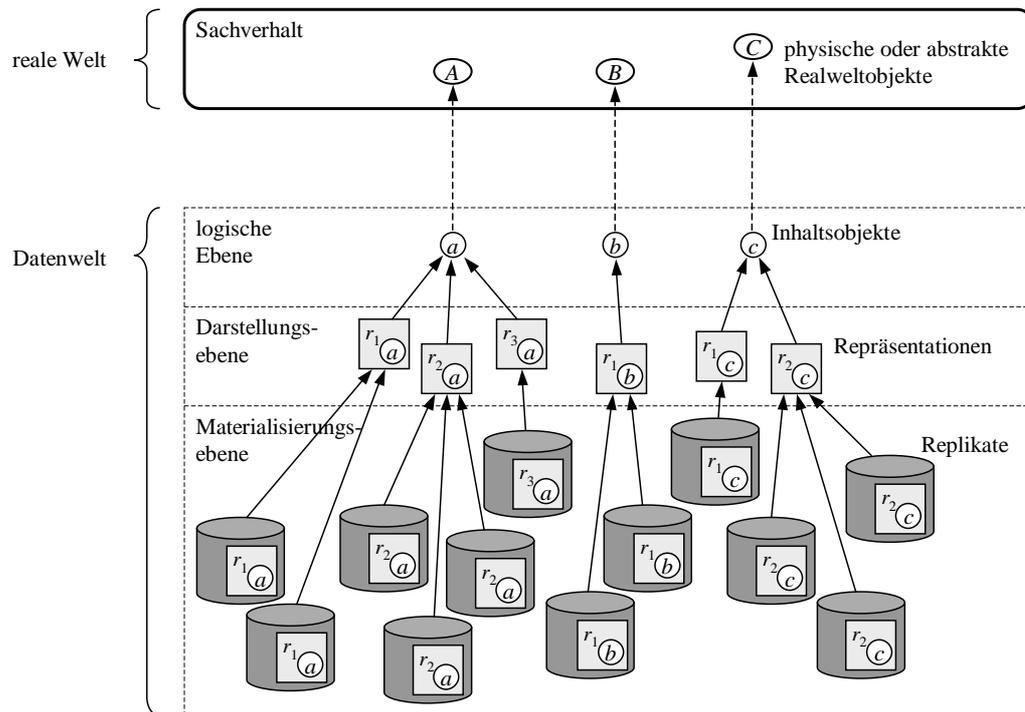
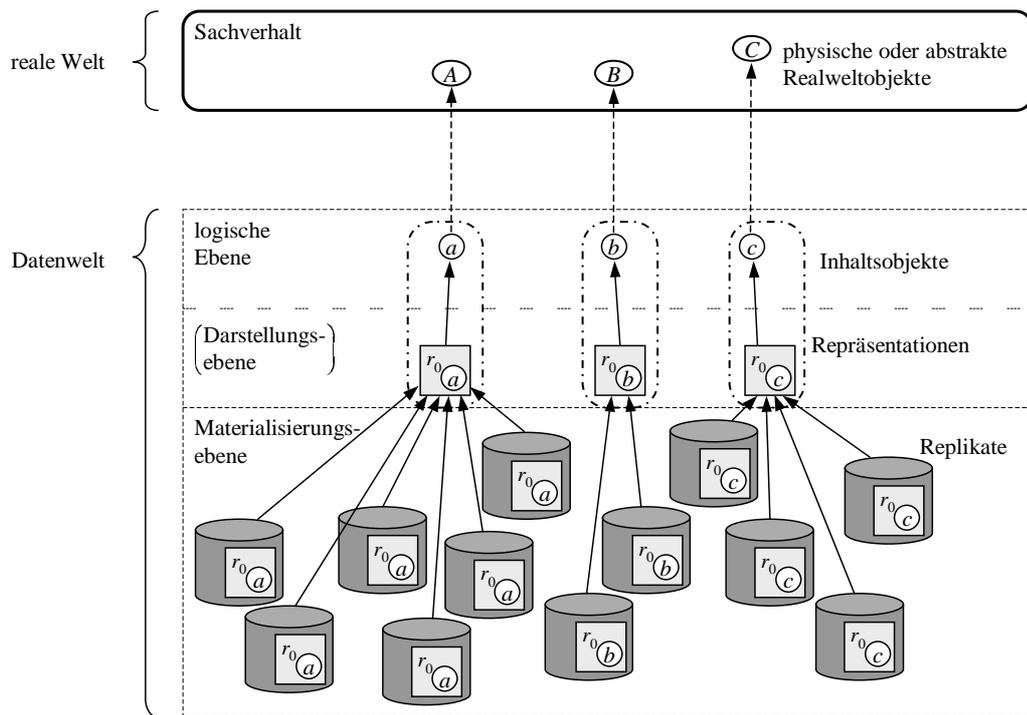


Abbildung 1.12: Ebenenmodell: Inhaltsobjekte, Repräsentationen und Replikate

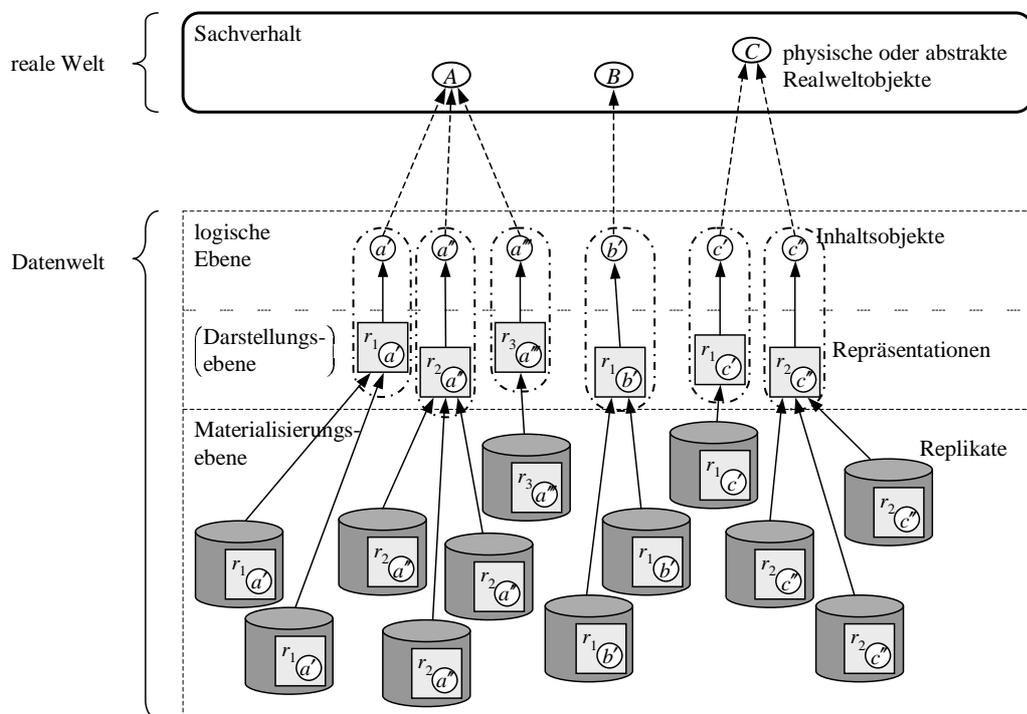
Das Problem der Adaptionkosten und Verzögerungen bei jeder Anfrage lässt sich vermeiden, wenn Inhaltsobjekte bereits vorab statisch adaptiert werden und in adaptierter Form in den Surrogaten des A-CDN repliziert werden. In diesem Fall findet die Adaption bereits bei der Verteilung der Inhalte statt, entweder durch das Verteilsystem des A-CDN (Abb. 1.11(b)), durch die Originalserver oder auch durch Adaptionsproxies eines Drittanbieters. Der Zusammenhang zwischen logischen Inhaltsobjekten, Repräsentationen und Replikaten kann durch das Ebenenmodell in Abbildung 1.12 verdeutlicht werden. Herkömmliche Platzierungsmechanismen basieren auf dem Ebenenmodell in Abbildung 1.5 auf Seite 5. Sie können nicht zwischen verschiedenen Repräsentationen eines Inhaltsobjektes unterscheiden. Soll ein solcher Platzierungsmechanismus in einem A-CDN eingesetzt werden, welches adaptierte Repräsentationen der Inhaltsobjekte repliziert, müssen die verschiedenen Repräsentationen aus der Sicht des Platzierungsmechanismus wie separate Objekte betrachtet werden. Die Sicht des Platzierungsmechanismus entspricht dann dem Ebenenmodell in Abbildung 1.13(b).

Wenn verschiedene Repräsentationen wie separate Objekte behandelt werden, so muss in einer Anfrage eines Client auch explizit eine konkrete Repräsentation spezifiziert werden. Generische Anfragen, wie in Abschnitt 1.2 gefordert, können nicht behandelt werden. Natürlich kann eine generische Anfrage durch das A-CDN in eine konkrete Anfrage überführt werden, indem das A-CDN bestimmt, welche Repräsentation zur Beantwortung der generischen Anfrage geeignet ist. Hierzu müssen aber bereits vor der Bestimmung einer Platzierung Annahmen über die Verfügbarkeit von Replikaten getroffen werden. Treffen diese Annahmen nicht zu, so ist die ermittelte Repräsentation suboptimal.

Neben dem Problem der mangelnden Unterstützung für generische Anfragen ist bei der Replikation statisch voradaptierter Inhalte zu beachten, dass jedes Replikat nur noch Anfragen nach genau einer Repräsentation des Inhaltsobjektes befriedigen kann. Anfragen nach anderen Repräsentationen erfordern weitere Replikate, wodurch der Speicherplatzbedarf in den Surrogaten steigt, oder müssen vom Originalserver beantwortet werden. Die Wiederverwendbarkeit der Replikate für verschiedene Anfragen ist eingeschränkt.



(a) Inhaltsobjekte, Repräsentationen und Replika bei der Replikation generischer Repräsentationen



(b) Inhaltsobjekte, Repräsentationen und Replika bei der Replikation statisch voradaptierter Repräsentationen

Abbildung 1.13: Ebenenmodelle für A-CDNs mit herkömmlichen Platzierungsmechanismen

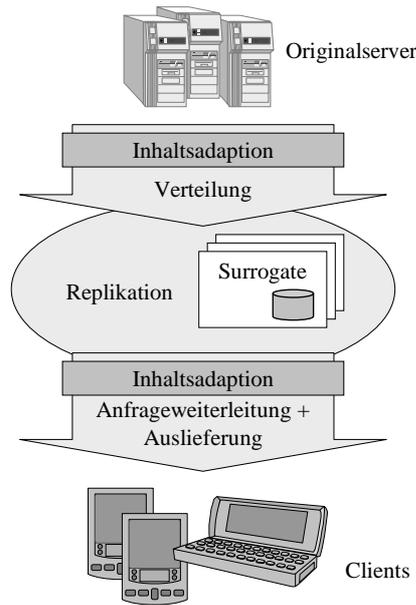


Abbildung 1.14: Kombination statischer und dynamischer Inhaltsadaption in A-CDNs

Dynamische Adaption ist mit dem Problem der Adaptionkosten und Verzögerungen bei jeder Anfrage verbunden. Bei statischer Adaption ist die Wiederverwendbarkeit der Replikate eingeschränkt, was den Speicherplatzbedarf in den Surrogaten erhöht. Deshalb ist es sinnvoll, statische und dynamische Adaption im A-CDN zu kombinieren: ausgewählte, sehr häufig benötigte Repräsentationen werden statisch adaptiert und im A-CDN repliziert, die übrigen Repräsentationen werden bei Bedarf dynamisch generiert. Dazu werden die benötigten generischen Repräsentationen gegebenenfalls¹ in den Surrogaten des A-CDN repliziert. Darüber hinaus können auch aus bereits statisch voradaptierten Repräsentationen bei der Auslieferung dynamisch weitere Repräsentationen erzeugt werden. Dieses Vorgehen basiert auf dem Konzept der Adaptionspfade, bei denen Inhaltsadaption aus mehreren elementaren Adaptionoperationen aufgebaut ist, die schrittweise nacheinander bei der Verteilung und Auslieferung ausgeführt werden, wobei das Zwischenergebnis nach Ausführung der Adaptionoperationen im Teilpfad bei der Verteilung als Replikat materialisiert wird. Inhaltsadaption findet also sowohl bei der Verteilung als auch bei Auslieferung der Inhalte statt (Abb. 1.14).

Platzierungsmechanismen für A-CDNs, die statische und dynamische Adaption flexibel kombinieren können, müssen die Adaptierbarkeit der Inhalte bei der Bestimmung einer Platzierung von Replikaten berücksichtigen. Das bedeutet, sie müssen in der Lage sein, verschiedene Repräsentationen jedes Objektes zu platzieren und dabei die Möglichkeit, die Repräsentationen durch Inhaltsadaption dynamisch ineinander umzuwandeln, in Betracht ziehen. Ihnen liegt das Ebenenmodell in Abbildung 1.12 auf Seite 16 zugrunde. Damit sind sie geeignet, generische Anfragen zu behandeln. Platzierungsmechanismen, die diese Anforderungen erfüllen, sollen als adaptivitätssensitive Platzierungsmechanismen bezeichnet werden.

Während die dynamische Inhaltsadaption für Anfragen, die durch das A-CDN beantwortet werden, durch das Auslieferungssystem des A-CDN (oder ggf. einen proxybasierten Adaptiondienst eines Drittanbieters) erfolgt, kann die Adaption für Anfragen, die durch einen Originalserver beantwortet werden müssen (ein relaying A-CDN vorausgesetzt), auf dem Originalserver, einem Adaptionproxy eines Drittanbieters oder einem Adaptionproxy des A-CDN ausgeführt

¹Die generischen Repräsentationen können auch von den Originalservern geladen werden (es sei denn, es handelt sich um ein *hosting* A-CDN). Durch Replikation werden aber die Zugriffszeiten, Server- und Netzwerklasten reduziert.

werden. Auch die Inhaltsadaption bei der Verteilung der Inhalte kann durch den Originalserver, einen Adaptionproxy eines Drittanbieters oder einen Adaptionproxy des A-CDN erfolgen. Verschiedene Ansätze für die Realisierung von A-CDNs lassen sich dementsprechend danach klassifizieren, an welchen Orten Adaptionmöglichkeiten zur Verfügung stehen. Ein weiteres Klassifikationsmerkmal ist der Zeitpunkt der Inhaltsadaption: bei der Auslieferung, bei der Verteilung oder sowohl als auch. Fernerhin kann die Klassifikation danach erfolgen, ob das A-CDN generische oder konkrete Anfragen beantworten kann. Nach diesen drei Kriterien ergibt sich das in Abbildung 1.15 dargestellte Klassifikationsschema für A-CDNs. Da die Inhaltsadaption bei der Auslieferung verteilte Adaptionmöglichkeiten erfordert, sind ausgewählte Kombinationen von Merkmalsausprägungen in der Praxis nicht umsetzbar (in Abb. 1.15 durch transparente Würfel gekennzeichnet).

Im Schema in Abbildung 1.15 sind die Ansätze, denen das Ebenenmodell in Abbildung 1.13(a) auf Seite 17 zugrunde liegt, hellgrau gekennzeichnet. Die dunkelgrau dargestellten Ansätze basieren auf dem Ebenenmodell in Abbildung 1.13(b) auf Seite 17. Die übrigen Ansätze (in weiß dargestellt) erfordern adaptivitätssensitive Platzierungsmechanismen auf Grundlage des Modells in Abbildung 1.12 auf Seite 16. Diesem Klassifikationsschema folgend, sollen die Ansätze in dieser Arbeit entsprechend der Art der Anfragen ($requests \in \{K, G\}$) sowie des Zeitpunktes ($time \in \{V, A, VA\}$) und des Ortes ($place \in \{O, C, OC\}$) der Inhaltsadaption bezeichnet werden:

$$requests/time/place \quad (1.1)$$

Zum Beispiel bezeichnet G/A/C ein A-CDN, das generische Anfragen berücksichtigen kann (G), bei dem Inhaltsadaption nur bei der Auslieferung der Inhalte stattfindet (A) und zwar durch Adaptionoperationen, die verteilt im A-CDN oder auf Proxies eines Drittanbieters zur Verfügung stehen (C). Durch einen Platzhalter * im Benennungsschema entsprechend Formel 1.1 können Klassen von A-CDN-Ansätzen bezeichnet werden. Zum Beispiel ist */*/OC die Klasse aller Ansätze, bei denen Inhaltsadaption sowohl verteilt im A-CDN als auch auf den Originalservern erfolgen kann.

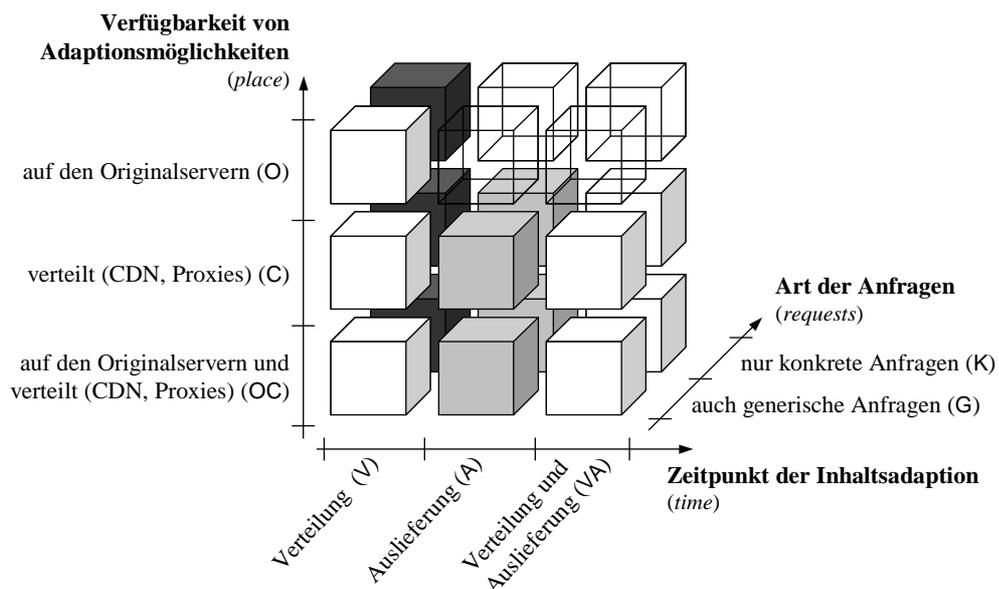


Abbildung 1.15: Klassifikation von A-CDNs (relaying A-CDNs)

1.4 Ziele der Arbeit

Wie in Abschnitt 1.3 dargestellt, haben adaptivitätssensitive Mechanismen zur Platzierung von Replikaten in CDNs den Vorteil, dass

1. unnötige Adaptionskosten und Verzögerungen infolge dynamischer Inhaltsadaption vermieden werden,
2. trotzdem von den Vorteilen dynamischer Inhaltsadaption zur Erhöhung der Wiederverwendbarkeit von Replikaten profitiert werden kann und
3. dabei generische Anfragen der Clients bei der Ermittlung einer Platzierung berücksichtigt werden.

Es ist jedoch unklar, inwieweit diese Vorteile in praktischen Einsatzszenarien relevant sind. Deshalb ist das Ziel der Arbeit, die Überlegenheit des adaptivitätssensitiven Ansatzes in der Praxis zu evaluieren. Dazu werden auf Grundlage eines Modells für ein System zum adaptiven Content Networking geeignete Verfahren zur adaptivitätssensitiven Platzierung von Replikaten entwickelt und in einem Simulator implementiert. Der Simulator bildet die Basis für die empirische Evaluation eines praktisch relevanten Einsatzszenarios. In den Untersuchungen wird ein statisch optimierender Platzierungsalgorithmus zugrunde gelegt, der eine globale Sicht auf das System voraussetzt. Auf diese Weise können nahezu optimale Platzierungen ermittelt werden (optimale Platzierungen sind aufgrund der Komplexität des Problems nicht möglich, vgl. Abs. 4.2.1).

Die praktische Einsetzbarkeit statisch optimierender Platzierungsalgorithmen, die eine globale Sicht auf das System voraussetzen, ist fraglich, da der globale Systemzustand und die Anfragelast in der Praxis nicht im Voraus bekannt sind, sich dynamisch ändern und deshalb nur grob abgeschätzt werden können. In dieser Arbeit werden trotzdem statisch optimierende Platzierungsalgorithmen auf Grundlage globaler System- und Lastinformationen gewählt, da diese erlauben, Möglichkeiten und Grenzen des Konzeptes der adaptivitätssensitiven Platzierung abzuschätzen. Eine Evaluation auf Grundlage grober, aber skalierbarer dynamischer Heuristiken hätte keine interpretierbaren Ergebnisse bezüglich der Vorteile des Konzeptes gebracht. Aus den Ergebnissen wäre nicht in jedem Fall ableitbar gewesen, ob adaptivitätssensitive Platzierung an sich sinnvoll oder nicht ist, oder ob die verwendete Heuristik ungeeignet war, die Vorteile eines adaptivitätssensitiven oder herkömmlichen Platzierungsmechanismus zu realisieren.

In der Arbeit sollen die folgenden Thesen belegt werden:

1. Adaptivitätssensitive Platzierung von Replikaten in A-CDNs ist möglich.
2. Platzierungen von Replikaten in A-CDNs, die durch adaptivitätssensitive Platzierungsmechanismen ermittelt werden, sind in vielen Fällen Platzierungen, die mittels herkömmlicher Platzierungsmechanismen bestimmt werden, überlegen.
3. Der Vorteil gegenüber herkömmlichen Platzierungsmechanismen, bei denen ausschließlich statisch voradaptierte Repräsentationen repliziert werden (Klasse K/V/*), hängt stark vom Anfrageverhalten der Clients und der Speicherkapazität der Surrogate des A-CDN ab. Bei hinreichend großer Speicherkapazität bringt der adaptivitätssensitive Ansatz keine Vorteile mit sich. Ebenso ist der Vorteil vernachlässigbar, wenn Clients unterschiedlicher Clientklassen nicht auf die selben Inhaltsobjekte zugreifen.
4. Aufgrund der Adaptionskosten und der adaptionsbedingten Verzögerungen bei ausschließlich dynamischer Inhaltsadaption sind A-CDNs mit adaptivitätssensitiven Platzierungsmechanismen solchen mit herkömmlichen Platzierungsmechanismen, die nur generische Repräsentationen replizieren (Klasse G/A/*), grundsätzlich überlegen.
5. Adaptivitätssensitive Platzierungsmechanismen in A-CDNs haben in Bezug auf die Leistungsfähigkeit der ermittelten Platzierungen in keinem Fall signifikante Nachteile gegenüber herkömmlichen Platzierungsmechanismen.

1.5 Inhaltsübersicht

Nach der Einführung in die Problematik der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs in diesem Kapitel wird in Kapitel 2 auf verwandte Forschungsarbeiten eingegangen und die vorliegende Arbeit gegenüber diesen Ansätzen abgegrenzt.

Kapitel 3 beschreibt die Basiskonzepte und Überlegungen zur Realisierung von A-CDNs. Darüber hinaus wird ein formales Modell zur Beschreibung des Problems der Platzierung von Replikaten in A-CDNs vorgestellt, welches der Diskussion von Platzierungsalgorithmen in dieser Arbeit zugrunde liegt.

Algorithmen zur adaptivitätssensitiven Platzierung von Replikaten in A-CDNs werden in Kapitel 4 entwickelt. Wie in Kapitel 3 gezeigt wird, ist die Komposition von Adaptionspfaden in A-CDNs ein inhärentes Teilproblem des Platzierungsproblems. Deshalb werden in Kapitel 4 neben den Platzierungsalgorithmen auch Algorithmen zur Komposition von Adaptionspfaden diskutiert.

Das Simulationsmodell zur Bewertung der Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen im Vergleich zum Einsatz herkömmlicher Platzierungsmechanismen in A-CDNs wird in Kapitel 5 vorgestellt.

Die Auswertung der Simulationsexperimente folgt in Kapitel 6. Dabei wird nicht nur die Leistungsfähigkeit der verschiedenen Verfahren zur Platzierung von Replikaten in A-CDNs entsprechend dem Simulationsmodell aus Kapitel 5 bewertet, sondern es werden darüber hinaus in Sensitivitätsanalysen die Einflussfaktoren auf die Leistungsfähigkeit der Platzierungsmechanismen bestimmt.

Kapitel 7 enthält zusammenfassende Bemerkungen und Vorschläge für weiterführende Arbeiten hinsichtlich weitergehender Analysen und der Weiterentwicklung der Ansätze in Richtung der praktischen Einsetzbarkeit von A-CDNs.

Kapitel 2

Verwandte Forschungsarbeiten

Die Platzierung von Replikaten in A-CDNs ist ein neues Forschungsgebiet. Bisher wurden die Vorteile einer global koordinierten, adaptivitätssensitiven Platzierung von Replikaten in CDNs nicht systematisch untersucht. Es gibt jedoch umfangreiche existierende Forschungsarbeiten zur Platzierung von Replikaten in herkömmlichen CDNs, d.h. ohne Berücksichtigung von Inhaltsadaption. Diese werden in Abschnitt 2.1 vorgestellt. Die Idee der Berücksichtigung der Adaptierbarkeit von Multimediainhalten bei der Replikation wurde bereits in vorangegangenen Arbeiten zum Caching von Multimediainhalten umgesetzt. Ein Überblick über Arbeiten dieser Kategorie wird in Abschnitt 2.2 gegeben. In Arbeiten zu diesem Thema wird aber grundsätzlich keine global koordinierte Platzierung von Replikaten untersucht.

Darüber hinaus soll in diesem Kapitel ein Überblick über verwandte Forschungsarbeiten zur Inhaltsadaption gegeben werden (Abschnitt 2.3). Dabei wird insbesondere auf das Konzept von Adaptionspfaden zur Beschreibung von Adaptionsvorgängen eingegangen, welches auch in der vorliegenden Arbeit zur Anwendung kommt. Fragen der Integration von Inhaltsadaption in CDNs, insbesondere in Bezug auf Standardisierungsbestrebungen, werden in Abschnitt 2.4 erörtert.

2.1 Platzierung von Replikaten in traditionellen Content Distribution Networks

Mit der Entwicklung von CDNs wurde das Problem der Platzierung von Replikaten in CDNs Gegenstand von Forschungsarbeiten. Dabei werden Algorithmen zur koordinierten Platzierung von Replikaten in erster Linie in wissenschaftlichen Arbeiten diskutiert. In der Praxis wird, soweit dies aus Veröffentlichungen der Anbieter, wie z.B. Akamai Technologies [LL00, DMP⁺02], ersichtlich ist, vornehmlich Verteilung bei Bedarf berücksichtigt, wobei die Platzierungsmechanismen durch das Anfrageweiterleitungssystem implementiert werden und ausschließlich Lastverteilungskriterien und ggf. regionale Lokalität berücksichtigen. Die aktuelle Platzierung von Replikaten wird nicht herangezogen. Anstelle dessen werden Replikate in jedem Surrogat erzeugt, welches gemäß Lastverteilungskriterien eine Anfrage erhält. Durch Cache-Ersetzungs-Strategien wird sichergestellt, dass die Surrogate hinreichend Speicherkapazität zur Verfügung stellen können.

In der wissenschaftlichen Literatur ist die koordinierte Platzierung von Replikaten viel diskutiert worden. Frühere Arbeiten zu dieser Problematik [LGI⁺99, KPR99, CKS01] basieren auf einer baumförmigen Netzwerktopologie. Der Grund für diese Annahme über die zugrunde liegende Topologie liegt darin, dass diese Arbeiten noch nicht von CDNs, bestehend aus gleichberechtigten Surrogaten, ausgegangen sind. In [LGI⁺99] und [KPR99] wird eine hierarchische Struktur der Surrogate wie beim hierarchischen Caching [CDN⁺96] angenommen. Im Gegensatz dazu geht [CKS01] nicht von einem WWW-Szenario im Internet aus, sondern untersucht das Problem der Platzierung von Replikaten in interaktiven Video-on-Demand-Systemen, in

denen das zugrunde liegende Netzwerk in einer Baumstruktur organisiert ist. Aufgrund der Beschränkung auf eine baumförmige Netzwerktopologie können in diesen Arbeiten Algorithmen angegeben werden, die eine exakte Lösung mit polynomialem Aufwand ermitteln.

Wird die Einschränkung auf eine baumförmige Netzwerktopologie aufgehoben, so ist eine exakte Lösung nicht mehr mit polynomialem Aufwand zu finden. Wie in [KRR01] bewiesen, ist das Platzierungsproblem bei beliebigen Netzwerktopologien NP-schwer. Aus diesem Grund werden in der wissenschaftlichen Literatur verschiedene heuristische Verfahren diskutiert, die nur eine Näherungslösung der optimalen Lösung ermitteln. Eine sehr umfangreiche Klassifikation dieser Heuristiken ist in [KKM02] dargestellt. An dieser Stelle sollen nur ausgewählte Arbeiten exemplarisch genannt werden. Zur Einordnung der darin vorgeschlagenen Heuristiken (vgl. Tab. 2.2 auf Seite 26) wird ein vereinfachtes Klassifikationsschema verwendet. Dabei werden die folgenden Kriterien unterschieden:

- **Parameterbereich.** Das ist die Menge der bei der Entscheidungsfindung berücksichtigten Clients, Surrogate und Objekte. Diese kann sowohl alle Clients, Surrogate und Objekte umfassen oder auf jeweils ein Objekt, ein Surrogat oder eine Teilmenge von Clients beschränkt sein.
- **Zielfunktion.** Welche Kriterien werden in der Zielfunktion berücksichtigt, entsprechend der die Optimierung erfolgt? Die allgemeinste in den Arbeiten verwendete Zielfunktion berücksichtigt die Anfragerate und die Kosten für die Zugriffe auf alle Objekte. Verschiedene Heuristiken beschränken sich aber darauf, nur die Anfragerate oder nur die Zugriffskosten einzubeziehen. Wieder andere Verfahren verwenden gänzlich andere Kriterien, wie Zeitpunkt des letzten Zugriffs oder topologische Informationen.
- **Nebenbedingungen.** Welche Nebenbedingungen schränken mögliche Platzierungen ein? Hier werden Arbeiten vorgestellt, die entweder die Gesamtzahl der Replikat eines Objektes einschränken (P-Bedingung) oder, wie auch die vorliegende Arbeit, die Speicherkapazität der Surrogate als beschränkte Ressource (SC-Bedingung) betrachten.
- **Näherungsverfahren.** Die hier vorgestellten Arbeiten benutzen einfache Ranking-Verfahren, Greedy-Verfahren, Relaxationsverfahren, schwellwert-basierte Verfahren oder Austauschverfahren. Eine Beschreibung der einzelnen Verfahren ist in Tabelle 2.1 gegeben.
- **Statisch vs. dynamisch.** Schließlich lassen sich die in der Literatur beschriebenen Verfahren auch danach unterscheiden, ob sie eine statische Platzierung bestimmen oder ob sie die Platzierung im Betrieb kontinuierlich dynamisch anpassen können.

Die häufig zitierte Arbeit von Qiu et al. [QPV01] untersucht Heuristiken zur statischen Platzierung von Replikaten, die auf einer Modellierung des Problems als p-Zentren-Problem, einem bekannten Problem aus der Graphentheorie, beruhen. Das bedeutet, dass nicht die Speicherkapazität der Surrogate als begrenzte Ressource betrachtet wird, sondern die Zahl der Replikat jedes Objektes beschränkt ist (P-Bedingung). Darüber hinaus werden keine Originalserver berücksichtigt. Alle Anfragen müssen also von einem Surrogat beantwortet werden. Für dieses Szenario werden zwei Heuristiken (Greedy-Algorithmus und *Hotspot*) zur Platzierung von Replikaten beschrieben und deren Leistungsfähigkeit in Simulationsexperimenten mit einer zufälligen Platzierung und einer superoptimalen Näherung der Leistung einer optimalen Platzierung verglichen. Der Greedy-Algorithmus verwendet die nach Anfrageraten gewichteten Zugriffskosten als Zielfunktion. Dabei wird die Zahl der benötigten Netzwerk-Hops als Maß für Zugriffskosten betrachtet. Beim *Hotspot*-Algorithmus handelt es sich um ein Ranking-Verfahren, bei dem in der Zielfunktion nur die Rate der Anfragen von Clients in unmittelbarer Umgebung der Surrogate herangezogen wird. Das heißt, Replikat werden in den Surrogaten repliziert, in deren Umgebung die meisten Anfragen für das entsprechende Objekt generiert werden. Die superoptimale Näherung wird mittels des iterativen Subgradientenverfahrens [HWC74] basierend auf einer Lagrange-Relaxation des Problems bestimmt.

Die Evaluation der Algorithmen zeigt, dass der Greedy-Algorithmus dem *Hotspot*-Algorithmus überlegen ist. Die Gesamtzahl der Netzwerk-Hops beim Greedy-Algorithmus liegt

Tabelle 2.1: Heuristische Näherungsverfahren

Verfahren	Beschreibung
Ranking	Bestimmt für jedes potentielle Replikate (Kombination von Objekt und Surrogat innerhalb des betrachteten Parameterbereichs) den Nutzen entsprechend der betrachteten Zielfunktion und sortiert sie nach ihrem Nutzen. Anschließend werden die potentiellen Replikate nacheinander gemäß der Sortierung zur Materialisierung ausgewählt. Wird dabei eine Nebenbedingung (z.B. Speicherkapazität im Surrogat) verletzt, wird das Replikate übergangen und mit dem nächsten in der sortierten Liste fortgefahren.
Greedy	Das Greedy-Verfahren ist ein iteratives Verfahren. Im ersten Iterationsschritt wird analog zum Ranking-Verfahren für jedes potentielle Replikate dessen Nutzen ermittelt. Anschließend wird das potentielle Replikate mit dem größten Nutzen zur Materialisierung ausgewählt. In den folgenden Iterationsschritten wird nun der Nutzen der verbleibenden potentiellen Replikate neu berechnet. Dabei wird berücksichtigt, dass bereits ausgewählte Replikate platziert wurden. Das wird so lange wiederholt, bis alle Replikate, die keine Nebenbedingung verletzen, platziert sind.
Relaxation	Relaxationsverfahren basieren auf der Idee, Nebenbedingungen, die das Problem verkomplizieren, aufzuheben oder abzuschwächen. Ein wichtiger Vertreter der Relaxationsverfahren in diesem Bereich ist die Lagrange-Relaxation [Geo74]. Hierbei werden verkomplizierende Nebenbedingungen aufgehoben und in abgeschwächter Form in die Zielfunktion eingebettet. Relaxationsverfahren sind im Gegensatz zu den anderen dargestellten Verfahren im Allgemeinen nicht geeignet, eine Platzierung von Replikaten zu bestimmen. Anstelle dessen werden sie verwendet, um eine obere Schranke der Leistung der optimalen Lösung abzuschätzen. Sie haben also eher theoretischen Wert, um die Leistungsfähigkeit anderer Verfahren zu beurteilen. Bei bestimmten Problemdefinitionen können aber auch Lösungen des relaxierten Problems auf realisierbare Platzierungen abgebildet werden [CFL77].
schwelligwert-basiert	Schwelligwert-basierte Verfahren definieren feste Schwelligwerte für den Nutzwert eines Replikates entsprechend der jeweiligen Zielfunktion. Liegt der Nutzwert über diesem Schwelligwert, wird das Replikate platziert, liegt der Nutzwert darunter, wird es entfernt. Schwelligwert-basierte Verfahren sind wegen ihrer Einfachheit gut für den Einsatz zur dynamischen Platzierung geeignet. Dabei müssen allerdings für die Platzierung und die Entfernung eines Replikates verschiedene Schwelligwerte definiert werden, um Oszillation zu vermeiden.
Austausch	Austausch-Verfahren dienen ausschließlich der Verbesserung einer bereits gefundenen Näherungslösung. Diese wird zufällig variiert, indem Replikate verschoben oder ausgetauscht werden, wobei die Einhaltung der Nebenbedingungen sicherzustellen ist. Wenn die neue Platzierung besser ist als die Ausgangslösung, wird sie beibehalten, andernfalls verworfen. Diese Verfahren können auch entsprechend der <i>Simulated-Annealing</i> -Heuristik oder evolutionärer Heuristiken erweitert werden [MF00]. Dabei werden entweder auch schlechtere Lösungen mit einer gewissen Wahrscheinlichkeit als Lösung akzeptiert oder aber mehrere Lösungen behalten, die zu neuen Lösungen kombiniert werden können. Beide Erweiterungen haben das Ziel, zu verhindern, dass sich das Verfahren in lokalen Optima verfängt.

Tabelle 2.2: Klassifikation der vorgestellten Heuristiken zur Platzierung von Replikaten

Heuristik	Parameterbereich			Zielfunktion	Nebenbeding.	Näherungsverfahren	statisch vs. dynamisch
	Client	Surr.	Obj.				
Greedy [QPV01]	alle	alle	ein	Anfragerate, Zugriffskosten	P	Greedy	statisch
Hotspot [QPV01]	lokal	alle	ein	Anfragerate	P	Ranking	statisch
superoptimale Näherung [QPV01]	alle	alle	ein	Anfragerate, Kosten	P	Relaxation	statisch
Popularität [KRR01]	lokal	ein	lokal	Anfragerate	SC	Ranking	statisch
Greedy-Sinlge [KRR01]	lokal	ein	lokal	Anfragerate, Zugriffskosten	SC	Ranking	statisch
Greedy-Global [KRR01]	alle	alle	alle	Anfragerate, Zugriffskosten	SC	Ranking	statisch
Fanout [RGE01]	kein	alle	kein	Topologie	P	Ranking	statisch
RaDaR [RA99]	lokal	alle	lokal	Anfragerate, Topologie	–	schwelligwertbasiert	dynamisch
LRU-Caching [KM02]	lokal	ein	lokal	Zeitpunkt des letzten Zugriffs	SC	Ranking	dynamisch
verzögertes Caching [KM02]	lokal	ein	lokal	Zeitpunkt des letzten Zugriffs	SC	Ranking	statisch

im Bereich zwischen 1,1 und 1,5 der superoptimalen Lösung, während sie beim *Hotspot*-Algorithmus zwischen 1,6 und 2,0 liegt. Beide Algorithmen produzieren aber deutlich bessere Ergebnisse als eine zufällige Platzierung. Außerdem wird gezeigt, dass beide Heuristiken nicht sehr empfindlich gegenüber ungenauen Eingabedaten sind. Sie sind damit auch dann einsetzbar, wenn in der Praxis die Anfragerlast und der Systemzustand nur grob abgeschätzt werden können.

Im Gegensatz zu [QPV01] betrachten Kangasharju et al. [KRR01] ein Szenario, welches dem dieser Arbeit zugrunde liegenden ähnelt. Die Arbeit geht von einem relaying CDN aus, wo Originalkopien der Inhalte auf Originalservern zur Verfügung stehen und unabhängig von der Platzierung der Replikate von diesen bezogen werden können. Als Nebenbedingung wird die Speicherkapazität der Surrogate als begrenzte Ressource betrachtet (SC-Bedingung), nicht die Zahl der Replikate jedes Objektes. In [KRR01] werden drei Heuristiken zur statischen Platzierung der Replikate vorgeschlagen: (1) ein lokales Ranking nach der Anfragerate (in [KRR01] als Popularitätsheuristik bezeichnet), (2) lokales Ranking entsprechend der nach Anfragerate gewichteten Zugriffskosten (*Greedy-Single*) und (3) ein Greedy-Algorithmus, der Anfragerate und Zugriffskosten in der Zielfunktion berücksichtigt (*Greedy-Global*). In Simulationsexperimenten wird gezeigt, dass die global koordinierte Platzierung von Replikaten mittels der *Greedy-Global*-Heuristik den beiden anderen Verfahren überlegen ist. Je nach Anfragerlast und Speicherkapazität der Surrogate wird ein Vorteil von bis zu 24%, gemessen an der Zahl der Netzwerk-Hops, nachgewiesen. Das Ergebnis zeigt, dass eine global koordinierte Platzierungsstrategie deutliche Vorteile bezüglich der Leistungsfähigkeit gegenüber einer ausschließlich lokal koordinierten mit sich bringen kann.

Ein gänzlich anderer Ansatz zur statischen Platzierung von Replikaten wird in [RGE01] verfolgt. Hierbei basiert die Platzierung von Replikaten ausschließlich auf der Topologie des

Netzwerkes. Die Anfragelast wird nicht zur Ermittlung einer Platzierung herangezogen. Als einschränkende Nebenbedingung wird die Zahl der Replikate betrachtet (P-Bedingung). Bei den vorgeschlagenen Heuristiken handelt es sich um Ranking-Heuristiken, die den Nutzen der Platzierung an einem bestimmten Knoten auf Grundlage des Knoten-Grades¹ (*Fanout*) bestimmen. Dabei wird der Knoten-Grad auf zwei unterschiedlichen Ebenen betrachtet: (1) auf der Ebene der Router (wobei implizit angenommen wird, dass Router auch Surrogat-Funktionalität wahrnehmen) und (2) auf der Ebene autonomer Systeme (AS-Topologie). Auch im letzteren Fall wird angenommen, dass Router als Surrogate fungieren. Dabei werden Replikate in den ausgewählten autonomen Systemen jeweils auf dem Router platziert, der in der Router-Topologie den höchsten Knoten-Grad aufweist. Zur Bewertung dieses Vorgehens werden die beiden Verfahren jeweils mit dem Greedy-Verfahren aus [QPV01] verglichen. Es wird gezeigt, dass die beiden topologiebasierten Verfahren vergleichbar gute Ergebnisse hervorbringen, welche unter den getroffenen Annahmen in den meisten Fällen in einer Größenordnung von Faktor 1,1 bis 1,2 schlechter sind als die der Greedy-Lösung.

Als Beispiel für dynamische Verfahren soll hier die Arbeit von Rabinovich et al. [RA99] vorgestellt werden. In dieser Arbeit wird die schwellwert-basierte *RaDaR*-Heuristik zur Verteilung der Replikate vorgestellt. Dabei werden zwei Ziele verfolgt: Verteilung der Anfragelast und Verbesserung der räumlichen Nähe von Replikaten zu Clients. Zur Verbesserung der räumlichen Nähe von Replikaten und Clients werden Replikate dynamisch zu anderen Surrogaten migriert, wenn sich diese näher an der Mehrzahl der Clients befinden. Die Auswahl eines geeigneten Replikates erfolgt dabei durch Inspektion der Routing-Pfade, auf denen Anfragen von den Clients gesendet werden. Übersteigt die Anfragerate an einem Surrogat einen bestimmten Schwellwert, so wird ein zusätzliches Replikat erzeugt, und zwar in dem am weitesten entfernten Surrogat, in dessen Umgebung eine Mindestanzahl von Anfragen generiert wird. Auf diese Weise wird implizit die räumliche Nähe von Replikaten und Clients verbessert. Fällt die Anfragerate für ein bestimmtes Replikat unter einen unteren Schwellwert, wird das Replikat gelöscht. Zur Überwachung der Last der einzelnen Surrogate und zur Koordination der Replikation wird ein systemweit verteilter Dienst aus hierarchisch organisierten Managementknoten, sog. Replikatoren, verwendet. In Simulationsexperimenten wird nachgewiesen, dass das dargestellte Verfahren die Netzwerklast gegenüber einem Szenario ohne Replikation um bis zu 52% reduziert. Durch die Nutzung des dynamischen schwellwert-basierten Verfahrens wird dieses Ergebnis bei guter globaler Skalierbarkeit erreicht. Ein quantitativer Vergleich mit statischen Verfahren ist in [RA99] nicht angegeben. In [KRR01] konnte für global koordinierte statische Verfahren jedoch eine Reduktion der Netzwerk-Hops um bis zu knapp 90% nachgewiesen werden. Aufgrund der unterschiedlichen Testumgebungen sind diese Ergebnisse aber nicht unmittelbar vergleichbar.

Der Mangel an Vergleichbarkeit der experimentellen Ergebnisse der verschiedenen Studien zu Platzierungsalgorithmen motivierte die Arbeit von Karlsson et al. [KM02, KKM02] an den HP Laboratories. Dabei werden verschiedene Heuristiken zur statischen Platzierung, welche in der Literatur vorgeschlagen wurden, unter Berücksichtigung der P- und der SC-Bedingung miteinander verglichen. Darüber hinaus wird LRU-Caching [PB03] als Beispiel für dynamische Platzierung in den Vergleich (für SC-Bedingung) einbezogen. Als Vergleichsmetrik wird dabei die Antwortzeit herangezogen. In Simulationsexperimenten wird gezeigt, dass für die P-Bedingung ein Greedy-Algorithmus die besten Ergebnisse erzielt. Auch durch die nachträgliche Optimierung der Greedy-Lösung mittels Austausch-Verfahren lassen sich keine nennenswerten Verbesserungen erzielen. Bei Berücksichtigung der SC-Bedingung erzeugt ein global optimierender Greedy-Algorithmus ebenfalls sehr gute Ergebnisse. Jedoch ermittelt [KM02] für eine lokal optimierende Popularitätsheuristik geringfügig niedrigere Antwortzeiten. Dieses Ergebnis steht jedoch im Widerspruch zu [KRR01], wo ein Vergleich der selben Heuristiken eindeutig einen Vorteil der Greedy-Heuristik nachgewiesen hat. Im Vergleich der statisch optimierenden Verfahren mit LRU-Caching zeigt sich eindeutig ein Vorteil der statischen Verfahren. Dies wird insbesondere darauf zurückgeführt, dass bei LRU-Caching Anfragen, die nicht aus dem lokalen Cache beantwortet werden können, an den Originalserver geschickt werden und

¹Der Knoten-Grad eines Netzwerkknotens gibt an, mit wie vielen anderen Knoten der Knoten durch einen direkten Netzwerkklink verbunden ist.

niemals an ein anderes, nahe gelegenes Surrogat. Wäre es hingegen möglich, Informationen über die Platzierung der Replikate zeitnah allen Surrogaten zur Verfügung zu stellen, so dass diese Anfragen an geeignete Surrogate weiterleiten können (optimales kooperatives Caching), dann wäre die Leistungsfähigkeit eines solchen Verfahrens den besten statischen Verfahren sogar geringfügig überlegen. Das scheitert jedoch an der mangelnden Skalierbarkeit eines solchen Verfahrens. Deshalb schlägt [KM02] vor, ein LRU-Verfahren zur statischen Platzierung einzusetzen, das heißt, die Platzierung nicht bei jeder Anfrage, sondern nur periodisch in festgelegten Intervallen anzupassen (*verzögertes Caching*).

Nicht unerwähnt soll die Verwandtschaft des Problems der Platzierung von Replikaten in CDNs zum Allokationsproblem in verteilten replizierten Datenbanken [ÖV99a, Len97] bleiben. Im Gegensatz zu den Arbeiten zur Platzierung von Replikaten in CDNs werden beim Allokationsproblem in verteilten replizierten Datenbanken grundsätzlich auch schreibende Zugriffe auf replizierte Daten mitberücksichtigt, was die Formulierung der Zielfunktion verkompliziert. Aus diesem Grund sind die Arbeiten zu dieser Problematik nur im beschränkten Maße in Forschungen zu CDNs eingeflossen. In der Klassifikation in [KKM02] werden aber auch solche Ansätze mitberücksichtigt.

2.2 Caching adaptierbarer Multimediainhalte

Die Erkenntnis, dass die Effektivität von Replikationstechniken durch Ausnutzung der Adaptierbarkeit von Multimediainhalten gesteigert werden kann, hat bereits vorangegangene Forschungsarbeiten motiviert. Die erste Arbeit dieser Art stellt das Soft-Caching-Projekt der University of Southern California in Zusammenarbeit mit der EPFL in Lausanne [OCAV97, KKO98] dar. In dieser Arbeit wird die Replikation innerhalb eines Proxy-Caches entsprechend des Systemmodells in Abbildung 2.1 betrachtet. Im Gegensatz zum herkömmlichen Caching-Ansatz, bei dem Objekte vollständig aus dem Cache verdrängt werden müssen, um Speicherplatz freizugeben, werden beim Soft-Caching-Ansatz Bilder nicht verdrängt, sondern dergestalt adaptiert, dass sie weniger Speicherplatz im Cache belegen. Konkret werden in den Untersuchungen JPEG-Bilder betrachtet, die durch erneute Kompression mit geringerem Qualitätsfaktor in eine kompaktere Repräsentation umgewandelt werden. Simulationen konnten eine beträchtliche Reduktion der Antwortzeiten durch den Einsatz dieser Technik belegen. Je nach Kapazität des Cache-Speichers reduzierten sich diese bis zu Faktor 1:2,84. Allerdings wird bei dieser Art der Bewertung, im Gegensatz zu den Betrachtungen in der vorliegenden Arbeit, der Qualitätsverlust infolge der Adaption nicht berücksichtigt.

Die Idee des Soft-Caching-Ansatzes wurde später erweitert und auf qualitäts-adaptives Caching von hierarchisch kodierten (*layered-encoded*) Videoströmen angewendet [KHRR01, RYHE00, PB02, SC02]. Hierarchisch kodierte Videoströme sind in mehreren separaten Strömen

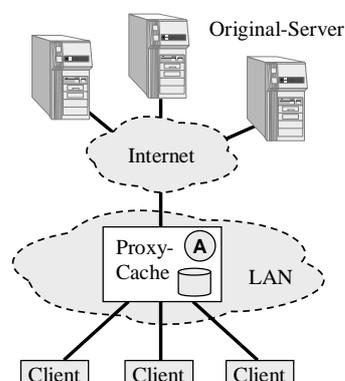


Abbildung 2.1: Systemmodell für Proxy-Caching

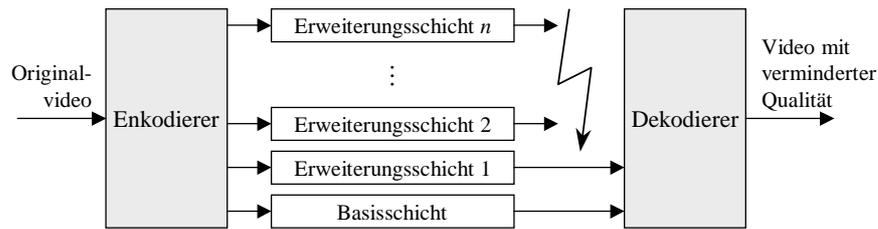


Abbildung 2.2: Hierarchische Kodierung von Videoströmen

kodiert, die schichtweise kombiniert werden können und ein Video mit steigender Qualität ergeben (Abb. 2.2). Dabei kann die Qualität sowohl in der zeitlichen Dimension (erhöhte Bildrate) als auch in der räumlichen Dimension (geringerer Rauschabstand) erhöht werden. Eine Basisschicht (*base layer*) kodiert das Video mit geringer Qualität. Eine oder mehrere Erweiterungsschichten (*enhancement layers*) enthalten weitere Detailinformationen und ergeben gemeinsam mit der Basisschicht eine höhere Qualität. Die Erweiterungsschichten sind dabei hierarchisch geordnet, d.h., eine Erweiterungsschicht kann nur dekodiert werden, wenn alle Schichten mit geringerer Qualität ebenfalls verfügbar sind. Andernfalls ist die Erweiterungsschicht nutzlos.

Sowohl Kangasharju et al. [KHRR01], Rejaie et al. [RYHE00] als auch Podlipnig et al. [PB02] schlagen Heuristiken vor, um zu bestimmen, welche Videos und welche Schichten der einzelnen Videos im Cache gespeichert werden sollen. Dabei berücksichtigen [KHRR01] und [RYHE00] neben der Speicherkapazität im Cache auch die verfügbare Datenrate der Anbindung des Cache an das Internet als begrenzte Ressource. Während [KHRR01] Heuristiken zur Vorab-Replikation basierend auf der stochastischen Knapsack-Theorie vorschlägt und vergleicht, wird in [RYHE00] und [PB02] der pragmatische Weg der dynamischen Verdrängung basierend auf Erweiterungen herkömmlicher Cache-Verdrängungs-Strategien (siehe [PB03]) gegangen. Im Gegensatz zu den anderen genannten Arbeiten zeichnet sich [RYHE00] dadurch aus, dass auch das Speichern von einzelnen Segmenten einer Schicht eines Videos betrachtet wird. In [KHRR01] und [PB02] hingegen wird von einem Lastmodell ausgegangen, bei dem Videos grundsätzlich komplett angefordert werden. Folglich werden auch nur komplette Schichten eines Videos repliziert. Eine Implementierung des Ansatzes aus [RYHE00] wird in einer gemeinsamen Arbeit von Rejaie und Kangasharju [RK01] beschrieben.

Die Arbeit von Sehgal und Chou [SC02] geht nicht den Weg einer heuristischen Lösung, sondern versucht auf Basis eines analytischen Modells eine exakte Lösung des Problems der Vorab-Replikation im Cache zu beschreiben und mittels iterativer Verfahren eine Näherungslösung zu bestimmen. Im Gegensatz zu [KHRR01] und [RYHE00] werden hier aber weder die Datenrate der Anbindung des Cache an das Internet noch die Speicherkapazität im Cache als begrenzte Ressource betrachtet. Anstelle dessen wird davon ausgegangen, dass Übertragungs- und Speicherkapazität zu bestimmten Kosten in hinreichendem Umfang zur Verfügung gestellt werden können. Dementsprechend erfolgt die Optimierung der Platzierung von Replikaten im Cache bezüglich der Kosten und der Qualität der ausgelieferten Videoströme. Kosten und Qualität werden dabei in der Zielfunktion linear kombiniert. Dieses Vorgehen entspricht dem kostenqualitäts-optimierten Ansatz in der vorliegenden Arbeit, wobei aber in der vorliegenden Arbeit im Gegensatz zu [SC02] die Speicherkapazität als begrenzte Ressource betrachtet wird.

Die Beschränkung auf hierarchisch kodierte Videoströme wird in einer jüngeren Arbeit der HP Laboratories Palo Alto zu Caching-Strategien für Medienströme in transcoding-fähigen Proxies [SLB04] aufgehoben. Anstelle dessen wird davon ausgegangen, dass verschiedene Repräsentationen der Medien-Ströme durch Anwendung von Adaptionsoptionen (Transcoding) ineinander überführt werden können. Dabei wird vorausgesetzt, dass eine totale Ordnung der verschiedenen Repräsentationen existiert, also jeweils paarweise für zwei Repräsentationen r_1 und r_2 entweder r_1 zu r_2 oder r_2 zu r_1 adaptierbar ist. Adaption in beliebigen Qualitätsdimensionen ist damit faktisch ausgeschlossen. Gegenstand der Untersuchungen in [SLB04] sind Heuristiken zur Auswahl, welche Repräsentation bzw. welche Repräsentationen durch einen

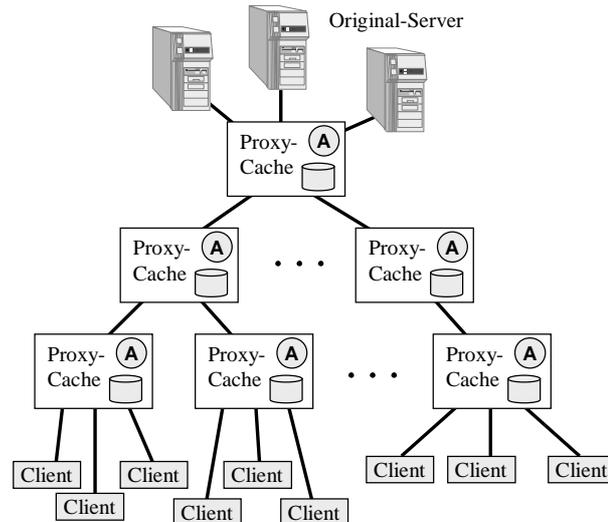


Abbildung 2.3: Hierarchisches Caching

Caching-Proxy mit Adaptionfähigkeiten gespeichert werden sollen. Es werden drei Heuristiken vorgeschlagen, die berücksichtigen, dass Repräsentationen durch Inhaltsadaption ineinander überführt werden können, also adaptivitätssensitiv sind. Die vorgeschlagenen Heuristiken werden mit zwei nicht-adaptivitätssensitiven Varianten aus [TZC02] verglichen. Bei den Varianten aus [TZC02] werden entweder nur die generischen oder nur die adaptierten Repräsentationen gespeichert. Die Analyse in [SLB04] zeigt, dass die vorgeschlagenen, adaptivitätssensitiven Heuristiken den Heuristiken aus [TZC02] überlegen sind. Dieser Vorteil wird insbesondere in Bezug auf die Last zur Durchführung der Inhaltsadaption deutlich, während in Bezug auf typische Metriken zur Evaluation von Caches, wie Byte-Hit-Rate, auch nicht-adaptivitätssensitive Verfahren konkurrenzfähig sind.

Die oben vorgestellten Arbeiten zeichnen sich dadurch aus, dass grundsätzlich nur die Adaption eines einzelnen Qualitätsmerkmals untersucht wird. Eine Adaption in verschiedenen Qualitätsdimensionen, wie in der vorliegenden Arbeit diskutiert, ist nicht Gegenstand der Betrachtungen. Inhaltsadaption wird in vielen vorangegangenen Arbeiten auch nicht als Mittel zur Anpassung an die heterogenen Geräteeigenschaften der Clients betrachtet. Inhaltsadaption dient in [OCAV97, KKO98, PB02, SC02] ausschließlich der effizienteren Nutzung der begrenzten Ressourcen des Proxy.

Im Gegensatz zur vorliegenden Arbeit betrachteten die vorgestellten Ansätze zum Caching von Multimediainhalten ausschließlich die Replikation innerhalb eines einzelnen, autonomen Proxy-Caches. Zwar wird in [RYHE00] die Einbindung des Proxy-Caches in eine verteilte Cache-Architektur, wie hierarchisches Caching [CDN⁺96], als Möglichkeit erwähnt, es erfolgt jedoch keine Koordination der Caching-Strategien der einzelnen Caches. In [SLB04] werden Proxy-Caches als Teil eines Streaming Media Distribution Networks verstanden. Aber auch hier arbeiten die Caching-Strategien lokal, ohne dass eine Koordination mit anderen Proxies stattfindet. Außerdem beschränkt sich in beiden Arbeiten die Evaluation auf ein Szenario mit einem einzelnen Cache.

Eine Ausnahme zu der rein lokalen Betrachtung des Problems stellt [CYH00] dar. Diese Arbeit beschäftigt sich mit Caching und Adaption von WWW-Inhalten in Cache-Hierarchien (Abb. 2.3, vgl. [CDN⁺96]). Dabei wird keine Einschränkung der Adaptionoperationen vorgenommen. Im Gegensatz zu den anderen vorgestellten Arbeiten können Adaptionen in verschiedenen Qualitätsdimensionen berücksichtigt werden. Der Ansatz in [CYH00] zeichnet sich dadurch aus, dass die Last zur Inhaltsadaption innerhalb eines Pfades in der Cache-Hierarchie koordiniert auf verschiedene Proxies verteilt wird. Bezüglich des Caching erfolgt jedoch auch

bei dieser Arbeit keine Koordination zwischen den einzelnen Proxies. Allerdings erfolgt hier die Evaluation anhand eines verteilten Szenarios mit mehreren Proxies. In Bezug auf Caching-Strategien wird gezeigt, dass das Caching der adaptierten Repräsentationen sowohl dem Caching der Ausgangsrepräsentationen bei der Adaption als auch dem gleichzeitigen Caching der adaptierten und der Ausgangsrepräsentationen überlegen ist.

2.3 Inhaltsadaption

Die Adaption von Inhalten an heterogene Anwendungsbereiche wurden in der Literatur sehr umfassend untersucht und beschrieben. Dabei wird ein sehr breites Feld von Anwendungsbereichen, Adaptionenzielen und Adaptionsmechanismen abgedeckt und verschiedene Aspekte der Inhaltsadaption untersucht. Eine Klassifikation der verfolgten Ansätze lässt sich anhand der folgenden Kriterien vornehmen (siehe auch Abb. 2.4):

- **Adaptionsziel.** Woran wird adaptiert? (vgl. [Spr04, Bru01])
- **Adaptionsgegenstand.** Was wird adaptiert? (vgl. [Spr04, Bru01])
- **Zeitpunkt der Adaption.** Wann findet die Adaption statt? (vgl. [LL02])
- **Ort der Adaption.** Wo findet die Adaption statt? (vgl. [FK03])

2.3.1 Adaptionsziel

Mittels Inhaltsadaption kann eine Anpassung der Repräsentation von Inhalten an verschiedene Adaptionsziele vorgenommen werden. Diese können den Nutzer, die aktuelle Situation und die technische Infrastruktur betreffen. Adaption an den Nutzer wurde bereits in früheren Arbeiten zu Hypertext und Hypermedia beschrieben [BKV98], stellt aber auch in aktuellen Arbeiten zu adaptiven WWW-Anwendungen ein wichtiges Thema dar [BM02, FHMW03]. Adaption an den Nutzer, seine Anforderungen, Kenntnisse, Ziele und Vorlieben, erfordert die Modellierung der relevanten Eigenschaften des Nutzers durch das System (Nutzermodellierung, [dK01]) und deren Speicherung in Nutzerprofilen. Auch wenn eine Anpassung an die Vorlieben und Anforderungen eines Nutzers durchaus anwendungsübergreifend erfolgen kann, werden in der Praxis vornehmlich Ansätze verfolgt, bei denen Adaption an den Nutzer anwendungsspezifisch ist.

Die aktuelle Situation umfasst Kontexteigenschaften, wie Ort, Zeit oder Umgebungsbedingungen (z.B. Wetter). Insbesondere Ortsinformationen wurden in vielen Arbeiten zur Erbringung adaptiver Anwendungen und Dienste herangezogen, wie z.B. für ortsabhängige Touristeninformationsdienste [CDMF00]. Allerdings findet bei situationsabhängiger Adaption weniger

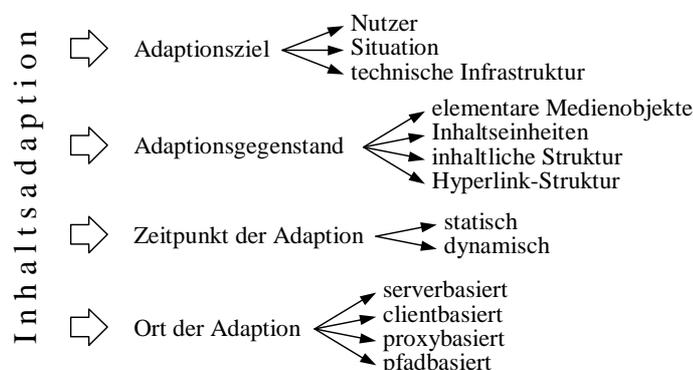


Abbildung 2.4: Schema zur Klassifikation von Ansätzen zur Inhaltsadaption

eine Adaption der Repräsentation der Inhalte, sondern eine Anpassung der anwendungsspezifischen Inhalte selbst statt. Dies geht über das Konzept der Inhaltsadaption, wie sie in der vorliegenden Arbeit adressiert wird, hinaus.

Die Adaption von Inhalten an die technische Infrastruktur ist mit dem Aufkommen mobiler Internetnutzung seit Mitte der 1990er Jahre zum wichtigen Gegenstand von Forschungsarbeiten geworden. Dabei werden zwei Ziele verfolgt: Reduktion der Datenmenge zur Anpassung an die geringe Datenrate mobiler Kommunikationssysteme und Anpassung der Repräsentation von Inhalten an die heterogenen Fähigkeiten mobiler Endgeräte. Die Fähigkeiten der Endgeräte werden dabei sowohl durch deren Hardware- (z.B. Display, Eingabegeräte, CPU-Leistung, Hauptspeicher) als auch durch deren Software-Plattform (z.B. verfügbare Anzeigeprogramme und Codecs) eingeschränkt. Exemplarisch für die Adaption an den technischen Kontext seien hier die Arbeiten im Daedalus-Projekt der UC Berkeley genannt [FB96a, FGCB98]. Das in diesem Projekt entwickelte Pythia-System [FB96a] benutzt verlustbehaftete Kompression von JPEG-Bildern zur Reduktion des Datenvolumens auf der drahtlosen Verbindung zum Client. In [FGCB98] wird zusätzlich die dynamische Umwandlung von Postscript-Dokumenten in Rich-Text (RTF) beschrieben, um die Repräsentation von Inhalten an die eingeschränkten Fähigkeiten des Endgerätes (3Com PalmPilot) anzupassen.

In der vorliegenden Arbeit wird die Inhaltsadaption an die technische Infrastruktur betrachtet. Nutzermodellierung und situationsabhängige Adaption von anwendungsspezifischen Inhalten gehen über das Konzept adaptiver CDNs, wie es in der vorliegenden Arbeit verfolgt wird, hinaus.

2.3.2 Adaptionsgegenstand

Den Arbeiten zur Inhaltsadaption in der wissenschaftlichen Literatur ist gemein, dass sie eine Adaption der Repräsentation von Inhalten beschreiben². Folglich ist der Adaptionsgegenstand aller im Folgenden betrachteten Arbeiten die Repräsentation der Inhaltsobjekte. Dieser kann jedoch auf unterschiedlichen Ebenen betrachtet werden. Bei der Klassifikation sollen in Bezug auf den Adaptionsgegenstand bei der Inhaltsadaption die folgenden Ebenen unterschieden werden:

- die Ebene der elementaren Medienobjekte (*media assets*),
- die Ebene der Inhaltseinheiten,
- die Ebene der inhaltlichen Struktur der Dokumente und
- die Ebene der Hyperlink-Struktur.

Diese Einteilung orientiert sich am komponenten-basierten Modell zur Beschreibung adaptiver WWW-Dokumente aus [FHMW03].

Adaption von Medienobjekten. Die Adaption von elementaren Medienobjekten umfasst Konvertierungen und Transformationen von einfachem Text, Bildern, Audio- und Videoströmen. Viele Forschungsarbeiten zur Adaption von Medienobjekten beschäftigten sich mit der Beschreibung von ausgewählten Adaptionsoperationen, z.B. zur Adaption von Bildern [HBR⁺98, CGSEV01, CSE99] oder Videoströmen [KW02, Cha02, NS03, Chu03, KWC03, LTEZG03]. Dabei wird der Zusammenhang zwischen den Adaptionsoperationen, dem Ressourcenbedarf bei der Adaption, der Qualität und der Größe des adaptierten Medienobjektes modelliert (Abb. 2.5(a)), um daraus Strategien abzuleiten, welche Adaptionen mit welchen Parametern unter welchen Bedingungen sinnvoll sind (Abb. 2.5(b)). Verschiedene Autoren beziehen dabei Metadaten über die zu adaptierenden Inhalte ein [NS03, KWC03]. In A-CDNs werden derartige Modelle benötigt, um den Effekt von Adaptionsoperationen auf die adaptierten Objekte vorherzusagen

²Natürlich werden in der Literatur auch andere Adaptionsgegenstände betrachtet (z.B. die Anwendungsstruktur, Protokollparameter usw., vgl. [Spr04]). Diese Formen der Adaption werden jedoch nicht durch den Begriff der Inhaltsadaption abgedeckt. Deshalb sollen Arbeiten zu anderen Adaptionsgegenständen hier unberücksichtigt bleiben.

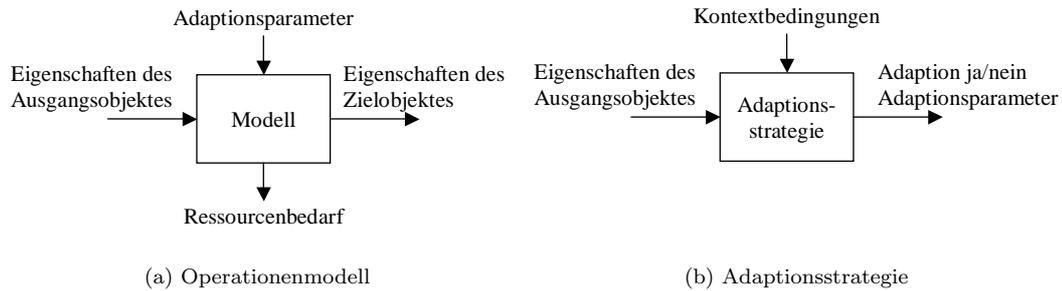


Abbildung 2.5: Modellierung von Adaptionsoptionen und Adaptionstrategien

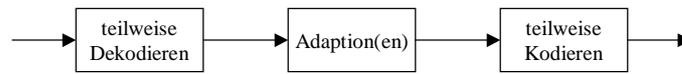
und den Ressourcenbedarf für die Adaption abzuschätzen. Auf Grundlage solcher Abschätzungen erfolgt die Platzierung der Replikat. In dieser Arbeit wird ein Modell zur Beschreibung von Adaptionsoptionen auf JPEG-Bildern verwendet, welches durch eigene Analysen gewonnen wurde (Anhang C.2). Eigene Analysen waren notwendig, da bisherige Publikationen in diesem Bereich [HBR⁺98, CGSEV01, CSE99] nur ausgewählte Aspekte der Adaption separat untersuchen. Sie stellen jedoch kein hinreichend umfassendes Modell der Operationen dar. Die Zusammenhänge, die bei den eigenen Analysen gefunden wurden, sind aber mit den in [HBR⁺98, CGSEV01, CSE99] beschriebenen konsistent.

In anderen Arbeiten werden neue Operationen entwickelt, wie z.B. eine Nachbearbeitung verlustbehaftet komprimierter JPEG-Bilder, um visuelle Artefakte (Block-Artefakte) zu reduzieren [LW99], oder aber Verfahren zur Verbesserung von Adaptionsoptionen durch die Nutzung semantischer Informationen über die Medienobjekte vorgeschlagen [CGP02, FXZM03].

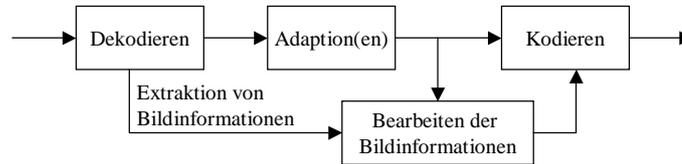
Eine weitere Gruppe von Forschungsarbeiten beschäftigt sich damit, die Implementierung von Adaptionsoptionen zu verbessern, um dadurch adaptionsbedingte Verzögerungen zu verringern und den Ressourcenverbrauch bei der Adaption zu senken. Dies ist insbesondere bei der Adaption von Videos relevant, da das Kodieren von Videos sehr ressourcenintensiv und langsam ist und somit die Adaption dekodierter Videos und anschließende erneute Kodierung nicht praktikabel ist. Deshalb wird nach Möglichkeiten gesucht, Adaptionsoptionen auf kodierten Medienobjekten auszuführen (*compressed-domain processing*), das heißt, ohne dass das Medienobjekt dazu komplett dekodiert und anschließend erneut kodiert werden muss. Anstelle dessen erfolgt nur eine teilweise Dekodierung (z.B. nur Entropiedekodierung), so dass nach erfolgter Adaption nur eine teilweise Kodierung (z.B. nur Entropiekodierung) erforderlich ist (Abb. 2.6(b)). Ein umfangreicher Überblick über solche Verfahren, z.B. zur Farbkonvertierung, zur räumlichen oder zeitlichen Skalierung, wird in [WSA02] gegeben. Für Adaptionsoptionen, die nicht komplett auf kodierten Daten ausgeführt werden können, wird an Möglichkeiten gearbeitet, Informationen des kodierten Videos (z.B. Bewegungsvektoren³) bei der Dekodierung zu bewahren und für eine erneute Kodierung nutzbar zu machen (Abb. 2.6(b)). Damit lässt sich eine signifikante Reduktion der Adaptionszeit erreichen. In [WAF99] wird gezeigt, wie mit Hilfe solcher Techniken eine effiziente Konvertierung von MPEG-2-kodierten Videos in das H.263-Format bei gleichzeitiger räumlicher und zeitlicher Skalierung möglich ist.

Eine Alternative zur Adaption kodierter Medienobjekte, um ebenfalls effektive, skalierbare Adaptionsoptionen zu ermöglichen, ergibt sich durch die Nutzung hierarchischer Kodierungsverfahren (vgl. Abb. 2.2 auf Seite 29) [VC94, CR99]. Dabei beschränkt sich die Adaption auf das Weglassen oder Hinzufügen einzelner Schichten des hierarchisch kodierten Medienobjektes. Im Gegensatz zur Adaption kodierter Medienobjekte ist die Adaption hierarchisch kodierter Medienobjekte auf eine einzelne Qualitätsdimension beschränkt.

³Bewegungsvektoren beschreiben räumlich-zeitliche Zusammenhänge zwischen aufeinander folgenden Einzelbildern (siehe [ES98] für weitere Informationen).



(a) Adaption kodierter Medienobjekte



(b) Wiederverwendung von Bildinformationen aus dem kodierten Ausgangsvideo

Abbildung 2.6: Effiziente Konvertierung von Medienobjekten

Auch wenn die Entwicklung effektiver Implementierungen von Adaptionenoperationen nicht Gegenstand der vorliegenden Arbeit ist, so ist doch die Verfügbarkeit solcher Implementierungen ein kritischer Erfolgsfaktor für A-CDNs. Die Problemmodellierung, die in dieser Arbeit zugrunde gelegt wird, schließt sowohl die Adaption kodierter Medien als auch das Weglassen von Schichten in hierarchisch kodierten Medienobjekten ein. Die Komposition eines hierarchisch kodierten Medienobjektes durch das Hinzufügen von Erweiterungsschichten wird jedoch nicht durch das Modell abgedeckt und geht damit über die Betrachtungen in dieser Arbeit hinaus.

Adaption von Inhaltseinheiten. Gemäß [FHMW03] beschreibt eine Inhaltseinheit die logische Einheit verschiedener Medienobjekte, z.B. ein Bild mit zugehörigem Untertitel. Die Adaption auf der Ebene von Inhaltseinheiten umfasst das Auslassen einzelner Medienobjekte oder die Ersetzung von Medienobjekten durch andere Medienobjekte, z.B. Bilder durch Text [FB96a, Jör99].

Adaption der inhaltlichen Struktur und der Hyperlink-Struktur. Die Adaption der inhaltlichen Struktur der Dokumente ist eng mit der Adaption der Hyperlink-Struktur verbunden. Neben der dokumentinternen Restrukturierung können Dokumente in mehrere Teildokumente dekomponiert werden, die durch Hyperlinks miteinander verknüpft sind, oder mehrere Teildokumente zu einem gemeinsamen Dokument zusammengeführt werden (Komposition). In der Literatur werden zwei verschiedene Ansätze zur Adaption der Struktur von Dokumenten verfolgt: Zum einen werden existierende HTML-Dokumente als Basis für die Adaption benutzt, zum anderen wird von einer darstellungsunabhängigen, generischen Dokumentbeschreibung ausgegangen. Im ersten Fall werden die nur implizit formulierten Strukturinformationen mittels Heuristiken aus dem HTML-Quelldokument gewonnen. Beispiele für diesen Ansatz sind das Digestor- [BGS99] und m-Links-System [STHK01] des Fuji Xerox's Palo Alto Laboratory (FXPAL) oder auch die Arbeiten von Buyukkokten et al. an der Stanford University [BGMP01]. Der zweite Ansatz erfordert neue Beschreibungssprachen für die explizite Beschreibung der inhaltlichen Struktur von Dokumenten. Durch das Hinzufügen von Metainformationen in der Dokumentbeschreibung kann außerdem die Adaption gezielt gesteuert werden. Aktuelle Arbeiten in diesem Bereich benutzen XML-basierte Sprachen zur Beschreibung der inhaltlichen Struktur der Dokumente [GBZS01, GSZ⁺03, FHMW03].

Die vorliegende Arbeit konzentriert sich auf die Adaption einzelner Medienobjekte, insbesondere multimedialer Medienobjekte wie Bilder, Audio und Video. Die Adaption der Inhaltseinheiten, der inhaltlichen Struktur sowie der Hyperlink-Struktur ist nicht Gegenstand der Betrachtungen.

2.3.3 Zeitpunkt der Adaption

In Bezug auf den Zeitpunkt der Adaption wird zwischen statischer und dynamischer Adaption unterschieden. Statische Adaption bedeutet, dass bereits zum Zeitpunkt der Inhaltserstellung verschiedene Repräsentationen der Inhalte erzeugt und auf dem Originalserver bereitgestellt werden. Zur Ausführungszeit wird dann nur noch eine voradaptierte Repräsentation ausgewählt und an den Client ausgeliefert. Statische Adaption wird zum Beispiel im Odyssey- [NPS95] und im InfoPyramid-System [MSL99] umgesetzt. Der Vorteil der statischen Adaption liegt darin, dass der Autor den Adaptionsprozess unmittelbar beeinflussen kann und die Darstellung für verschiedene Endgeräte von Hand optimieren kann. Andererseits ist statische Adaption mit einem erhöhten Aufwand für Verwaltung und Wartung der verschiedenen Repräsentationen verbunden. Außerdem besteht ein erhöhter Bedarf an Speicherplatz, um die verschiedenen Repräsentationen der Inhaltsobjekte zu speichern.

Bei dynamischer Adaption wird nur eine einzige generische Repräsentation auf dem Originalserver gespeichert. Adaptierte Repräsentationen werden erst zum Zeitpunkt des Zugriffs erzeugt. Dieser Ansatz wird z.B. im Daedalus- [FB96a, FGCB98] und Ninja-Projekt [BFK⁺00, GWB⁺01], dem Digestor-System [BGS99] oder dem DDL-Transcoding-Framework [GBZS01] verwendet. Im Gegensatz zur statischen Adaption bringt der dynamische Ansatz Antwortzeitverzögerungen und Adaptionskosten (insbes. CPU-Last) zur Laufzeit mit sich, da Adaptionoperationen sehr rechenintensiv sein können.

Aufgrund der dargestellten Probleme bei beiden Ansätzen schlägt [LL02] vor, beide Ansätze zu kombinieren: Ausgewählte, häufig benötigte Repräsentationen werden vorab erzeugt und stehen ohne zusätzliche Adaptionskosten und Verzögerung auf dem Server bereit. Weitere Repräsentationen, die nicht auf dem Server gespeichert werden können, werden dynamisch beim Zugriff erzeugt. Die Auswahl, welche Repräsentationen statisch erzeugt werden, erfolgt durch das System auf Grundlage von Abschätzungen des Zugriffsverhaltens mittels eines Greedy-Algorithmus. Damit wägt das System selbstständig zwischen den Adaptionskosten der dynamischen Adaption und dem Speicherplatzbedarf der statischen Adaption ab.

Adaptivitätssensitive Platzierung von Replikaten in einem A-CDN ist diesem Verfahren sehr ähnlich. Auch hier wird statische und dynamische Adaption kombiniert. Allerdings findet die statische Adaption in A-CDNs nicht zum Zeitpunkt der Inhaltserstellung, sondern ebenfalls zur Laufzeit, aber bereits vor dem Zugriff der Clients statt. Zusätzlich zu der Entscheidung über statische oder dynamische Adaption wird in der vorliegenden Arbeit der Aspekt der Replikation und Verteilung betrachtet.

2.3.4 Ort der Adaption

Inhaltsadaption kann an verschiedenen Orten im Netzwerk vorgenommen werden. In Bezug auf den Ort der Adaption werden die folgenden Ansätze unterschieden:

- **Serverbasierte Adaption.** Bei diesem Ansatz findet die Adaption nur auf dem Server statt (Abb. 2.7(a)). Dieser Ansatz wurde z.B. in Odyssey [NPS95], dem InfoPyramid-System [MSL99], dem DDL-Transcoding-Framework [GBZS01] und dem AMACONT-Projekt [FHMW03] verfolgt. Auch viele Systeme in der Praxis nutzen serverbasierte Adaption auf einem Application-Server (z.B. [oV02a]).
- **Clientbasierte Adaption.** Adaption auf dem Client (Abb. 2.7(b)) spielt bei mobilen Internetszenarien eine untergeordnete Rolle, da clientbasierte Adaption keine Adaption der angeforderten Inhalte an die Eigenschaften der Netzwerkverbindung erlaubt und darüber hinaus die begrenzten Ressourcen mobiler Geräte die Möglichkeiten zur Adaption auf dem Client einschränken. Clientbasierte Adaption wird zum Beispiel im Smiley-System der UCLA [BFK⁺00] oder durch die Smart Clients der UC Berkeley [YCE⁺97] umgesetzt.
- **Proxybasierte Adaption.** Bei proxybasierter Adaption findet die Adaption nicht auf dem Server, sondern auf einer Zwischeninstanz zwischen Server und Client, einem Proxy,

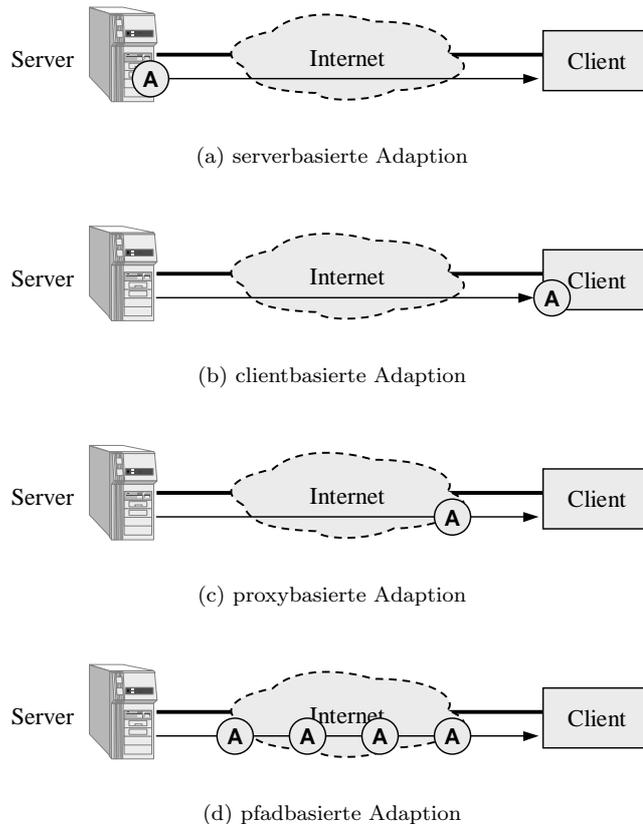


Abbildung 2.7: Ort der Adaption

statt (Abb. 2.7(c)). Der Proxy kann dabei von mehreren Clients gemeinsam genutzt werden, um auf Inhalte von verschiedenen Servern zuzugreifen. Somit entfällt die Notwendigkeit, dass jeder Server in der Lage sein muss, Inhalte für alle Clients zu adaptieren. Aufgrund der einfachen Integrierbarkeit proxybasierter Lösungen in existierende Infrastrukturen (weder Server noch Client müssen dafür geändert werden) wurde dieser Ansatz im wissenschaftlichen Umfeld intensiv verfolgt, z.B. im Daedalus-Projekt [FB96a, FGCB98] oder beim Digestor- [STHK01] und m-Links-System [BGS99].

- **Pfadbasierte Adaption.** Im Gegensatz zu den anderen Ansätzen ist beim pfadbasierten Ansatz Inhaltsadaption nicht auf einen einzelnen Knoten beschränkt, sondern wird verteilt auf verschiedenen Knoten durchgeführt (Abb. 2.7(d)). Diesem Ansatz liegt die Annahme zugrunde, dass Inhaltsadaption oft mehrere elementare Adaptionsschritte umfasst, die nacheinander ausgeführt werden, um komplexe Adaptionen zu ermöglichen. Außerdem gehen viele Arbeiten in diesem Bereich davon aus, dass nicht nur eine Adaption an das Endgerät, dessen unmittelbare Anbindung an das Internet und den Nutzer erfolgt, sondern auch eine Adaption an die wechselnden Übertragungseigenschaften im Internet zwischen Server und Client notwendig ist, z.B. durch Protokollanpassungen, Verschlüsselung oder Fehlerkorrekturmaßnahmen. Pfadbasierte Adaption wurde insbesondere im Kontext aktiver programmierbarer Netze verfolgt, z.B. sind Conductor [YRP99, YREP01], Panda [FRE⁺02], CANS [FSAK01], Active Pipes [KRW01, CTW01] und End-to-End Media Paths [NPB01] Vertreter solcher Arbeiten, bei denen Inhaltsadaption in den aktiven Routern eines aktiven programmierbaren Netzes durchgeführt wird. Ninja [GWB⁺01, CMI00] hingegen betrachtet pfadbasierte Adaption in lokalen Clustern von Workstations.

Fu und Karamcheti [FK03] haben in einer umfassenden Studie die Vor- und Nachteile der verschiedenen Ansätze zur räumlichen Verteilung der Adaption untersucht. Als Anwendungsszenario wurde dabei der Zugriff von Clients mit unterschiedlichen Netzwerkanbindungen (breitbandige Festnetzverbindung, Modemverbindung, WLAN 802.11b) auf Videoströme auf verschiedenen im Internet verteilten Videosevernen gewählt. Es wird gezeigt, dass je nach Netzwerkumgebung verschiedene Ansätze zu bevorzugen sind. Der pfadbasierte Ansatz zeigt aber in sehr unterschiedlichen Umgebungen gute Ergebnisse und ist damit am besten in der Lage, sich flexibel unterschiedlichen Anforderungen anzupassen.

Auch in der vorliegenden Arbeit wird ein pfadbasierter Ansatz zur Adaption verwendet. Als Knoten für die Adaption werden dabei aber nicht die aktiven Router eines aktiven programmierbaren Netzes berücksichtigt, sondern die Surrogate im CDN.

Ein zentrales Problem bei pfadbasierter Adaption besteht in der Komposition der Adaptionspfade. In der wissenschaftlichen Literatur werden dafür sehr unterschiedliche Verfahren dargestellt. Viele Ansätze unterscheiden dabei zwischen einem logischen Pfad und einem physischen Pfad [CMI00, YREP01, NPB01, FSAK01]. Der logische Pfad beschreibt die Abfolge von elementaren Adaptionsschritten unabhängig vom Ort, an dem die Adaption stattfindet. Der physische Pfad hingegen ist die Folge von Netzwerkknoten zwischen Quelle (Server) und Senke (Client) (vgl. Abb. 2.8). Diese Unterscheidung erlaubt, den logischen und physischen Pfad getrennt zu ermitteln und damit die Komplexität der Suche nach einem optimalen Pfad zu reduzieren. Bei Conductor [YREP01], End-to-End Media Paths [NPB01] und CANS [FSAK01] wird zuerst der physische Pfad ermittelt und anschließend ein logischer Adaptionspfad auf diesen abgebildet. Conductor [YREP01] und End-to-End Media Paths [NPB01] verwenden dazu eine Bibliothek vordefinierter logischer Adaptionspfade (*templates*), die durch den Anwendungsentwickler vorgegeben werden müssen. Aus der Bibliothek wird ein geeigneter logischer Adaptionspfad ausgewählt und anschließend Knoten im physischen Pfad gesucht, an denen die einzelnen Adaptionsoptionen im logischen Adaptionspfad ausgeführt werden. Auch im Panda-System [FRE⁺02] wurde unter anderem ein solches Verfahren realisiert. Im CANS-System [FSAK01] hingegen werden keine vordefinierten logischen Adaptionspfade verwendet. Anstelle dessen wird die Menge korrekter logischer Adaptionspfade durch getypte Ein- und Ausgabeports der Adaptionsoptionen definiert. Ein näherungsweise optimaler Adaptionspfad wird durch dynamische Programmierung ermittelt, wobei schrittweise Adaptionsoptionen hinzugefügt werden und ausgehend vom Zielknoten den physischen Knoten im physischen Pfad zugeordnet werden. Bei der Auswahl und Zuordnung von Adaptionsoptionen werden die verfügbaren Ressourcen der physischen Knoten berücksichtigt.

Bei Ninja Paths [GWB⁺01, CMI00] und Active Pipes [KRW01, CTW01] wird zuerst der logische Adaptionspfad ermittelt und anschließend ein physischer Pfad gesucht, in dem der logische Adaptionspfad instantiiert werden kann. Bei Ninja Paths wird die Menge korrekter logischer Adaptionspfade durch streng getypte Ein- und Ausgabeports der Adaptionsoptionen

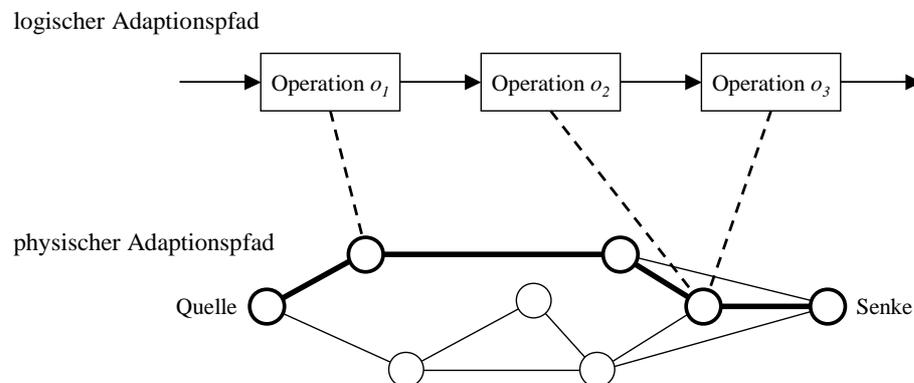


Abbildung 2.8: Logische und physische Adaptionspfade

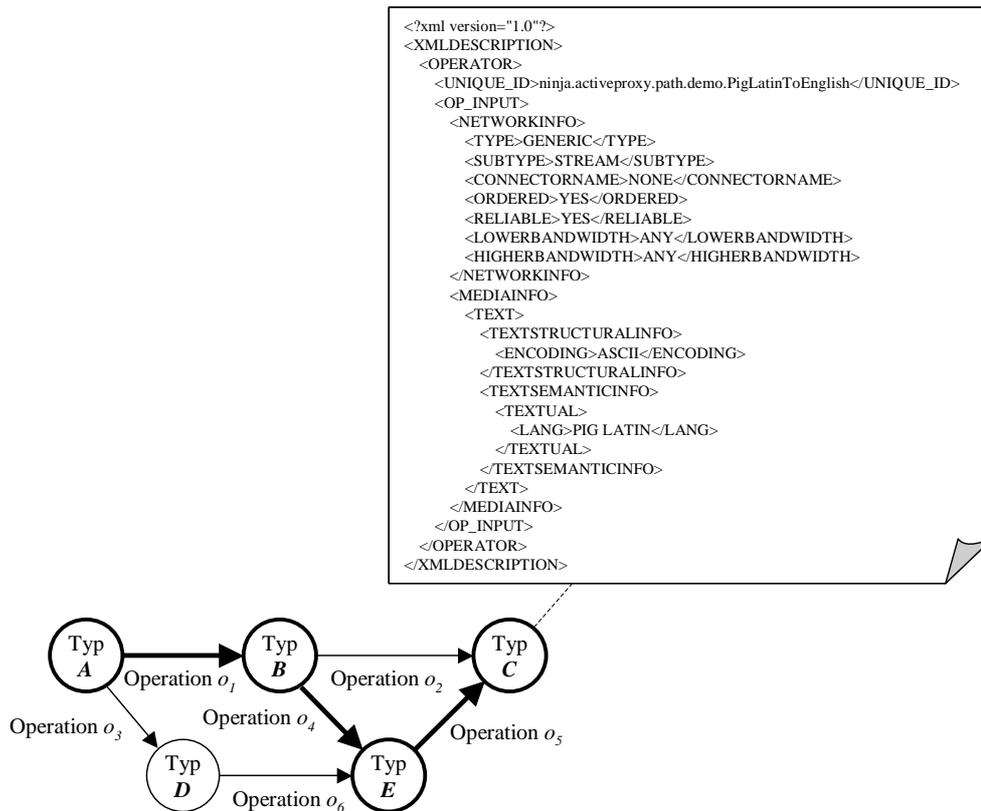


Abbildung 2.9: Beispiel zur Konstruktion eines logischen Adaptionspfades und einer XML-Typbeschreibung in Ninja [CMI00]

nen definiert. Für die Konstruktion eines logischen Adaptionspfades wird Breitensuche (*breadth first search*) in einer Graphrepräsentation des Problems verwendet. Hierzu wird auf Basis des Typsystems ein gerichteter Graph konstruiert, bei dem die Knoten Datentypen repräsentieren und die Kanten Operationen, die den Datentyp des jeweiligen Startknotens in den des Zielknotens überführen (Abb. 2.9). Das Optimierungskriterium bei der Breitensuche ist die Zahl der Adaptionsoptionen im Graph. Nachdem ein korrekter logischer Pfad minimaler Länge ermittelt wurde, wird dieser auf einen physischen Pfad abgebildet. Dazu werden physische Knoten gesucht, die die einzelnen Adaptionsoptionen ausführen können. Für die Suche nach geeigneten Knoten zur Ausführung der Adaptionsoptionen verwendet Ninja einen *Service Discovery Service*. Wird mehr als ein geeigneter Knoten für eine Adaptionsoption gefunden, erfolgt eine gleichverteilt zufällige Auswahl [CMI00]. Durch die zufällige Auswahl wird Lastverteilung zwischen den verschiedenen Knoten erreicht. Eine spätere Publikation zum Ninja-Projekt [GWB⁺01] beschreibt außerdem die Möglichkeit, die verfügbare Rechenkapazität und die Ausführungskosten in die Auswahl von physischen Knoten einzubeziehen. Welche Kostenmetrik dabei zugrunde gelegt wird, wird jedoch nicht erörtert. Da bei diesem Verfahren die Eigenschaften und Kosten der Verbindungen zwischen den Adaptionsoptionen nicht berücksichtigt werden, ist der Ninja-Paths-Ansatz weniger für den Weitverkehrsbereich geeignet, sondern für die Anwendung in lokalen Clustern, wie den Ninja Bases [GWB⁺01].

Bei Active Pipes [KRWP00, KRW01, CTW01] wird der logische Pfad durch die Anwendung vorgegeben. Dazu wird in einer speziellen Grammatik, der Active-Pipe-Grammatik, die Abfolge der Operationen im Pfad definiert. Allerdings erlaubt die Grammatik die Definition von optionalen Operationen in einem logischen Pfad. Optionale Operationen können im Adaptionspfad ausgeführt werden, müssen aber nicht. Sie verändern weder die Repräsentation der Daten noch

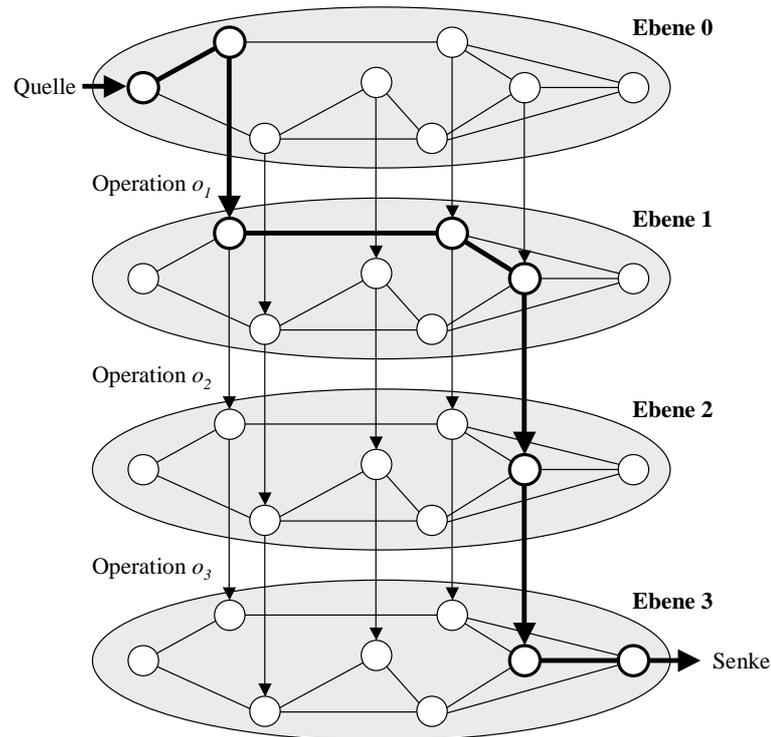


Abbildung 2.10: Graphmodell der Pfadsuche bei Active Pipes [CTW01]

die Semantik des logischen Pfades (sind also entsprechend der Betrachtung in dieser Arbeit keine Adaptionsoptionen). Als Beispiel für optionale Operationen nennt [KRWP00] Monitoring-Funktionen oder Staukontrolle im Netz. Darüber hinaus kann die Definition eines logischen Pfades auch Einschränkungen bzgl. der Abbildung des logischen Pfades auf einen physischen Pfad beinhalten, z.B. dass bestimmte Operationen nur auf bestimmten Knoten oder in einer bestimmten Netzwerkdomäne ausgeführt werden dürfen. Die Abbildung des logischen Pfades auf einen physischen Pfad erfolgt durch Abbildung des Problems auf die Suche nach dem kürzesten Pfad in einem gewichteten gerichteten Graphen. Zur besseren Anschauung wird der Graph dreidimensional als geschichteter Graph dargestellt (Abb. 2.10). Für einen logischen Pfad mit n obligatorischen Operationen wird ein Graph mit $n + 1$ Ebenen erzeugt. Jede Ebene repräsentiert die physische Topologie des Netzwerkes. Die Kanten sind mit den Kommunikationskosten der Netzwerklinks gewichtet. Die Ausführbarkeit der k -ten Operation im logischen Pfad an einem physischen Knoten wird durch eine vertikale Kante ausgedrückt, die die Repräsentation des physischen Knotens in der k -ten Ebene mit der in der $k + 1$ -ten Ebene verbindet. Die Bewertung dieser Kanten entspricht den Verarbeitungskosten zur Ausführung der Operation. Die Kosten für optionale Operationen, die nur an bestimmten Knoten ausgeführt werden, werden einfach der Bewertung aller vom Ausführungsknoten ausgehenden Kanten hinzugerechnet. In einem solchen Graphen stellt sich die Suche nach einem optimalen physischen Adaptionspfad als Suche nach dem kürzesten Pfad zwischen dem Repräsentanten des Quellknotens in der ersten Ebene und dem Repräsentanten der Senke in der n -ten Ebene dar. Dafür können beliebige Kürzester-Pfad-Algorithmen verwendet werden. Da dieser Ansatz die Kenntnis der gesamten Infrastruktur voraussetzt, skaliert er nicht mit sehr großen Systemen. Deshalb wird in [KRW01] eine Erweiterung vorgeschlagen, wobei das System hierarchisch partitioniert wird (Abb. 2.11). Die Pfadsuche wird dann in einem Teile-und-Herrsche-Ansatz zuerst auf der obersten Hierarchieebene auf Grundlage aggregierter Informationen ausgeführt und die Details der Pfadsuche werden anschließend auf Systempartitionen in unteren Hierarchieebenen delegiert.

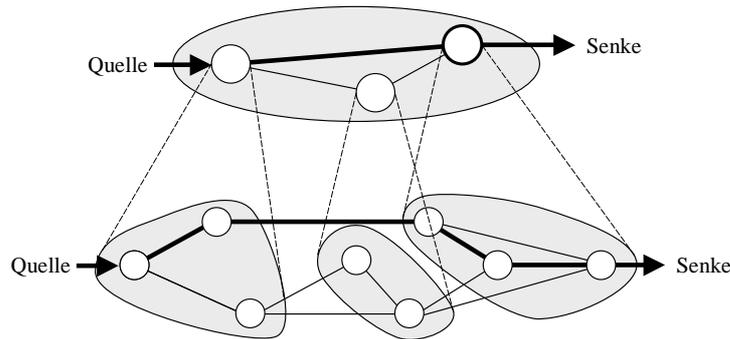


Abbildung 2.11: Hierarchisches Modell der Pfadsuche bei Active Pipes [KRW01]

Die Idee aus [CTW01], das Problem auf die Suche nach dem kürzesten Pfad in einem gewichteten gerichteten Graphen abzubilden, bildet die Grundlage des Verfahrens zur Suche nach optimalen Adaptionspfaden in A-CDNs, das in der vorliegenden Arbeit entwickelt wird. Allerdings wird sie dort nicht nur zur Suche nach einem physischen Pfad verwendet, sondern auch zur Suche nach einem logischen Pfad.

Die oben genannten Verfahren zur Komposition von Adaptionspfaden haben gemeinsam, dass Sie bei der Komposition zwischen logischem und physischem Adaptionspfad trennen. Auf diese Weise wird die Komplexität der Pfadsuche eingeschränkt. Andererseits muss ein optimaler Adaptionspfad in der Praxis nicht immer auf dem kürzesten physischen Pfad instantiierbar sein. Ebenso kann die physische Platzierung der Operationen Auswirkungen auf eine sinnvolle Abfolge von Operationen im logischen Adaptionspfad haben. Eine unter beliebigen Bedingungen optimale Lösung kann nur gefunden werden, wenn logischer und physischer Pfad gemeinsam optimiert werden. Ein solches Verfahren wird in einer Arbeit zur Adaption von Medienobjekten in verteilten, kooperativen Multimediasystemen [CSR98] beschrieben. In einem verteilten, kooperativen Multimediasystem werden Medienobjekte an verschiedenen Knoten im System in verschiedenen Repräsentationen gespeichert, angezeigt und bearbeitet. Die Knoten verfügen über unterschiedliche Fähigkeiten zum Anzeigen, Bearbeiten und Konvertieren von Repräsentationen. Eine Repräsentation wird in [CSR98] durch einen einfachen Dateityp gekennzeichnet. Adaptionspfade, die hier als Objekt-Synthese-Sequenzen bezeichnet werden, werden als eine Abfolge von Konvertierungen (Adaptionoperationen) und Netzwerktransfers verstanden (Abb. 2.12(a)). Diese Sichtweise wird auch in der vorliegenden Arbeit übernommen. Zur Ermittlung einer Objekt-Synthese-Sequenz wird ein Algorithmus (*DOptOSA*-Algorithmus) vorgeschlagen, der mittels dynamischer Programmierung eine optimale Lösung ermittelt. Die Optimierung kann entweder nach den Kosten der Operationen in der Objekt-Synthese-Sequenz, nach der Qualität der generierten Repräsentation des Objektes oder nach beiden Kriterien erfolgen. Dazu werden die Kosten einer Operation o_k und die Änderung der Objektgröße und -qualität in einfacher Weise über die Faktoren $\text{cost_ratio}(o_k)$ bzw. $\text{size_ratio}(o_k)$ und $\text{QF}(o_k)$ definiert (Abb. 2.12(b)). Für die Optimierung bzgl. Qualität und Kosten werden Kosten ($\text{TotCost}(SS)$) und Qualität ($\text{FinQual}(SS)$) einer Objekt-Synthese-Sequenz in der Zielfunktion linear kombiniert (Abb. 2.12(c)).

Ähnliche Konzepte wie die zur Ermittlung von verteilten Adaptionspfaden werden in manchen Arbeiten auch ohne Berücksichtigung des Verteilungsaspektes verwendet. In diesem Fall wird nur ein logischer Adaptionspfad ermittelt, der an einem lokalen Knoten instantiiert wird. Zum Beispiel wird in [CSR98] neben dem *DOptOSA*-Algorithmus auch eine vereinfachte Form (*OptOSA*-Algorithmus) vorgeschlagen, bei dem alle Operationen lokal ausgeführt werden. In [Mar03] wird ein ähnliches Verfahren zur Adaption von Multimediaströmen in Medienservern vorgeschlagen. Einzelne Adaptionoperationen, die hier als Medienfilter bezeichnet werden, werden zu Filtergraphen kombiniert, die komplexe Adaptionen ausführen können. Im Gegensatz zu den Arbeiten zu Adaptionspfaden berücksichtigt diese Arbeit auch Dekompositions- und Kompositionsoperationen, die Medienobjekte in Teilobjekte zerlegen oder mehrere Teil-

$$SS = (N_1, obj, typ_1) \xrightarrow{o_1} (N_2, obj, typ_2) \xrightarrow{o_2} (N_3, obj, typ_3) \xrightarrow{o_3} \dots \xrightarrow{o_{m-1}} (N_m, obj, typ_m)$$

(a) Objekt-Synthese-Sequenzen

$$\begin{aligned} \text{cost}(o_k, obj, typ_k) &= \text{size}(obj, typ_k) \cdot \text{cost_ratio}(o_k) && \text{(Operationskosten)} \\ \text{size}(obj, typ_{k+1}) &= \text{size}(obj, typ_k) \cdot \text{size_ratio}(o_k) && \text{(Objektgröße)} \\ \text{QA}(obj, typ_{k+1}) &= \text{QA}(obj, typ_k) \cdot \text{QF}(o_k) && \text{(Objektqualität)} \end{aligned}$$

(b) Beschreibung der Operationen

$$w_c \cdot \text{TotCost}(SS) - w_q \cdot \text{FinQual}(SS) = \min!$$

(c) Zielfunktion für die Optimierung bzgl. Qualität und Kosten

Abbildung 2.12: Modellierung nach [CSR98]

objekte zu einem Medienobjekt zusammenfügen können. Die Korrektheit eines Filtergraphen wird durch getypte Ein- und Ausgabeports der Filter definiert. Die Komposition optimaler, korrekter Filtergraphen funktioniert folgendermaßen: Das Problem, für das ein Filtergraph gesucht wird, wird in Form eines inkorrekten Filtergraphen beschrieben, d.h. ein Filtergraph, bei dem Ein- und Ausgangsports mit unterschiedlichen Typen miteinander verbunden sind. Solche Verbindungen werden als *magische Kanäle* bezeichnet. Nun wird der Filtergraph durch das Hinzufügen von weiteren Filtern und durch andere Umformungen schrittweise in einen korrekten Filtergraphen überführt. Während [Mar03] die dafür notwendigen Transformationsregeln definiert, wird eine algorithmische Lösung nicht präsentiert.

Inhaltsadaption, die nicht an den Endpunkten der Kommunikation, also nicht auf Originalservern oder Clients, ausgeführt wird, ist in der Praxis sowohl mit juristischen als auch mit Sicherheitsproblemen verbunden. Die Modifikation von Inhalten durch Dritte ist im Allgemeinen durch die Urheberrechte des Inhaltsanbieters untersagt. Aus diesem Grund können nur Lösungen den Weg in die Praxis finden, bei denen zwischen dem Inhaltsanbieter und dem Betreiber des Adaptiondienstes eine Vertragsbeziehung existiert, die dem Betreiber des Adaptiondienstes explizit das Recht zur Adaption der Inhalte einräumt. Im Kontext von Content Distribution Networks ist im Allgemeinen eine Vertragsbeziehung zwischen Inhaltsanbieter und CDN-Provider gegeben, die solche Rechte einschließen kann.

In Bezug auf Sicherheit ist festzustellen, dass Inhaltsadaption durch Zwischeninstanzen, wie z.B. Proxies, die Ende-zu-Ende-Sicherheit zwischen Inhaltsanbieter und Client aufbricht. Sollen trotzdem WWW-Dienste, die vertrauliche Informationen bereitstellen, von verteilten Adaptiondiensten profitieren können, so muss der Betreiber des Adaptiondienstes durch vertragliche Regelungen als Dritter in die Vertrauensbeziehung zwischen Inhaltsanbieter und Client einbezogen werden. Eine Alternative dazu stellt der Ansatz von [CHLS02] dar. Diese Arbeit schlägt einen Ansatz vor, der Filterung von Inhalten ermöglicht, ohne dass diese dazu entschlüsselt und anschließend erneut verschlüsselt werden müssen. Damit bleiben auch digitale Signaturen über den Filterungsprozess hinweg gültig. Dazu müssen Inhalte vom Inhaltsanbieter in Inhaltskomponenten zerlegt werden, die anschließend einzeln verschlüsselt und signiert werden. Unverschlüsselte Metainformationen geben dann einem proxy-basierten Filter Hinweise über Prioritäten und Abhängigkeiten zwischen den Inhaltskomponenten. Auf deren Grundlage kann eine Adaption erfolgen, ohne dass die Inhalte entschlüsselt werden müssen. Die Möglichkeiten dieses Ansatzes beschränken sich aber auf Filterung von Inhalten. Weitergehende Adaptionoperationen werden nicht unterstützt. Außerdem können Metainformationen und die Menge von Inhaltskomponenten einem Angreifer bereits Rückschlüsse auf den Inhalt ermöglichen, was die Anwendbarkeit des Ansatzes auch aus Sicherheitssicht fraglich erscheinen lässt.

2.4 Inhaltsadaption in CDNs

Während in der Forschungsgemeinde verschiedene Ansätze zur Inhaltsadaption diskutiert wurden, hat sich in der Praxis bisher in erster Linie der serverbasierte Adaptionansatz durchgesetzt. Wenn Inhalte auf Originalservern an die Nutzungsbedingungen konkreter Nutzer angepasst werden, können diese Inhalte jedoch nicht mehr ohne weiteres in Proxy-Caches oder in CDNs gespeichert werden. Dieses Problem motivierte Betreiber von Content Distribution Networks und Anbieter von CDN-Technologien, nach Lösungen zu suchen, wie Inhaltsadaption von Originalservern in ein CDN verlagert werden kann. Zu diesem Zweck haben sich Betreiber und Hersteller, wie Akamai Technologies und Network Appliance, im Jahr 1999 im ICAP-Forum (<http://www.i-cap.org/>) zusammengefunden. Ziel des Forums war die Definition von standardisierten Protokollen und Strategien, um Mehrwertdienste von Originalservern in CDNs und zu WWW-Caches zu verlagern. Standardisierte Schnittstellen sollten dabei von Anfang an erlauben, Dienste verschiedener Anbieter einzubinden. Beispiele für solche Mehrwertdienste sind:

- Überprüfen von Inhalten auf Viren,
- Einfügen von Werbe-Bannern,
- Filtern von Informationen,
- Übersetzung von Markup-Sprachen (für mobile Geräte),
- Übersetzung von natürlichen Sprachen,
- Datenkompression und
- Skalierung und Formatumwandlung elementarer Medienobjekte.

Das Ergebnis dieser Bestrebungen ist die Spezifikation des Internet Content Adaptation Protocol (ICAP) [EC03], eines leichtgewichtigen Protokolls zum Aufruf von Mehrwertdiensten auf sogenannten ICAP-Servern von Surrogaten oder Proxies aus. Die Surrogate bzw. Proxies treten dabei gegenüber den ICAP-Servern als ICAP-Clients auf (Abb. 2.13).

ICAP definiert Primitive, um sowohl HTTP-Anfragen als auch HTTP-Antworten – und somit WWW-Inhalte – anpassen zu können. Damit erlauben sie Operationen, die über Inhaltsadaption, wie in der vorliegenden Arbeit diskutiert, hinausgehen. Allerdings definiert die ICAP-Spezifikation nur ein Protokoll zum Aufruf von Adaptiondiensten auf ICAP-Servern aus einem CDN heraus. Die Nutzung der Adaptiondienste und die Koordination mit den Platzierungsmechanismen im CDN ist außerhalb des Gegenstandsbereiches von ICAP.

Während ICAP aufgrund der Unterstützung vieler Hersteller und Betreiber (z.Zt. engagieren sich 85 Firmen im ICAP-Forum) bereits eine gewisse Verbreitung in der Praxis erreicht hat, sind die Standardisierungsbestrebungen in die Aktivitäten der Arbeitsgruppe zu Open Pluggable Edge Services (OPES) der IETF (<http://www.ietf.org/html.charters/opes-charter.html>)

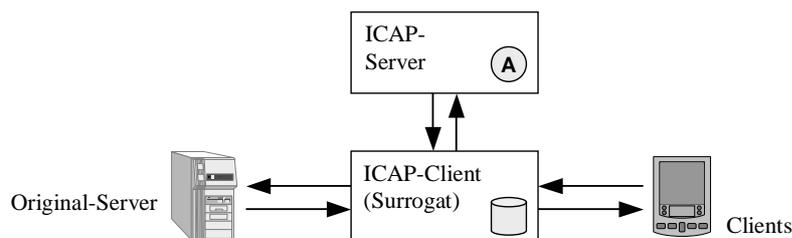


Abbildung 2.13: ICAP-Modell

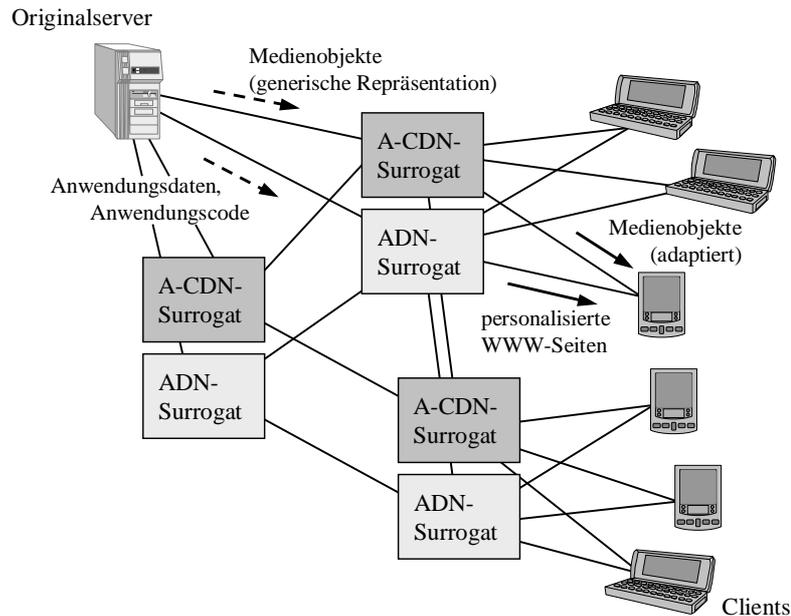


Abbildung 2.14: Zusammenspiel von ADN und A-CDN

eingeflossen. OPES definiert in Anlehnung an ICAP ein Protokoll [Rou04] zur Einbindung von externen Mehrwertdiensten in CDNs. Dabei soll das OPES-Protokoll aber deutlich flexibler sein als ICAP. So wird z.B. die Bindung an das HTTP-Protokoll aufgehoben. Darüber hinaus spezifiziert OPES auch ein Architekturmodell [BCH⁺02], welches neben der Einbindung von externen Mehrwertdiensten auch die Ausführung von Mehrwertdiensten in den Surrogaten selbst definiert. Die Koordination der Inhaltsadaption und der Platzierungsmechanismen im CDN ist aber ebenfalls nicht Gegenstand der Aktivitäten der OPES-Arbeitsgruppe.

Ein weiteres Forschungsthema im Bereich von CDNs ist die Auslagerung ganzer Anwendungslogik in ein CDN. Dieser Ansatz wird teilweise auch als Application Delivery Network (ADN) [Kin03] oder Application Content Delivery Network (ACDN) [RXA03] bezeichnet. Damit können nicht nur generische Adaptionoperationen auf einzelnen Medienobjekten im CDN ausgeführt werden, sondern auch komplexe WWW-Seiten im CDN individuell für eine konkrete Anfrage erzeugt werden [RS02]. Anstatt zum Beispiel eine Anfrage an eine Online-Enzyklopädie auf einem zentralen Anwendungsserver auszuwerten und nur das Ergebnis der Anfrage (bzw. die statischen Komponenten des Ergebnisses) im CDN zu replizieren, kann auch die Enzyklopädie-Anwendung auf den Surrogaten selbst ausgeführt werden. In diesem Fall muss die Anwendungslogik auf den Surrogaten verfügbar sein, und die Anwendungsdaten müssen im CDN verteilt werden. Bei ausschließlich lesenden Zugriffen auf die Anwendungsdaten gelten dafür die gleichen Anforderungen wie bei der Verteilung von nicht-adaptierbaren Inhalten in CDNs. Wird auf die Anwendungsdaten auch schreibend zugegriffen, erfordert die Replikation der Anwendungsdaten transaktionsgeschützte Mechanismen, wie sie im Bereich verteilter Datenbanken diskutiert werden, um One-Copy-Serialisierbarkeit zu gewährleisten [Len97].

ADNs sind nicht als alternativer Ansatz zu A-CDNs zu sehen. Vielmehr können sich A-CDNs und ADNs bei der Gewährleistung von skalierbaren Web-Diensten ergänzen. Während ADNs erlauben, dynamische, evtl. personalisierte WWW-Seiten bei Bedarf zu erzeugen, wird eine Wiederverwendung der generierten Inhalte nicht betrachtet. Deshalb bietet sich an, die weniger dynamischen, wiederverwendbaren Medienobjekte durch andere Technologien, wie z.B. A-CDNs, zu verteilen. Dieses Zusammenspiel zwischen ADNs und A-CDNs wird in Abbildung 2.14 illustriert.

2.5 Fazit

Der Überblick über verwandte Forschungsarbeiten hat gezeigt, dass koordinierte Platzierungsmechanismen für CDNs, die die Adaptierbarkeit der Inhalte bei der Ermittlung einer Platzierung berücksichtigen, um dadurch statische und dynamische Adaption flexibel zu kombinieren, bisher nicht untersucht worden sind. Arbeiten zur Platzierung von Replikaten in herkömmlichen CDNs berücksichtigen keine Inhaltsadaption. Die vorgestellten Platzierungsalgorithmen sind deshalb grundsätzlich nicht adaptivitätssensitiv. Zwar können, wie in Abschnitt 1.3 dargestellt, auch nicht-adaptivitätssensitive Platzierungsmechanismen in einer Umgebung eingesetzt werden, in der Inhalte an die Eigenschaften und Fähigkeiten der Endgeräte und Netzwerkanbindungen adaptiert werden müssen, aber diese Platzierungsmechanismen sind nicht in der Lage, statische und dynamische Adaption flexibel miteinander zu kombinieren. Für traditionelle Einsatzszenarien, in denen Inhaltsadaption keine Rolle spielt, konnte (in einem statischen Systemmodell) gezeigt werden, dass global koordinierte Platzierung deutliche Vorteile gegenüber einer ausschließlich lokal koordinierten Vorgehensweise mit sich bringen kann. Greedy-Algorithmen haben sich für die global koordinierte Platzierung von Replikaten in CDNs bewährt. Deshalb wird in der vorliegenden Arbeit ein global koordinierender Greedy-Algorithmus zur adaptivitätssensitiven Platzierung von Replikaten in A-CDNs favorisiert.

Die Idee der Berücksichtigung der Adaptierbarkeit von Inhalten bei der Platzierung von Replikaten ist bereits in vorangegangenen Arbeiten zum Caching von Multimediainhalten umgesetzt worden. Viele dieser Arbeiten berücksichtigen aber nur Adaption eines einzelnen Qualitätsmerkmals, z.B. durch Weglassen einzelner Schichten eines hierarchisch kodierten Videostroms. Eine Adaption in verschiedenen Qualitätsdimensionen, wie sie in der vorliegenden Arbeit diskutiert wird, wird nur in ausgewählten Publikationen adressiert. Darüber hinaus untersucht keine der Arbeiten auf diesem Gebiet eine global koordinierte Platzierung von Replikaten. Die Mehrzahl der Arbeiten beschränkt sich auf die Betrachtung eines einzelnen, autonomen Proxy-Caches. Einzig [CYH00] betrachtet ein Szenario mit mehreren, verteilten Proxies. Eine Koordination der Platzierungsmechanismen der einzelnen Proxies findet aber nicht statt.

Der Aspekt der Inhaltsadaption wird in der wissenschaftlichen Literatur sehr umfangreich beschrieben. Dabei wird ein breites Feld von Anwendungsbereichen, Adaptionenzielen und Adaptionmechanismen abgedeckt und verschiedene Aspekte der Inhaltsadaption untersucht. Im Kontext von Platzierungsmechanismen für A-CDNs ist insbesondere die Arbeit von Lum und Lau [LL02] zur Kombination von statischer und dynamischer Inhaltsadaption interessant. Die Auswahl, welche Repräsentationen statisch erzeugt und gespeichert werden, erfolgt dabei mittels eines Greedy-Algorithmus auf Grundlage von Abschätzungen des Zugriffsverhaltens der Clients. Dieses Vorgehen ist den Verfahren zur adaptivitätssensitiven Platzierung von Replikaten in A-CDNs, die in der vorliegenden Arbeit entwickelt werden, sehr ähnlich. Allerdings wird in [LL02] die Möglichkeit der Verteilung und Replikation der verschiedenen Repräsentationen nicht berücksichtigt.

Forschungsarbeiten zur Beschreibung und Optimierung von Adaptionvorgängen durch Adaptionspfade sind für die vorliegende Arbeit ebenfalls von besonderer Relevanz, da die Inhaltsadaption in A-CDNs als pfadbasierte Adaption beschrieben wird. Die Verfahren zur Komposition von Adaptionspfaden in dieser Arbeit bauen auf der Grundidee des Active-Pipes-Ansatzes auf, das Problem auf die Suche nach dem kürzesten Pfad in einem gewichteten gerichteten Graphen abzubilden. Allerdings müssen Adaptionspfade in A-CDNs bezüglich ihrer Kosten und Qualität optimiert werden und es kommen verschiedene Replikate als Ausgangspunkt und verschiedene Repräsentationen als Zielpunkt eines Adaptionspfades in Frage. Deshalb werden die Verfahren aus dem Active-Pipes-Projekt entsprechend erweitert. Außerdem wird in der vorliegenden Arbeit, anders als beim Active-Pipes-Ansatz, nicht zwischen logischem und physischem Adaptionspfad unterschieden, sondern beide werden in einem Optimierungsschritt ermittelt.

Industriekonsortien, wie das ICAP-Forum und die OPES-Arbeitsgruppe der IETF, beschäftigen sich mit der Entwicklung und Standardisierung von Protokollen zur Integration von Inhaltsadaption in CDNs. Die Koordination der Inhaltsadaption mit den Platzierungsmechanismen im CDN wird durch die Arbeit dieser Gruppen jedoch nicht adressiert.

Eine Alternative zur Inhaltsadaption in A-CDNs besteht in der Auslagerung ganzer Anwendungslogik in ein CDN. Dieser als Application Delivery Network (ADN) bezeichnete Ansatz ist gut geeignet, wenn eventuell personalisierte WWW-Seiten dynamisch erzeugt und adaptiert werden müssen. Eine Wiederverwendung der dynamisch generierten Inhalte wird in ADNs aber nicht betrachtet. Deshalb bietet sich an, weniger dynamische Inhaltsobjekte einer adaptiven WWW-Anwendung, wie Multimediaobjekte, durch A-CDNs, die eine Wiederverwendung der adaptierten Repräsentationen erlauben, zu verteilen und auszuliefern. Somit können sich ADNs und A-CDNs bei der Realisierung skalierbarer adaptiver WWW-Anwendungen ergänzen.

Kapitel 3

Grundzüge und Modellierung von A-CDNs

Adaptive Content Distribution Networks (A-CDNs) sind CDNs, die Inhalte ausliefern, deren Repräsentation an die heterogenen Eigenschaften und Fähigkeiten der Clients und deren Netzwerkanbindung adaptiert sind. Im ersten Teil dieses Kapitels (Abs. 3.1) sollen die grundlegenden Konzepte von A-CDNs und Überlegungen zu deren Realisierung dargestellt werden. Darauf aufbauend wird in Abschnitt 3.2 ein formales Modell für A-CDNs entwickelt und auf Grundlage dieses Modells das Problem der Platzierung von Replikaten in A-CDNs formalisiert.

3.1 Basiskonzepte und Realisierung

3.1.1 Basiskonzepte

A-CDNs liefern Inhalte aus, deren Repräsentation an die Eigenschaften und Fähigkeiten der Clients und deren Netzwerkanbindung adaptiert sind. Adaption an den Nutzer und die aktuelle Situation wird in dieser Arbeit nicht berücksichtigt, da sie über das Konzept einer anwendungsübergreifenden Infrastruktur zur skalierbaren Auslieferung von Inhalten hinausgehen. Es ist festzustellen, dass heute eine zunehmende Zahl von WWW-Diensten dynamisch personalisierte Hypertext- und Hypermedia-Dokumente generieren, die für Replikation in einem A-CDN ungeeignet sind [BJS02]. Multimediaobjekte (*Assets*), wie Bilder, Audio- und Videoströme, hingegen sind typischerweise nicht personalisiert, nur die Auswahl dieser Medienobjekte durch das referenzierende Hypertext- oder Hypermedia-Dokument wird ggf. an ein Nutzermodell angepasst. Folglich sind Multimediaobjekte perfekt geeignet, in einem A-CDN repliziert und adaptiert zu werden. Aus diesem Grund wird in den Betrachtungen im Folgenden davon ausgegangen, dass A-CDNs zur Verteilung und Auslieferung von Multimediaobjekten eingesetzt werden. Personalisierte Hypertext- und Hypermedia-Dokumente sollen entweder auf den Originalservern oder durch die Nutzung von Application Delivery Networks (ADNs) verteilt generiert werden.

Eine Anfrage eines Client umfasst einen *Uniform Resource Identifier* (URI) und eine Spezifikation der Eigenschaften und Fähigkeiten des Endgerätes und der Netzwerkanbindung des Client. Der URI identifiziert das angeforderte logische Inhaltsobjekt. Die Identität der logischen Inhaltsobjekte sei an deren URI gebunden. Die Spezifikation der Eigenschaften und Fähigkeiten des Endgerätes und der Netzwerkanbindung kann zum Beispiel auf Basis von CC/PP-[KRW⁺04] oder CSCP-Profilen [BHH04] erfolgen. Eigenschaften und Fähigkeiten des Endgerätes definieren die Menge an Repräsentationen, die der Client akzeptieren kann. Dabei wird gemäß Abschnitt 1.2 davon ausgegangen, dass ein Client mehrere verschiedene Repräsentationen eines Inhaltsobjektes akzeptieren kann, wobei sich die Qualität verschiedener Repräsentationen gegebenenfalls unterscheidet. Die Qualität der verschiedenen Repräsentationen kann durch die Eigenschaften des Client mitbestimmt sein, da Clients mit unterschiedlichen Eigenschaften die Qualität der gleichen Repräsentation unterschiedlich bewerten können. Zum Beispiel ist die

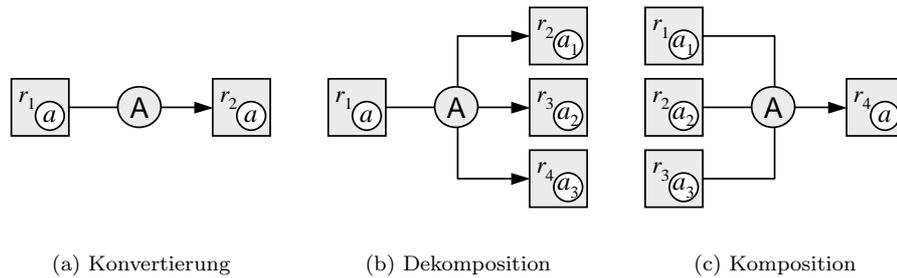


Abbildung 3.1: Konvertierung und Strukturadaption

Qualität (bzgl. der räumlichen Auflösung) eines Videos mit einer Auflösung von 240×180 Bildpunkten aus der Sicht eines Client mit hochauflösendem Bildschirm eher als gering einzuschätzen. Auf einem Pocket PC (Displayauflösung 240×320 Bildpunkte) hingegen ist besagte Auflösung die größte, die sich vollständig und ohne Verzerrungen darstellen lässt. Aus Sicht des Nutzers eines Pocket PC ist die Qualität bzgl. der räumlichen Auflösung also optimal. Die Angabe der Eigenschaften der Netzwerkanbindung gibt Aufschluss über Kosten, Verzögerungen, Datenratenbeschränkung usw. bei der Übertragung von Inhalten an den Client. Darüber hinaus kann auch das subjektive Qualitätsempfinden der Nutzer von der Netzwerkanbindung des Client abhängen, z.B. ist die Toleranz mobiler Nutzer (mit mobilen Geräten in zellularen Mobilfunknetzen) gegenüber Antwortzeitverzögerungen größer als bei Festnetznutzern.

Aus der Sicht der Nutzer ist die Qualität ausgelieferter Inhalte zu maximieren. Auch die Inhaltsanbieter haben ein Interesse an einer Maximierung der Qualität. Eine Studie des Verhaltens von Nutzern von Shopping-Angeboten im Internet am Georgia Tech [oV98] hat gezeigt, dass der Verkaufserfolg von E-Commerce-Anbietern unmittelbar durch Qualitätsfaktoren, wie die Antwortzeit, beeinflusst wird. Ma et al. [MBC⁺00] argumentieren, dass weitere Qualitätsfaktoren, wie ein ästhetisches Erscheinungsbild, ebenfalls zu mehr Nutzerzufriedenheit, somit zu mehr wiederkehrenden Nutzern und im Endeffekt zu mehr Verkäufen von E-Commerce-Anbietern und höheren Werbeeinnahmen für werbefinanzierte WWW-Angebote führen. Qualität wirkt sich also unmittelbar auf den wirtschaftlichen Erfolg des Diensteanbieters aus. Folglich ist davon auszugehen, dass die Zahlungsbereitschaft der Inhaltsanbieter als Kunden des A-CDN von der Qualität der Inhaltsauslieferung abhängt.

Die Inhaltsadaption in A-CDNs soll gemäß dem Paradigma pfadbasierter Adaption erfolgen. In der vorliegenden Arbeit werden die möglichen Adaptionoperationen in einem Adaptionspfad auf Konvertierungsoperationen beschränkt. Konvertierungsoperationen wandeln ein Inhaltsobjekt von einer Repräsentation in eine andere Repräsentation um (Abb. 3.1(a)). Strukturadaptionen, die ein Inhaltsobjekt in mehrere unabhängige Objekte dekomponieren (Abb. 3.1(b)), z.B. Fragmentierung von WWW-Seiten [GBZS01], oder ein Objekt aus mehreren Objekten komponieren (Abb. 3.1(c)), z.B. Komposition eines hierarchisch kodierten Videostroms aus einer Basisschicht und separaten Erweiterungsschichten [KHRR01], werden nicht berücksichtigt. Die Mannigfaltigkeit der Konvertierungsoperationen, die in A-CDNs eingesetzt werden können, ist nicht eingeschränkt. Es können beliebige Eigenschaften der Inhaltsobjekte in verschiedenen Qualitätsdimensionen adaptiert werden (multidimensionale Adaption).

Der Dienst der Auslieferung von Inhalten verursacht Kosten, wie Netzwerkkosten für den Datentransfer über das Internet oder zu den Clients und Kosten für die Ausführung von Adaptionoperationen. Ziel eines A-CDN-Anbieters ist es, diese Betriebskosten zu minimieren. Bei der Betrachtung der Kosten in der vorliegenden Arbeit wird nicht unterschieden, wem die Kosten entstehen, dem A-CDN-Betreiber, dem Inhaltsanbieter (Betreiber der Originalserver) oder dem Nutzer. Als Optimierungsziel wird die Minimierung der Gesamtkosten (Summe aller Kosten) berücksichtigt. Für eine differenziertere Betrachtung der Kosten müsste ein konkretes Geschäfts- und Abrechnungsmodell zugrunde gelegt werden. Darüber hinaus müssen die

Wechselwirkungen zwischen den Kosten für die Nutzer und deren Nutzungsverhalten sowie die Auswirkungen des Nutzungsverhalten auf die Zahlungsbereitschaft der Inhaltsanbieter als Kunden des A-CDN-Betreibers modelliert werden. Das geht über den Rahmen dieser Arbeit hinaus und soll nicht betrachtet werden.

Dem Ziel der Kostenminimierung steht das Ziel der Qualitätsmaximierung gegenüber. Deshalb wird in der vorliegenden Arbeit ein kosten-qualitäts-optimierter Ansatz zur Platzierung von Replikaten und Auslieferung von Inhalten verfolgt.

3.1.2 Realisierung

Entsprechend dem CDN-Referenzmodell ist ein A-CDN aus den Surrogaten, auf denen Inhalte repliziert werden, einem Verteilsystem zur Verteilung von Replikaten, einem Anfrageweiterleitungs- und Auslieferungssystem zur Weiterleitung von Anfragen an geeignete Surrogate im A-CDN und Auslieferung der Inhalte an die Clients, einem Monitoringsystem zur Überwachung des aktuellen Systemzustandes und einem Accountingsystem zur Protokollierung

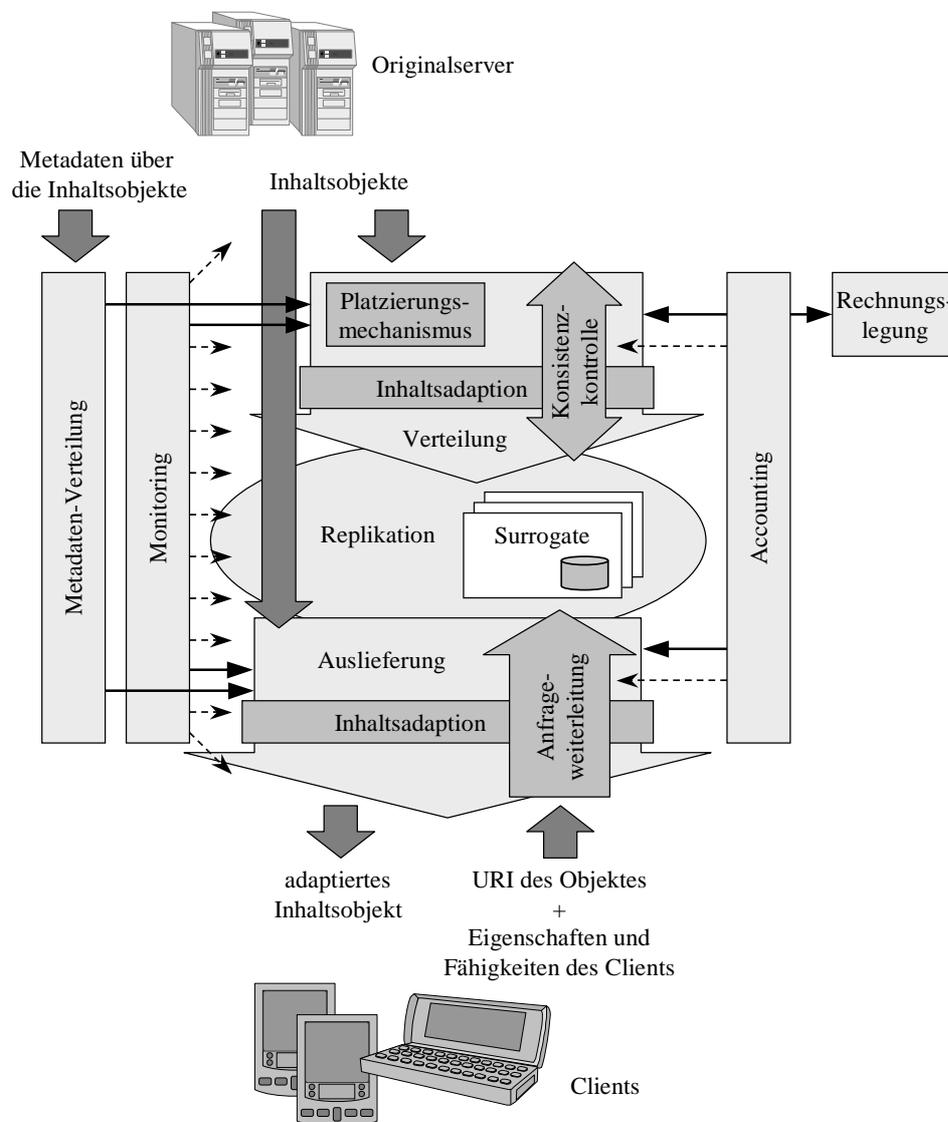


Abbildung 3.2: Architektur von A-CDNs

der Anfragen und weiterer zur Abrechnung relevanter Aktivitäten aufgebaut. Darüber hinaus wird in einem A-CDN ein Metadaten-Verteilssystem benötigt, durch welches dem Verteilssystem sowie dem Anfrageweiterleitungs- und Auslieferungssystem Metadaten über die Inhaltsobjekte zur Verfügung gestellt werden. Ein Gesamtbild der Architektur von A-CDNs, entsprechend den Ausführungen in diesem Abschnitt, ist in Abbildung 3.2 dargestellt.

Replikation

A-CDNs replizieren Inhalte in Surrogaten. Ein Surrogat ist ein Rechnerknoten im Internet, der Speicherkapazität und geeignete Speicherverwaltungsmechanismen zur Replikation von Inhalten sowie eine Netzwerkschnittstelle zur Auslieferung replizierter Inhalte zur Verfügung stellt. Typischerweise werden durch einen CDN-Anbieter mehrere Surrogate an einem Standort in einem Rechenzentrum (*Data Center*) betrieben. Die Surrogate in einem Rechenzentrum arbeiten dabei in einem Cluster zusammen, um eine transparente Lastverteilung zu erreichen. Ein Cluster von Surrogaten erscheint virtuell wie ein Surrogat. Deshalb soll bei den abstrakten Betrachtungen im Folgenden ein Cluster von Surrogaten in einem Rechenzentrum wie ein einziges Surrogat betrachtet werden, dessen Kapazität der kumulierten Kapazität der Surrogate im Cluster entspricht.

Inhaltsadaption

Die Inhaltsadaption in einem A-CDN wird durch Adaptionskomponenten realisiert, die jeweils eine elementare Adaptionsoption in einem Adaptionspfad ausführen. Die Adaptionskomponenten können sowohl für die Inhaltsadaption bei der Auslieferung als auch bei der Verteilung der Inhalte verwendet werden (Abb. 3.3(a)). Somit können Adaptionskomponenten gleichzeitig Teil des Verteil- und des Auslieferungssystems sein. Die Adaptionskomponenten können auf den Surrogaten, aber auch auf separaten Adaptionsservern installiert sein. Auf diese Weise können auch spezialisierte Hardware-Plattformen für die Inhaltsadaption genutzt werden. In der Arbeit wird davon ausgegangen, dass die Adaptionsserver innerhalb der Rechenzentren der CDN-Anbieter betrieben werden. Befinden sich Adaptionsserver und Surrogate im selben Hochgeschwindigkeits-LAN innerhalb eines Rechenzentrums, werden die Kommunikationskosten und Verzögerungen zwischen Surrogaten und Adaptionsservern minimiert. Auch administrative Gründe sprechen dafür, Surrogate und Adaptionsserver in Rechenzentren zu konzentrieren. Sind die Kommunikationskosten und Verzögerungen zwischen Surrogaten und Adaptionsservern in einem Rechenzentrum vernachlässigbar, so können sie virtuell als ein Knoten betrachtet werden. Dementsprechend sollen bei den abstrakten Betrachtungen im Folgenden alle Surrogate und Adaptionsserver innerhalb des selben Rechenzentrums wie ein Surrogatknoten betrachtet werden, auf dem alle Adaptionskomponenten aller Adaptionsserver des Rechenzentrums installiert sind (Abb. 3.4).

Zusätzlich zu den Adaptionskomponenten in den Surrogaten bzw. Rechenzentren des A-CDN können auch die Originalserver bzw. die Rechenzentren der Inhaltsanbieter (in denen die Originalserver betrieben werden) über Adaptionskomponenten verfügen. Diese können sowohl für die Inhaltsadaption bei der Verteilung der Inhalte auf die Surrogate als auch für die dynamische Adaption bei der Auslieferung von Inhalten von den Originalservern genutzt werden (Abb. 3.3(b)). Statische Inhaltsadaption in den Originalservern wird in dieser Arbeit nicht betrachtet.

Dynamische Inhaltsadaption bei der Auslieferung von Inhalten erfordert hohe Rechenkapazitäten und gegebenenfalls spezialisierte Hardware und somit eine kostenintensive Infrastruktur. Das CDN-Geschäftsmodell [BEH⁺01] hingegen geht davon aus, dass der Inhaltsanbieter nicht in eine kostenintensive Infrastruktur investiert, sondern die Infrastruktur für die Auslieferung durch den CDN-Anbieter bereitgestellt und betrieben wird. Deshalb sind in der Praxis Lösungen zu bevorzugen, bei denen die Originalserver nicht über Adaptionmöglichkeiten verfügen. Die Adaption von Inhalten, die von einem Originalserver ausgeliefert werden müssen, erfolgt dann durch Adaptionskomponenten im A-CDN.

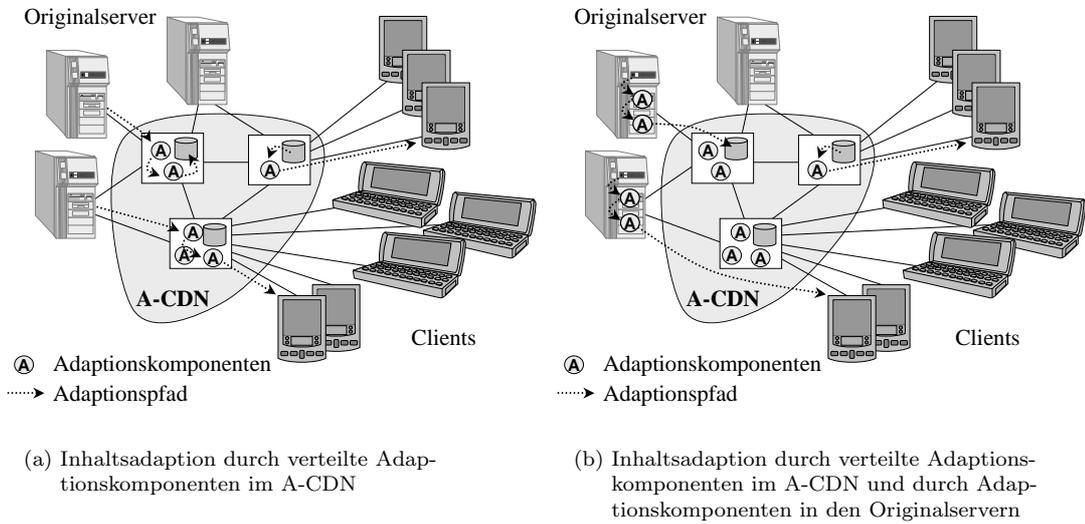


Abbildung 3.3: Ort der Inhaltsadaption

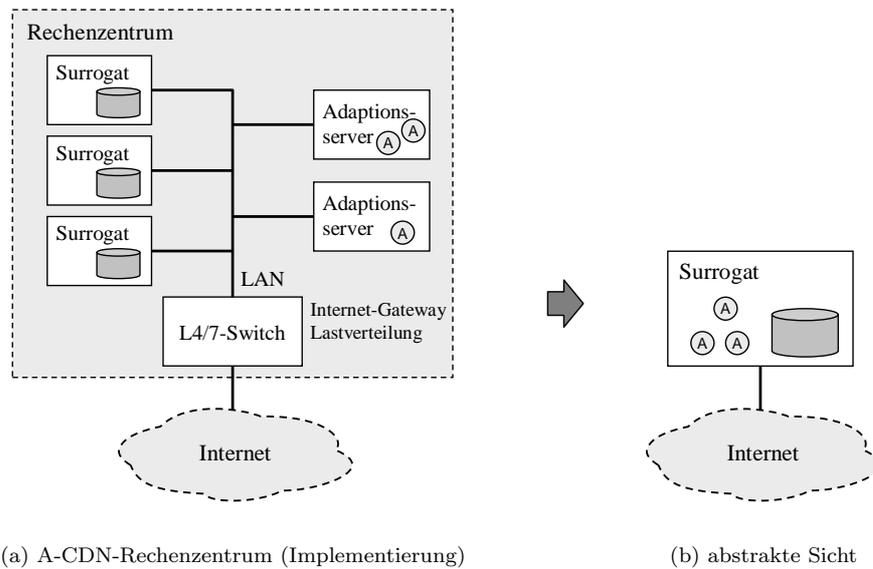


Abbildung 3.4: Abstraktion von A-CDN-Rechenzentren

Verteilung

Das Verteilsystem eines A-CDN kann sowohl das Internet als auch ein privates Netzwerk für die Verteilung von Inhalten nutzen. Während bei der Benutzung des Internets zur Verteilung auf die Investition in ein privates Netzwerk verzichtet werden kann, erlauben private Netze die Verkehrsbedingungen im Verteilsystem besser zu kontrollieren und gegebenenfalls Broadcast- oder Multicast-Fähigkeiten des zugrunde liegenden Netzes zu nutzen.

In einem A-CDN können verschiedene Repräsentationen eines Inhaltsobjektes repliziert werden. Diese müssen bei der Verteilung durch Inhaltsadaption gewonnen werden. Die Ermittlung der Adaptionpfade zur Generierung der verschiedenen Repräsentationen der Replikat aus der Repräsentation des Originalreplikates ist Aufgabe des Verteilsystems eines A-CDN.

Die Bestimmung einer Platzierung von Replikaten in einem A-CDN ist ebenfalls Aufgabe des Verteilsystems. In der vorliegenden Arbeit wird aus den in Abschnitt 1.4 dargelegten Gründen ein statisch optimierender Platzierungsalgorithmus entwickelt, der eine globale Sicht auf das System voraussetzt. Ein solcher Algorithmus kann durch eine zentrale Instanz des Verteilsystems, einen Platzierungsmanager, implementiert werden, welchem durch das Monitoring- und das Accountingsystem eine globale Sicht auf das System und die Anfragelast zur Verfügung gestellt werden. Im Anschluss an die Ausführung des Platzierungsalgorithmus stellt der Platzierungsmanager die ermittelte Platzierung von Replikaten allen Surrogaten zur Verfügung. Die Skalierbarkeit einer solchen Lösung ist jedoch eingeschränkt. Außerdem führt ein Ausfall der zentralen Instanz zu einem Ausfall des gesamten Platzierungsmechanismus. Der Betrieb der anderen Funktionen des A-CDN (Auslieferung, Verteilung entsprechend einer zuvor bestimmten Platzierung, Accounting) ist durch einen solchen Ausfall jedoch nicht eingeschränkt. Ist der Platzierungsmanager in mehrfacher Ausführung realisiert, so kann im Fehlerfall auf ein Notssystem umgestellt werden. Sinnvoller als ein zentraler Ansatz ist jedoch eine verteilte Realisierung des Platzierungsalgorithmus. Deshalb wird in der Arbeit auch eine Möglichkeit zur verteilten Realisierung des adaptivitätssensitiven Platzierungsalgorithmus vorgeschlagen. Dabei kooperieren verteilte Platzierungsmanager, die in allen Surrogaten bzw. in allen Rechenzentren des A-CDN installiert sind, bei der Ermittlung einer Platzierung von Replikaten.

Die Implementierung einer Platzierung von Replikaten in einem A-CDN, d.h. die Verteilung der Replikat auf die Surrogate, kann im Voraus oder bei Bedarf erfolgen. Die erste Variante ermöglicht, dass die Verteilung zu Zeitpunkten geringer Last ausgeführt wird und die Replikat beim Zugriff eines Client bereits in den Surrogaten vorliegen. Somit entstehen beim Zugriff keine weiteren Verzögerungen, weil Inhalte erst verteilt werden müssen. Außerdem können bei der Verteilung im Voraus Broadcast- oder Multicast-Fähigkeiten des Verteilsystems ausgenutzt werden. Wird die Platzierung jedoch häufig zu Zeiten hoher Last angepasst, führt eine sofortige Verteilung im Voraus zu zusätzlichen Lastspitzen (Adaptionlast und Netzwerklast).

Bei der Verteilung bei Bedarf wird die Erzeugung eines Replikates bis zum ersten Zugriff eines Client auf dieses Replikat verzögert. Somit verteilt sich die Last der Verteilung zeitlich, Lastspitzen werden vermieden. Allerdings führt dieses Verfahren zu zusätzlichen Antwortzeitverzögerungen beim ersten Zugriff auf das Replikat, da dieses erst vom Originalserver angefordert und gegebenenfalls adaptiert werden muss.

Durch Kombination der beiden Vorgehensweisen lassen sich die Vorteile der Verfahren sinnvoll kombinieren: Replikat werden im Voraus erzeugt, allerdings wird die Erzeugung auf Phasen geringerer Last verzögert. Soll auf ein Replikat, welches aufgrund der verzögerten Verteilung noch nicht erzeugt wurde, zugegriffen werden, so wird es bedarfsgetrieben erzeugt.

Bei der Diskussion von Platzierungsmechanismen im Folgenden werden keine zusätzlichen Antwortzeitverzögerungen berücksichtigt, die durch Verteilung bei Bedarf entstünden. Das entspricht einem A-CDN, welches auf Verteilung im Voraus basiert. Allerdings ist davon auszugehen, dass die Ergebnisse auch auf A-CDNs mit bedarfsgetriebener Verteilung übertragbar sind, weil die zusätzlichen Verzögerungen jeweils nur beim ersten Zugriff auf ein Replikat auftreten und damit für das Gesamtergebnis vernachlässigbar sind. Die Kosten für die Verteilung der Inhalte bleiben ebenfalls unberücksichtigt. Diese werden gegenüber den Auslieferungskosten als vernachlässigbar angesehen. Bei der Modellierung werden also keine Annahmen über das verwendete Verteilsystem zugrunde gelegt.

Die Wahrung der Konsistenz der verschiedenen Replikate ist ebenfalls Aufgabe des Verteilsystems. Die Clients eines A-CDN greifen nur lesend auf Inhalte zu. Änderungen erfolgen ausschließlich durch die Inhaltsanbieter auf den Originalreplikaten, die als Primärkopien fungieren. Die Serialisierbarkeit der Änderungen ist durch den Primärkopie-Ansatz sichergestellt. Konsistenzkontrolle für strikte Konsistenz kann durch

- Validierung (Überprüfen der Aktualität bei jedem Zugriff, [SSPS04, FGM⁺99]),
- proaktive Verteilung von Aktualisierungen (*Push* vom Originalserver, [SSPS04]) oder
- Invalidierung (Änderungsnachricht vom Originalserver ohne Aktualisierung, [SSPS04])

erfolgen. Abgeschwächte zeitbasierte Konsistenz kann durch Spezifikation einer Gültigkeitsdauer für jedes Replikat realisiert werden (vgl. [SSPS04, FGM⁺99]). Sind Änderungen selten¹, verursacht Konsistenzkontrolle durch Validierung einen unnötigen Kommunikationsmehraufwand, die proaktive Verteilung von Aktualisierungen hingegen den geringsten Aufwand. Die Kosten für die Aktualisierung sind bei seltenen Änderungen vernachlässigbar. Deshalb wird bei der Diskussion von Mechanismen zur Platzierung von Replikaten in A-CDNs im Folgenden keine Konsistenzkontrolle berücksichtigt.

Anfrageweiterleitung, Auslieferung

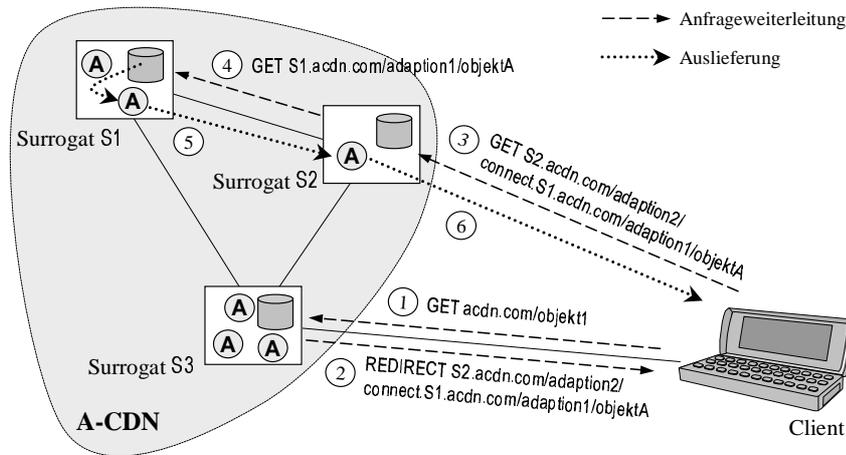
Durch die Anfrageweiterleitung wird eine Anfrage eines Client an ein geeignetes Surrogat oder an einen Originalserver dirigiert, von welchem das angeforderte Inhaltsobjekt durch das Auslieferungssystem des A-CDN an den Client ausgeliefert und dabei gegebenenfalls adaptiert werden kann. In dieser Arbeit wird grundsätzlich die Möglichkeit in Betracht gezogen, auch Originalserver bei der Auslieferung von Inhalten mit einzubeziehen. Das heißt, die betrachteten A-CDNs sind relaying CDNs (vgl. Abs. 1.1.4).

Anfrageweiterleitung umfasst die Weiterleitungsstrategie und die Weiterleitungsmechanismen [SSPS04]. Die Weiterleitungsstrategie entscheidet, welches Replikat zur Befriedigung einer Anfrage verwendet wird und wie dieses bei der Auslieferung gegebenenfalls an die Bedürfnisse des Client adaptiert wird. Das heißt, die Weiterleitungsstrategie eines A-CDN ist für die Komposition der Adaptionspfade zur Beantwortung der Anfragen der Clients verantwortlich.

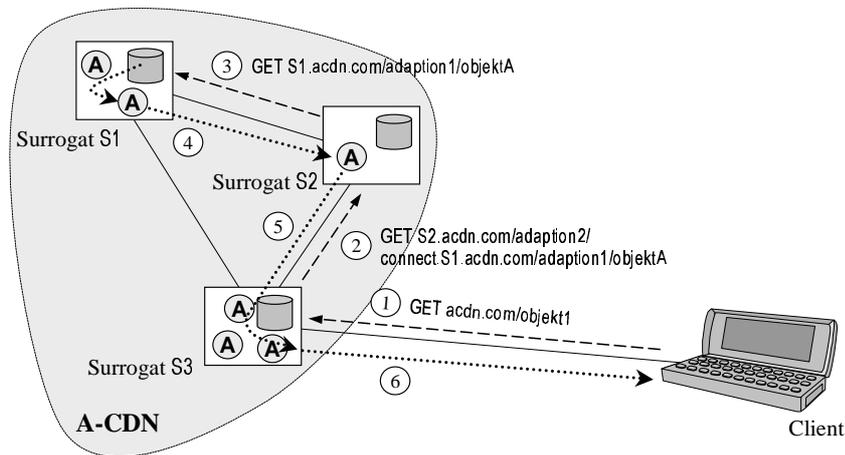
Die Weiterleitungsmechanismen sind für die transparente Weiterleitung der Anfragen an ein gemäß Weiterleitungsstrategie geeignetes Surrogat des A-CDN verantwortlich. In einem A-CDN müssen den Weiterleitungsmechanismen neben dem URI des angeforderten Inhaltsobjektes auch Informationen über Eigenschaften und Fähigkeiten des Endgerätes und der Netzwerkanbindung zur Verfügung stehen. Diese Informationen sind nur in der Anwendungsschicht verfügbar.

Eine einfache Möglichkeit zur Anfrageweiterleitung in A-CDNs besteht in der Nutzung von Umleitungsmechanismen des Anwendungsschicht-Protokolls, wie der HTTP-Umleitung. Hierbei wird die Anfrage des Client zunächst an ein beliebiges Surrogat (sinnvollerweise das nächstgelegene) gesendet (Abb. 3.5(a)). Dieses Surrogat wertet alle Informationen in der Anfrage aus und sendet dann eine HTTP-Antwort (Statuscode „307 Temporary Redirect“ [FGM⁺99]) mit dem URI des Surrogates, welches das letzte im Adaptionspfad ist, an den Client. Der Client baut daraufhin eine neue Verbindung zu diesem Surrogat auf, sendet ihm die Anfrage und erhält die Antwort. Im URI kann bereits der komplette Adaptionspfad für die Auslieferung des Objektes spezifiziert sein. Andere Protokolle, wie das Real Time Streaming Protokoll (RTSP) zur Auslieferung von Video- und Audioströmen [SRL98], stellen Mechanismen ähnlich der HTTP-Umleitung zur Verfügung.

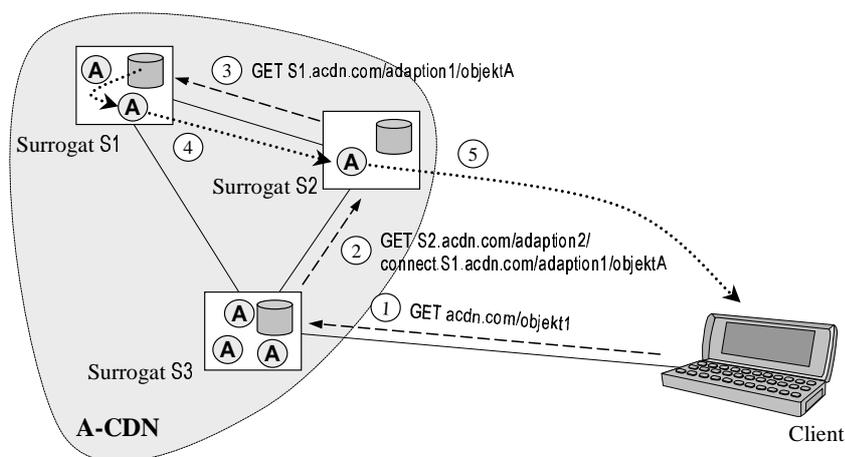
¹Empirische Untersuchungen der Änderungsrate von WWW-Objekten im HTTP-Verkehr [DFKM97] haben gezeigt, dass sich Bilder im WWW fast nie ändern. Auch wenn vergleichbare Untersuchungen der Änderungsrate von anderen Multimediaobjekten, wie Audio- und Videostreamen, nicht publiziert wurden, scheint diese Beobachtung generell auf Multimediaobjekte zutreffend zu sein. Bei Änderungen von Multimediainhalten wird im Allgemeinen nicht das Inhaltsobjekt geändert, sondern ein neues Inhaltsobjekt mit einem anderen URI erzeugt. Dieses Verfahren wird in [SSPS04] unter dem Namen Versionierung auch als Variante zur Konsistenzkontrolle vorgeschlagen. Akamai verwendet Versionierung zur Konsistenzkontrolle in seinem CDN [DMP⁺02].



(a) HTTP-Umleitung



(b) Proxy im Adaptionpfad



(c) Dreieckskommunikation

Abbildung 3.5: Anfrageweiterleitung in A-CDNs

Die Anfrageweiterleitung mittels der Umleitungsmechanismen des Anwendungsschicht-Protokolls ist einfach zu implementieren, hat aber signifikante Antwortzeitverzögerungen durch den zusätzlichen Verbindungsaufbau des Client zur Folge. Dies gilt insbesondere für den Verbindungsaufbau über die schmalbandige, drahtlose Netzwerkanbindung mobiler Clients. Eine bessere Alternative zur Anfrageweiterleitung ist, wenn das Surrogat, welches ursprünglich die Anfrage des Client erhält, als Proxy agiert und die Anfrage selbst an das letzte Surrogat im Adaptionpfad weiterleitet (Abb. 3.5(b)). Die Antwort wird bei dieser Variante über den Proxy an den Client geleitet. Dadurch entfällt nicht nur der zweite Verbindungsaufbau über die potentiell schmalbandige Verbindung zum Client. Die Surrogate können untereinander auch persistente Verbindungen [FGM⁺99] unterhalten, so dass der zweite Verbindungsaufbau komplett entfällt. Das Proxy-Verfahren kann noch weiter verbessert werden, wenn die Antwort nicht über ein als Proxy agierendes Surrogat geleitet, sondern vom letzten Surrogat im Adaptionpfad direkt an den Client gesendet wird (Dreieckskommunikation, Abb. 3.5(c)).

Bei allen dargestellten Weiterleitungsmechanismen kann der Client auch anstatt des nächstgelegenen Surrogates zunächst einen speziellen Knoten des A-CDN kontaktieren, der nur für die Anfrageweiterleitung zuständig ist, selbst aber keine Anfragen beantwortet (Weiterleitungsknoten).

Die oben beschriebenen Weiterleitungsmechanismen setzen voraus, dass der Client zunächst ein beliebiges, möglichst das nächstgelegene, Surrogat oder einen Weiterleitungsknoten des A-CDN kontaktiert, welcher einen Adaptionpfad zur Beantwortung der Anfrage ermittelt und anschließend eine modifizierte Anfrage zur Beantwortung weiterleitet. Hierzu muss die ursprüngliche Anfrage des Client zunächst transparent für den Nutzer an ein solches Surrogat bzw. einen solchen Weiterleitungsknoten geleitet werden. Hierzu können beliebige inhaltsunabhängige (*content-blind*) Weiterleitungsmechanismen [RS02] eingesetzt werden. Inhaltsunabhängige Weiterleitungsmechanismen sind solche, bei denen bei der Weiterleitung keine Informationen der Anwendungsschicht berücksichtigt werden. DNS-basierte Verfahren sind hierfür besonders geeignet, weil sie einfach und performant zu implementieren sind. DNS-basierte inhaltsunabhängige Anfrageweiterleitung in A-CDNs kann wie folgt realisiert werden:

1. Der Client sendet eine Anfrage nach der IP-Adresse des Host im URI des gesuchten Objektes an seinen lokalen DNS-Server.
2. Dieser sendet die Anfrage an den autorisierten DNS-Server des A-CDN weiter (dazu muss der Inhaltsanbieter die Verantwortung für die Domäne an den A-CDN-Betreiber delegiert haben).
3. Der DNS-Server des A-CDN schätzt aus der Lokation des Client-DNS-Servers die Lokation des Client.
4. Auf Grundlage des Modells des Internets, welches dem DNS-Server durch das Monitoringsystem des A-CDN zur Verfügung gestellt wird, ermittelt der DNS-Server das nächstgelegene Surrogat und gibt dessen IP-Adresse an den Client zurück.

Die Skalierbarkeit dieses Verfahrens lässt sich durch Verwendung einer mehrstufigen DNS-Namensauflösung wie im CDN von Akamai [DMP⁺02] erhöhen.

Bei der Modellierung von A-CDNs in Abschnitt 3.2 wird vom Anfrageweiterleitungssystem des A-CDN abstrahiert. Ein A-CDN, welches ein beliebiges der aufgezeigten Verfahren verwendet, kann entsprechend Abschnitt 3.2 modelliert werden.

Monitoring

Die Algorithmen zur Anfrageweiterleitung und Platzierung von Replikaten in einem A-CDN, die in der vorliegenden Arbeit vorgeschlagen werden, gehen von einem vollständigen Modell des Systemzustandes aus. Die Ermittlung des Systemzustandes ist Aufgabe des Monitoringsystems.

Die Implementierung des Monitoringsystems geht über den Fokus der Arbeit hinaus und soll hier nicht diskutiert werden. Die Anforderungen in Bezug auf das Monitoring des Internets entsprechen denen an das Monitoringsystem in herkömmlichen CDNs, so dass davon ausgegangen werden kann, dass die gleichen Verfahren wie in herkömmlichen CDNs zum Einsatz kommen können. Das Monitoring der Eigenschaften und Fähigkeiten der Endgeräte und deren Netzwerkanbindungen zur Modellierung der Clients erfolgt durch Auswertung der Informationen in den Anfragen. Deshalb ist in A-CDNs eine enge Integration von Monitoring- und Accountingsystem angebracht.

Accounting

Die Mechanismen zur statischen Platzierung von Replikaten in dieser Arbeit basieren auf der Kenntnis der zukünftigen Anfragelast. Deshalb muss das Accountingsystem eines A-CDN die Anfragen der Clients protokollieren, aus den protokollierten Informationen die zukünftige Anfragelast abschätzen und diese in Form der erwarteten mittleren Anfrageraten dem Platzierungsalgorithmus bereitstellen. Für die Abrechnung gegenüber den Kunden eines A-CDN ist fernerhin die Qualität der ausgelieferten Inhaltsobjekte und die Servicequalität der Auslieferung aufzuzeichnen.

Um diesen Aufgaben nachzukommen, kann das Accountingsystem durch je einen Agenten auf jedem Surrogat bzw. Weiterleitungsknoten des Anfrageweiterleitungssystems, das bzw. der die Anfragen der Clients entgegennimmt und einen Adaptionpfad zur Auslieferung bestimmt, implementiert werden. An dieser Stelle können auch die Eigenschaften und Fähigkeiten der Endgeräte und deren Netzwerkanbindungen erfasst werden, die mit der Anfrage (z.B. in Form eines CC/PP-Profiles) übermittelt werden und vom Monitoringsystem zur Modellierung der Clients benötigt werden.

Metadaten zu den Inhaltsobjekten

Um eine optimale Platzierung von Replikaten in einem A-CDN zu bestimmen, muss der Platzierungsalgorithmus in der Lage sein, die Kosten für die Ausführung der Adaptionoperationen sowie deren Auswirkungen auf die Objektmerkmale und die Qualität der einzelnen Inhaltsobjekte vorherzusagen. In gleicher Weise erfordert die Komposition optimaler Adaptionpfade durch die Weiterleitungsstrategie des Anfrageweiterleitungs- und Auslieferungssystems eine Vorhersage der Kosten und Auswirkungen der Adaptionoperationen. Eine solche Vorhersage kann entweder generisch oder aber spezifisch für jedes Inhaltsobjekt erfolgen. Während der generische Ansatz nur eine sehr grobe Abschätzung erlaubt, werden für die objektspezifische Vorhersage Metadaten über die Inhaltsobjekte benötigt (vgl. Abs. 3.2.1). Die Metadaten müssen durch den Inhaltsanbieter zum Beispiel als MPEG-7-Deskriptoren [MSS02] bereitgestellt und innerhalb des A-CDN den Platzierungsmanagern sowie den Knoten des Anfrageweiterleitungssystems, in denen die Adaptionpfade ermittelt werden (Surrogate bzw. Weiterleitungsknoten), zur Verfügung gestellt werden. Diese Funktionalität wird durch das Metadaten-Verteilssystem des A-CDN realisiert (Abb. 3.2 auf Seite 49). Da Metadaten sowohl bei der Verteilung der Inhalte als auch bei der Anfrageweiterleitung und Auslieferung benötigt werden, wird das Metadaten-Verteilssystem als eigenständiges Subsystem eines A-CDN definiert.

Das Metadaten-Verteilssystem muss die Metadaten A-CDN-weit mit minimalen Verzögerungen zur Verfügung stellen können, da sich Verzögerungen beim Zugriff auf Metadaten unmittelbar als Antwortzeitverzögerungen bemerkbar machen können. Deshalb ist eine verteilte Replikation der Metadaten im A-CDN notwendig. Dazu können im Metadaten-Verteilssystem Verfahren eingesetzt werden, die sich in herkömmlichen CDNs zur verteilten Replikation nicht adaptierbarer Inhalte bewährt haben.

Die Problematik der Verteilung von Metadaten über Inhaltsobjekte soll in dieser Arbeit nicht weiter vertieft werden. Bei der Modellierung im folgenden Abschnitt werden keine Verzögerungen, die beim Zugriff auf Metadaten entstehen, berücksichtigt.

3.2 Modellierung

In diesem Abschnitt wird ein formales Modell zur Beschreibung des Problems der Platzierung von Replikaten in A-CDNs entwickelt. Die formale Beschreibung umfasst die folgenden Teilm-odelle:

- ein **Systemmodell** (Abs. 3.2.1), welches die Menge an Inhaltsobjekten, die Surrogate des A-CDN, die Originalserver der Inhaltsanbieter, die Clients, die Netzwerkinfrastruktur und die Adaptionsoptionen modelliert. Da das Systemmodell neben dem A-CDN auch Originalserver, Clients und Netzwerkinfrastruktur beschreibt, welche nicht Teil des A-CDN sind, wird im Folgenden von einem System zum adaptiven Content Networking gesprochen.
- ein **Lastmodell** (Abs. 3.2.2), welches die Anfragen der Clients beschreibt.
- ein **Problemmodell** (Abs. 3.2.3), welches die Optimierungsziele beim Betrieb von A-CDNs in Form der Probleme der Platzierung von Replikaten und der Komposition von Adaptionspfaden formalisiert.

Das Modell verwendet als formale Beschreibungssprache die klassische Algebra und Prädikatenlogik erster Ordnung (ohne Sorten). Die Anwendung des formalen Modells auf ein konkretes Anwendungsszenario wird in Kapitel 5 illustriert.

3.2.1 Systemmodell

Ein System zum adaptiven Content Networking ist ein Tupel

$$\Sigma = (K, (T, M, S), sf, N, sc, P_0, mhc, qf, O, \wp), \quad (3.1)$$

bestehend aus

- einer Menge K von Inhaltsobjekten,
- einem Tripel (T, M, S) , bestehend aus einer Menge T von Typmerkmalen, einer Menge M von Medienqualitätsmerkmalen und einer Menge S von Servicequalitätsmerkmalen,
- einer Größenfunktion sf ,
- einer Menge N von Knoten, die Surrogate, Originalserver und Clientknoten einschließt,
- eine Funktion sc der Speicherkapazität der Surrogate,
- einer Relation P_0 als Platzierung von Originalreplikaten,
- einer Funktion mhc der Clientfähigkeiten,
- einer Qualitätsfunktion qf ,
- einer Menge O von Operationen und
- einer Zuordnung \wp von Kostenfunktionen zu Operationen.

In den folgenden Abschnitten werden die einzelnen Elemente des Systemmodells im Detail definiert und erläutert.

Inhaltsobjekte, Merkmale von Objektausprägungen

Ein logisches Inhaltsobjekt ist eine eindeutig identifizierbare logische Entität k , welche ein physisches oder abstraktes Objekt aus dem darzustellenden Sachverhalt der realen Welt repräsentiert. Gemäß Abschnitt 3.1.1 ist die Objektidentität an einen *Uniform Resource Identifier* (URI) [BLFIM98] des Inhaltsobjektes gebunden. Die Menge aller Inhaltsobjekte sei K .

Physisch kann ein Objekt verschiedene Ausprägungen besitzen. Die verschiedenen Ausprägungen können sich dabei durch den Ort, an dem sie vorliegen, und durch ihre Repräsentation unterscheiden. Die Repräsentation einer Objektausprägung wird durch die Ausprägung ihrer Objektmerkmale beschrieben. Im Folgenden werden Objektmerkmale in Typmerkmale (z.B. MIME-Inhaltstyp, Kodierung) und Medienqualitätsmerkmale (z.B. Bildgröße, Frame-Rate, Rauschabstand) unterschieden (vgl. [Mar03]). Während die Ausprägungen der Typmerkmale einzig Kompatibilität sicherstellt, hat die Ausprägung der Medienqualitätsmerkmale darüber hinaus Einfluss auf die Qualität, die durch den Nutzer wahrgenommen wird. Neben den Medienqualitätsmerkmalen des empfangenen Objektes bestimmen weitere Merkmale (z.B. Verzögerung, Jitter), die nicht dem Objekt selbst, sondern dem Vorgang der Auslieferung des Objektes zuzuordnen sind, die durch den Nutzer subjektiv wahrgenommene Qualität des Dienstes mit. Diese Merkmale werden im Folgenden als Servicequalitätsmerkmale bezeichnet.

Formal ist ein Typmerkmal ein Symbol t_ξ aus einem endlichen Symbolvorrat T_ξ . Das τ -Tupel

$$\mathbf{t} = (t_1, \dots, t_\tau) \quad (3.2)$$

der Ausprägungen aller Typmerkmale definiert den Typ der Ausprägung eines Objektes aus einer endlichen Menge

$$T \subseteq T_1 \times \dots \times T_\tau \quad (3.3)$$

von Typen.

Beispiel 3.1 (Typmerkmale) *Der Typ eines Inhaltsobjektes kann durch die folgenden Typmerkmale beschrieben werden:*

- **type.** *Haupttyp eines Inhaltsobjektes im Sinne der MIME-Inhaltstypen [FB96b] aus der Menge {image, video, ...}*
- **subtype.** *Subtyp eines Inhaltsobjektes im Sinne der MIME-Inhaltstypen [FB96b] aus der Menge {gif, jpeg, png, mpeg, H261, ...}*

Ein Medienqualitätsmerkmal m_ξ ist ein Symbol aus einem (potentiell unendlichen) Symbolvorrat $M_\xi \subseteq \mathbb{R}_{0+}$. Das μ -Tupel

$$\mathbf{m} = (m_1, \dots, m_\mu) \quad (3.4)$$

der Ausprägungen aller Medienqualitätsmerkmale ist die Medienqualität der Ausprägung eines Objektes aus der (potentiell unendlichen) Menge

$$M \subseteq M_1 \times \dots \times M_\mu \quad (3.5)$$

von Medienqualitäten. Das Tupel (\mathbf{t}, \mathbf{m}) beschreibt die Repräsentation der Objektausprägung.

Beispiel 3.2 (Medienqualitätsmerkmale) *Ein Bild kann durch die folgenden Medienqualitätsmerkmale beschrieben werden:*

- **pix-x.** *Auflösung (Zahl der Bildpunkte) in x-Dimension aus der Menge \mathbb{N}*
- **pix-y.** *Auflösung (Zahl der Bildpunkte) in y-Dimension aus der Menge \mathbb{N}*
- **col-depth.** *Farbtiefe (in Bits pro Bildpunkt) aus der Menge {1, 2, 4, 8, 16, 24, 32}*
- **q-factor.** *Qualitätsfaktor bei der verlustbehafteten Kompression aus der Menge $\mathbb{Z}[0, 100]$*
- **size.** *Objektgröße (in Bytes) aus der Menge \mathbb{N}*

Bei einem Videostrom sind zusätzlich die folgenden Medienqualitätsmerkmale relevant:

- **framerate.** Bildwiederholrate (in Bilder pro Sekunde) aus der Menge \mathbb{R}_{0+}
- **bitrate.** Datenrate² (in Bytes pro Sekunde) aus der Menge \mathbb{R}_{0+}

Die Objektgröße stellt ein ausgezeichnetes Merkmal einer Repräsentation dar, da sie direkt in der Formulierung der Nebenbedingung zur beschränkten Speicherkapazität in den Surrogaten berücksichtigt wird. Deshalb wird die Größenfunktion

$$sf : K \times (T \times M) \rightarrow \mathbb{N} \quad (3.6)$$

definiert, die einer Repräsentation $(\mathbf{t}, \mathbf{m}) \in (T \times M)$ eines Objektes $k \in K$ die Größe in Bytes zuordnet. Ist, wie in Beispiel 3.2, die Objektgröße als Medienqualitätsmerkmal modelliert, so ist sf eine Projektionsfunktion:

$$sf(k, (\mathbf{t}, (m_1, \dots, \mathbf{size}, \dots, m_\mu))) = \mathbf{size} \quad (3.7)$$

Ein Servicequalitätsmerkmal s_ξ ist ein Symbol aus einem (potentiell unendlichen) Symbolvorrat $S_\xi \subseteq \mathbb{R}_{0+}$. Das σ -Tupel

$$\mathbf{s} = (s_1, \dots, s_\sigma) \quad (3.8)$$

der Ausprägungen aller Servicequalitätsmerkmale ist die Servicequalität bei der Auslieferung eines Objektes aus einer (potentiell unendlichen) Menge

$$S \subseteq S_1 \times \dots \times S_\sigma \quad (3.9)$$

von Servicequalitäten.

Beispiel 3.3 (Servicequalitätsmerkmale) Die Auslieferung eines Bildes kann durch das Servicequalitätsmerkmal

- **delay.** Verzögerungszeit (in Sekunden) aus der Menge \mathbb{R}_{0+}

charakterisiert werden. Bei der Auslieferung eines Videostroms ist das Servicequalitätsmerkmal

- **jitter.** Verzögerungsschwankung (in Sekunden) aus der Menge \mathbb{R}_{0+}

relevant.

Surrogate

Die Surrogate des A-CDN sind die Knoten, an denen Replikate der Inhaltsobjekte gespeichert werden können. Sie werden im Systemmodell durch Knoten h einer Menge $H \subset N$ von Surrogaten repräsentiert. Gemäß Abschnitt 3.1.2 kann ein Knoten h ein ganzes Rechenzentrum des A-CDN, bestehend aus einem Cluster von Surrogaten und Adaptionsservern, repräsentieren.

Die Zahl der Replikate, die in einem Surrogat $h \in H$ gespeichert werden kann, ist durch die Speicherkapazität des Surrogats eingeschränkt. Die Funktion

$$sc : N \rightarrow \mathbb{N} \quad (3.10)$$

ordnet jedem Knoten $n \in N$ eine Speicherkapazität in Bytes zu. Dabei ist für alle Knoten, die keine Surrogate sind ($n \notin H$), die Speicherkapazität $sc(n) = 0$. Somit ist die Menge H der Surrogate im formalen Modell implizit über die Funktion sc definiert:

$$H = \{n \mid (n \in N) \wedge (sc(n) > 0)\} \quad (3.11)$$

²Bei Videostromen mit variabler Datenrate muss gegebenenfalls zwischen maximaler (*peak bitrate*) und effektiver Datenrate (*effective bitrate*) unterschieden werden.

Ein Replikat im A-CDN ist eine (nicht-volatile) Materialisierung einer Ausprägung eines Objektes $k \in K$ mit den Typ- und Medienqualitätsmerkmalen $\mathbf{t} \in T$ bzw. $\mathbf{m} \in M$ in einem Surrogat $h \in H$. Es wird formal durch ein Tupel

$$(k, h, (\mathbf{t}, \mathbf{m})) \in K \times H \times (T \times M) \quad (3.12)$$

beschrieben. Die dreistellige Relation

$$P_R \subseteq K \times H \times (T \times M) \quad (3.13)$$

beschreibt die Platzierung aller Replikate in den Surrogaten des A-CDN.

Servicequalitätsmerkmale sind für die Beschreibung von Replikaten nicht relevant. Die Ausprägungen der Servicequalitätsmerkmale werden durch den Vorgang der Auslieferung bestimmt und werden folglich nicht durch Materialisierung permanent. Somit entsprechen die Servicequalitätsmerkmale von Replikaten Initialwerten $\mathbf{s} = \mathbf{s}_{\text{init}} = (0, \dots, 0)$ ³.

Speicherkosten für das Halten von Replikaten werden im Modell nicht berücksichtigt. Es wird davon ausgegangen, dass ein A-CDN-Betreiber bestimmte, begrenzte Speicherkapazitäten für die Replikation von Inhalten bereithält, die zu fixen Kosten zur Verfügung stehen. Fixkosten können durch die Platzierungsmechanismen nicht beeinflusst werden und finden deshalb im Modell keine Berücksichtigung.

Originalserver

Inhaltsobjekte werden durch die Inhaltsanbieter auf Originalservern bereitgestellt. Typischerweise wird ein Inhaltsanbieter genau eine Kopie, die Primärkopie, jeden Inhaltsobjektes auf einem Originalserver bereitstellen. Im Modell ist aber auch beschreibbar, dass ein Inhaltsanbieter mehrere Kopien in verschiedenen Repräsentationen oder auf verschiedenen Originalservern bereitstellt. Diese sind aus Sicht des A-CDN als gleichberechtigte Kopien zu betrachten (d.h., der Inhaltsanbieter muss strikte Konsistenz sicherstellen).

Ein Originalserver wird im formalen Modell durch einen Knoten b aus einer Menge $B \subset N$ von Originalservern dargestellt. Die Bereitstellung einer Kopie eines Objektes $k \in K$ in einem Originalserver $b \in B$ in einer Repräsentation mit den Typ- und Medienqualitätsmerkmalen $\mathbf{t} \in T$ bzw. $\mathbf{m} \in M$ kann formal als permanent materialisiertes Replikat $(k, b, (\mathbf{t}, \mathbf{m}))$ beschrieben werden. Ein permanent materialisiertes Replikat in einem Originalserver wird auch als Originalreplikat bezeichnet. Die Platzierung der Originalreplikate aller Objekte $k \in K$ in Knoten $n \in N$ wird durch die dreistellige Relation

$$P_0 \subseteq K \times N \times (T \times M) \quad (3.14)$$

dargestellt. Somit ist die Menge B der Originalserver implizit über die Platzierung der Originalreplikate definiert:

$$B = \{n \mid (k, n, (\mathbf{t}, \mathbf{m})) \in P_0\} \quad (3.15)$$

Die Vereinigung $P = P_R \cup P_0$ steht für die Platzierung aller Replikate (Originalreplikate in den Originalservern und Replikate in den Surrogaten des A-CDN).

Clients

Clients werden im Systemmodell durch Clientknoten c einer Menge $C \subset N$ (mit $C \cap H = \emptyset$ und $C \cap B = \emptyset$) von Clientknoten repräsentiert. Dabei steht ein Clientknoten c nicht für einen individuellen Client, sondern vertritt eine Gruppe von Clients, die über die gleichen Fähigkeiten zur

³Die Annahme, dass die Initialwerte \mathbf{s}_{init} der Servicequalitätsmerkmale von Replikaten $(0, \dots, 0)$ sind, gilt bei manchen Servicequalitätsmerkmalen nur näherungsweise. Zum Beispiel der Initialwert des Servicequalitätsmerkmals `delay` ergibt sich aus den Verzögerungen, die bei der Anfrageweiterleitung entstehen. Diese können sich je nach verwendeten Anfrageweiterleitungsmechanismen (vgl. Abs. 3.1.2) unterscheiden. Verzögerungen bei der Anfrageweiterleitung werden bei der Modellierung vernachlässigt. Mit Ausnahme der Auslieferung sehr kleiner Objekte sind sie bei der Auslieferung an mobile Clients mit schmalbandiger Netzwerkanbindung gegenüber den Verzögerungen bei der Auslieferung vernachlässigbar.

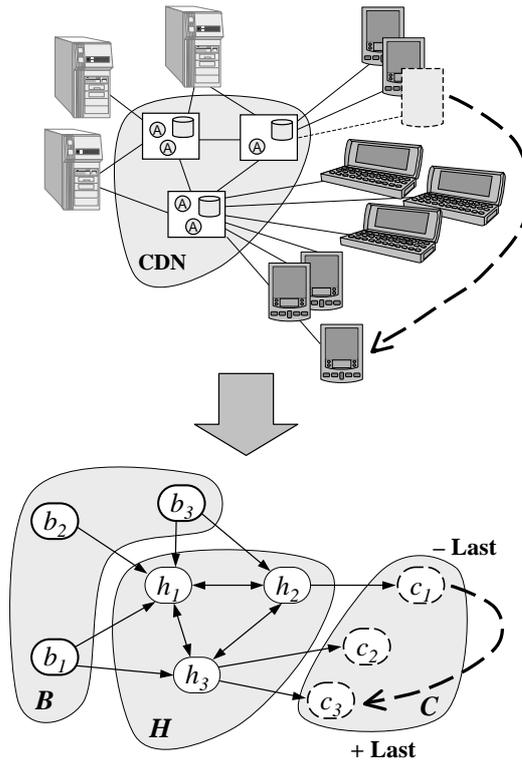


Abbildung 3.6: Mobilität individueller Clients im formalen Modell

Darstellung von Objekten verfügen, die gleichen Bewertungsmaßstäbe für Qualität haben (einem Objekt bei gleichen Ausprägungen der Medien- und Servicequalitätsmerkmale den gleichen Qualitätswert zuweisen, siehe Abschnitt „Qualität“ auf Seite 62) und über Netzwerkverbindungen mit gleichen Eigenschaften (d.h. gleiche Kosten und Auswirkungen auf Merkmalsausprägungen, vgl. Abschnitt „Operationen“ auf Seite 63) mit den gleichen Knoten $n \in N$ verbunden sind. Alle Anfragen aller Clients, die durch den Clientknoten c repräsentiert werden, werden c zugerechnet.

Dieser Modellierung liegt die Annahme zugrunde, dass im Allgemeinen mehrere Clients physikalisch Endgeräte des gleichen Typs sind und bei vergleichbarer Software-Ausstattung die gleichen Fähigkeiten zur Darstellung von Objekten aufweisen. Weiterhin sind die Geräte, wenn sie sich im gleichen Zugangnetz oder gar in der gleichen Funkzelle eines zellularen Zugangnetzes befinden, auch über eine Netzwerkverbindung mit vergleichbaren Eigenschaften mit dem selben Gateway-Knoten zum Internet verbunden.

Durch die Zusammenfassung von individuellen Clients zu Gruppen wird erreicht, dass sich die Mobilität individueller Clients nicht als Dynamik im Systemmodell widerspiegelt. Der Wechsel eines Client in eine andere Funkzelle oder zu einem anderen Zugangnetz schlägt sich dann in einem Wechsel des Client von einer Gruppe, die im Modell durch den Knoten c vertreten wird, in eine andere Gruppe von Clients, die durch c' repräsentiert wird, nieder, was zu einer Änderung der Anfragelast von c und c' führt (Abb. 3.6). Darüber hinaus wird durch die Zusammenfassung von Clients die Komplexität des Modells reduziert.

Die Fähigkeiten der Clients, die durch einen Clientknoten $c \in C$ repräsentiert werden, zur Darstellung von Medienobjekten mit verschiedenen Typ-, Medienqualitäts- und Servicequalitätsmerkmalen (*Media Handling Capabilities*) werden durch eine Menge von Tupeln $(t, \mathbf{m}, \mathbf{s}) \in (T \times M \times S)$ ausgedrückt. Die Abbildung

$$mhc : N \rightarrow 2^{(T \times M \times S)} \quad (3.16)$$

ordnet jedem Clientknoten c in der Menge $C \subset N$ eine solche Menge zu, so dass

$$mhc(c) = \{(\mathbf{t}, \mathbf{m}, \mathbf{s}) \mid c \text{ kann Medienobjekte, die mit den Typmerkmalen } \mathbf{t}, \text{ den Medienqualitätsmerkmalen } \mathbf{m} \text{ und den Servicequalitätsmerkmalen } \mathbf{s} \text{ ausgeliefert werden, darstellen}\}. \quad (3.17)$$

Implizit definiert der Definitionsbereich der Abbildung mhc formal die Menge C der Clientknoten: $C = \mathbf{D}(mhc)$.

Beispiel 3.4 ($mhc(c)$) *Die Fähigkeit eines Client, Videos in den Formaten MPEG und H.261 bis zu einer Auflösung von 240×320 Punkten und einer maximalen Farbtiefe von 8 Bits pro Bildpunkt darstellen zu können, wobei die Puffergröße des Client für maximale Verzögerungsschwankungen von 5 Sekunden ausgelegt ist, wird durch folgende Menge ausgedrückt:*

$$mhc(c) = \left\{ \left(\left(\begin{array}{c} \text{type,} \\ \text{subtype} \end{array} \right), \begin{array}{c} \text{pix-x,} \\ \text{pix-y,} \\ \text{col-depth,} \\ \text{q-factor,} \\ \text{size,} \\ \text{framerate,} \\ \text{bitrate} \end{array}, \left(\begin{array}{c} \text{delay,} \\ \text{jitter} \end{array} \right) \right) \mid \right. \\ \left. \left((\text{type} = \text{video}) \wedge ((\text{subtype} = \text{mpeg}) \vee (\text{subtype} = \text{H261})) \right) \right. \\ \left. \wedge (\text{pix-x} \leq 240) \wedge (\text{pix-y} \leq 320) \wedge (\text{col-depth} \leq 8) \wedge (\text{jitter} \leq 5) \right\}$$

Qualität

Die Qualität beschreibt, wie gut die Ausprägungen der Medien- und Servicequalitätsmerkmale bei der Auslieferung eines Inhaltsobjektes an einen Client den im Idealfall möglichen Merkmalsausprägungen bei der Auslieferung des Objektes an diesen Client entsprechen.

Die im Idealfall möglichen Ausprägungen der Medienqualitätsmerkmale sind diejenigen, die die höchste Wiedergabetreue des Objektes unter Berücksichtigung der Eigenschaften des Client ermöglichen. Die Wiedergabetreue kann dabei sowohl im Vergleich zum Originalreplikat auf dem Originalserver als auch im Vergleich zum Original aus der realen Welt definiert werden. Letzteres bedeutet, dass die Originalreplikate auf den Originalservern bereits mit eingeschränkter Qualität zur Verfügung stehen können. Auf diese Weise lässt sich die unterschiedliche Qualität verschiedener Originalreplikate einfach erfassen. Andererseits ist es bei vielen Realwelt-Objekten, z.B. bei Grafiken, die abstrakte Zusammenhänge aus der realen Welt illustrieren, nicht möglich, die Merkmalsausprägungen des Originals zu beschreiben und damit mit denen einer konkreten physischen Ausprägung vergleichbar zu machen. Eine praktikable Lösung ist, für jedes Inhaltsobjekt eine primäre Repräsentation zu identifizieren und die Wiedergabetreue im Vergleich zur primären Repräsentation zu betrachten. Dabei muss die primäre Repräsentation aber nicht notwendigerweise auch als Originalreplikat materialisiert sein.

Beispiel 3.5 (ideale Ausprägungen von Medienqualitätsmerkmalen) *Wenn ein Client durch die Auflösung seines Displays nur Bilder und Videos mit einer Auflösung bis zu 240×320 Punkten darstellen kann, dann sind 240 und 320 die im Idealfall möglichen Ausprägungen der Medienqualitätsmerkmale pix-x bzw. pix-y , vorausgesetzt, die primäre Repräsentation hat mindestens eine Auflösung von 240×320 Punkten. Bei einem Piktogramm hingegen, dessen primäre Repräsentation nur eine Auflösung von 50×50 Punkten aufweist, wäre die im Idealfall mögliche Ausprägung $\text{pix-x} = 50$ und $\text{pix-y} = 50$.*

Für Servicequalitätsmerkmale lassen sich die im Idealfall möglichen Ausprägungen als diejenigen definieren, die keinerlei Beeinträchtigungen der Wiedergabe im Vergleich zur Wiedergabe des Originals darstellen.

Beispiel 3.6 (ideale Ausprägungen von Servicequalitätsmerkmalen) *Im Idealfall sollen bei der Auslieferung eines Objektes an einen Clientknoten keinerlei Verzögerungen auftreten. Das heißt, die ideale Ausprägung des Servicequalitätsmerkmals `delay` ist `delay = 0`.*

Formal sei die Qualität einer bestimmten Ausprägung der Medien- und Servicequalitätsmerkmale bei der Auslieferung eines Objektes durch einen Qualitätswert im abgeschlossenen Intervall $\mathbb{R}[0, 1]$ beschrieben. Ein Qualitätswert von 1 steht für erschöpfende Qualität, die immer dann zugewiesen wird, wenn die Medien- und Servicequalität durch den Nutzer nicht von der im Idealfall möglichen Medien- und Servicequalität unterscheidbar ist. Dabei müssen die Merkmalsausprägungen nicht notwendigerweise denen im Idealfall entsprechen, da geringe Abweichungen durch den Nutzer nicht immer wahrnehmbar sind.

Beispiel 3.7 (erschöpfende Qualität) *Bei verlustbehafteter JPEG-Kompression eines Bildes sind durch einen menschlichen Beobachter bei geringem Kompressionsfaktor keinerlei Unterschiede zwischen dem Originalbild und dem komprimierten Bild wahrnehmbar. Damit wird dem komprimierten Bild ein Qualitätswert von 1 zugeordnet, obwohl die Ausprägungen einzelner Objektmerkmale (z.B. `q-factor`) objektiv von denen des Originals abweichen.*

Ein Qualitätswert von 0 (unzureichende Qualität) drückt aus, dass das ausgelieferte Objekt aufgrund seiner Medien- oder Servicequalitätsmerkmale für den Client unbrauchbar ist. Deshalb sei gefordert, dass ein Objekt niemals mit unzureichender Qualität ausgeliefert wird.

Beispiel 3.8 (unzureichende Qualität) *Ist ein Bild durch verlustbehaftete Kompression (z.B. JPEG-Kompression) so stark verfälscht, dass ein Nutzer keine Informationen aus dem verfälschten Bild entnehmen kann, so ist es für den Client unbrauchbar und bekommt einen Qualitätswert von 0 zugewiesen. Gleichsam ist die Qualität unzureichend, wenn die Verzögerung bei der Objektauslieferung einen Schwellwert für untolerierbare Verzögerung überschreitet.*

Der Zusammenhang zwischen den Ausprägungen der Medien- und Servicequalitätsmerkmale \mathbf{m} und \mathbf{s} und dem zugeordneten Qualitätswert wird durch die Qualitätsfunktion qf erfasst. Da die Definition der Qualität auf der Wiedergabetreue unter Berücksichtigung der Eigenschaften des Clientknoten c basiert und Wiedergabetreue im Vergleich zum Original des Objektes k definiert ist, gehen c und k als Parameter in die Qualitätsfunktion qf ein⁴. Damit ist die Qualitätsfunktion eine Abbildung

$$qf : C \times K \times (M \times S) \rightarrow \mathbb{R}[0, 1], \quad (3.18)$$

die Tupeln $(c, k, (\mathbf{m}, \mathbf{s}))$ den Qualitätswert der Ausprägungen der Medien- und Servicequalitätsmerkmale \mathbf{m} und \mathbf{s} des Objektes k aus Sicht des Clientknoten c zuweist.

Operationen

Im formalen Modell wird ein umspannender Operationsbegriff zugrunde gelegt, der Adaptionenoperationen und Transferoperationen, die den Transfer von Daten über Netzwerkverbindungen beschreiben, einschließt. Entsprechend diesem umspannenden Operationsbegriff ist eine Operation eine partielle Abbildung

$$o : K \times N \times (T \times M \times S) \rightarrow N \times (T \times M \times S), \quad (3.19)$$

⁴In der Praxis kann die Abhängigkeit des Qualitätswertes einer Ausprägung der Medien- und Servicequalitätsmerkmale vom konkreten Objekt k und vom konkreten Clientknoten c als Funktion von Metadaten über den Client und das Objekt beschrieben werden (vgl. Abschnitt 5.3). Für das formale Modell ist eine Modellierung der Metadaten aber nicht notwendig. Anstelle dessen ist es hinreichend, die Identität des Objektes k und des Clientknoten c als Parameter der Abbildung qf zu berücksichtigen.

die einem Tupel $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ ein Tupel $(n', (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$ zuordnet. Damit können zum einen die Änderungen $((\mathbf{t}, \mathbf{m}, \mathbf{s}) \mapsto (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$ der Typ-, Medienqualitäts- und Servicequalitätsmerkmale durch eine Adaptionsoption an einem Knoten $n = n'$ beschrieben werden. Darüber hinaus lässt sich auch die Ortsänderung $(n \mapsto n')$ durch eine Transferoperation beschreiben. Dabei erfasst $((\mathbf{t}, \mathbf{m}, \mathbf{s}) \mapsto (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$ mit $\mathbf{t} = \mathbf{t}'$ die Auswirkungen des Transfers auf die Medien- und Servicequalitätsmerkmale des Objektes, z.B. die Auswirkungen von Paketverlusten auf das Medienqualitätsmerkmal **framerate** eines Videostroms bei Übertragung über eine unzuverlässige Verbindung oder die Auswirkungen von Übertragungsverzögerungen auf das Servicequalitätsmerkmal **delay**. Der Parameter $k \in K$ der Abbildung o erlaubt, die unterschiedlichen Auswirkungen einer Operation auf die Ausprägungen der Merkmale verschiedener Objekte zu erfassen⁵. Der Definitionsbereich $\mathbf{D}(o) \subset (K \times N \times (T \times M \times S))$ gibt an, an welchen Knoten $n \in N$ und bei welchen Ausprägungen $(\mathbf{t}, \mathbf{m}, \mathbf{s}) \in (T \times M \times S)$ der Typ-, Medienqualitäts- und Servicequalitätsmerkmale die Operation o auf ein Objekt $k \in K$ anwendbar ist. Er spezifiziert somit, welche Operationen an welchen Knoten installiert sind und welche Vorbedingungen für die Anwendbarkeit der Operationen gelten.

In der Praxis können Adaptionsoptionen, die an einem Knoten parallel ausgeführt werden, miteinander wechselwirken, da sie um die gleichen Ressourcen (z.B. CPU) konkurrieren. So wird zum Beispiel die Ausführungszeit einer Operation von der Zahl konkurrierender Operationsausführungen abhängen. Folglich kann die Last an einem Knoten Auswirkungen auf Medien- und Servicequalitätsmerkmale eines adaptierten Objektes haben. Gleiches gilt auch für Transferoperationen. Parallele Netzwerktransfers konkurrieren zum Beispiel um die Bandbreite der Netzwerkverbindung. Bei der Modellierung werden die Wechselwirkungen paralleler Operationsausführungen vernachlässigt. Es werden keine lastabhängigen Auswirkungen auf Typ-, Medienqualitäts- und Servicequalitätsmerkmale berücksichtigt. Ebenso werden Adaptionsoptionen und Transferoperationen als nicht kapazitätsbeschränkt betrachtet.

Jede Installation einer Adaptionsoption an einem Knoten $n \in N$ definiert eine Operation

$$o_{\text{adapt}} : K \times N \times (T \times M \times S) \rightarrow N \times (T \times M \times S) \quad (3.20)$$

als Abbildung

$$\begin{aligned} o_{\text{adapt}} = \{ & ((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \mapsto (n, (\mathbf{t}', \mathbf{m}', \mathbf{s}'))) \mid (k \in K) \wedge (n \in N) \wedge \\ & ((\mathbf{t}, \mathbf{m}, \mathbf{s}) \in (T \times M \times S)) \wedge ((\mathbf{t}', \mathbf{m}', \mathbf{s}') \in (T \times M \times S)) \wedge \\ & (\text{die Adaptionsoption ist am Knoten } n \text{ installiert}) \wedge (\text{wenn} \\ & \text{das Objekt } k \text{ vor der Anwendung der Adaptionsoption mit} \\ & \text{den Merkmalsausprägungen } (\mathbf{t}, \mathbf{m}, \mathbf{s}) \text{ vorliegt, so ändern sich die} \\ & \text{Merkmalsausprägungen durch die Ausführung der Operation} \\ & \text{zu } (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \}. \end{aligned} \quad (3.21)$$

Die Menge O_{adapt} sei die Menge aller Adaptionsoptionen.

Analog definieren die Netzwerkverbindungen Operationen

$$o_{\text{trans}} : K \times N \times (T \times M \times S) \rightarrow N \times (T \times M \times S) \quad (3.22)$$

⁵Die unterschiedlichen Auswirkungen einer Operation auf die Ausprägungen der Typ-, Medienqualitäts- und Servicequalitätsmerkmale können in der Praxis durch die Einbeziehung von Metadaten über das Objekt bei der Ermittlung der Merkmalsausprägungen im Ergebnis der Anwendung einer Operation berücksichtigt werden (vgl. Abschnitt 5.4). Für das formale Modell ist eine Modellierung der Metadaten aber nicht notwendig. Anstelle dessen ist es hinreichend, die Identität des Objektes k als Parameter der Abbildung o einzubeziehen.

mit

$$o_{\text{trans}} = \{((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \mapsto (n', (\mathbf{t}, \mathbf{m}', \mathbf{s}')) \mid (k \in K) \wedge (n \in N) \wedge (n' \in N) \wedge ((\mathbf{t}, \mathbf{m}, \mathbf{s}) \in (T \times M \times S)) \wedge ((\mathbf{t}, \mathbf{m}', \mathbf{s}') \in (T \times M \times S)) \wedge (\text{die Transferoperation ermöglicht die Datenübertragung vom Knoten } n \text{ zum Knoten } n') \wedge (\text{wenn das Objekt } k \text{ vor dem Transfer mit den Merkmalsausprägungen } (\mathbf{t}, \mathbf{m}, \mathbf{s}) \text{ an } n \text{ vorliegt, so ändern sich die Merkmalsausprägungen durch den Transfer von } n \text{ nach } n' \text{ zu } (\mathbf{t}, \mathbf{m}', \mathbf{s}'))\}. \quad (3.23)$$

Dabei beschreibt eine Transferoperation o_{trans} jeweils einen unidirektionalen Netzwerkdienst bzw. eine Richtung eines bidirektionalen Netzwerkdienstes zwischen zwei Endknoten n und n' . Folglich werden bidirektionale Netzwerkdienste im Modell durch zwei Transferoperationen repräsentiert. Wenn zwischen zwei Knoten n und n' mehrere Netzwerkdienste (z.B. TCP und RTP) genutzt werden können, dann wird jeder Netzwerkdienst als eine separate Transferoperation (bei bidirektionalen Diensten als jeweils zwei separate Transferoperationen) modelliert.

Die Menge aller Operationen, die durch die Gesamtheit aller Netzwerkverbindungen definiert wird, sei O_{trans} . Die Menge aller Operationen im Sinne des umfassenden Operationsbegriffs ergibt sich als:

$$O = O_{\text{adapt}} \cup O_{\text{trans}} \quad (3.24)$$

Inhaltsadaption verursacht dem A-CDN-Betreiber Kosten in Form von Ressourcenverbrauch. In der vorliegenden Arbeit werden Adaptionkosten als der monetäre Aufwand des Betreibers bei der Anwendung der einzelnen Adaptionoperationen definiert. Hier wird davon ausgegangen, dass sich jeder Ausführung einer Adaptionoperation konkrete monetäre Kosten zuordnen lassen. Dies entspricht Abrechnungsmodellen, die in Hochleistungsrechenzentren verwendet werden, wo eine Abrechnung nach CPU-Zeit erfolgt (vgl. Anhang D). In der Praxis werden die Adaptionkosten oft nicht von den ausgeführten Adaptionoperationen abhängen, sondern fixe Unterhaltskosten für den Betrieb des A-CDN sein. Um die vorgeschlagenen Modelle auch in diesen Fällen einsetzen zu können, müssen die fixen Unterhaltskosten unter Abschätzung der mittleren Auslastung der Ressourcen auf Kosten für die einzelnen Operationsausführungen projiziert werden. Auch Transferoperationen sind mit Kosten verbunden, die im Modell als monetäre Aufwendungen für die einzelnen Netzwerktransfers berücksichtigt werden. Die Kosten für die Anwendung einer Adaption- oder Transferoperation o werden durch eine Kostenfunktion

$$cf_o : K \times N \times (T \times M \times S) \rightarrow \mathbb{R}_{0+}$$

beschrieben. Die Abbildung

$$\wp : O \rightarrow (K \times N \times (T \times M \times S) \rightarrow \mathbb{R}_{0+}) \quad (3.25)$$

ist die Zuordnung von Kostenfunktionen cf_o zu den Operationen $o \in O$.

Beispiel 3.9 (Kostenfunktion einer Adaptionoperation) *Die Kosten einer Adaptionoperation lassen sich als Produkt der CPU-Zeit und der Basiskosten c_{cpu} pro Zeiteinheit CPU-Nutzung definieren. Damit ergibt sich für die Kostenfunktion einer Adaptionoperation zur Umwandlung eines Bildes im GIF-Format in ein JPEG-Bild folgender Zusammenhang:*

$$cf_{\text{gif} \rightarrow \text{jpeg}}(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = c_{\text{cpu}} \left(t_{\text{fix}} + t_{\text{LZW}} \cdot \text{size} + t_{\text{JPEG}} \cdot \left(\left\lceil \frac{\text{pix-x}}{8} \right\rceil \left\lceil \frac{\text{pix-y}}{8} \right\rceil \right)^{1.07} \right)$$

Dabei sind pix-x , pix-y und size Medienqualitätsmerkmale in \mathbf{m} . Der Summand t_{fix} deckt die datenunabhängigen Zeitanteile (z.B. zum Erzeugen und Initialisieren einer Instanz der Adaptionoperation) ab. Der Faktor t_{LZW} beschreibt den Zeitaufwand pro Byte des Ausgangsbildes für die GIF-Dekomprimierung. Der Zeitaufwand für die JPEG-Kodierung wird durch den Summanden $t_{\text{JPEG}} \cdot \left(\left\lceil \frac{\text{pix-x}}{8} \right\rceil \left\lceil \frac{\text{pix-y}}{8} \right\rceil \right)^{1.07}$ beschrieben. Das entspricht der empirisch ermittelten Ausführungszeit einer JPEG-Kodieroperation (vgl. Anhang C.2).

Beispiel 3.10 (Kostenfunktion einer Transferoperation) *Bei der Datenübertragung zwischen einem Surrogat und einem mobilen Client über eine GPRS-Verbindung fallen Übertragungskosten proportional zur Zahl der übertragenen Datenbytes an. Dementsprechend lässt sich die Kostenfunktion für die Übertragung eines Objektes zwischen Surrogat und mobilem Client folgendermaßen definieren:*

$$cf_{\text{gprs}}(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = c_{\text{gprs}} \cdot \text{size}$$

Dabei steht c_{gprs} für die Basiskosten pro übertragenem Datenbyte und schließt etwaige Kosten für die Übertragung eines Datenbyte zwischen dem Gateway des GPRS-Netzes zum Internet-Backbone und dem Surrogat mit ein.

Adaptionspfade

Ein Adaptionspfad beschreibt die sukzessive Anwendung von Adaptions- und Transferoperationen auf einem Replikat (Originalreplikat oder Replikat in einem A-CDN). In dieser Arbeit soll der Begriff des Adaptionspfades auch für die sukzessive Anwendung von ausschließlich Transferoperationen (ohne Adaptionsoperation) verwendet werden, auch wenn in einem solchen Pfad genau genommen keine Inhaltsadaptation stattfindet. Auch die Anwendung einer einzelnen Adaptions- oder Transferoperation sei ein Adaptionspfad. Ein Adaptionspfad kann sowohl zur Auslieferung eines Objektes zu einem Clientknoten als Antwort auf eine Anfrage als auch zur Verteilung eines Replikates bestimmt sein.

Formal sei ein Adaptionspfad ein gerichteter, unärer Baum⁶

$$\pi = (k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} (k, n_1, (\mathbf{t}_1, \mathbf{m}_1, \mathbf{s}_1)) \xrightarrow{o_2} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)), \quad (3.26)$$

dessen Knotenmenge $V(\pi)$ eine Teilmenge von $K \times N \times (T \times M \times S)$ ist, dessen Kanten mit Operationen (Adaptions- und Transferoperationen) $o_i \in O$ beschriftet sind und der die folgenden Bedingungen erfüllt:

- (i) Für den Wurzelknoten⁷ $root(\pi) = (k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0))$ gilt:
 - Das Objekt k ist am Knoten n_0 als Replikat oder Originalreplikat mit den Typ- und Medienqualitätsmerkmalen \mathbf{t}_0 und \mathbf{m}_0 materialisiert, d.h. $(k, n_0, (\mathbf{t}_0, \mathbf{m}_0)) \in P$.
 - Die Servicequalitätsmerkmale haben Initialwerte, d.h., $\mathbf{s}_0 = \mathbf{s}_{\text{init}}$.
- (ii) Für jede Kante $(k, n_i, (\mathbf{t}_i, \mathbf{m}_i, \mathbf{s}_i)) \xrightarrow{o_{i+1}} (k, n_{i+1}, (\mathbf{t}_{i+1}, \mathbf{m}_{i+1}, \mathbf{s}_{i+1})) \in E(\pi)$ gilt:
 - Der Endknoten $(k, n_{i+1}, (\mathbf{t}_{i+1}, \mathbf{m}_{i+1}, \mathbf{s}_{i+1}))$ ergibt sich aus dem Anfangsknoten $(k, n_i, (\mathbf{t}_i, \mathbf{m}_i, \mathbf{s}_i))$ durch Anwendung der Operation o_{i+1} , mit der die Kante beschriftet ist, d.h. $(n_{i+1}, (\mathbf{t}_{i+1}, \mathbf{m}_{i+1}, \mathbf{s}_{i+1})) = o_{i+1}(k, n_i, (\mathbf{t}_i, \mathbf{m}_i, \mathbf{s}_i))$.

Die Menge Π_P aller Adaptionspfade, die sich bei einer bestimmten Platzierung P von Replikaten in einem System Σ konstruieren lassen, ist induktiv definiert durch:

- (i) Für jedes Replikat $(k, n, (\mathbf{t}, \mathbf{m})) \in P$ existiert ein Adaptionspfad $\pi \in \Pi_P$, dessen Knotenmenge nur einen Knoten (als Wurzel- und Blattknoten) umfasst und dessen Kantenmenge leer ist. Formal ausgedrückt heißt das:

$$\forall (k, n, (\mathbf{t}, \mathbf{m})) \in P. (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}_{\text{init}})) \in \Pi_P \quad (3.27)$$

⁶Ein gerichteter, unärer Baum ist ein zusammenhängender, gerichteter, zyklener Graph, dessen Knoten mit Ausnahme zweier ausgezeichneter Knoten (des Wurzelknotens und des Blattknotens) eine Eingangsvalenz von 1 und eine Ausgangsvalenz von 1 haben. Der Wurzelknoten hat eine Eingangsvalenz von 0, die Ausgangsvalenz beträgt 1. Der Blattknoten hingegen hat eine Ausgangsvalenz von 0 und eine Eingangsvalenz von 1.

⁷In dieser Arbeit werden die Funktionen $root$ und $leaf$ verwendet, die einen Adaptionspfad π auf seinen Wurzelknoten ($root(\pi)$) bzw. seinen Blattknoten ($leaf(\pi)$) abbilden.

- (ii) Wenn Π_P einen Adaptionpfad $\pi = (k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p))$ umfasst, so sind alle Adaptionpfade $(k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)) \xrightarrow{o_q} (k, n_q, (\mathbf{t}_q, \mathbf{m}_q, \mathbf{s}_q))$, die sich aus π durch Anwendung einer Operation $o_q \in O$ auf den Blattknoten $(k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p))$ von π ergeben (d.h., $(n_q, (\mathbf{t}_q, \mathbf{m}_q, \mathbf{s}_q)) = o_q(k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p))$), ebenfalls Element der Menge Π_P . Formal ausgedrückt:

$$\begin{aligned} \forall (k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)) \in \Pi_P. & \quad (3.27') \\ \forall (k, n_q, (\mathbf{t}_q, \mathbf{m}_q, \mathbf{s}_q)) \in K \times N \times (T \times M \times S). & \\ (\exists o_q \in O. (n_q, (\mathbf{t}_q, \mathbf{m}_q, \mathbf{s}_q)) = o_q(k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p))) \Rightarrow & \\ ((k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)) \xrightarrow{o_q} (k, n_q, (\mathbf{t}_q, \mathbf{m}_q, \mathbf{s}_q)) \in \Pi_P) & \end{aligned}$$

- (iii) Es gibt keinen Adaptionpfad in Π_P , der sich nicht nach Regel (i) oder (ii) ergibt.

Da ein Adaptionpfad grundsätzlich von genau einem Replikat $(k, n, (\mathbf{t}, \mathbf{m})) \in P$ ausgeht, gilt für zwei Platzierungen P_1 und P_2 :

$$\Pi_{P_1} \cup \Pi_{P_2} = \Pi_{P_1 \cup P_2} \quad (3.28)$$

Wenn es einen Adaptionpfad $\pi \in \Pi_P$ gibt, dessen Blattknoten $leaf(\pi) = (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ ist, so wird dadurch ausgedrückt, dass das Objekt k in einem System Σ bei der Platzierung P zum Knoten n mit den Typ-, Medienqualitäts- und Servicequalitätsmerkmalen \mathbf{t} , \mathbf{m} und \mathbf{s} zugestellt werden kann.

Wenn ein Objekt k durch einen Adaptionpfad $\pi \in \Pi_{P_0}$ zu einem Surrogat $h \in H$ mit den Typ-, Medienqualitäts- und Servicequalitätsmerkmalen \mathbf{t} , \mathbf{m} und \mathbf{s} zugestellt werden kann, so kann es im Surrogat h auch mit den Typ- und Medienqualitätsmerkmalen \mathbf{t} und \mathbf{m} materialisiert werden (vorausgesetzt, h hat hinreichend freien Speicherplatz). Eine Objektausprägung, die als Replikat materialisiert werden kann, wird als potentielles Replikat bezeichnet. Die Menge der potentiellen Replikate bei einer vorgegebenen Platzierung P_0 von Originalreplikaten sei $PR_{P_0} \subseteq K \times H \times (T \times M)$.

Da die Ausprägungen \mathbf{s} der Servicequalitätsmerkmale durch Materialisierung nicht permanent werden, sind sie bei der Verteilung nicht relevant. Deshalb wird PR_{P_0} ohne Berücksichtigung von Servicequalitätsmerkmalen definiert. Für die Definition von PR_{P_0} wird eine Menge $\widehat{PR}_{P_0} \subseteq K \times N \times (T \times M)$ zu Hilfe genommen, die induktiv definiert ist:

- (i) Alle Originalreplikate sind Element der Menge \widehat{PR}_{P_0} :

$$P_0 \subseteq \widehat{PR}_{P_0} \quad (3.29)$$

- (ii) Wenn es eine Adaption- oder Transferoperation $o \in O$ gibt, die eine Ausprägung eines Objektes k am Knoten n mit den Typ- und Medienqualitätsmerkmalen \mathbf{t} und \mathbf{m} , die durch ein Tupel $(k, n, (\mathbf{t}, \mathbf{m})) \in \widehat{PR}_{P_0}$ repräsentiert wird, in eine Ausprägung von k am Knoten n' mit den Typ- und Medienqualitätsmerkmalen \mathbf{t}' und \mathbf{m}' überführt, dann ist auch $(k, n', (\mathbf{t}', \mathbf{m}')) \in \widehat{PR}_{P_0}$. Formal ausgedrückt:

$$\begin{aligned} \forall (k, n, (\mathbf{t}, \mathbf{m})) \in \widehat{PR}_{P_0}. \forall (n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \in (N \times (T \times M \times S)). & \\ (\exists o \in O. (n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) = o(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}_{init}))) \Rightarrow ((k, n', (\mathbf{t}', \mathbf{m}')) \in \widehat{PR}_{P_0}) & \quad (3.29') \end{aligned}$$

- (iii) Es gibt kein Tupel $(k, n, (\mathbf{t}, \mathbf{m})) \in \widehat{PR}_{P_0}$, das sich nicht nach Regel (i) oder (ii) ergibt.

Die Menge PR_{P_0} der potentiellen Replikate bei einer vorgegebenen Platzierung P_0 von Originalreplikaten umfasst alle Tupel in \widehat{PR}_{P_0} , die Ausprägungen von Objekten in einem Surrogat repräsentieren:

$$PR_{P_0} = \widehat{PR}_{P_0} \cap (K \times H \times (T \times M)) \quad (3.30)$$

3.2.2 Lastmodell

Das Lastmodell beschreibt die Anfragen der Clients im System zum adaptiven Content Networking. Eine Anfrage ist ein geordnetes Paar

$$(c, k) \in C \times K. \quad (3.31)$$

Durch (c, k) wird zum Ausdruck gebracht, dass ein Client, der der Gruppe von Clients zuzuordnen ist, die durch den Clientknoten c repräsentiert wird, ein Objekt k anfordert. In der Praxis wird eine Anfrage noch weitere Informationen, wie die Beschreibung der Eigenschaften und Fähigkeiten des Endgerätes und der Netzwerkanbindung, enthalten. Diese Informationen sind im formalen Modell über die Funktion mhc der Clientfähigkeiten, die Qualitätsfunktion qf und die Menge der Operationen O definiert und deshalb nicht Teil der Anfrage.

Eine Anfrage wird durch die Konstruktion eines geeigneten Adaptionspfades $\pi \in \Pi_P$ beantwortet. Ein Adaptionspfad π mit $leaf(\pi) = (k_\pi, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p))$ ist geeignet, eine Anfrage (c, k) von einem Clientknoten c zu beantworten, wenn

- der Adaptionspfad die Anwendung von Operationen auf einem Replikat des angeforderten Objektes k beschreibt, d.h. $k_\pi = k$,
- der Blattknoten eine Ausprägung des angeforderten Objektes im Clientknoten c repräsentiert, d.h. $n_p = c$,
- der Client, der durch c repräsentiert wird, in der Lage ist, eine Objektausprägung mit den Typ-, Medienqualitäts- und Servicequalitätsmerkmalen des Blattknotens des Adaptionspfades darzustellen, d.h. $(\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p) \in mhc(c)$, und
- die Qualität der Ausprägung der Medien- und Servicequalitätsmerkmale des Blattknotens aus Sicht des Client nicht unzureichend ist, d.h. $qf(c, k, (\mathbf{m}_p, \mathbf{s}_p)) > 0$.

Die Eigenschaft eines Adaptionspfades π , eine Anfrage (c, k) beantworten zu können, sei formal durch die Relation

$$satisfies \subseteq \Pi \times (C \times K) \quad (3.32)$$

beschrieben.

Die Anfragerate $\lambda_{c,k} \in \mathbb{R}_{0+}$ ist die mittlere kumulierte Zahl von Anfragen (c, k) , die von den Clients der Gruppe, die durch den Clientknoten c repräsentiert wird, innerhalb eines normierten Betrachtungszeitraums initiiert werden. Die Lastmatrix

$$\Lambda = (\lambda_{c,k}) \in \mathbb{R}_{0+}^{(|C|, |K|)} \quad (3.33)$$

der Anfrageraten aller Clientknoten $c \in C$ und aller Objekte $k \in K$ beschreibt die Last eines Systems Σ zum adaptiven Content Networking.

3.2.3 Problemmodell

Ein Ziel bei der Auslieferung von adaptierten Inhalten durch ein A-CDN ist, die Medienqualität der ausgelieferten Repräsentationen und die Servicequalität der Auslieferung zu maximieren. Demgegenüber steht das Ziel, die Betriebskosten für die Dienstleistung zu minimieren.

Beim Betrieb eines A-CDN entstehen Adaption- und Netzwerktransferkosten für die Verteilung und Auslieferung von Inhalten durch Adaptionspfade. Die Kosten $cost_\pi$ eines Adaptionspfades π umfassen alle Kosten, die dem A-CDN-Betreiber, dem Inhaltsanbieter oder dem Nutzer durch die Auslieferung eines Objektes über den Adaptionspfad π entstehen (vgl. Abs. 3.1.1). Sie ergeben sich als die Summe der Kosten für alle Anwendungen von Adaption- und Transferoperationen innerhalb des Adaptionspfades π :

$$cost_\pi = \sum_{(k, n_i, (\mathbf{t}_i, \mathbf{m}_i, \mathbf{s}_i)) \xrightarrow{o_j} (k, n_j, (\mathbf{t}_j, \mathbf{m}_j, \mathbf{s}_j)) \in E(\pi)} cf_{o_j}(k, n_i, (\mathbf{t}_i, \mathbf{m}_i, \mathbf{s}_i)) \quad (3.34)$$

Da die Verteilung von Inhalten im Vergleich zur Auslieferung selten vorkommt (weil Aktualisierungen von Multimediaobjekten sehr selten sind), können die Kosten für die Verteilung gegenüber den Auslieferungskosten vernachlässigt werden. Deshalb werden in der Problemformulierung keine Verteilungskosten berücksichtigt.

Die Gesamtkosten für die Auslieferung von Inhalten sind die Summe der Kosten, die bei der Beantwortung aller Anfragen entstehen. Es sei $\pi_{c,k}$ ein Adaptionpfad zur Beantwortung einer Anfrage (c, k) bei einer Platzierung P_R von Replikaten in den Surrogaten im System Σ . Dann beschreibt $cost_{\pi_{c,k}}$ die Kosten zur Beantwortung der Anfrage (c, k) . Die Gesamtkosten zur Beantwortung aller Anfragen bei der Last $\Lambda = (\lambda_{c,k})$ sind somit

$$Cost_{\Sigma, \Lambda}(P_R) = \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot cost_{\pi_{c,k}}. \quad (3.35)$$

Die Gesamtkosten $Cost_{\Sigma, \Lambda}(P_R)$ für die Beantwortung aller Anfragen sind zu minimieren.

Die Qualität eines Adaptionpfades π lässt sich auf Grundlage der Medien- und Servicequalitätsmerkmale des Blattknotens $leaf(\pi)$ des Adaptionpfades beurteilen. Die Qualität eines Adaptionpfades π ist jeweils nur in Bezug auf einen konkreten Clientknoten c definiert, dessen Anfrage durch den Adaptionpfad π beantwortet werden soll:

$$qual_{\pi,c} = qf(c, k, (\mathbf{m}_p, \mathbf{s}_p)) \quad \text{mit} \quad (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)) = leaf(\pi) \quad (3.36)$$

Aus Sicht der Nutzer sollte jede Anfrage (c, k) durch einen Adaptionpfad $\pi_{c,k}$ mit möglichst guter Qualität ($qual_{\pi_{c,k},c} = \max!$) beantwortet werden. Auch der A-CDN-Betreiber hat Interesse an einer guten Qualität bei der Auslieferung der Inhalte, da davon auszugehen ist, dass die Zahlungsbereitschaft der Kunden mit der Qualität korreliert ist. Folglich lässt sich der Erlös, den ein A-CDN-Anbieter durch die Beantwortung einer Anfrage (c, k) erzielt, als Funktion der Qualität des Adaptionpfades $\pi_{c,k}$, mittels dessen die Anfrage beantwortet wird, beschreiben:

$$rev_{\pi_{c,k},c} = price_{c,k} \cdot qual_{\pi_{c,k},c} \quad (3.37)$$

Dabei ist $price_{c,k}$ der nominelle Preis des Objektes k , der sich bei der Auslieferung von k mit maximaler Qualität an einen Client, der durch c repräsentiert wird, erzielen ließe. Der nominelle Preis $price_{c,k}$ ist objektspezifisch, kann sich also von Inhaltsobjekt zu Inhaltsobjekt unterscheiden. Zum Beispiel kann für Videostreams ein gewisser Basispreis pro Spielminute festgelegt werden, was zu unterschiedlichen nominellen Preisen für Videostreams mit unterschiedlicher Länge führt. Allerdings ist der nominelle Preis per Definition über alle Ausprägungen eines Inhaltsobjektes konstant. Ebenso kann sich der nominelle Preis von Client zu Client unterscheiden. Auf diese Weise können zum Beispiel die unterschiedlichen Kosten für unterschiedliche Netzverbindungen der Clients berücksichtigt werden. Darüber hinaus erlaubt diese Modellierung eine Preisdiskriminierung für unterschiedliche Nutzer bzw. Gruppen von Nutzern. So können Nutzern mit höherer Zahlungsbereitschaft Premiumdienste angeboten werden, bei denen aufgrund höherer nomineller Preise die Qualität gegenüber den Kosten höher gewichtet ist. Diese Form der Preisdiskriminierung ist insbesondere dann sinnvoll, wenn die Nutzer als Kunden für den Service des CDN zur Kasse gebeten werden. Vertragspartner von CDNs sind aber üblicherweise nicht die Nutzer, sondern die Inhaltsanbieter. In diesem Fall ist nutzerabhängige Preisdiskriminierung nicht sinnvoll.

Der Gesamterlös, der durch Beantwortung aller Anfragen erzielt wird, ist

$$Rev_{\Sigma, \Lambda}(P_R) = \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot rev_{\pi_{c,k},c}. \quad (3.38)$$

Der Gewinn des A-CDN-Betreibers ist die Differenz zwischen dem Gesamterlös und den Gesamtkosten zur Beantwortung aller Anfragen:

$$Profit_{\Sigma, \Lambda}(P_R) = Rev_{\Sigma, \Lambda}(P_R) - Cost_{\Sigma, \Lambda}(P_R) \quad (3.39)$$

Ziel des A-CDN-Betreibers ist, seinen Gewinn zu maximieren: $Profit_{\Sigma, \Lambda}(P_R) = \max!$ Dazu ist eine Platzierung P_R von Replikaten in den Surrogaten des A-CDN sowie je ein Adaptionpfad $\pi_{c,k}$ zur Beantwortung jeder Anfrage (c, k) bei der Platzierung P_R zu ermitteln, wobei die

begrenzte Speicherkapazität in den Surrogaten zu berücksichtigen ist. Damit kann das Optimierungsproblem folgendermaßen zusammengefasst werden:

Problem 3.1 (Optimierung des Betriebs eines A-CDN)

gegeben:

- ein System $\Sigma = (K, (T, M, S), sf, N, sc, P_0, mhc, qf, O, \wp)$
- die Anfragelast $\Lambda = (\lambda_{c,k})$ der Clients in Σ

gesucht:

- eine Platzierung $P_R \subseteq PR_{P_0}$ von Replikaten in den Surrogaten des A-CDN
- für jede Anfrage (c, k) mit $\lambda_{c,k} > 0$ ein Adaptionpfad $\pi_{c,k} \in \Pi_{P_0 \cup P_R}$

Zielfunktion:

- Maximierung des Gesamtgewinns:

$$Profit_{\Sigma, \Lambda}(P_R) = \max!$$

Nebenbedingungen:

- Jede Anfrage (c, k) mit einer Anfragerate $\lambda_{c,k} > 0$ kann durch den entsprechenden Adaptionpfad $\pi_{c,k}$ beantwortet werden:

$$\forall (c, k) \in C \times K. (\lambda_{c,k} > 0) \Rightarrow (\pi_{c,k} \text{ satisfies } (c, k))$$

- Die Gesamtgröße aller Replikate in jedem einzelnen Surrogat überschreitet nicht die Speicherkapazität des Surrogates:

$$\forall h \in H. \sum_{(k,h,(t,m)) \in P_R} sf(k, (t, m)) \leq sc(h)$$

Durch Einsetzen von Gleichung 3.35 und 3.38 in Gleichung 3.39 erhält man:

$$\begin{aligned} Profit_{\Sigma, \Lambda}(P_R) &= Rev_{\Sigma, \Lambda}(P_R) - Cost_{\Sigma, \Lambda}(P_R) \\ &= \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot rev_{\pi_{c,k}, c} - \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot cost_{\pi_{c,k}} \\ &= \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot (rev_{\pi_{c,k}, c} - cost_{\pi_{c,k}}) \\ &= \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot profit_{\pi_{c,k}, c} \end{aligned} \quad (3.40)$$

Die Differenz

$$profit_{\pi_{c,k}, c} = rev_{\pi_{c,k}, c} - cost_{\pi_{c,k}} \quad (3.41)$$

ist der Gewinn aus der Beantwortung einer einzelnen Anfrage (c, k) durch den Adaptionpfad $\pi_{c,k}$ ⁸. Der maximale Gewinn, der sich bei einer gegebenen Platzierung P_R von Replikaten im A-CDN aus der Beantwortung einer Anfrage (c, k) erzielen lässt, sei

$$Profit_{\Sigma, (c,k)}(P_R) = \max \{ profit_{\pi, c} \mid (\pi \in \Pi_{P_0 \cup P_R}) \wedge (\pi \text{ satisfies } (c, k)) \}. \quad (3.42)$$

Die Zielfunktion $\sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot profit_{\pi_{c,k}, c}$ wird maximal, wenn der Gewinn aus der Beantwortung jeder Anfrage (c, k) mit $\lambda_{c,k} > 0$ der bezüglich der Platzierung P_R maximal mögliche Gewinn $Profit_{\Sigma, (c,k)}(P_R)$ ist. Somit lässt sich durch Einsetzen von Gleichung 3.42 in die Zielfunktion 3.40 das Optimierungsproblem der Suche nach einer Platzierung P_R von Replikaten in den Surrogaten des A-CDN unabhängig von der Suche nach Adaptionspfaden $\pi_{c,k}$ zur Beantwortung der Anfragen (c, k) formulieren:

⁸Man beachte: Aus Gleichung 3.37 und 3.41 folgt $profit_{\pi_{c,k}, c} = price_{c,k} \cdot qual_{\pi_{c,k}, c} - cost_{\pi_{c,k}}$. Somit wichten die nominellen Preise $price_{c,k}$ Kosten und Qualität in der Zielfunktion.

Problem 3.2 (Suche nach einer Platzierung von Replikaten)

gegeben:

- ein System $\Sigma = (K, (T, M, S), sf, N, sc, P_0, mhc, qf, O, \wp)$
- die Anfragelast $\Lambda = (\lambda_{c,k})$ der Clients in Σ

gesucht:

- eine Platzierung $P_R \subseteq PR_{P_0}$ von Replikaten in den Surrogaten des A-CDN

Zielfunktion:

- Maximierung des Gesamtgewinns:

$$\sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot Profit_{\Sigma, (c,k)}(P_R) = \max!$$

Nebenbedingungen:

- Die Gesamtgröße aller Replikate in jedem einzelnen Surrogat überschreitet nicht die Speicherkapazität des Surrogates:

$$\forall h \in H. \sum_{(k,h,(\mathbf{t},\mathbf{m})) \in P_R} sf(k, (\mathbf{t}, \mathbf{m})) \leq sc(h)$$

Algorithmen zur Lösung des Problems 3.2 werden in Abschnitt 4.2 entwickelt.

Der maximale Gewinn $Profit_{\Sigma, (c,k)}(P_R)$, der sich bei einer gegebenen Platzierung P_R von Replikaten im A-CDN aus der Beantwortung einer Anfrage (c, k) erzielen lässt, ist der Gewinn $profit_{\pi_{c,k}, c}$ eines bei der Platzierung P_R von Replikaten im A-CDN optimalen Adaptionspfades $\pi_{c,k}$ zur Beantwortung der Anfrage (c, k) :

$$Profit_{\Sigma, (c,k)}(P_R) = profit_{\pi_{c,k}, c} \quad (3.43)$$

Das Problem der Suche nach einem optimalen Adaptionspfad $\pi_{c,k}$ ist folglich ein inhärentes Teilproblem bei der Suche nach einer optimalen Platzierung von Replikaten im A-CDN. Darüber hinaus ist dieses Problem aber auch durch das Anfrageweiterleitungssystem des A-CDN bei der Planung eines Adaptionspfades zur Beantwortung einer Anfrage zu lösen. Es lässt sich folgendermaßen formalisieren:

Problem 3.3 (Suche nach einem optimalen Adaptionspfad)

gegeben:

- ein System $\Sigma = (K, (T, M, S), sf, N, sc, P_0, mhc, qf, O, \wp)$
- eine Platzierung P_R von Replikaten in den Surrogaten des A-CDN
- eine Anfrage (c, k)

gesucht:

- ein Adaptionspfad $\pi_{c,k} \in \Pi_{P_0 \cup P_R}$

Zielfunktion:

- Maximierung des Gewinns, der durch die Beantwortung der Anfrage (c, k) erzielt wird:

$$profit_{\pi_{c,k}, c} = \max!$$

Nebenbedingung:

- Die Anfrage (c, k) kann durch $\pi_{c,k}$ beantwortet werden:

$$\pi_{c,k} \text{ satisfies } (c, k)$$

Algorithmen zur Lösung des Problems 3.3 werden in Abschnitt 4.1 diskutiert.

Kapitel 4

Algorithmen zur Komposition von Adaptionspfaden und zur Platzierung von Replikaten

Wie im vorangegangenen Kapitel gezeigt, hat ein A-CDN-Betreiber das Ziel, durch den Betrieb des A-CDN die Zielfunktion $Profit_{\Sigma, \Lambda}(P_R)$ zu maximieren. Um dieses Ziel zu erreichen, wird durch die Platzierungsmechanismen die Platzierung der Replikate entsprechend der Problemformulierung 3.2 auf Seite 71 optimiert und durch das Anfrageweiterleitungssystem für jede Anfrage ein entsprechend der Problemstellung 3.3 auf Seite 71 optimaler Adaptionspfad zur Auslieferung des angeforderten Inhaltsobjektes bestimmt. Da die Problemformulierung 3.2 voraussetzt, dass der maximale Gewinn zur Beantwortung jeder Anfrage bekannt ist, ist die Komposition eines optimalen Adaptionspfades (Problem 3.3) für jede Anfrage auch ein inhärentes Teilproblem des Platzierungsproblems.

In diesem Kapitel werden Algorithmen zur Lösung der Probleme 3.2 und 3.3 vorgestellt. Da die Algorithmen die Grundlage für die Evaluation der Möglichkeiten und Grenzen adaptivitätssensitiver Platzierung von Replikaten in A-CDNs bilden sollen, werden hier statisch optimierende Algorithmen vorgeschlagen, die auf Grundlage globaler System- und Lastinformationen nahezu optimale Lösungen ermitteln.

4.1 Komposition optimaler Adaptionspfade

Die hier vorgeschlagene Lösung des Problems der Suche nach einem optimalen Adaptionspfad basiert auf der Idee von Choi et al. [CTW01] aus dem Active-Pipes-Projekt, das Problem auf die Suche nach dem kürzesten Pfad in einem gewichteten gerichteten Graphen abzubilden (vgl. Abs. 2.3.4). Anders als in [CTW01] ist der logische Adaptionspfad aber nicht als gegeben anzusehen, sondern soll in einem Optimierungsschritt gemeinsam mit dem physischen Adaptionspfad ermittelt werden. Darüber hinaus müssen Adaptionspfade in A-CDNs bezüglich ihrer Kosten und Qualität entsprechend der Zielfunktion $profit_{\pi_{c,k}, c}$ (Formel 3.41 auf Seite 70) optimiert werden und es kommen verschiedene Replikate als Ausgangspunkt sowie verschiedene Objektausprägungen als Zielpunkt eines Adaptionspfades in Frage, da eine Anfrage durch verschiedene Repräsentationen beantwortet werden kann. Deshalb müssen die Ideen aus dem Active-Pipes-Projekt entsprechend erweitert werden.

Der Ansatz erfordert, dass das Systemmodell Σ bzw. die für die Lösung des Problems für eine konkrete Anfrage (c, k) relevanten Teile des Systemmodells auf einen gewichteten gerichteten Graphen abgebildet werden, in dem die Suche nach einem kürzesten Pfad mit der Suche nach einem optimalen Adaptionspfad $\pi_{c,k}$ in Σ korrespondiert. Eine solche Graphrepräsentation des Systems wird als Problemgraph $\Gamma_{c,k}$ bezeichnet. Die Konstruktion des Problemgraphen $\Gamma_{c,k}$ für eine Anfrage (c, k) ist im folgenden Abschnitt 4.1.1 dargestellt. Algorithmen zur Suche nach dem kürzesten Pfad in $\Gamma_{c,k}$ werden in Abschnitt 4.1.2 diskutiert.

Die in Abschnitt 4.1.1 und 4.1.2 vorgeschlagenen Verfahren gehen von der allgemeinen Definition von Adaptionsoptionen entsprechend dem Modell in Kapitel 3 aus. Sie ermitteln den optimalen Adaptionspfad zur Beantwortung einer Anfrage aus der Menge aller Adaptionspfade, die sich auf Basis des Systemmodells konstruieren lassen. In der Praxis kann die Menge der relevanten Adaptionspfade aber durch bestimmte Eigenschaften der Adaptionsoptionen oder durch deren Platzierung im A-CDN eingeschränkt sein. In Abschnitt 4.1.3 wird diskutiert, wie Heuristiken, die die Menge der relevanten Adaptionspfade einschränken, in den Verfahren aus Abschnitt 4.1.1 und 4.1.2 berücksichtigt werden können.

Während in Problemstellung 3.3 auf Seite 71 das Problem der Suche nach einem optimalen Adaptionspfad zur Beantwortung einer einzelnen Anfrage definiert wird, erfordert die Bestimmung einer optimalen Platzierung von Replikaten in einem A-CDN gemäß Problemformulierung 3.2, dass für jede Anfrage $(c, k) \in C \times K$ der Gewinn $profit_{\pi_{c,k},c}$ des optimalen Adaptionspfades $\pi_{c,k}$ ermittelt wird. In Abschnitt 4.1.4 wird diskutiert, wie die in Abschnitt 4.1.1 bis 4.1.3 entwickelten Verfahren hierfür genutzt werden können und durch welche Optimierungen dabei die Komplexität reduziert werden kann.

4.1.1 Abbildung des Problems

Die Grundidee bei der Abbildung des Problems auf einen Problemgraphen beruht darauf, dass die Kanten im Problemgraphen $\Gamma_{c,k}$ zur Ermittlung des optimalen Adaptionspfades zur Beantwortung einer Anfrage (c, k) die Anwendung von Adaption- und Transferoperationen $o \in O$ auf Ausprägungen des angeforderten Objektes k repräsentieren. Die Knoten in $\Gamma_{c,k}$ stehen für Ausprägungen von k , die durch einen Ort (den Netzwerkknoten¹ $n \in N$, an dem die Ausprägung vorliegt) sowie durch Typ-, Medienqualitäts- und Servicequalitätsmerkmale \mathbf{t} , \mathbf{m} und \mathbf{s} gekennzeichnet sind. Eine Kante

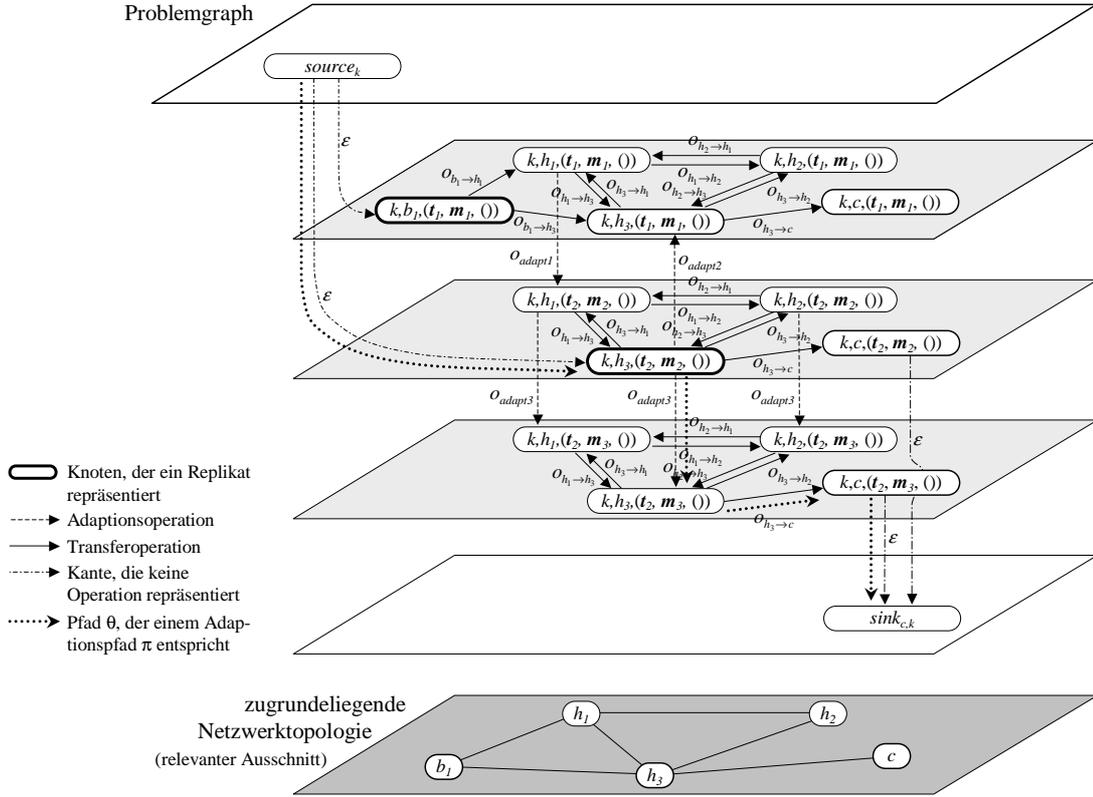
$$(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{o} (k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \quad (4.1)$$

repräsentiert die Anwendung einer Operation $o \in O$ auf die Ausprägung von k mit den Typ-, Medienqualitäts- und Servicequalitätsmerkmalen \mathbf{t} , \mathbf{m} und \mathbf{s} im Netzwerkknoten n , wobei $o(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = (n', (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$.

Ein einfaches Beispiel für einen solchen Problemgraphen $\Gamma_{c,k}$ ist in Abbildung 4.1 illustriert. Er ist dreidimensional als horizontal geschichteter Graph dargestellt. Übereinander liegende Knoten in den hellgrau dargestellten Ebenen stehen für unterschiedliche Ausprägungen von k am selben Netzwerkknoten $n \in N$. Knoten innerhalb der selben hellgrauen Ebene repräsentieren Ausprägungen von k mit den selben Typ-, Medienqualitäts- und Servicequalitätsmerkmalen \mathbf{t} , \mathbf{m} und \mathbf{s} an unterschiedlichen Netzwerkknoten. Der für die Suche nach einem optimalen Adaptionspfad für eine gegebene Anfrage (c, k) relevante Teil der Netzwerktopologie (alle Originalserver, die kein Replikat von k halten, und alle Clientknoten außer c wurden weggelassen) ist zur Orientierung unterhalb des Problemgraphen gezeigt (dunkelgraue Ebene). Die in Weiß dargestellten Ebenen mit den Knoten $source_k$ und $sink_{c,k}$ werden weiter unten erläutert. Kanten, die als durchgezogene Pfeile dargestellt sind, repräsentieren die Anwendung von Transferoperationen. Im Beispiel in Abbildung 4.1 werden keine Servicequalitätsmerkmale berücksichtigt (d.h. $S = \{()\}$) und es wird davon ausgegangen, dass Netzwerktransfers sich nicht auf Medienqualitätsmerkmale auswirken. Deshalb verlaufen alle Kanten, die die Anwendung von Transferoperationen repräsentieren, horizontal innerhalb einer Ebene. Vertikale Kanten, die die einzelnen Ebenen miteinander verbinden, stehen für die Anwendung von Adaptionsoptionen. Sie sind durch gestrichelte Pfeile dargestellt.

Ein komplexeres Beispiel, in welchem auch Servicequalitätsmerkmale berücksichtigt werden, ist in Abbildung 4.2 auf Seite 76 dargestellt. Zur besseren Übersicht sind die (hellgrau dargestellten) Ebenen, in denen Knoten liegen, die Ausprägungen des Inhaltsobjektes repräsentieren, gruppiert, wobei die Ausprägungen der Typ- und Medienqualitätsmerkmale \mathbf{t} und \mathbf{m} aller Knoten innerhalb einer Gruppe identisch sind. Sie repräsentieren also die gleichen Repräsentationen

¹Um Verwechslungen zwischen den Knoten $n \in N$ im Systemmodell Σ und den Knoten des Problemgraphen $\Gamma_{c,k}$ zu vermeiden, werden die Knoten $n \in N$ im Systemmodell Σ hier als Netzwerkknoten bezeichnet.

Abbildung 4.1: Problemgraph $\Gamma_{c,k}$, Beispiel 1 (keine Servicequalitätsmerkmale berücksichtigt)

des Objektes an unterschiedlichen Knoten mit unterschiedlicher Servicequalität. Beim Graphen in Abbildung 4.2 handelt es sich genau genommen nur um einen Ausschnitt des gesamten Problemgraphen. Wenn Transferoperationen wiederholt angewendet werden können, was jeweils zu einer Erhöhung der Verzögerung (Servicequalitätsmerkmal `delay`, vgl. Beispiel 3.3 auf Seite 59) führt, kann die Zahl der Ebenen und somit die Zahl der Knoten im Graphen unendlich sein. In der Praxis sind jedoch im Allgemeinen nicht unendlich viele Knoten relevant. Wenn zum Beispiel für das Servicequalitätsmerkmal `delay` ein Schwellwert für untolerierbare Verzögerungen existiert (vgl. Beispiel 3.8 auf Seite 63), ist es nicht sinnvoll, in $\Gamma_{c,k}$ Knoten zu berücksichtigen, deren Servicequalitätsmerkmal `delay` einen Wert oberhalb dieses Schwellwertes hat. Eine weitere interessante Eigenschaft von Problemgraphen, die ein Systemmodell repräsentieren, das Verzögerungszeiten als Qualitätsmerkmal berücksichtigt, ist Zyklensfreiheit. Die Zyklensfreiheit liegt darin begründet, dass jede Adaptionsoperation und jeder Netzwerktransfer Zeit kostet und die dadurch entstehenden Verzögerungen durch keine Operation reversibel sind.

Ein Pfad

$$(k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} (k, n_1, (\mathbf{t}_1, \mathbf{m}_1, \mathbf{s}_1)) \xrightarrow{o_2} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)) \quad (4.2)$$

in einem Problemgraphen $\Gamma_{c,k}$ ist eine Folge von Anwendungen von Adaptionsoperationen. Repräsentiert der Startknoten $(k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0))$ des Pfades ein Replikat des Objektes k (d.h. $((k, n_0, (\mathbf{t}_0, \mathbf{m}_0)) \in P) \wedge (\mathbf{s}_0 = \mathbf{s}_{\text{init}})$), entspricht der Pfad im Problemgraphen $\Gamma_{c,k}$ einem Adaptionspfad π in Σ . Wenn darüber hinaus der Zielknoten $(k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p))$ eine Ausprägung von k

- im Clientknoten c ($n_p = c$),
- die der Client, der durch c repräsentiert wird, darstellen kann $((\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p) \in \text{mhc}(c))$,

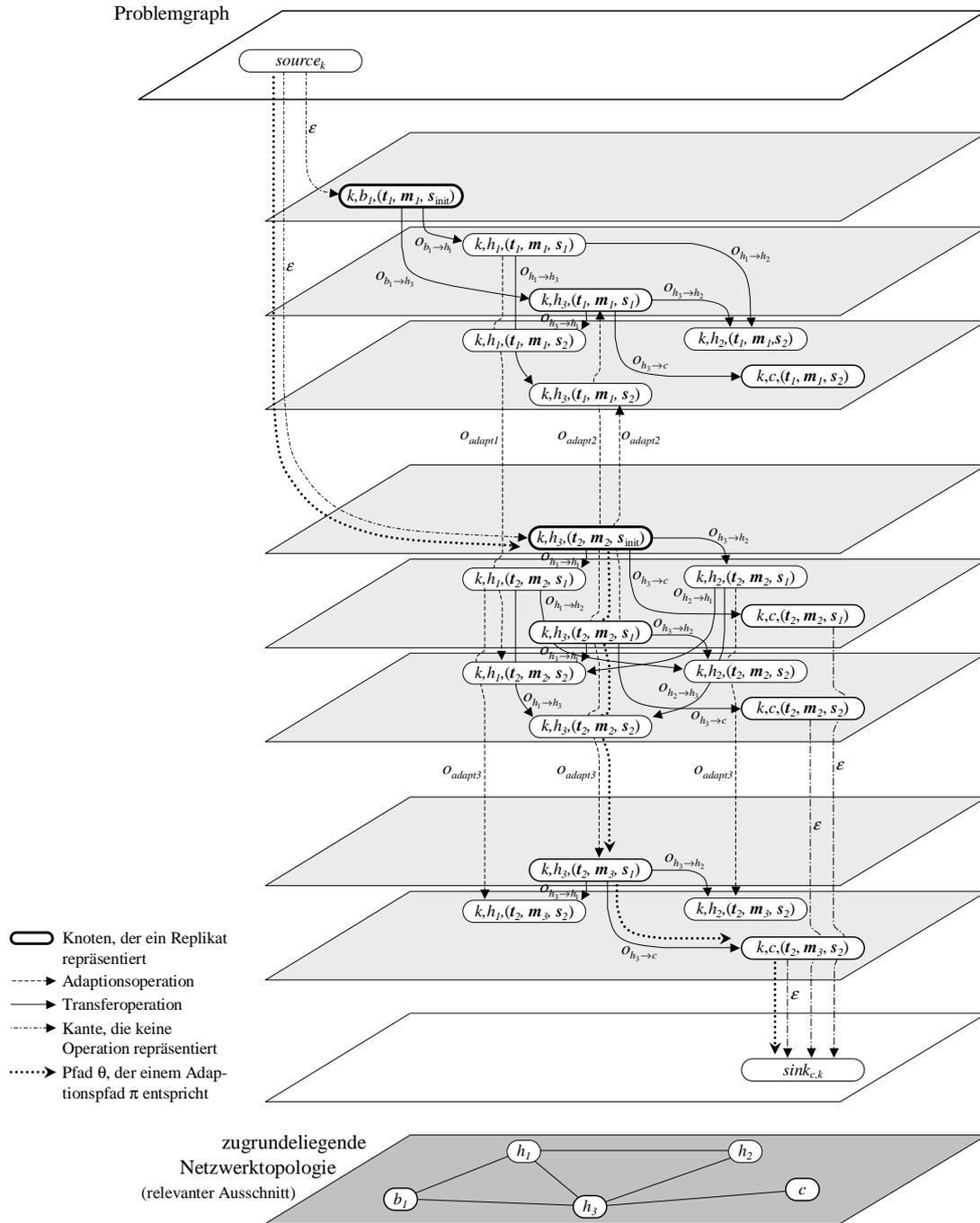


Abbildung 4.2: Problemgraph $\Gamma_{c,k}$, Beispiel 2 (Servicequalitätsmerkmale berücksichtigt)

- mit hinreichender Qualität ($qf(c, k, (\mathbf{m}_p, \mathbf{s}_p)) > 0$)

beschreibt, dann ist der entsprechende Adaptionspfad π in der Lage, die Anfrage (c, k) zu beantworten (π *satisfies* (c, k)). Deshalb wird im Folgenden davon gesprochen, dass ein Knoten $(k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p))$ eine Anfrage (c, k) befriedigt, wenn er die genannten Bedingungen erfüllt.

Mögliche Lösungen für das Problem der Suche nach einem optimalen Adaptionspfad $\pi_{c,k}$ sind alle Pfade im Problemgraphen $\Gamma_{c,k}$ zwischen einem der Knoten, die ein Replikat repräsentieren, und einem der Knoten, die die Anfrage (c, k) befriedigen. Um das Problem als das klassische graphentheoretische Problem der Suche nach dem kürzesten Pfad zwischen einem gegebenen Startknoten und einem gegebenen Zielknoten lösen zu können, wird der Problemgraph $\Gamma_{c,k}$ um die Knoten $source_k$ und $sink_{c,k}$ ergänzt, die bei der Pfadsuche als Start- und Zielknoten dienen sollen. Die Knoten $source_k$ und $sink_{c,k}$ repräsentieren im Gegensatz zu den anderen Knoten des Problemgraphen $\Gamma_{c,k}$ keine Ausprägungen des Objektes k . Sie werden deshalb im Folgenden als virtueller Startknoten bzw. virtueller Zielknoten bezeichnet. Somit ist die Knotenmenge $V(\Gamma_{c,k})$ des Problemgraphen $\Gamma_{c,k}$ als die Menge aus

- allen Tupeln $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$, die eine Ausprägung des angeforderten Objektes k repräsentieren, die durch einen Adaptionspfad $\pi \in \Pi_P$ zu einem für das Problem relevanten Netzwerkknoten n mit den Typ-, Medienqualitäts- und Servicequalitätsmerkmalen \mathbf{t} , \mathbf{m} und \mathbf{s} zugestellt werden kann,
- einem virtuellen Startknoten $source_k$ und
- einem virtuellen Zielknoten $sink_{c,k}$

definiert:

$$V(\Gamma_{c,k}) = \{ (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \mid (n \in ((N - C) \cup \{c\})) \wedge (\exists \pi \in \Pi_P. (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = leaf(\pi)) \} \cup \{ source_k, sink_{c,k} \} \quad (4.3)$$

Der Knoten $source_k$ sei durch eine ausgehende Kante mit jedem Knoten des Problemgraphen $\Gamma_{c,k}$ verbunden, der ein Replikat von k repräsentiert. Der Knoten $sink_{c,k}$ habe eine eingehende Kante von jedem Knoten in $\Gamma_{c,k}$, der die Anfrage (c, k) befriedigt. Die Kanten, die im Knoten $source_k$ entspringen oder im Knoten $sink_{c,k}$ münden, repräsentieren – anders als die übrigen Kanten im Problemgraphen – keine Anwendungen von Adaption- oder Transferoperationen $o \in O$. Sie werden deshalb mit ε beschriftet. Somit umfasst die Kantenmenge $E(\Gamma_{c,k})$ des Problemgraphen

- je eine Kante $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{o} (k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$ für jedes Paar von Knoten $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ und $(k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$, die Ausprägungen des Inhaltsobjektes k repräsentieren, und jede Adaption- oder Transferoperation $o \in O$, die $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ in $(k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$ überführt,
- je eine Kante $source_k \xrightarrow{\varepsilon} (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ für jedes Replikat des Objektes k , die den virtuellen Startknoten mit dem Knoten verbindet, der das Replikat in $\Gamma_{c,k}$ repräsentiert, und
- je eine Kante $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{\varepsilon} sink_{c,k}$ für jeden Knoten $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ in $\Gamma_{c,k}$, der die Anfrage (c, k) befriedigt, die $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ mit dem virtuellen Zielknoten in $\Gamma_{c,k}$ verbindet.

Formal ist die Kantenmenge des Problemgraphen $\Gamma_{c,k}$ folgendermaßen definiert:

$$\begin{aligned}
E(\Gamma_{c,k}) = & \{ (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{o} (k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \mid & (4.4) \\
& ((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in V(\Gamma_{c,k})) \wedge ((k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \in V(\Gamma_{c,k})) \wedge \\
& (\exists o \in O. o(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = (n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')))) \} & (*) \\
\cup & \{ source_k \xrightarrow{\varepsilon} (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \mid \\
& ((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in V(\Gamma_{c,k})) \wedge ((k, n, (\mathbf{t}, \mathbf{m})) \in P) \wedge (\mathbf{s} = \mathbf{s}_{init})) \} & (**) \\
\cup & \{ (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{\varepsilon} sink_{c,k} \mid \\
& ((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in V(\Gamma_{c,k})) \wedge (n = c) \wedge \\
& ((\mathbf{t}, \mathbf{m}, \mathbf{s}) \in mh(c)) \wedge (qf(c, k, (\mathbf{m}, \mathbf{s})) > 0) \} & (***)
\end{aligned}$$

Wenn es eine Kante $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{\varepsilon} sink_{c,k} \in E(\Gamma_{c,k})$ (gemäß Formel 4.4***) gibt, ist der resultierende Problemgraph $\Gamma_{c,k}$ ein zusammenhängender Graph, in dem jeder Knoten $v \in V(\Gamma_{c,k})$ vom virtuellen Startknoten $source_k$ erreichbar ist. Andernfalls gibt es keinen Adaptionpfad, der die Anfrage (c, k) beantworten kann.

Jeder Pfad

$$\theta = source_k \xrightarrow{\varepsilon} (k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)) \xrightarrow{\varepsilon} sink_{c,k} \quad (4.5)$$

in $\Gamma_{c,k}$ zwischen dem virtuellen Start- und Zielknoten entspricht einem Adaptionpfad

$$\pi = (k, n_0, (\mathbf{t}_0, \mathbf{m}_0, \mathbf{s}_0)) \xrightarrow{o_1} \dots \xrightarrow{o_p} (k, n_p, (\mathbf{t}_p, \mathbf{m}_p, \mathbf{s}_p)), \quad (4.6)$$

der die Anfrage (c, k) beantworten kann. Durch eine geeignete Wahl der Gewichtsfunktion

$$w : E(\Gamma_{c,k}) \rightarrow \mathbb{R}, \quad (4.7)$$

die jeder Kante $e \in E(\Gamma_{c,k})$ im Problemgraphen $\Gamma_{c,k}$ eine Gewichtung $w(e)$ zuordnet, kann das Problem der Suche nach einem optimalen Adaptionpfad $\pi_{c,k}$ zur Beantwortung der Anfrage (c, k) als das klassische graphentheoretische Problem der Suche nach dem kürzesten Pfad zwischen dem virtuellen Startknoten $source_k$ und dem virtuellen Zielknoten $sink_{c,k}$ in $\Gamma_{c,k}$ gelöst werden. Eine solche Gewichtsfunktion ist durch die folgende Definition gegeben:

- Jede Kante, die für die Anwendung einer Adaption- oder Transferoperation steht (Formel 4.4*), ist mit der Summe aus den Kosten der Operation und den Einbußen an Erlös infolge des Qualitätsverlustes durch die Operationsanwendung gewichtet.
- Die Kanten, die den virtuellen Startknoten $source_k$ mit einem Knoten $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ verbinden (Formel 4.4**), der ein Replikat repräsentiert, sind mit dem negierten Erlös gewichtet, der durch die Auslieferung des Inhaltsobjektes k mit den Typ-, Medienqualitäts- und Servicequalitätsmerkmalen \mathbf{t} , \mathbf{m} und \mathbf{s} erwirtschaftet werden kann. Auf diese Weise wird der unterschiedlichen Qualität der verschiedenen Replikate Rechnung getragen.
- Die Gewichtung der übrigen Kanten (Formel 4.4***) ist 0.

Formal ausgedrückt:

$$w(e) = \begin{cases} cf_o(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) + price_{c,k} \cdot (qf(c, k, (\mathbf{m}, \mathbf{s})) - qf(c, k, (\mathbf{m}', \mathbf{s}'))) & \text{falls } e = (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{o} (k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \\ -price_{c,k} \cdot qf(c, k, (\mathbf{m}, \mathbf{s})) & \text{falls } e = source_k \xrightarrow{\varepsilon} (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \\ 0 & \text{falls } e = (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{\varepsilon} sink_{c,k} \end{cases} \quad (4.8)$$

Die Länge eines Pfades θ in $\Gamma_{c,k}$ (Formel 4.5), der einem Adaptionspfad π in Σ (Formel 4.6) entspricht, ist:

$$\begin{aligned}
l(\theta) &= \sum_{e \in E(\theta)} w(e) \\
&= -price_{c,k} \cdot qf(c, k, (\mathbf{m}_0, \mathbf{s}_0)) \\
&\quad + \sum_{i=1}^p \left(cf_{o_i}(k, n_{i-1}, (\mathbf{t}_{i-1}, \mathbf{m}_{i-1}, \mathbf{s}_{i-1})) \right. \\
&\quad \quad \left. + price_{c,k} \cdot (qf(c, k, (\mathbf{m}_{i-1}, \mathbf{s}_{i-1})) - qf(c, k, (\mathbf{m}_i, \mathbf{s}_i))) \right) \\
&\quad + 0 \\
&= -price_{c,k} \cdot qf(c, k, (\mathbf{m}_0, \mathbf{s}_0)) \\
&\quad + \sum_{i=1}^p cf_{o_i}(k, n_{i-1}, (\mathbf{t}_{i-1}, \mathbf{m}_{i-1}, \mathbf{s}_{i-1})) \\
&\quad + \sum_{i=1}^p price_{c,k} \cdot (qf(c, k, (\mathbf{m}_{i-1}, \mathbf{s}_{i-1})) - qf(c, k, (\mathbf{m}_i, \mathbf{s}_i))) \\
&= -price_{c,k} \cdot qf(c, k, (\mathbf{m}_0, \mathbf{s}_0)) + cost_\pi \\
&\quad + price_{c,k} \cdot (qf(c, k, (\mathbf{m}_0, \mathbf{s}_0)) - qf(c, k, (\mathbf{m}_p, \mathbf{s}_p))) \\
&= cost_\pi - price_{c,k} \cdot qf(c, k, (\mathbf{m}_p, \mathbf{s}_p)) \\
&= cost_\pi - price_{c,k} \cdot qual_{\pi,c} \\
&= cost_\pi - rev_{\pi,c} \\
&= -profit_{\pi,c}
\end{aligned} \tag{4.9}$$

Folglich entspricht ein kürzester Pfad in $\Gamma_{c,k}$ zwischen dem virtuellen Startknoten $source_k$ und dem virtuellen Zielknoten $sink_{c,k}$ einem Adaptionspfad π in Σ , der die Anfrage (c, k) beantworten kann und bei dem der Gewinn $profit_{\pi,c}$ größer oder gleich dem Gewinn aller anderen Adaptionspfade ist, die (c, k) beantworten können. Ein solcher Adaptionspfad ist gemäß der Problemformulierung 3.3 ein optimaler Adaptionspfad $\pi_{c,k}$ zur Beantwortung der Anfrage (c, k) .

4.1.2 Algorithmen zur Ermittlung eines kürzesten Pfades

Auf Grundlage der im letzten Abschnitt vorgestellten Abbildung des Problems kann ein herkömmlicher Algorithmus zur Ermittlung eines kürzesten Pfades in einem gewichteten gerichteten Graphen zur Komposition eines optimalen Adaptionspfades verwendet werden. Je nach Eigenschaften des Problemgraphen sind dafür verschiedene Kürzester-Pfad-Algorithmen geeignet. Im Folgenden werden eine Variante des Dijkstra-Algorithmus, eine Variante des Bellman-Ford-Algorithmus und ein DAG-SP-Algorithmus (Bestimmung eines kürzesten Pfades in gerichteten zyklenfreien Graphen) beschrieben und ihre Einsatzmöglichkeiten diskutiert.

Dijkstra-Algorithmus

Der Dijkstra-Algorithmus [Dij59] ermittelt einen kürzesten Pfad in einem Graphen Γ unter der Bedingung, dass keine der Kanten im Graphen negativ gewichtet ist, d.h. $\forall e \in E(\Gamma). w(e) \geq 0$. In einem Problemgraphen $\Gamma_{c,k}$ zur Bestimmung eines optimalen Adaptionspfades ist diese Bedingung jedoch nicht erfüllt. Zumindest alle Kanten $source_k \xrightarrow{\varepsilon} (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in E(\Gamma_{c,k})$, die vom virtuellen Startknoten $source_k$ ausgehen, sind gemäß Formel 4.8 mit $-price_{c,k} \cdot qf(c, k, (\mathbf{m}, \mathbf{s}))$ negativ gewichtet.

Darüber hinaus können auch Kanten $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{o} (k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$, die die Anwendung einer Adaption- oder Transferoperation repräsentieren, negativ gewichtet sein. Das setzt voraus, dass $qf(c, k, (\mathbf{m}', \mathbf{s}')) > qf(c, k, (\mathbf{m}, \mathbf{s}))$, das heißt, dass sich die Qualität durch die Anwendung einer Operation erhöht (vgl. Formel 4.8). Lazzaro und Wawrzyniek zum Beispiel schlagen in [LW99] eine Operation vor, durch die die Qualität von komprimierten JPEG-Bildern durch die Reduktion von Block-Artefakten erhöht wird. Im Allgemeinen erhöht sich die Qualität durch die Anwendung einer Adaptionoperation jedoch nicht, da durch vorherige Adaptionoperationen verloren gegangene Informationen nicht wiederherstellbar sind.

Algorithmus 4.1 Dijkstra-Algorithmus

```

1   $Perm := \emptyset$  Initialisierung;
2   $Temp := \{source_k\}$ 
3   $d_{source_k} := 0$ 
4   $p_{source_k} := nil$ 
5  while  $Temp \neq \emptyset$  do solange es Knoten in der Menge Temp gibt,
6     suche  $x \in Temp$ , für das  $d_x = \min \{d_y \mid y \in Temp\}$  einen neuen Arbeitsknoten x suchen,
7      $Temp := Temp - \{x\}$  aus der Menge Temp entnehmen und
8      $Perm := Perm \cup \{x\}$  in die Menge Perm übernehmen;
9     if  $x = sink_{c,k}$  then wenn der Arbeitsknoten der Zielknoten sink_{c,k} ist,
10        halt dann wurde ein kürzester Pfad gefunden;
11    end if
12    foreach  $x \xrightarrow{\xi} y \in E(\Gamma_{c,k})$  do über allen Kanten, die vom Arbeitsknoten ausgehen, iterieren;
13        if  $y \notin Perm$  then wenn für den Endknoten y der Kante
noch kein kürzester Pfad ermittelt wurde,
14        if  $(\neg(\text{defined } d_y)) \vee (d_y > d_x + w(x \xrightarrow{\xi} y))$  then den jeweils kürzesten Pfad von
source_k nach y ermitteln, der außer y
nur Knoten aus der Menge Perm enthält,
und diesen Pfad in d_y und p_y übernehmen
15             $d_y := d_x + w(x \xrightarrow{\xi} y)$ 
16             $p_y := x$ 
17        end if
18         $Temp := Temp \cup \{y\}$ 
19    end if
20 end foreach
21 end while

```

Wenn ausschließlich die Kanten $source_k \xrightarrow{\varepsilon} (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in E(\Gamma_{c,k})$, die vom virtuellen Startknoten $source_k$ ausgehen, negativ gewichtet sind, alle anderen Kanten eine nichtnegative Wichtung haben und $source_k$ (wie in $\Gamma_{c,k}$) keine eingehenden Kanten aufweist, so ist der Dijkstra-Algorithmus aber ebenfalls anwendbar, wie in Anhang A bewiesen wird.

In dieser Arbeit wird eine Variante des Dijkstra-Algorithmus verwendet, die sich an [Tin76] orientiert. Bei dieser Variante muss die Knotenmenge $V(\Gamma_{c,k})$ des Graphen nicht im Voraus bekannt sein. Anstelle dessen wird die Teilmenge der relevanten Knoten durch den Algorithmus konstruiert. Knoten, die für die Lösung nicht relevant sind, werden nicht ermittelt. Somit kann der Algorithmus auch bei einer unendlichen Knotenmenge $V(\Gamma_{c,k})$ eingesetzt werden.

Die Formulierung des Dijkstra-Algorithmus ist durch Algorithmus 4.1 gegeben. Algorithmus 4.1 verwendet zwei Mengen von Knoten, $Perm$ und $Temp$. $Perm$ enthält nach jeder Iteration der **while**-Schleife in Zeile 5 die Menge der Knoten x , für die bereits ein kürzester Pfad der Länge d_x vom Startknoten $source_k$ ermittelt wurde. Für die Knoten $x \in Temp$ wurde jeweils ein Pfad (der Länge d_x) vom Startknoten $source_k$ ermittelt, ohne dass dieser bereits der kürzeste Pfad sein muss. Für alle Knoten $x \in (Perm \cup Temp)$ verweist p_x jeweils auf den Vorgängerknoten von x in dem Pfad, dessen Länge in d_x gespeichert ist. Für alle anderen Knoten ($x \notin (Perm \cup Temp)$) wurde bis zum aktuellen Iterationsschritt noch kein Pfad ermittelt (d_x und p_x nicht definiert). Sie waren bis zum aktuellen Iterationsschritt für die Lösung irrelevant. Die Menge dieser Knoten ist unbenannt und potentiell unendlich.

Die Menge $Temp$ wird als Menge $\{source_k\}$ mit $d_{source_k} = 0$ und $p_{source_k} = nil$ (kein Vorgängerknoten) initialisiert. In jedem Iterationsschritt wird aus der Menge $Temp$ ein Knoten x mit dem kleinsten d_x als Arbeitsknoten ausgewählt und in die Menge $Perm$ übernommen. Für jeden Nachfolgerknoten y des Arbeitsknoten x , für den noch kein kürzester Pfad ermittelt wurde ($y \notin Perm$), wird der kürzeste Pfad bestimmt, der außer y nur Knoten aus der Menge $Perm$ enthält. Die Länge dieses Pfades wird in d_y gespeichert, p_y verweist anschließend auf den Vorgänger von y in diesem Pfad. Darüber hinaus werden die Nachfolgerknoten von x , die bisher noch keine Relevanz für die Lösung hatten, in die Menge $Temp$ aufgenommen.

Der Algorithmus terminiert, wenn der Zielknoten $sink_{c,k}$ zum Arbeitsknoten und damit Element der Menge $Perm$ wird. In diesem Fall ist $d_{sink_{c,k}}$ die Länge des kürzesten Pfades von $source_k$ nach $sink_{c,k}$. Der Pfad ist durch die Rückwärtsverweise p_x der Knoten $x \in Perm$ ausgehend von $p_{sink_{c,k}}$ gegeben. Existiert kein Adaptionpfad, der die Anfrage (c, k) beantworten kann, terminiert der Algorithmus nur, wenn die Knotenmenge $V(\Gamma_{c,k})$ endlich ist. In diesem Fall wird $Temp = \emptyset$ bevor $sink_{c,k}$ zum Arbeitsknoten wird.

Um die Suche nach einem neuen Arbeitsknoten in $Temp$ effektiv zu gestalten, kann $Temp$ als Fibonacci-Heap [FT87] implementiert werden, bei dem die einzelnen Knoten $x \in Temp$ entsprechend d_x sortiert sind. Bei dieser Implementierung ist die Zeitkomplexität der Suche nach einem neuen Arbeitsknoten $O(\log |Temp|)$. Das Hinzufügen eines Knotens zur Menge $Temp$ hat konstante Komplexität ($O(1)$). Die Werte d_x und p_x können in einer Hash-Tabelle abgelegt werden, die mit x indiziert ist. Damit ergibt sich für die Zeitkomplexität des Algorithmus eine obere Schranke von

$$O(|V_{rel}(\Gamma_{c,k})| \cdot (\log |V_{rel}(\Gamma_{c,k})| + |O|)). \quad (4.10)$$

Dabei steht $V_{rel}(\Gamma_{c,k}) \subseteq V(\Gamma_{c,k})$ für die Menge aller Knoten, die für die Suche nach dem kürzesten Pfad in $\Gamma_{c,k}$ relevant sind. Das sind alle Knoten, die während der Abarbeitung des Algorithmus in die Menge $Temp$ aufgenommen werden:

$$V_{rel}(\Gamma_{c,k}) = \{source_k\} \cup \{y \mid (x \xrightarrow{\xi} y \in E(\Gamma_{c,k})) \wedge (\delta(source_k, x) \leq \delta(source_k, sink_{c,k}))\} \quad (4.11)$$

Dabei gibt die Funktion δ die jeweilige Distanz zwischen zwei Knoten an:

$$(\delta(x, y) = l(\theta)) \Leftrightarrow (\theta \text{ ist der kürzeste Pfad von } x \text{ nach } y) \quad (4.12)$$

O ist die Menge von Operationen im System Σ . Die Kardinalität $|O|$ ist eine obere Grenze der Zahl der ausgehenden Kanten eines Knotens in $\Gamma_{c,k}$ (mit Ausnahme des virtuellen Startknotens $source_k$, der bis zu $|V_{rel}(\Gamma_{c,k})|$ ausgehende Kanten haben kann). Wegen $|V_{rel}(\Gamma_{c,k})| \leq (|B| + |H|)|T||M||S|$ (aus Formel 4.11 und Formel 4.3 auf Seite 77) ist auch

$$O(|B||H||T||M||S| \cdot (\log(|B||H||T||M||S|) + |O|)) \quad (4.13)$$

eine obere Schranke der Zeitkomplexität des Dijkstra-Algorithmus.

Bellman-Ford-Algorithmus

Im Gegensatz zum Dijkstra-Algorithmus kann mittels des Bellman-Ford-Algorithmus [Bel57, FF62] ein kürzester Pfad auch dann korrekt ermittelt werden, wenn beliebige Kanten im Problemgraphen negativ gewichtet sind².

Um den Aufwand zur Konstruktion des Problemgraphen im Voraus zu vermeiden, wird hier eine Variante des Bellman-Ford-Algorithmus nach [Tin76] verwendet, bei der die Knotenmenge $V(\Gamma_{c,k})$ des Problemgraphen schrittweise durch den Algorithmus konstruiert wird. Anders als beim Dijkstra-Algorithmus sind für die Ermittlung eines kürzesten Pfades durch den Bellman-Ford-Algorithmus alle Knoten des Problemgraphen relevant. Deshalb ist er nicht einsetzbar, wenn $|V(\Gamma_{c,k})| = \infty$.

Die Formulierung des Bellman-Ford-Algorithmus ist in Algorithmus 4.2 dargestellt. Der Algorithmus verwaltet eine Arbeitsmenge $Wset$ von Knoten, die die Knoten des Problemgraphen $\Gamma_{c,k}$ enthält, welche im aktuellen Iterationsschritt examiniert werden. $Wset$ wird mit $\{source_k\}$ initialisiert. In jeder Iteration i der **while**-Schleife in Zeile 4 wird die Menge der Nachfolgerknoten aller Knoten der Arbeitsmenge $Wset$ bestimmt. Für jeden Nachfolgerknoten y wird der kürzeste Pfad vom Startknoten $source_k$ ermittelt, der maximal $(i + 1)$ Knoten enthält (Beweis s. [Tin76]). Die Länge des Pfades wird in d_y und der Vorgänger von y in diesem Pfad in p_y gespeichert. Hat sich d_y im aktuellen Iterationsschritt i verringert, so wird y in eine Menge

²Allerdings muss sichergestellt sein, dass der Graph keine negativ gewichteten Zyklen hat. Andernfalls ist das Problem nicht lösbar, denn der negativ gewichtete Zyklus müsste in jedem kürzesten Pfad unendlich oft durchlaufen werden.

Algorithmus 4.2 Bellman-Ford-Algorithmus

```

1   $Wset := \{source_k\}$  Initialisierung;
2   $d_{source_k} := 0$ 
3   $p_{source_k} := nil$ 
4  while  $Wset \neq \emptyset$  do solange die Arbeitsmenge nicht leer ist,
5     $Wtmp := \emptyset$ 
6    foreach  $x \in Wset$  do über allen Knoten der Arbeitsmenge iterieren;
7      foreach  $x \xrightarrow{\xi} y \in E(\Gamma_{c,k})$  do für jede Kante, die von einem
Knoten der Arbeitsmenge ausgeht,
8      if  $(\neg(\text{defined } d_y)) \vee (d_y > d_x + w(x \xrightarrow{\xi} y))$  then den jeweils kürzesten Pfad zum
jeweiligen Endknoten  $y$  der Kante ermitteln,
der maximal  $i + 1$  Knoten umfasst
( $i =$  Zahl der abgeschlossenen Itera-
tionen der while-Schleife in Zeile 4),
und diesen Pfad in  $d_y$  und  $p_y$  übernehmen;
9       $d_y := d_x + w(x \xrightarrow{\xi} y)$ 
10      $p_y := x$ 
11      $Wtmp := Wtmp \cup \{y\}$  Endknoten  $y$  ggf. in die Arbeitsmenge
für die nächste Iteration aufnehmen
12   end if
13   end foreach
14 end foreach
15    $Wset := Wtmp$ 
16 end while

```

$Wtmp$ aufgenommen, die im nachfolgenden Iterationsschritt $(i + 1)$ zur Arbeitsmenge $Wset$ wird. Wenn der Algorithmus terminiert, ist für alle Knoten $x \in V(\Gamma_{c,k})$, für die ein d_x definiert ist (**defined** x), d_x gleich der Länge des kürzesten Pfades zwischen $source_k$ und x . Für die übrigen Knoten existiert kein Pfad.

Auf den Beweis der Korrektheit des Algorithmus kann hier verzichtet werden. Die Korrektheit der verwendeten Variante wird in [Tin76] gezeigt.

Die Zahl der Iterationen des Bellman-Ford-Algorithmus entspricht der maximalen Zahl von Knoten in einem kürzesten Pfad. Da kürzeste Pfade zyklensfrei sind, gibt es maximal $|V(\Gamma_{c,k})|$ Iterationen. In jedem Iterationsschritt wird jede Kante des Graphen $\Gamma_{c,k}$ maximal einmal verfolgt. Folglich ist die Zeitkomplexität des Algorithmus durch

$$O(|V(\Gamma_{c,k})| \cdot |E(\Gamma_{c,k})|) \leq O(|V(\Gamma_{c,k})|^2 \cdot |O|) \quad (4.14)$$

nach oben beschränkt. Dabei ist O die Menge von Operationen im System Σ . Die Kardinalität $|O|$ ist eine obere Grenze der Zahl der ausgehenden Kanten eines Knotens in $\Gamma_{c,k}$ (mit Ausnahme des virtuellen Startknotens $source_k$, der bis zu $|V(\Gamma_{c,k})|$ ausgehende Kanten haben kann; da der virtuelle Startknoten $source_k$ keine eingehenden Kanten hat, wird er aber nur einmal, in der ersten Iteration, examiniert). Wegen $|V(\Gamma_{c,k})| \leq (|B| + |H|)|T||M||S|$ (Formel 4.3 auf Seite 77) gibt auch

$$O(|B|^2|H|^2|T|^2|M|^2|S|^2|O|) \quad (4.15)$$

eine obere Schranke der Zeitkomplexität des Bellman-Ford-Algorithmus an.

DAG-SP-Algorithmus

Der DAG-SP-Algorithmus (*Directed Acyclic Graph Shortest Path Algorithm*) dient der Bestimmung von kürzesten Pfaden in gerichteten zyklensfreien Graphen. Durch Ausnutzung der Zyklensfreiheit kann ein kürzester Pfad mit linearer Komplexität bestimmt werden. Ebenso wie der Bellman-Ford-Algorithmus kann der DAG-SP-Algorithmus auch eingesetzt werden, wenn der Problemgraph negativ gewichtete Kanten enthält. Der Algorithmus terminiert nur, wenn die Knotenmenge $V(\Gamma_{c,k})$ endlich ist.

Algorithmus 4.3 DAG-SP-Algorithmus

```

1  sub  $dfs(x)$  Unterprogramm: rekursive Tiefensuche;
2    foreach  $x \xrightarrow{\xi} y \in E(\Gamma_{c,k})$  do über allen Kanten, die vom Arbeitsknoten  $x$  ausgehen, iterieren;
3      if  $y \notin Temp$  then wenn der Endknoten  $y$  der Kante noch nicht examiniert wurde,
4         $dfs(y)$  dann Tiefensuche ausgehend vom Knoten  $y$  durchführen;
5      end if
6    end foreach nachdem alle Nachfolger des Arbeitsknoten examiniert worden sind,
7     $Temp := Temp \cup \{x\}$  den Arbeitsknoten in die topologisch sortierte Liste aufnehmen;
8     $t_x := cnt$ 
9     $cnt := cnt + 1$ 
10 end sub
11  $Temp := \emptyset$  Initialisierung;
12  $Perm := \emptyset$ 
13  $cnt := 0$ 
14  $dfs(source_k)$  topologisch sortierte Liste aller von  $source_k$  erreichbaren Knoten des
Problemgraphen konstruieren (im Anschluss ist  $Temp$  die Menge der Knoten
in der Liste und  $t_x$  der jeweilige Listenindex für jedes  $x \in Temp$ );
15 while  $Temp \neq \emptyset$  do solange es Knoten in  $Temp$  gibt,
16    $suche\ x \in Temp$ , für das  $t_x = \max\{t_y \mid y \in Temp\}$  einen Arbeitsknoten  $x$  suchen,
17    $Temp := Temp - \{x\}$  aus der Menge  $Temp$  entnehmen und
18    $Perm := Perm \cup \{x\}$  in die Menge  $Perm$  übernehmen;
19   if  $x = sink_{c,k}$  then wenn der Arbeitsknoten der Zielknoten  $sink_{c,k}$  ist,
20     halt dann wurde ein kürzester Pfad gefunden;
21   end if
22   foreach  $x \xrightarrow{\xi} y \in E(\Gamma_{c,k})$  do über allen Kanten, die vom Arbeitsknoten ausgehen, iterieren
23     if  $(\neg(\text{defined } d_y)) \vee (d_y > d_x + w(x \xrightarrow{\xi} y))$  then und dabei den jeweils kürzesten Pfad
zum Endknoten  $y$  ermitteln, der außer  $y$ 
nur Knoten aus der Menge  $Perm$  enthält,
und diesen Pfad in  $d_y$  und  $p_y$  übernehmen
24        $d_y := d_x + w(x \xrightarrow{\xi} y)$ 
25        $p_y := x$ 
26     end if
27   end foreach
28 end while

```

Der DAG-SP-Algorithmus (nach [CLRS01]) ist in Algorithmus 4.3 dargestellt. Er umfasst zwei Phasen. In der ersten Phase (Zeile 1 bis 14) wird durch rekursive Tiefensuche (Unterprogramm dfs) eine topologische Sortierung aller vom virtuellen Startknoten $source_k$ erreichbaren Knoten der Knotenmenge $V(\Gamma_{c,k})$ ermittelt. Knoten, die nicht vom virtuellen Startknoten $source_k$ erreichbar sind, sind bei der Suche nach einem kürzesten Pfad nicht relevant. Nach Abschluss der ersten Phase ist $Temp$ die Menge aller von $source_k$ erreichbaren Knoten und für alle $x \in Temp$ ist t_x der Index von x in der topologischen Sortierung, wobei der Startknoten $source_k$ den größten Index hat.

In der zweiten Phase des Algorithmus (Zeile 15 bis 28) wird nacheinander in umgekehrter Reihenfolge der topologischen Sortierung beginnend mit $source_k$ für jeden Knoten x der kürzeste Pfad von $source_k$ nach x bestimmt und x aus der Menge $Temp$ in die Menge $Perm$ übernommen. $Perm$ enthält zu jedem Zeitpunkt alle Knoten, für die bereits ein kürzester Pfad ermittelt wurde. Der Algorithmus terminiert, wenn der Zielknoten $sink_{c,k}$ in die Menge $Perm$ übernommen wird. Dann ist $d_{sink_{c,k}}$ die Länge des kürzesten Pfades von $source_k$ nach $sink_{c,k}$. Wenn es keinen Pfad von $source_k$ nach $sink_{c,k}$ gibt, terminiert der Algorithmus mit $sink_{c,k} \notin Perm$.

Auf den Beweis der Korrektheit des Algorithmus 4.3 soll hier verzichtet werden. Die Korrektheit des Verfahrens, den kürzesten Pfad auf Grundlage einer topologischen Sortierung des Graphen zu ermitteln, wobei die Sortierung mittels Tiefensuche im Graphen bestimmt wird, ist in [CLRS01] bewiesen.

In der ersten Phase des Algorithmus wird jede ausgehende Kante jedes Knotens im Problemgraphen $\Gamma_{c,k}$ genau einmal examiniert. Da die Zahl der ausgehenden Kanten je Knoten kleiner oder gleich der Zahl $|O|$ der Operationen ist (mit Ausnahme des virtuellen Startknotens $source_k$, der bis zu $|V(\Gamma_{c,k})|$ ausgehende Kanten haben kann), ist die Zeitkomplexität der ersten Phase durch $O(|V(\Gamma_{c,k})| \cdot |O|)$ beschränkt. In der zweiten Phase wird über allen Knoten $x \in V(\Gamma_{c,k})$ iteriert und in jeder Iteration werden alle ausgehenden Kanten von x examiniert. Durch Implementierung der topologisch sortierten Liste als rückwärts verkettete Liste, hat die Suche nach einem neuen Arbeitsknoten eine Komplexität von $O(1)$. Die Zeitkomplexität in der zweiten Phase des Algorithmus ist also auch durch $O(|V(\Gamma_{c,k})| \cdot |O|)$ beschränkt. Somit beträgt die obere Schranke der Zeitkomplexität des DAG-SP-Algorithmus:

$$O(|V(\Gamma_{c,k})| \cdot |O|) \quad (4.16)$$

Wegen $|V(\Gamma_{c,k})| \leq (|B| + |H|)|T||M||S|$ (Formel 4.3 auf Seite 77) ist auch

$$O(|B||H||T||M||S||O|) \quad (4.17)$$

eine obere Schranke der Zeitkomplexität des DAG-SP-Algorithmus.

Fazit

Ist der Problemgraph azyklisch und seine Knotenmenge endlich, ist aufgrund der linearen Zeitkomplexität in jedem Fall der DAG-SP-Algorithmus (Alg. 4.3) zu bevorzugen. Bei Zyklen im Problemgraphen muss auf den Dijkstra- (Alg. 4.1) oder Bellman-Ford-Algorithmus (Alg. 4.2) ausgewichen werden. Der Dijkstra-Algorithmus weist zwar eine geringere Zeitkomplexität auf, er ist aber nur dann einsetzbar, wenn die Kanten, die von einem anderen als dem virtuellen Startknoten ausgehen, nichtnegativ gewichtet sind. Dafür ist der Dijkstra-Algorithmus – im Gegensatz zu den anderen beiden Algorithmen – auch dann in der Lage, einen kürzesten Pfad zu ermitteln (vorausgesetzt, es existiert ein solcher), wenn die Knotenmenge des Algorithmus unendlich ist. Für beliebige unendliche Problemgraphen mit negativ gewichteten Kanten kann ein kürzester Pfad nicht in endlicher Zeit bestimmt werden (vgl. Abb. 4.3).

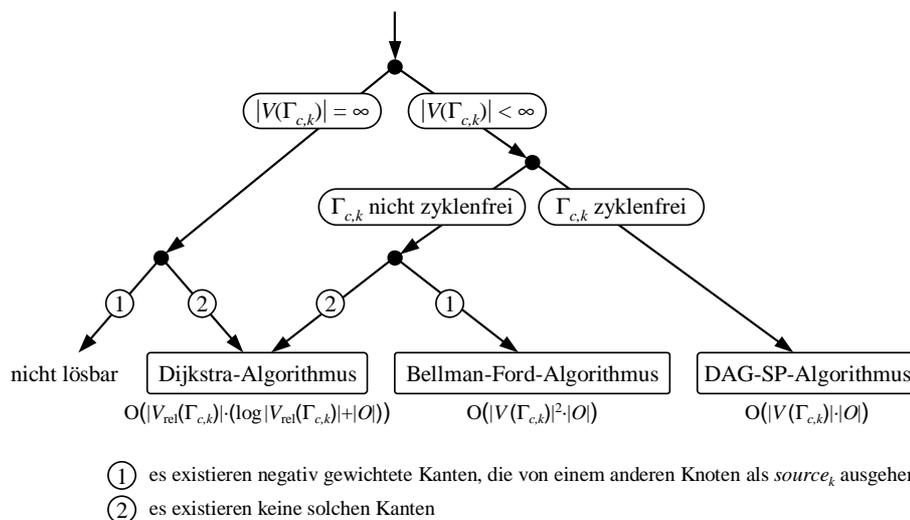


Abbildung 4.3: Entscheidungsgraph zur Auswahl eines geeigneten Pfadalgorithmus

4.1.3 Heuristische Einschränkung des Lösungsraums

Die oben vorgeschlagenen Verfahren zur Komposition von Adaptionspfaden in A-CDNs gehen von der allgemeinen Definition von Adaptionoperationen entsprechend dem Modell in Kapitel 3 aus. In der Praxis ist die Menge der Adaptionspfade, die bei der Suche nach einem optimalen Adaptionspfad berücksichtigt werden müssen, oft durch die Eigenschaften der Adaptionoperationen oder durch deren Platzierung im A-CDN eingeschränkt. Durch Ausnutzung der Kenntnisse über solche Einschränkungen kann die Komplexität der Komposition von optimalen Adaptionspfaden reduziert werden.

Aufgrund der Eigenschaften bestimmter Adaptionoperationen ist es oft nicht sinnvoll, eine konkrete Adaptionoperation in einem Adaptionspfad mehrfach auszuführen oder bestimmte verschiedene Adaptionoperationen in einem Adaptionspfad zu kombinieren.

Beispiel 4.1 *Ein typisches Beispiel dafür sind die Operationen zur räumlichen Skalierung von JPEG-Bildern, die im Simulationsmodell in Abschnitt 5.3 verwendet werden. Anstatt eine Skalierung in mehreren Schritten auszuführen, ist es aus Kostengründen und in der Praxis auch im Hinblick auf die Qualität des Ergebnisses sinnvoller, wenn die Skalierung in einem Schritt durch Anwendung einer einzigen Adaptionoperation erfolgt. Aus diesem Grund kann die Menge der Adaptionspfade, die bei der Suche nach einem optimalen Adaptionspfad berücksichtigt werden, auf solche Adaptionspfade beschränkt werden, bei denen nur eine Operation zur räumlichen Skalierung angewendet wird.*

Das folgende Beispiel 4.2 zeigt, wie die Menge der Adaptionspfade, die bei der Suche nach einem optimalen Adaptionspfad relevant sind, durch die Platzierung von Adaptionoperationen im A-CDN eingeschränkt sein kann:

Beispiel 4.2 *Wenn alle Adaptionoperationen in allen Surrogaten des A-CDN installiert sind und sich die Adaptionkosten sowie die Auswirkungen der Adaptionoperationen auf Medien- und Servicequalitätsmerkmale der Objekte (z.B. Antwortzeitverzögerungen) in den verschiedenen Surrogaten nicht unterscheiden, findet im optimalen Adaptionspfad Inhaltsadaption höchstens in den folgenden zwei Surrogaten statt:*

1. *in dem Surrogat, welches dem verwendeten Replikat am nächsten ist (entweder das Surrogat, in dem das Replikat gespeichert ist, oder jenes, das dem Originalserver, in dem das Replikat gespeichert ist, am nächsten liegt) und*
2. *in dem Surrogat, das dem Client am nächsten ist.*

Adaption in anderen Surrogaten ist unter den beschriebenen Bedingungen grundsätzlich sub-optimal. Deshalb kann die Menge der betrachteten Adaptionspfade bei der Suche nach einem optimalen Adaptionspfad auf die Pfade beschränkt werden, in denen nur in zwei Surrogaten Adaptionoperationen ausgeführt werden.

Solche heuristischen Einschränkungen des Lösungsraums können bei den in Abschnitt 4.1.1 und 4.1.2 beschriebenen Verfahren berücksichtigt werden, indem bei der Konstruktion des Problemgraphen zusätzliche Servicequalitätsmerkmale betrachtet werden. Für Beispiel 4.1 wird als zusätzliches Servicequalitätsmerkmal eine binäre Variable `scale-flag` benötigt, die anzeigt, ob im aktuellen Adaptionspfad bereits eine Skalierungsoperation angewendet wurde. Eine Vorbedingung für die Anwendbarkeit einer Skalierungsoperation (im Systemmodell durch den Definitionsbereich $\mathbf{D}(o)$ der Skalierungsoperation o repräsentiert) ist, dass die Variable nicht gesetzt ist (`scale-flag = false`). Bei Anwendung einer Skalierungsoperation wird `scale-flag := true` gesetzt.

Um für Beispiel 4.2 die Zahl der Surrogate in einem Adaptionspfad, in denen Adaptionoperationen ausgeführt werden, auf zwei zu beschränken, werden zwei zusätzliches Servicequalitätsmerkmale benötigt: eine Integervariable `anode-cnt` und eine binäre Variable `anode-flag`. Die Variable `anode-cnt` gibt an, in wie vielen Surrogaten im Adaptionspfad bereits Adaptionoperationen ausgeführt wurden. Durch `anode-flag` wird angezeigt, ob im aktuellen Surrogat bereits

eine Adaptionsoption angewendet wurde. Durch jede Anwendung einer Adaptionsoption in einem Surrogat wird `anode-flag := true`, nach Anwendung einer Transferoperation ist `anode-flag := false`. Wenn `anode-flag = false`, dann wird durch Anwendung einer Adaptionsoption auf einem Surrogat `anode-cnt` inkrementiert. Bei `anode-flag = true` bleibt `anode-cnt` unverändert. Vorbedingung für die Anwendbarkeit einer Adaptionsoption ist $((\text{anode-cnt} < 2) \vee (\text{anode-flag} = \text{true}))$.

Die hier vorgestellten heuristischen Vereinfachungen, um die Komplexität der Suche nach optimalen Adaptionspfaden zu reduzieren, wurden bei der Implementierung des Simulators zur Evaluation der Vorteile adaptivitätssensitiver Platzierung von Replikaten in A-CDNs entsprechend dem Simulationsmodell in Kapitel 5 berücksichtigt.

4.1.4 Optimale Adaptionspfade zur Beantwortung aller Anfragen

In den vorangegangenen Abschnitten wurden Verfahren zur Konstruktion eines optimalen Adaptionspfades zur Beantwortung einer einzelnen Anfrage (c, k) beschrieben. Bei der Bestimmung einer optimalen Platzierung von Replikaten in einem A-CDN gemäß Problemformulierung 3.2 muss jedoch für jede Anfrage $(c, k) \in C \times K$ mit $\lambda_{c,k} > 0$ der Gewinn $profit_{\pi_{c,k},c}$ des optimalen Adaptionspfades $\pi_{c,k}$ ermittelt werden.

Eine triviale Lösung für dieses Problem ist, einen der in Abschnitt 4.1.2 vorgestellten Algorithmen für jede Anfrage auszuführen. Sowohl der Dijkstra- als auch Bellman-Ford-Algorithmus können aber nicht nur verwendet werden, um den kürzesten Pfad von einem Startknoten zu einem Zielknoten zu ermitteln, sondern ebenso, um den kürzesten Pfad von einem Startknoten zu einer Menge von Zielknoten zu bestimmen. Das eröffnet die Möglichkeit, in einem Lauf des Algorithmus den optimalen Adaptionspfad für mehrere Anfragen (c', k) aus einer Menge $Req_{c,k}$ von Anfragen nach einem Objekt $k \in K$ von verschiedenen Clientknoten $c' \in C$ zu ermitteln. Adaptionspfade für Anfragen nach verschiedenen Objekten in einem Lauf des Algorithmus zu ermitteln ist jedoch nicht sinnvoll, da die Knotenmengen von Adaptionspfaden für verschiedene Objekte elementfremd sind.

Sollen in einem Durchlauf des Algorithmus optimale Adaptionspfade für alle Anfragen $(c', k) \in Req_{c,k}$ ermittelt werden, muss ein Problemgraph $\Gamma_{Req_{c,k}}$ konstruiert werden, der neben den Knoten, die Ausprägungen $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ des Objektes k beschreiben, einen virtuellen Startknoten $source_k$ und für jede Anfrage $(c', k) \in Req_{c,k}$ einen virtuellen Zielknoten $sink_{c',k}$ enthält:

$$\begin{aligned} V(\Gamma_{Req_{c,k}}) = & \{ (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \mid \\ & (n \in ((N - C) \cup \{c' \mid (c', k) \in Req_{c,k}\})) \wedge \\ & (\exists \pi \in \Pi_P. (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = leaf(\pi)) \} \\ & \cup \{ source_k \} \\ & \cup \{ sink_{c',k} \mid (c', k) \in Req_{c,k} \} \end{aligned} \quad (4.18)$$

Analog zu Formel 4.4 auf Seite 78 ist die Kantenmenge des Problemgraphen $\Gamma_{Req_{c,k}}$:

$$\begin{aligned} E(\Gamma_{Req_{c,k}}) = & \{ (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{o} (k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \mid \\ & ((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in V(\Gamma_{Req_{c,k}})) \wedge ((k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')) \in V(\Gamma_{Req_{c,k}})) \wedge \\ & (\exists o \in O. o(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = (n', (\mathbf{t}', \mathbf{m}', \mathbf{s}')))) \} \\ & \cup \{ source_k \xrightarrow{\varepsilon} (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \mid \\ & ((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in V(\Gamma_{Req_{c,k}})) \wedge ((k, n, (\mathbf{t}, \mathbf{m})) \in P) \wedge (\mathbf{s} = \mathbf{s}_{init}) \} \\ & \cup \{ (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{\varepsilon} sink_{c',k} \mid \\ & ((k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \in V(\Gamma_{Req_{c,k}})) \wedge ((c', k) \in Req_{c,k}) \wedge (n = c') \wedge \\ & ((\mathbf{t}, \mathbf{m}, \mathbf{s}) \in mhc(c')) \wedge (qf(c', k, (\mathbf{m}, \mathbf{s})) > 0) \} \end{aligned} \quad (4.19)$$

Die Gewichtsfunktion entspricht Formel 4.8 auf Seite 78.

Bei der Konstruktion eines solchen Problemgraphen ist zu beachten, dass die Wichtungen $w(e)$ der Kanten $e = (k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) \xrightarrow{o} (k, n', (\mathbf{t}', \mathbf{m}', \mathbf{s}'))$ in einem virtuellen Problemgraphen vom nominellen Preis $price_{c',k}$ und von den Qualitätswerten $qf(c', k, (\mathbf{m}, \mathbf{s}))$ und $qf(c', k, (\mathbf{m}', \mathbf{s}'))$ der Objektausprägungen, die durch den Anfangs- und Endknoten der Kante e repräsentiert werden, aus Sicht des anfragenden Client c' abhängen. Folglich können nur für solche Anfragen optimale Adaptionspfade in einem gemeinsamen Problemgraphen $\Gamma_{Req_{c,k}}$ ermittelt werden, für die paarweise (für alle Paare $((c', k), (c'', k)) \in (Req_{c,k} \times Req_{c,k})$) die folgenden Bedingungen erfüllt sind:

$$price_{c',k} = price_{c'',k} \quad (4.20)$$

$$\forall(\mathbf{m}, \mathbf{s}) \in M \times S. qf(c', k, (\mathbf{m}, \mathbf{s})) = qf(c'', k, (\mathbf{m}, \mathbf{s})) \quad (4.21)$$

Die Überprüfung, ob die erste Bedingung (Formel 4.20) erfüllt ist, ist trivial. Eine formale Überprüfung der zweiten Bedingung (Formel 4.21) paarweise für alle Anfragen in $Req_{c,k}$ ist aber nicht praktikabel. Anstelle dessen ist es sinnvoller, Kenntnisse aus der Anwendungsdomäne, auf deren Grundlage die Qualitätsfunktion qf definiert ist, heranzuziehen, um Clientknoten entsprechend der Äquivalenzrelation \equiv_q mit

$$(c' \equiv_q c'') \Leftrightarrow (\forall k \in K. \forall(\mathbf{m}, \mathbf{s}) \in M \times S. qf(c', k, (\mathbf{m}, \mathbf{s})) = qf(c'', k, (\mathbf{m}, \mathbf{s}))) \quad (4.22)$$

in Äquivalenzklassen $[c]_{\equiv_q} \subseteq C$ einzuteilen. Einfach gesagt, drückt die Äquivalenzrelation \equiv_q aus, dass zwei Clientknoten die gleichen Bewertungsmaßstäbe für Qualität haben. In der Praxis hängen die Bewertungsmaßstäbe eines Client für die Qualität von Objektausprägungen von den Eigenschaften des verwendeten Endgerätes und ggf. vom Typ der Netzwerkanbindung des Client ab. Somit sind Clientknoten, die Clients des gleichen Endgerätetyps mit Netzwerkanbindungen des gleichen Typs repräsentieren, äquivalent bezüglich \equiv_q .

Analog zur Äquivalenzrelation \equiv_q soll durch eine Äquivalenzrelation \equiv_p ausgedrückt werden, dass zwei Clientknoten aus Sicht des Preissystems äquivalent sind:

$$(c' \equiv_p c'') \Leftrightarrow (\forall k \in K. price_{c',k} = price_{c'',k}) \quad (4.23)$$

Bei der Modellierung in Kapitel 3 wurden keine Annahmen über das Preissystem gemacht. In der Praxis ist zu erwarten, dass für Clients des gleichen Endgerätetyps mit Netzwerkanbindungen des gleichen Typs im Allgemeinen keine Preisdifferenzierung erfolgt, so dass zwei Clients, die bezüglich \equiv_q äquivalent sind, auch bezüglich \equiv_p äquivalent sind.

Formal wird die Eigenschaft zweier Clients c' und c'' sowohl bezüglich der Qualitätsfunktion als auch bezüglich des Preissystems äquivalent zu sein, durch die Relation $\equiv_{q,p}$ ausgedrückt:

$$(c' \equiv_{q,p} c'') \Leftrightarrow ((c' \equiv_q c'') \wedge (c' \equiv_p c'')) \quad (4.24)$$

Die Menge der Anfragen (c', k) nach einem Inhaltsobjekt k von verschiedenen Clients $c' \in C$, die paarweise äquivalent bezüglich $\equiv_{q,p}$ sind, sei

$$Req_{c,k} = \left\{ (c', k) \mid c' \in [c]_{\equiv_{q,p}} \right\}. \quad (4.25)$$

Die Anfragen in $Req_{c,k}$ erfüllen paarweise die Bedingungen in Formel 4.20 und 4.21. Somit lässt sich ein Problemgraph $\Gamma_{Req_{c,k}}$ mit einem virtuellen Startknoten $source_k$ und einer Menge

$$Sink_{Req_{c,k}} = \{ sink_{c',k} \mid (c', k) \in Req_{c,k} \} \quad (4.26)$$

von virtuellen Zielknoten konstruieren, in dem die Suche nach den optimalen Adaptionspfaden für alle Anfragen $(c', k) \in Req_{c,k}$ der Suche nach den kürzesten Pfaden zwischen dem virtuellen Startknoten und allen virtuellen Zielknoten entspricht.

Zur Ermittlung der kürzesten Pfade von $source_k$ zu allen $sink_{c',k} \in Sink_{Req_{c,k}}$ kann der Bellman-Ford-Algorithmus, so wie in Algorithmus 4.2 auf Seite 82 dargestellt, unverändert verwendet werden. Nach Terminierung des Algorithmus enthält d_x für alle $x \in Temp$ die Länge des kürzesten Pfades. Für Knoten $x \notin Temp$ gibt es keinen Pfad.

Der Dijkstra-Algorithmus (Alg. 4.1 auf Seite 80) und der DAG-SP-Algorithmus (Alg. 4.3 auf Seite 83) hingegen terminieren sofort, wenn der kürzeste Pfad zum Zielknoten $sink_{c,k}$ gefunden wurde. Soll der kürzeste Pfad zu allen Zielknoten $sink_{c',k} \in Sink_{Req_{c,k}}$ ermittelt werden, muss die Abbruchbedingung in Zeile 9 des Algorithmus 4.1 bzw. in Zeile 19 des Algorithmus 4.3 folgendermaßen geändert werden:

if $Sink_{Req_{c,k}} \subseteq Perm$ **then**

Wenn der Dijkstra- oder DAG-SP-Algorithmus terminiert, gibt $d_{sink_{c',k}}$ für alle virtuellen Zielknoten $sink_{c',k} \in Perm$ die Länge des kürzesten Pfades vom Startknoten $source_k$ an. Für virtuelle Zielknoten $sink_{c',k} \notin Perm$ gibt es keinen Pfad.

Werden Problemgraphen $\Gamma_{Req_{c,k}}$ herangezogen, kann der Aufwand zur Ermittlung der optimalen Adaptionpfade für alle Anfragen gegenüber dem trivialen Ansatz, für jede Anfrage einen einzelnen Problemgraphen $\Gamma_{c,k}$ zu verwenden, deutlich reduziert werden. In diesem Fall muss ein Kürzester-Pfad-Algorithmus (Dijkstra-, Bellman-Ford- oder DAG-SP-Algorithmus) nur noch einmal pro Äquivalenzklasse $[c]_{\equiv_{q,p}}$ und Inhaltsobjekt $k \in K$ ausgeführt werden.

4.2 Adaptivitätssensitive Platzierungsalgorithmen

Adaptivitätssensitive Platzierungsalgorithmen haben das Ziel, eine Platzierung P_R von Replikaten in den Surrogaten eines A-CDN zu ermitteln, bei der der Wert der Zielfunktion $Profit_{\Sigma,\Lambda}(P_R)$ maximal ist (Problem 3.2 auf Seite 71). Wie im folgenden Abschnitt gezeigt wird, ist dieses Problem NP-schwer. Eine exakte Lösung kann demzufolge nicht mit polynomialem Aufwand bestimmt werden³. Deshalb werden in Abschnitt 4.2.3 Greedy-Algorithmen entwickelt, die in polynomialer Zeit eine Näherungslösung ermitteln. Zur Bewertung der Güte des Näherungsverfahrens sollen in Kapitel 6 die Näherungslösungen für ausgewählte Beispielszenarien jeweils mit der entsprechenden exakten Lösung verglichen werden. Ein Algorithmus zur Ermittlung einer exakten Lösung wird in Abschnitt 4.2.2 gezeigt.

Wie in Abschnitt 3.1.2 dargestellt wurde, ist in einem A-CDN eine verteilte Realisierung des Platzierungsalgorithmus sinnvoll, um die Skalierbarkeit und Ausfallsicherheit zu erhöhen. Deshalb wird in Abschnitt 4.2.4 eine Möglichkeit zur verteilten Realisierung des Algorithmus aus Abschnitt 4.2.3 vorgestellt.

4.2.1 Komplexität des Platzierungsproblems

Das Problem der adaptivitätssensitiven Platzierung von Replikaten in einem A-CDN ist NP-schwer. Dies lässt sich durch Zurückführen eines Spezialfalls der adaptivitätssensitiven Platzierung auf das k -Median-Optimierungsproblem zeigen.

Beweis 4.1 (NP-Schwere von Problem 3.2)

Man betrachte den Spezialfall, dass nur ein Surrogat \tilde{h} zur Verfügung steht und nur ein einzelnes Inhaltsobjekt \tilde{k} durch das A-CDN ausgeliefert werden soll:

$$H = \{\tilde{h}\} \quad |H| = 1 \quad (4.27)$$

$$K = \{\tilde{k}\} \quad |K| = 1 \quad (4.28)$$

Ferner haben alle Repräsentationen des Objektes \tilde{k} eine Größe von 1:

$$\forall(\mathbf{t}, \mathbf{m}) \in T \times M. sf(\tilde{k}, (\mathbf{t}, \mathbf{m})) = 1 \quad (4.29)$$

³Vorausgesetzt, dass $\mathbf{P} \neq \mathbf{NP}$ gilt. \mathbf{P} ist die Komplexitätsklasse, die alle Probleme umfasst, die in polynomialer Zeit gelöst werden können. Die Komplexitätsklasse \mathbf{NP} umfasst alle Probleme, für die in polynomialer Zeit für eine potentielle Lösung (Zertifikat) geprüft werden kann, ob sie tatsächlich eine Lösung ist. Es ist jedoch nicht Voraussetzung, dass ein Algorithmus existiert, der eine Lösung in polynomialer Zeit findet. Offensichtlich gilt $\mathbf{P} \subseteq \mathbf{NP}$. Dass \mathbf{P} eine echte Teilmenge von \mathbf{NP} ist, ist wahrscheinlich, aber bis heute nicht bewiesen. Für eine umfassende Diskussion der Problematik sei auf [CLRS01] verwiesen.

Somit lässt sich die Nebenbedingung aus Problem 3.2 folgendermaßen umformulieren:

$$\begin{aligned} \forall h \in H. \quad \sum_{(k,h,(\mathbf{t},\mathbf{m})) \in P_R} sf(k, (\mathbf{t}, \mathbf{m})) &\leq sc(h) \\ \sum_{(k,\tilde{h},(\mathbf{t},\mathbf{m})) \in P_R} 1 &\leq sc(\tilde{h}) \\ |P_R| &\leq sc(\tilde{h}) \end{aligned} \quad (4.30)$$

Für die Zielfunktion aus Problem 3.2 ergibt sich durch Einsetzen von Formel 3.42 auf Seite 70 folgende äquivalente Umformung:

$$\begin{aligned} \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot Profit_{\Sigma,(c,k)}(P_R) &= \max! \\ \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot \max \{ profit_{\pi,c} \mid (\pi \in \Pi_{P_0 \cup P_R}) \wedge (\pi \text{ satisfies } (c,k)) \} &= \max! \\ \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot \min \{ -profit_{\pi,c} \mid (\pi \in \Pi_{P_0 \cup P_R}) \wedge (\pi \text{ satisfies } (c,k)) \} &= \min! \end{aligned}$$

Mit Formel 3.28 auf Seite 67 gilt:

$$\sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot \min \left\{ \min \left\{ -profit_{\pi,c} \mid (\pi \in \Pi_{P_0 \cup \{(k,h,(\mathbf{t},\mathbf{m}))\}}) \wedge (\pi \text{ satisfies } (c,k)) \right\} \mid (k,h,(\mathbf{t},\mathbf{m})) \in P_R \right\} = \min!$$

Wird hier wiederum Formel 3.42 auf Seite 70 eingesetzt, folgt:

$$\begin{aligned} \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot \min \left\{ -Profit_{\Sigma,(c,k)}(\{(k,h,(\mathbf{t},\mathbf{m}))\}) \mid (k,h,(\mathbf{t},\mathbf{m})) \in P_R \right\} &= \min! \\ \sum_{(c,k) \in C \times K} \min \left\{ -\lambda_{c,k} \cdot Profit_{\Sigma,(c,k)}(\{(k,h,(\mathbf{t},\mathbf{m}))\}) \mid (k,h,(\mathbf{t},\mathbf{m})) \in P_R \right\} &= \min! \end{aligned} \quad (4.31)$$

Durch eine eindeutige Abbildung $ri : PR_{P_0} \rightarrow F$ werde jedem potentiellen Replikate $(k,h,(\mathbf{t},\mathbf{m})) \in PR_{P_0}$ ein Element i aus einer Menge F zugewiesen. Die Teilmenge $I \subseteq F$ entspreche der Menge der Replikate in der Platzierung P_R :

$$I = \{ i \mid (i = ri(k,h,(\mathbf{t},\mathbf{m}))) \wedge ((k,h,(\mathbf{t},\mathbf{m})) \in P_R) \} \quad (4.32)$$

Eine eindeutige Abbildung $cj : (C \times K) \rightarrow C$ bilde jede Anfrage $(c,k) \in C \times K$ auf ein Element j einer Menge C ab. Es sei:

$$c_{ij} = -\lambda_{c,k} \cdot Profit_{\Sigma,(c,k)}(\{(k,h,(\mathbf{t},\mathbf{m}))\}) \quad \text{mit } i = ri(k,h,(\mathbf{t},\mathbf{m})) \text{ und } j = cj(c,k) \quad (4.33)$$

Durch Einsetzen von Formel 4.33 in Formel 4.31 erhält man:

$$\sum_{j \in C} \min \{ c_{ij} \mid i \in I \} = \min! \quad (4.34)$$

Setzt man $k = sc(\tilde{h})$, ist wegen $|I| = |P_R|$ (Formel 4.32) die Nebenbedingung in Formel 4.30 äquivalent zu:

$$|I| \leq k \quad (4.35)$$

Die Zielfunktion in Formel 4.34 in Zusammenhang mit der Nebenbedingung in Formel 4.35 entspricht dem k -Median-Optimierungsproblem, von welchem bekannt ist, dass es NP-schwer ist [JV99]. Da das NP-schwere k -Median-Optimierungsproblem, wie gezeigt, polynomial auf das Problem der adaptivitätssensitiven Platzierung von Replikaten (Problem 3.2 auf Seite 71) reduzierbar ist, ist auch das Problem der adaptivitätssensitiven Platzierung NP-schwer. \square

Algorithmus 4.4 Erschöpfende Suche

```

1  sub  $res(P, PR)$  Unterprogramm: rekursive erschöpfende Suche
2  if  $PR = \emptyset$  then wenn für alle potentiellen Replikate entschieden ist, ob sie bei
   der aktuellen Platzierung materialisiert werden oder nicht,
3      $p = Profit_{\Sigma, \Lambda}(P)$  den Gesamtgewinn bei der aktuellen Platzierung ermitteln;
4     if  $p > Profit_{\max}$  then ist dieser größer, als bei allen vorher getesteten Platzierungen,
   den Gesamtgewinn und die aktuelle Platzierung merken;
5        $Profit_{\max} := p$ 
6        $P_R := P$ 
7     end if
8  else
9     suche ein beliebiges  $(k', h', (\mathbf{t}', \mathbf{m}')) \in PR$ 
10     $res(P, PR - \{(k', h', (\mathbf{t}', \mathbf{m}'))\})$  rekursive erschöpfende Suche in allen Platzierungen, in denen
   alle  $(k, h, (\mathbf{t}, \mathbf{m})) \in P$  materialisiert sind, aber  $(k', h', (\mathbf{t}', \mathbf{m}'))$  und
   alle  $(k, h, (\mathbf{t}, \mathbf{m})) \in (PR_{P_0} - (PR \cup P))$  nicht materialisiert sind;
11    if  $\sum_{(k, h', (\mathbf{t}, \mathbf{m})) \in (P \cup \{(k', h', (\mathbf{t}', \mathbf{m}'))\})} sf(k, (\mathbf{t}, \mathbf{m})) \leq sc(h')$  then
   wenn durch Materialisierung von  $(k', h', (\mathbf{t}', \mathbf{m}'))$  die
   Speicherkapazität von  $h'$  nicht überschritten wird,
12     $res(P \cup \{(k', h', (\mathbf{t}', \mathbf{m}'))\}, PR - \{(k', h', (\mathbf{t}', \mathbf{m}'))\})$ 
   rekursive erschöpfende Suche in allen Platzierungen,
   in denen  $(k', h', (\mathbf{t}', \mathbf{m}'))$  und alle  $(k, h, (\mathbf{t}, \mathbf{m})) \in P$ , aber
   kein  $(k, h, (\mathbf{t}, \mathbf{m})) \in (PR_{P_0} - (PR \cup P))$  materialisiert sind;
13    end if
14  end if
15 end sub
16  $Profit_{\max} := -\infty$  Initialisierung;
17  $P_R := \emptyset$ 
18  $res(\emptyset, PR_{P_0})$  rekursive erschöpfende Suche im kompletten Lösungsraum

```

4.2.2 Erschöpfende Suche zur Bestimmung einer optimalen Lösung

Da das Problem NP-schwer ist, gibt es keinen Algorithmus, der eine exakte Lösung mit geringerer Komplexität als durch erschöpfende Suche im gesamten Lösungsraum findet. Der Lösungsraum

$$Sol_{\Sigma} = \left\{ P_R \mid (P_R \subseteq PR_{P_0}) \wedge \left(\forall h \in H. \sum_{(k, h, (\mathbf{t}, \mathbf{m})) \in P_R} sf(k, (\mathbf{t}, \mathbf{m})) \leq sc(h) \right) \right\} \quad (4.36)$$

umfasst alle Platzierungen, für die die Nebenbedingung aus Problemformulierung 3.2 auf Seite 71 erfüllt ist. Algorithmus 4.4 zeigt einen Backtracking-Algorithmus für die erschöpfende Suche im Lösungsraum Sol_{Σ} zur Bestimmung einer optimalen Lösung. Wenn der Algorithmus terminiert, ist P_R die optimale Platzierung von Replikaten im A-CDN. $Profit_{\max}$ ist dann der maximale Gesamtgewinn. Die Zeitkomplexität des Algorithmus beträgt

$$O(|Sol_{\Sigma}| \cdot (|PR_{P_0}| + CPX_{Profit_{\Sigma, \Lambda}}(\Sigma))). \quad (4.37)$$

Dabei ist $CPX_{Profit_{\Sigma, \Lambda}}(\Sigma)$ die Komplexität des verwendeten Algorithmus zur Komposition optimaler Adaptionenpfade für alle Anfragen (vgl. Abschnitt 4.1.4), der zur Berechnung von $Profit_{\Sigma, \Lambda}$ in Zeile 3 verwendet wird. Da $|Sol_{\Sigma}| = O(2^{|PR_{P_0}|})$, kann die erschöpfende Suche nur bei sehr kleinen Problemen angewendet werden, bei denen aufgrund einer geringen Zahl von Surrogaten, Objekten und Repräsentationen sowie durch eine begrenzte Speicherkapazität der Surrogate der Lösungsraum stark eingeschränkt ist.

Beispiel 4.3 Wenn durch ein A-CDN zehn Objekte verteilt werden sollen, die in drei verschiedenen Repräsentationen in drei Surrogaten repliziert werden können, wobei aufgrund der

begrenzten Speicherkapazität der Surrogate jeweils genau drei Replikate pro Surrogat materialisierbar sind, so ist $|Sol_\Sigma| \approx 9 \cdot 10^{10}$. Wenn für die Berechnung des Wertes der Zielfunktion $Profit_{\Sigma,\Lambda}$ in Zeile 3 des Algorithmus im Mittel $10 \mu s$ benötigt werden, so erfordert die Bestimmung einer optimalen Platzierung mehr als 10 Tage.

4.2.3 Greedy-Heuristik zur Bestimmung einer Näherungslösung

Zur Ermittlung einer Näherungslösung für das Problem der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs soll eine Greedy-Heuristik verwendet werden. Greedy-Heuristiken haben sich bei einer Reihe von verwandten Problemen, wie dem Problem der Platzierung von Replikaten ohne Berücksichtigung von Inhaltsadaption [KRR01, QPV01, KKM02], der Auswahl von statisch voradaptierten Repräsentationen in einem Server für adaptierbare Inhalte [LL02] oder der Auswahl von materialisierten Sichten in einem Data-Warehouse-System [HRU96, Gup97, GM99, BL03], bewährt.

Greedy-Heuristiken basieren auf einer linear rekursiven Zerlegung des Problems in Teilprobleme SP_{i1} und SP_{i2} , wobei auf jeder Stufe i der Rekursion ein Teilproblem SP_{i1} durch Wahl der lokal optimalen Option heuristisch gelöst wird und das zweite Teilproblem SP_{i2} rekursiv weiter zerlegt wird, bis schließlich auch das Teilproblem SP_{j2} trivial oder heuristisch gelöst werden kann (Abb. 4.4).

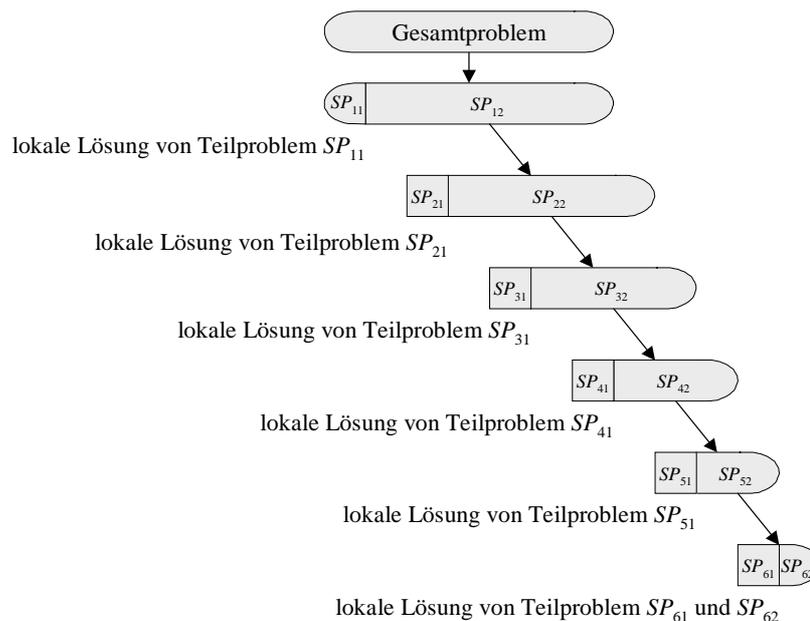


Abbildung 4.4: Prinzip der Greedy-Heuristik

Für kapazitätsbeschränkte Auswahlprobleme, wie das Problem der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs und die genannten verwandten Probleme, die sich durch

Problem 4.1 (kapazitätsbeschränkte Auswahl)

gegeben: eine Grundmenge M ,
ein Prädikat \mathcal{P} mit $\forall X \subseteq M. \forall Y \subseteq X. \mathcal{P}(X) \Rightarrow \mathcal{P}(Y)$ (Kapazitätsbeschränkung)

gesucht: eine Teilmenge $X \subseteq M$

Zielfunktion: $f(X) = \max!$

Nebenbedingung: das Prädikat $\mathcal{P}(X)$ muss erfüllt sein

abstrahieren lassen, kann die Greedy-Heuristik folgendermaßen konkretisiert werden: In jeder Stufe i der Rekursion wird für ein Element x_i der Grundmenge (ggf. auch für eine Menge ΔX_i von Elementen) entschieden, ob es Teil der Ergebnismenge X sein wird. Es sei $M_i \subseteq M$ die Menge der Elemente, für die nach $i - 1$ Rekursionsstufen noch nicht entschieden ist, ob sie Teil der Ergebnismenge X sein werden. Es gilt $M_1 = M$.

In jeder Rekursionsstufe i wird die Menge M_i durch Auswahl eines Elementes x_i in zwei disjunkte Teilmengen $\Delta X_i = \{x_i\}$ und M_{i+1} zerlegt. Während für das Element x_i der ersten Teilmenge sofort entschieden wird, ob es Element der Ergebnismenge sein wird oder nicht, wird das Teilproblem SP_{i2} , zu entscheiden, welche Elemente der Menge M_{i+1} in die Ergebnismenge aufgenommen werden, in der Rekursionsstufe $i + 1$ gelöst. Wenn die Menge X_{i-1} alle Elemente von M umfasst, die in den vorangegangenen Stufen der Rekursion in die Ergebnismenge aufgenommen wurden, dann wird das Element x_i der Menge ΔX_i in die Ergebnismenge aufgenommen, falls die Vereinigung $X_{i-1} \cup \Delta X_i$ die Nebenbedingung nicht verletzt:

$$X_i = \begin{cases} X_{i-1} \cup \Delta X_i & \text{falls } \mathcal{P}(X_{i-1} \cup \Delta X_i) \text{ erfüllt ist} \\ X_{i-1} & \text{sonst} \end{cases} \quad (4.38)$$

Die Auswahl eines Elementes $x_i \in M_i$ erfolgt heuristisch auf Grundlage einer Nutzwertfunktion

$$\text{benefit} : (2^{M_i} \times 2^{M-M_i}) \rightarrow \mathbb{R}, \quad (4.39)$$

die jeder Teilmenge $\Delta X \subseteq M_i$ einen Nutzwert $\text{benefit}(\Delta X, X_{i-1})$ zuordnet, der den Nutzen der Auswahl der Elemente in ΔX zusätzlich zu den bereits ausgewählten Elementen in $X_{i-1} \subseteq (M - M_i)$ bzgl. der Maximierung der Zielfunktion unter Einhaltung der Nebenbedingung repräsentiert. Die Rekursion terminiert, wenn M_i keine Elemente enthält, die einen Nutzen gemäß benefit haben: $\forall \Delta X \subseteq M_i. \text{benefit}(\Delta X, X_{i-1}) \leq 0$.

Für Problemstellungen, in denen das Prädikat \mathcal{P} ausdrückt, dass genau eine Ressource kapazitätsbeschränkt ist, wie z.B. die Speicherkapazität eines zentralisierten Data-Warehouse-Systems (wie in [HRU96, Gup97, GM99]) oder eines Servers (wie in [LL02]), wird in [HRU96] eine untere Schranke für den relativen Wert der Zielfunktion der Greedy-Lösung im Vergleich zur optimalen Lösung nachgewiesen. Der Beweis der unteren Schranke geht allerdings davon aus, dass die Nutzwertfunktion die Monotonieeigenschaft nach [GM99] aufweist:

Definition 4.1 (Monotonie von Nutzwertfunktionen) Eine Nutzwertfunktion benefit ist monoton, wenn für alle Teilmengen $\Delta X \subseteq M$, $\Delta X' \subseteq M$ und $X \subseteq M$ gilt:

$$\text{benefit}(\Delta X \cup \Delta X', X) \leq \max\{\text{benefit}(\Delta X, X), \text{benefit}(\Delta X', X)\} \quad (4.40)$$

In [GM99] wird gezeigt, dass Monotonie der Nutzwertfunktion nicht nur hinreichend, sondern auch notwendig für die Existenz einer unteren Schranke ist. Ist die Monotonieeigenschaft nicht erfüllt, kann eine Greedy-Heuristik beliebig schlechte Lösungen liefern.

Für Problemstellungen, in denen das Prädikat \mathcal{P} ausdrückt, dass mehrere Ressourcen beschränkt sind, wie zum Beispiel die Speicherkapazitäten in verschiedenen Knoten eines verteilten Data-Warehouse-Systems oder in verschiedenen Surrogaten eines A-CDN, gilt die in [HRU96] nachgewiesene untere Schranke nicht [BL03]. Da der Fall einer beschränkten Ressource ein Spezialfall des Falls mehrerer beschränkter Ressourcen ist, ist Monotonie der Nutzwertfunktion aber eine notwendige Bedingung für die Existenz einer solchen Schranke.

Für das Problem der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs ist die Grundmenge die Menge der potentiellen Replikate: $M = PR_{P_0}$. Ein Lösung $X \subseteq M$ ist eine Platzierung $P_R \subseteq PR_{P_0}$ von Replikaten. Das Prädikat \mathcal{P} drückt die begrenzte Speicherkapazität der Surrogate aus:

$$\mathcal{P}(P_R) = \left(\forall h \in H. \sum_{(k,h,(\mathbf{t},\mathbf{m})) \in P_R} sf(k, (\mathbf{t}, \mathbf{m})) \leq sc(h) \right) \quad (4.41)$$

Die Bedingung $\forall P_R \subseteq PR_{P_0}. \forall P'_R \subseteq P_R. \mathcal{P}(P_R) \Rightarrow \mathcal{P}(P'_R)$ ist offensichtlich erfüllt, da $P'_R \subseteq P_R$ impliziert, dass $\forall h \in H. \sum_{(k,h,(\mathbf{t},\mathbf{m})) \in P'_R} sf(k, (\mathbf{t}, \mathbf{m})) \leq \sum_{(k,h,(\mathbf{t},\mathbf{m})) \in P_R} sf(k, (\mathbf{t}, \mathbf{m}))$. Die Zielfunktion (vgl. Problem 3.2 auf Seite 71) ist:

$$f(P_R) = \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot Profit_{\Sigma,(c,k)}(P_R) \quad (4.42)$$

Der Nutzwert $benefit_{\Sigma,\Lambda}(\Delta P_R, P_{R(i-1)})$ einer Menge ΔP_R von Replikaten bzgl. einer Menge $P_{R(i-1)}$ bereits zur Materialisierung ausgewählter Replikate wird analog zum Vorgehen in [HRU96, Gup97, GM99, BL03] als Verhältnis der Steigerung des Wertes der Zielfunktion durch Materialisierung der Replikate in ΔP_R zur Reduktion der verfügbaren Ressourcen (Speicherkapazität) definiert:

$$benefit_{\Sigma,\Lambda}(\Delta P_R, P_{R(i-1)}) = \frac{f(\Delta P_R \cup P_{R(i-1)}) - f(P_{R(i-1)})}{\sum_{(k,h,(\mathbf{t},\mathbf{m})) \in \Delta P_R \cup P_{R(i-1)}} sf(k, (\mathbf{t}, \mathbf{m})) - \sum_{(k,h,(\mathbf{t},\mathbf{m})) \in P_{R(i-1)}} sf(k, (\mathbf{t}, \mathbf{m}))} \quad (4.43)$$

Wie in Anhang B nachgewiesen wird, ist die Nutzwertfunktion $benefit_{\Sigma,\Lambda}$ monoton entsprechend Definition 4.1.

Die Greedy-Heuristik kann durch den Greedy-Algorithmus in Algorithmus 4.5 implementiert werden. In Algorithmus 4.5 wird die linear rekursive Struktur der Heuristik durch eine Iteration (vgl. Zeile 19) über der Menge *Ranking* aller potentiellen Replikate implementiert, die ohne Verletzung einer Nebenbedingung (vgl. Zeile 10 und 30) materialisiert werden können. Um den Aufwand zur Berechnung der Nutzwertfunktion für alle $\{(k, h, (\mathbf{t}, \mathbf{m}))\} \subseteq \textit{Ranking}$ zu reduzieren, werden nur vor dem ersten Iterationsschritt die Nutzwerte aller potentiellen Replikate ermittelt (Zeile 9–18). Später werden nur für solche Replikate die Nutzwerte neu berechnet, für die sich durch die Auswahl im vorangegangenen Iterationsschritt eine Veränderung ergeben haben könnte (Zeile 29–39).

Der Algorithmus terminiert, wenn *Ranking* keine Replikate enthält, die einen positiven Nutzwert haben. Das ist der Fall, wenn selbst der Nutzwert des Replikates mit dem maximalen Nutzwert in *Ranking* nicht größer als 0 ist (vgl. Zeile 21) oder wenn *Ranking* leer ist. Wenn Algorithmus 4.5 terminiert, dann ist P_R die näherungsweise optimale Platzierung von Replikaten im A-CDN.

Die Zeitkomplexität des Algorithmus 4.5 ist durch

$$O(|PR_{P_0}|^2 \cdot (|C| \cdot CPX_{Profit_{\Sigma,(c,k)}}(\Sigma) + \log |PR_{P_0}|)) \quad (4.44)$$

beschränkt. Dabei ist $CPX_{Profit_{\Sigma,(c,k)}}(\Sigma)$ die Komplexität des Algorithmus zur Komposition optimaler Adaptionspfade (vgl. Abschnitt 4.1), der zur Berechnung von $Profit_{\Sigma,(c,k)}$ in Zeile 7, 13 und 33 verwendet wird. In Formel 4.44 wird davon ausgegangen, dass *Ranking* als binärer Heap [CLRS01] implementiert wird, in dem die Replikate entsprechend des Schlüsselwertes $b_{k,h,(\mathbf{t},\mathbf{m})}$ sortiert sind, so dass die Suche nach einem Replikat in Zeile 20 eine konstante Zeitkomplexität ($O(1)$) und das Einfügen eines Eintrags (Zeile 16), Löschen eines Eintrags (Zeile 24 und 37) und Aktualisieren eines Schlüsselwertes (Zeile 35) eine Komplexität von $O(\log |PR_{P_0}|)$ hat.

Wegen $PR_{P_0} \subseteq K \times H \times T \times M$ und aufgrund der **if**-Bedingung in Zeile 29 ist

$$O(|K||H|^2|T|^2|M|^2 \cdot (|C| \cdot CPX_{Profit_{\Sigma,(c,k)}}(\Sigma) + \log(|K||H||T||M|))) \quad (4.45)$$

ebenfalls eine obere Schranke der Zeitkomplexität des Greedy-Algorithmus 4.5. Somit ergeben sich für die Komplexität inklusive des Aufwands zur Komposition optimaler Adaptionspfade in Zeile 7, 13 und 33 je nach verwendetem Kürzester-Pfad-Algorithmus im Problemgraphen $\Gamma_{c,k}$ (vgl. Abs. 4.1.2) die Schranken in Tabelle 4.1(a) auf Seite 96.

Eine alternative Implementierung (Algorithmus 4.6 auf Seite 95) der Greedy-Heuristik er-

Algorithmus 4.5 Greedy-Algorithmus, Variante 1

```

1   $Ranking := \emptyset$  Initialisierung;
2   $P_R := \emptyset$ 
3  foreach  $h \in H$  do
4     $capacity_h := sf(h)$ 
5  end foreach
6  foreach  $(c, k) \in C \times K$  do für jedes Objekt  $k \in K$  den Gesamtgewinn  $p_{\emptyset k}$ 
7     $p_{\emptyset k} := p_{\emptyset k} + \lambda_{c,k} \cdot Profit_{\Sigma, (c,k)}(\emptyset)$  über alle Anfragen nach  $k$  bei  $P_R = \emptyset$  ermitteln;
8  end foreach
9  foreach  $(k, h, (\mathbf{t}, \mathbf{m})) \in PR_{P_0}$  do Ermittlung der Nutzwerte aller Replikate,
10   if  $capacity_h \geq sf(k, (\mathbf{t}, \mathbf{m}))$  then deren Materialisierung keine Nebenbedingung verletzt;
11      $p_{k,h,(\mathbf{t},\mathbf{m})} := 0$ 
12     foreach  $c \in C$  do
13        $p_{k,h,(\mathbf{t},\mathbf{m})} := p_{k,h,(\mathbf{t},\mathbf{m})} + \lambda_{c,k} \cdot Profit_{\Sigma, (c,k)}(\{(k, h, (\mathbf{t}, \mathbf{m}))\})$ 
14     end foreach
15      $b_{k,h,(\mathbf{t},\mathbf{m})} := (p_{k,h,(\mathbf{t},\mathbf{m})} - p_{\emptyset k}) / sf(k, (\mathbf{t}, \mathbf{m}))$ 
16      $Ranking := Ranking \cup \{(k, h, (\mathbf{t}, \mathbf{m}))\}$ 
17   end if
18 end foreach
19 while  $Ranking \neq \emptyset$  do Iteration über allen materialisierbaren potentiellen Replikaten;
20   suche  $(k, h, (\mathbf{t}, \mathbf{m})) \in Ranking$ , für das Auswahl eines Replikates auf
21      $b_{k,h,(\mathbf{t},\mathbf{m})} = \max \{b_{k',h',(\mathbf{t}',\mathbf{m}')} \mid (k', h', (\mathbf{t}', \mathbf{m}')) \in Ranking\}$  Basis der Nutzwertfunktion;
22   if  $b_{k,h,(\mathbf{t},\mathbf{m})} = 0$  then wenn selbst das beste potentielle Replikat
23     in Ranking keinen Nutzen hat,
24     terminiert der Greedy-Algorithmus;
25   halt
26   end if
27    $Ranking := Ranking - \{(k, h, (\mathbf{t}, \mathbf{m}))\}$ 
28   if  $capacity_h \geq sf(k, (\mathbf{t}, \mathbf{m}))$  then wenn es ohne Verletzung einer
29     Nebenbedingung materialisierbar ist,
30     das Replikat  $(k, h, (\mathbf{t}, \mathbf{m}))$  materialisieren;
31      $P_R := P_R \cup \{(k, h, (\mathbf{t}, \mathbf{m}))\}$ 
32      $capacity_h := capacity_h - sf(k, (\mathbf{t}, \mathbf{m}))$ 
33     foreach  $(k', h', (\mathbf{t}', \mathbf{m}')) \in Ranking$  do für alle potentiellen Replikate  $(k', h', (\mathbf{t}', \mathbf{m}'))$ ,
34       deren Nutzwert sich verändert haben könnte und
35       die ohne Verletzung einer Nebenbedingung
36       materialisiert werden können,
37       den Nutzwert neu berechnen;
38        $p_{k',h',(\mathbf{t}',\mathbf{m}')} := 0$ 
39       foreach  $c' \in C$  do
40          $p_{k',h',(\mathbf{t}',\mathbf{m}')} := p_{k',h',(\mathbf{t}',\mathbf{m}')} + \lambda_{c',k'} \cdot Profit_{\Sigma, (c',k')} (P_R \cup \{(k', h', (\mathbf{t}', \mathbf{m}'))\})$ 
41       end foreach
42        $b_{k',h',(\mathbf{t}',\mathbf{m}')} := (p_{k',h',(\mathbf{t}',\mathbf{m}')} - p_{k,h,(\mathbf{t},\mathbf{m})}) / sf(k', (\mathbf{t}', \mathbf{m}'))$ 
43     else
44        $Ranking := Ranking - \{(k', h', (\mathbf{t}', \mathbf{m}'))\}$ 
45     end if
46   end if
47 end while

```

Algorithmus 4.6 Greedy-Algorithmus, Var. 2 (Wiederverwendung von Zwischenergebnissen)

```

1   $Ranking := \emptyset$  Initialisierung;
2   $P_R := \emptyset$ 
3  foreach  $h \in H$  do
4     $capacity_h := sc(h)$ 
5  end foreach
6  foreach  $(c, k) \in C \times K$  do für jedes Objekt  $k \in K$  den Gesamtgewinn  $p_{\emptyset k}$ 
7     $pcp_{c,k} := Profit_{\Sigma, (c,k)}(\emptyset)$  über alle Anfragen nach  $k$  bei  $P_R = \emptyset$  ermitteln;
8     $p_{\emptyset k} := p_{\emptyset k} + \lambda_{c,k} \cdot pcp_{c,k}$ 
9  end foreach
10 foreach  $(k, h, (\mathbf{t}, \mathbf{m})) \in PR_{P_0}$  do Ermittlung der Nutzwerte aller Replikate,
11   if  $capacity_h \geq sf(k, (\mathbf{t}, \mathbf{m}))$  then deren Materialisierung keine Nebenbedingung verletzt;
12      $p_{k,h,(\mathbf{t},\mathbf{m})} := 0$ 
13     foreach  $c \in C$  do
14        $psr_{c,k,h,(\mathbf{t},\mathbf{m})} := Profit_{\Sigma, (c,k)}(\{(k, h, (\mathbf{t}, \mathbf{m}))\})$ 
15        $p_{k,h,(\mathbf{t},\mathbf{m})} := p_{k,h,(\mathbf{t},\mathbf{m})} + \lambda_{c,k} \cdot psr_{c,k,h,(\mathbf{t},\mathbf{m})}$ 
16     end foreach
17      $b_{k,h,(\mathbf{t},\mathbf{m})} := (p_{k,h,(\mathbf{t},\mathbf{m})} - p_{\emptyset k}) / sf(k, (\mathbf{t}, \mathbf{m}))$ 
18      $Ranking := Ranking \cup \{(k, h, (\mathbf{t}, \mathbf{m}))\}$ 
19   end if
20 end foreach
21 while  $Ranking \neq \emptyset$  do Iteration über allen materialisierbaren potentiellen Replikaten;
22   suche  $(k, h, (\mathbf{t}, \mathbf{m})) \in Ranking$ , für das Auswahl eines Replikates auf
23      $b_{k,h,(\mathbf{t},\mathbf{m})} = \max \{b_{k',h',(\mathbf{t}',\mathbf{m}')} \mid (k', h', (\mathbf{t}', \mathbf{m}')) \in Ranking\}$  Basis der Nutzwertfunktion;
24     if  $b_{k,h,(\mathbf{t},\mathbf{m})} = 0$  then wenn selbst das beste potentielle Replikat
25       halt in Ranking keinen Nutzen hat,
26       end if terminiert der Greedy-Algorithmus;
27      $Ranking := Ranking - \{(k, h, (\mathbf{t}, \mathbf{m}))\}$ 
28     if  $capacity_h \geq sf(k, (\mathbf{t}, \mathbf{m}))$  then wenn das Replikat ohne Verletzung einer
29        $P_R := P_R \cup \{(k, h, (\mathbf{t}, \mathbf{m}))\}$  Nebenbedingung materialisierbar ist,
30        $capacity_h := capacity_h - sf(k, (\mathbf{t}, \mathbf{m}))$  das Replikat  $(k, h, (\mathbf{t}, \mathbf{m}))$  materialisieren
31       foreach  $c \in C$  do und für alle Anfragen, für die sich
32          $pcp_{c,k} := \max \{pcp_{c,k}, psr_{c,k,h,(\mathbf{t},\mathbf{m})}\}$  eine Änderung ergeben haben könnte,
33         end foreach  $pcp_{c,k} = Profit_{\Sigma, (c,k)}(P_R)$  neu berechnen;
34       foreach  $(k', h', (\mathbf{t}', \mathbf{m}')) \in Ranking$  do für alle potentiellen Replikate  $(k', h', (\mathbf{t}', \mathbf{m}'))$ ,
35         if  $k' = k$  then deren Nutzwert sich verändert haben könnte und
36         if  $capacity_{h'} \geq sf(k', (\mathbf{t}', \mathbf{m}'))$  then die ohne Verletzung einer Nebenbedingung
37            $p_{k',h',(\mathbf{t}',\mathbf{m}')} := 0$  materialisiert werden können,
38           foreach  $c' \in C$  do den Nutzwert neu berechnen;
39              $p_{k',h',(\mathbf{t}',\mathbf{m}')} := p_{k',h',(\mathbf{t}',\mathbf{m}')} + \lambda_{c',k'} \cdot \max \{pcp_{c',k'}, psr_{c',k',h',(\mathbf{t}',\mathbf{m}')} \}$ 
40           end foreach
41            $b_{k',h',(\mathbf{t}',\mathbf{m}')} := (p_{k',h',(\mathbf{t}',\mathbf{m}')} - p_{k,h,(\mathbf{t},\mathbf{m})}) / sf(k', (\mathbf{t}', \mathbf{m}'))$ 
42         else
43            $Ranking := Ranking - \{(k', h', (\mathbf{t}', \mathbf{m}'))\}$ 
44         end if
45       end if
46     end if
47   end while

```

Tabelle 4.1: Zeitkomplexität des Greedy-Algorithmus

Algorithmus zur Berechnung von $Profit_{\Sigma,(c,k)}(P_R)$	Zeitkomplexität (gesamt)
Dijkstra-Algorithmus	$O(C \cdot K \log K \cdot B \log B \cdot H ^3 \log H \cdot T ^3 \log T \cdot M ^3 \log M \cdot S \log S \cdot O)$
Bellman-Ford-Algorithmus	$O(C \cdot K \log K \cdot B ^2 \cdot H ^4 \cdot T ^4 \cdot M ^4 \cdot S ^2 \cdot O)$
DAG-SP-Algorithmus	$O(C \cdot K \log K \cdot B \cdot H ^3 \cdot T ^3 \cdot M ^3 \cdot S \cdot O)$

(a) Variante 1 (Algorithmus 4.5)

Algorithmus zur Berechnung von $Profit_{\Sigma,(c,k)}(P_R)$	Zeitkomplexität (gesamt)
Dijkstra-Algorithmus	$O(C \cdot K \log K \cdot B \log B \cdot H ^2 \log H \cdot T ^2 \log T \cdot M ^2 \log M \cdot S \log S \cdot O)$
Bellman-Ford-Algorithmus	$O(C \cdot K \log K \cdot B ^2 \cdot H ^3 \cdot T ^3 \cdot M ^3 \cdot S ^2 \cdot O)$
DAG-SP-Algorithmus	$O(C \cdot K \log K \cdot B \cdot H ^2 \log H \cdot T ^2 \log T \cdot M ^2 \log M \cdot S \cdot O)$

(b) Variante 2 (Algorithmus 4.6)

laubt, durch die Wiederverwendung von Zwischenergebnissen die Zeitkomplexität des Algorithmus auf

$$O(|C| \cdot (|K| + |PR_{P_0}|) \cdot CPX_{Profit_{\Sigma,(c,k)}}(\Sigma) + |PR_{P_0}|^2 \cdot (|C| + \log|PR_{P_0}|)) \quad (4.46)$$

zu reduzieren. Bei Berücksichtigung des Aufwands zur Komposition optimaler Adaptionspfade in Zeile 7 und 14 ergeben sich für die Zeitkomplexität je nach verwendetem Kürzester-Pfad-Algorithmus die Schranken in Tabelle 4.1(b).

In Algorithmus 4.6 wird die Beziehung

$$Profit_{\Sigma,(c,k)}(P_1 \cup P_2) = \max \{ Profit_{\Sigma,(c,k)}(P_1), Profit_{\Sigma,(c,k)}(P_2) \} \quad (4.47)$$

(aus Formel 3.28 auf Seite 67 und Formel 3.42 auf Seite 70) ausgenutzt, um $Profit_{\Sigma,(c,k)}(P_R \cup \{(k, h, (\mathbf{t}, \mathbf{m}))\})$ und $Profit_{\Sigma,(c',k')}(P_R \cup \{(k', h', (\mathbf{t}', \mathbf{m}'))\})$ in Zeile 31 bzw. Zeile 38 aus den bereits vorher ermittelten Zwischenergebnissen

$$\begin{aligned} pcp_{c,k} &= Profit_{\Sigma,(c,k)}(P_R) \quad \text{und} \quad psr_{c,k,h,(\mathbf{t},\mathbf{m})} = Profit_{\Sigma,(c,k)}(\{(k, h, (\mathbf{t}, \mathbf{m}))\}) \quad \text{bzw.} \\ pcp_{c',k'} &= Profit_{\Sigma,(c',k')}(P_R) \quad \text{und} \quad psr_{c',k',h',(\mathbf{t}',\mathbf{m}')} = Profit_{\Sigma,(c,k)}(\{(k', h', (\mathbf{t}', \mathbf{m}'))\}) \end{aligned}$$

zu berechnen. Algorithmus 4.6 setzt jedoch voraus, dass dem Platzierungsmanager hinreichend Speicherkapazität zur Verfügung steht, um

- für alle Anfragen $(c, k) \in C \times K$ den Gewinn $pcp_{c,k} = Profit_{\Sigma,(c,k)}(P_R)$ des optimalen Adaptionspfades zur Beantwortung von (c, k) bei der aktuellen Platzierung P_R sowie
- den Gewinn $psr_{c,k,h,(\mathbf{t},\mathbf{m})} = Profit_{\Sigma,(c,k)}(\{(k, h, (\mathbf{t}, \mathbf{m}))\})$ aus der Beantwortung jeder Anfrage $(c, k) \in C \times K$ bei der Platzierung jeweils eines einzelnen Replikates $(k, h, (\mathbf{t}, \mathbf{m})) \in PR_{P_0}$

zu speichern. Die Speicherplatzkomplexität steigt dadurch gegenüber Algorithmus 4.5 auf Seite 94 von $O(|PR_{P_0}|)$ auf $O(|C| \cdot (|K| + |PR_{P_0}|))$ (ohne Berücksichtigung des Speicherplatzbedarfs des verwendeten Algorithmus zur Komposition optimaler Adaptionspfade).

4.2.4 Verteilte Realisierung der Greedy-Algorithmen

In den vorangegangenen Abschnitten wurden keine Annahmen darüber getroffen, wo die Algorithmen zur Platzierung von Replikaten in einem A-CDN ausgeführt werden. Die Algorithmen 4.5 auf Seite 94 und 4.6 auf Seite 95 können sowohl zentralisiert in einem zentralen Platzierungsmanager als auch verteilt in kooperierenden Platzierungsmanagern realisiert werden. Die zentralisierte Implementierung ist trivial. Wie in Abschnitt 3.1.2 dargestellt, ist sie jedoch mit Problemen in Bezug auf die Skalierbarkeit verbunden. Deshalb soll in diesem Abschnitt dargestellt werden, wie die Greedy-Heuristik aus Abschnitt 4.2.3 verteilt implementiert werden kann.

Bei der verteilten Implementierung handelt eine Menge lokaler Platzierungsmanager in Kooperation eine Platzierung von Replikaten aus. Jeder lokale Platzierungsmanager vertritt dabei ein oder mehrere Surrogate. Die Menge der lokalen Platzierungsmanager sei \mathcal{Loc} . Die Teilmenge der Surrogate, die durch einen lokalen Platzierungsmanager $loc \in \mathcal{Loc}$ vertreten wird, sei $H_{loc} \subseteq H$. Die Menge

$$PR_{P_0}^{H_{loc}} = PR_{P_0} \cap (K \times H_{loc} \times (T \times M)) \quad (4.48)$$

umfasst alle potentiellen Replikate, die in einem der Surrogate $h \in H_{loc}$ materialisiert werden können. Auf Grundlage dieser Definitionen kann die verteilte Implementierung der Greedy-Heuristik wie folgt beschrieben werden:

1. Zu Beginn bestimmt jeder lokale Platzierungsmanager auf Basis der Kenntnis des globalen Systemmodells Σ und Lastmodells Λ den Nutzwert $benefit_{\Sigma, \Lambda}(\{(k, h, (\mathbf{t}, \mathbf{m}))\}, \emptyset)$ aller potentiellen Replikate $(k, h, (\mathbf{t}, \mathbf{m})) \in PR_{P_0}^{H_{loc}}$.
2. Jeder Platzierungsmanager teilt allen anderen Platzierungsmanagern über einen zuverlässigen Multicast [TS03] das Replikat $(k_{loc1}, h_{loc1}, (\mathbf{t}_{loc1}, \mathbf{m}_{loc1}))$ mit dem größten (positiven) Nutzwert aller potentiellen Replikate in $PR_{P_0}^{H_{loc}}$, das ohne Verletzung einer Nebenbedingung materialisiert werden kann, sowie dessen Nutzwert $benefit_{\Sigma, \Lambda}(\{(k_{loc1}, h_{loc1}, (\mathbf{t}_{loc1}, \mathbf{m}_{loc1}))\}, \emptyset)$ mit.
3. Über den Multicast-Kanal empfängt jeder lokale Platzierungsmanager die Menge $Cand_1 = \{(k_{loc1}, h_{loc1}, (\mathbf{t}_{loc1}, \mathbf{m}_{loc1})) \mid loc \in \mathcal{Loc}\}$ aller potentiellen Replikate mit dem jeweils lokal maximalen Nutzwert inklusive deren Nutzwerte $benefit_{\Sigma, \Lambda}(\{(k_{loc1}, h_{loc1}, (\mathbf{t}_{loc1}, \mathbf{m}_{loc1}))\}, \emptyset)$. Aus der Menge $Cand_1$ kann jeder Platzierungsmanager lokal bestimmen, welches Replikat $(k_{global1}, h_{global1}, (\mathbf{t}_{global1}, \mathbf{m}_{global1}))$ global den größten Nutzwert aufweist, und dieses zur Platzierung auswählen⁴:

$$P_{R1} := \{(k_{global1}, h_{global1}, (\mathbf{t}_{global1}, \mathbf{m}_{global1}))\}$$

4. In den folgenden Iterationsschritten i berechnet jeder lokale Platzierungsmanager $loc \in \mathcal{Loc}$ den Nutzwert $benefit_{\Sigma, \Lambda}(\{(k, h, (\mathbf{t}, \mathbf{m}))\}, P_{R(i-1)})$ der verbleibenden potentiellen Replikate $(k, h, (\mathbf{t}, \mathbf{m})) \in PR_{P_0}^{H_{loc}} - P_{R(i-1)}$ (für die sich eine Änderung ergeben haben könnte) neu⁵ und teilt wiederum allen anderen Platzierungsmanagern das Replikat $(k_{loci}, h_{loci}, (\mathbf{t}_{loci}, \mathbf{m}_{loci}))$ mit dem lokal größten (positiven) Nutzwert, das ohne Verletzung einer Nebenbedingung materialisiert werden kann, sowie dessen Nutzwert $benefit_{\Sigma, \Lambda}(\{(k_{loci}, h_{loci}, (\mathbf{t}_{loci}, \mathbf{m}_{loci}))\}, P_{R(i-1)})$ mit.

⁴Damit auch bei mehreren gleichnützlichen potentiellen Replikaten alle lokalen Platzierungsmanager die gleiche Auswahl treffen, muss für diesen Fall eine systemweit einheitliche Auswahlstrategie verwendet werden, z.B.: Allen Surrogaten wird eine eindeutige Identifikationsnummer zugewiesen und bei gleichnützlichen potentiellen Replikaten wird jenes in dem Surrogat mit der kleinsten Identifikationsnummer zur Materialisierung ausgewählt.

⁵Die Neubestimmung der Nutzwerte kann dabei wie in Algorithmus 4.5 auf Seite 94 durch Berechnung des optimalen Adaptionspfades für jede Anfrage bei der Platzierung $P_{Ri} \cup \{(k, h, (\mathbf{t}, \mathbf{m}))\}$ oder wie in Algorithmus 4.6 auf Seite 95 durch Wiederverwendung der vorher ermittelten Zwischenergebnisse $Profit_{\Sigma, (c, k)}(P_{Ri})$ und $Profit_{\Sigma, (c, k)}(\{(k, h, (\mathbf{t}, \mathbf{m}))\})$ erfolgen.

5. Aus der Menge $Cand_i = \{(k_{loc\ i}, h_{loc\ i}, (\mathbf{t}_{loc\ i}, \mathbf{m}_{loc\ i})) \mid loc \in \mathcal{Loc}\}$, die alle lokalen Platzierungsmanager im Iterationsschritt i über den Multicast-Kanal empfangen, kann jeder Platzierungsmanager lokal bestimmen, welches Replikat $(k_{global\ i}, h_{global\ i}, (\mathbf{t}_{global\ i}, \mathbf{m}_{global\ i}))$ global den größten Nutzwert aufweist, und dieses zur Platzierung auswählen:

$$P_{Ri} := P_{R(i-1)} \cup \{(k_{global\ i}, h_{global\ i}, (\mathbf{t}_{global\ i}, \mathbf{m}_{global\ i}))\}$$

6. Kann auf keinem Surrogat $h \in H_{loc}$, das durch einen Platzierungsmanager loc vertreten wird, ein weiteres Replikat platziert werden, weil andernfalls eine Nebenbedingung verletzt wird, $PR_{P_0}^{H_{loc}} - P_{R(i-1)} = \emptyset$ ist oder kein potentielles Replikat einen positiven Nutzwert aufweist, sendet loc den anderen Platzierungsmanagern eine *fertig*-Nachricht über den Multicast-Kanal. Ein Platzierungsmanager, der *fertig* gemeldet hat, empfängt in den folgenden Iterationsschritten weiterhin die Nachrichten der anderen Platzierungsmanager und ermittelt jeweils $(k_{global\ i}, h_{global\ i}, (\mathbf{t}_{global\ i}, \mathbf{m}_{global\ i}))$, sendet aber selbst keine weiteren Nachrichten.
7. Die Schritte 4 bis 6 werden solange wiederholt, bis alle Platzierungsmanager eine *fertig*-Nachricht gesendet haben. Nachdem der letzte Platzierungsmanager im Iterationsschritt j eine *fertig*-Nachricht gesendet hat, ist $P_R = P_{R(j-1)}$ die gemäß der Greedy-Heuristik näherungsweise optimale Platzierung von Replikaten im A-CDN.

Ist ein Platzierungsmanager $loc \in \mathcal{Loc}$ ausgefallen, so scheidet loc aus der Multicast-Gruppe aus. Das Ausscheiden aus der Multicast-Gruppe wird von den anderen Platzierungsmanagern als *fertig*-Meldung gewertet. Somit werden auf den Surrogaten, die durch den ausgefallenen Platzierungsmanager vertreten werden, keine Replikate platziert.

Bei einer Partitionierung des A-CDN nehmen die Platzierungsmanager in jeder Partition an, dass die Platzierungsmanager in der jeweils anderen Partition ausgefallen sind und damit aus der Multicast-Gruppe ausscheiden. Sie bestimmen also in jeder Partition jeweils unabhängig eine Platzierung von Replikaten. Aufgrund der Partitionierung des Accountingsystems können dabei auch nur die Anfragen der Clients in der jeweiligen Netzwerkpartition berücksichtigt werden. Somit ermitteln die Platzierungsmanager in jeder Partition des A-CDN eine Platzierung, die für die jeweilige Partition optimiert ist.

Die Zahl der Multicast-Nachrichten bei der dargestellten verteilten Realisierung ist durch $O(|\mathcal{Loc}| \cdot |PR_{P_0}|)$ beschränkt. Allerdings müssen dabei alle Multicast-Nachrichten global zwischen allen Platzierungsmanagern ausgetauscht werden. Dieses Problem lässt sich umgehen, wenn die Platzierungsmanager nach Regionen gruppiert werden und der Nachrichtenaustausch hierarchisch strukturiert wird (Abb. 4.5):

1. Zunächst senden alle Platzierungsmanager innerhalb einer Gruppe die jeweiligen potentiellen Replikate mit lokal maximalem Nutzwert an einen ausgezeichneten Platzierungsmanager in ihrer Gruppe, den Hauptplatzierungsmanager. Dieser ermittelt aus den empfangenen Nachrichten das potentielle Replikat mit dem größten Nutzwert innerhalb der Gruppe.
2. Daraufhin tauschen die Hauptplatzierungsmanager über Multicast die jeweiligen potentiellen Replikate mit dem größten Nutzwert innerhalb jeder Gruppe inklusive deren Nutzwerte aus. Auf Grundlage der ausgetauschten Informationen kann jeder Hauptplatzierungsmanager das Replikat mit dem global maximalen Nutzwert bestimmen.
3. Zum Abschluss teilt jeder Hauptplatzierungsmanager den Platzierungsmanagern innerhalb seiner Gruppe durch lokalen Multicast mit, welches Replikat schließlich ausgewählt worden ist.

Durch Verwendung einer mehrstufigen Hierarchie ließe sich die Lokalität der Nachrichten noch weiter steigern. Allerdings ist eine größere Tiefe der Hierarchie auch mit erhöhten Verzögerungen verbunden. Deshalb ist eine tiefere Strukturierung nicht sinnvoll.

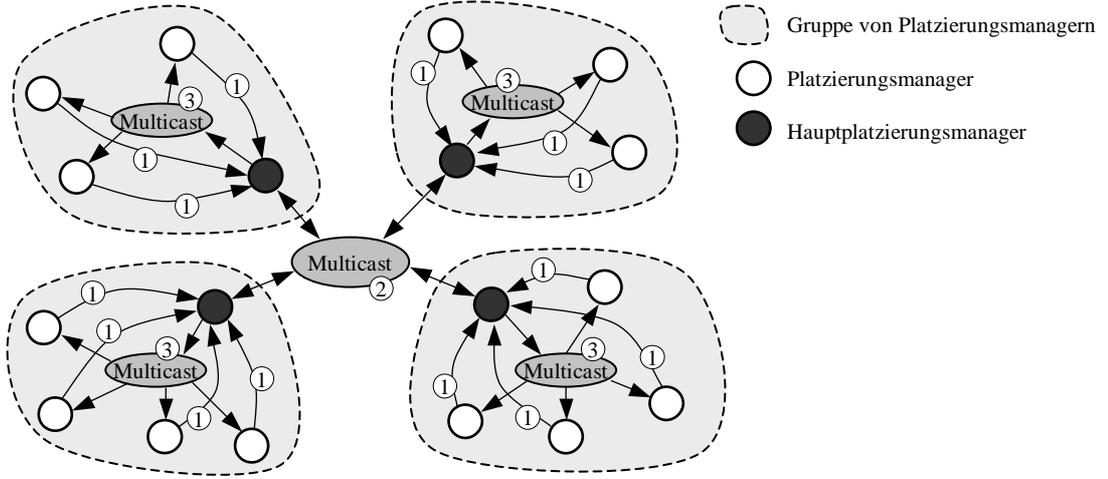


Abbildung 4.5: Hierarchischer Nachrichtenaustausch zwischen verteilten Platzierungsmanagern

Die Nachrichtenkomplexität bei der hierarchischen Kommunikation zwischen Platzierungsmanagern beträgt immer noch $O(|Loc| \cdot |PR_{P_0}|)$, jedoch ist die Zahl der Multicast-Nachrichten im Weitverkehrsbereich in jedem Iterationsschritt durch die Zahl der Gruppen von Platzierungsmanagern beschränkt.

Die Zeitkomplexität des Algorithmus 4.5 auf Seite 94 reduziert sich durch die teilweise parallele Ausführung bei der verteilten Realisierung gegenüber Formel 4.44 auf Seite 93 auf

$$O\left(|PR_{P_0}| |PR_{P_0}^{H_{loc}}| \cdot (|C| \cdot CPX_{Profit_{\Sigma, (c, k)}}(\Sigma) + \log |PR_{P_0}^{H_{loc}}|)\right), \quad (4.49)$$

die des Algorithmus 4.6 auf Seite 95 gegenüber Formel 4.46 auf Seite 96 auf

$$O\left(|C| (|K| + |PR_{P_0}^{H_{loc}}|) \cdot CPX_{Profit_{\Sigma, (c, k)}}(\Sigma) + |PR_{P_0}| |PR_{P_0}^{H_{loc}}| (|C| + \log |PR_{P_0}^{H_{loc}}|)\right). \quad (4.50)$$

Es ist zu beachten, dass auch bei den verteilten Implementierungen der Greedy-Heuristik in jedem Platzierungsmanager ein globales Wissen über den aktuellen Systemzustand Σ und die Anfragelast Λ vorausgesetzt wird. Die Skalierbarkeit des Gesamtsystems ist somit durch die Skalierbarkeit der Implementierung des Monitoring- und Accountingsystems eingeschränkt. Die Implementierung von Monitoring- und Accountingsystem geht jedoch über den Gegenstand dieser Arbeit hinaus und soll hier nicht betrachtet werden.

Kapitel 5

Simulationsszenario

Zur Evaluation der Vorteile adaptivitätssensitiver Platzierung von Replikaten in einem A-CDN soll die Leistungsfähigkeit eines A-CDN mit adaptivitätssensitiven Platzierungsalgorithmen entsprechend Abschnitt 4.2 experimentell evaluiert werden und mit der von A-CDNs mit herkömmlichen Platzierungsmechanismen verglichen werden. In diesem Kapitel wird ein Beispielszenario eingeführt, welches die Grundlage für die experimentelle Evaluation bildet. Das Beispielszenario stellt eine konkrete Instanz des in Abschnitt 3.2 eingeführten formalen Modells dar. Somit wird die Anwendbarkeit des Modells für praktische Anwendungsszenarien illustriert. Die Durchführung der Simulationsexperimente und deren Ergebnisse werden in Kapitel 6 dargestellt und diskutiert.

Im Beispielszenario wird ein global im Internet verteiltes A-CDN betrachtet, welches zur Auslieferung von JPEG-Bildern an mobile und stationäre Internetnutzer verwendet wird. Es werden zwei Formen der Adaption von JPEG-Bildern berücksichtigt:

- Skalierung (genauer gesagt, Verringerung) der räumlichen Auflösung unter Beibehaltung des Seitenverhältnisses
- Erhöhung der Komprimierung (verlustbehaftet) durch Requantisierung mit vermindertem JPEG-Qualitätsfaktor

Die Skalierung der räumlichen Auflösung ist notwendig, um ein Bild der Displayauflösung eines bestimmten Endgerätes anzupassen. Deshalb wird jeweils eine Skalierungsoperation zur Anpassung an die Displayauflösung eines konkreten Endgerätes berücksichtigt. Darüber hinaus führt die Skalierung der räumlichen Auflösung aber auch zu einer Reduktion des Datenvolumens des Bildes. Sie kann somit auch zur Anpassung an die Übertragungsrate der Netzwerkanbindung des mobilen Gerätes oder an die Speicherkapazität der Surrogate verwendet werden. Requantisierung reduziert das Datenvolumen des Bildes. Sie wird zur Anpassung an die Übertragungsrate der Netzwerkanbindung des mobilen Gerätes oder an die Speicherkapazität der Surrogate verwendet. Eine Konvertierung in andere Bildformate oder eine Anpassung der Farbtiefe (Umwandlung in ein Graustufenbild) wird im Beispielszenario nicht berücksichtigt.

5.1 Beschreibung der Objektmerkmale

Entsprechend dem Modell, welches in Abschnitt 3.2 eingeführt wurde, wird die Ausprägung eines Objektes durch die Ausprägung von Typ- und Medienqualitätsmerkmalen beschrieben. Der Typ eines Objektes wird im Beispielszenario durch ein Tupel $\mathbf{t} = (\text{type}, \text{subtype})$ aus der Menge $T = \{(\text{image}, \text{jpeg})\}$ beschrieben. Da nur JPEG-Bilder und keine Formatkonvertierungen berücksichtigt werden, ist die Typmenge T einelementig. Der Typ ist für die Betrachtungen im Folgenden also nicht relevant, soll hier aber aus Gründen der Vollständigkeit genannt werden.

Über den Typ hinaus wird die Repräsentation der Inhaltsobjekte durch die folgenden Medienqualitätsmerkmale beschrieben:

- **pix-x**. räumliche Auflösung in x-Dimension aus der Menge $T_{\text{pix-x}} = \mathbb{N}$
- **pix-y**. räumliche Auflösung in y-Dimension aus der Menge $T_{\text{pix-y}} = \mathbb{N}$
- **q-factor**. JPEG-Qualitätsfaktor aus der Menge $T_{\text{q-factor}} = \mathbb{Z}[0 - 100]$ entsprechend der Definition der Independent JPEG Group (IJG) [LGB⁺98]
- **size**. Objektgröße (in Bytes) aus der Menge $T_{\text{size}} = \mathbb{N}$

Da im Beispielszenario die Objektgröße als Medienqualitätsmerkmal definiert ist, ist die Größenfunktion sf , die das Datenvolumen der Bilder beschreibt, eine Projektionsfunktion

$$sf(k, ((\text{type}, \text{subtype}), (\text{pix-x}, \text{pix-y}, \text{q-factor}, \text{size}))) = \text{size}.$$

Die Servicequalität s bei der Auslieferung von Bild-Objekten wird durch die Antwortzeit bestimmt:

- **delay**. Antwortzeit aus der Menge $S_{\text{delay}} = \mathbb{R}_{0+}$

5.2 Netzwerktopologie

Die Simulationsexperimente werden auf Grundlage verschiedener Netzwerktopologien ausgeführt. Diese basieren auf synthetischen Internettologien, die mittels des Internettologie-Generators Inet-3.0 [JWJ02] von der University of Michigan erzeugt werden. Inet-3.0 erzeugt Internettologien auf der Ebene von autonomen Systemen (AS). Die synthetischen Topologien sind zufällig erzeugte Graphen, die typische Eigenschaften des Internet auf AS-Ebene widerspiegeln. Die Modellierung der Internettologien in Inet-3.0 basiert auf empirischen Untersuchungen des Internets im Zeitraum von November 1997 bis Februar 2002. Inet-3.0 stellt jedoch nur Verbindungsinformationen bereit. Die generierten Topologien enthalten keine direkten Informationen bezüglich Latenz, Datenrate o.ä.

Topologien auf AS-Ebene wurden in vielen existierenden Arbeiten zur Platzierung von Replikaten in CDNs verwendet [QPV01, KRR01, KM02, KK03], weil die Zahl der Hops auf AS-Ebene eine angemessene Näherung für die Ende-zu-Ende-Verzögerung [KM02], die Verlässlichkeit und die Stabilität [QPV01] einer Netzwerkverbindung darstellt. Aus diesem Grund war auch die Wahl von AS-Internettologien in dieser Arbeit naheliegend.

Die Zahl der Knoten in den von Inet-3.0 generierten Topologien kann als Parameter gewählt werden, muss jedoch mindestens 3037 (die Zahl der AS zum Beginn der Untersuchungen, auf denen Inet-3.0 basiert) betragen. Für die Simulationsexperimente im Rahmen dieser Arbeit werden fünf verschiedene Topologien mit jeweils 3037 Knoten erzeugt. Originalserver werden zufällig gleichverteilt 10 der 3037 Knoten in der AS-Topologie zugeordnet. Weitere 15 Knoten werden ebenfalls zufällig gleichverteilt als Standort für Zugangspunkte für die Clients ausgewählt¹. Die Clients selbst werden jedoch nicht durch Knoten in der AS-Topologie repräsentiert. Clientknoten werden später zum Modell hinzugefügt (siehe unten).

Die Surrogate eines A-CDN sollten sinnvollerweise in der Nähe der Clients liegen. Deshalb wird im Beispielszenario angenommen, dass in jedem AS, in dem sich ein Zugangspunkt für Clients befindet, ein Surrogat platziert wird. Somit umfasst das A-CDN im Beispielszenario 15 Surrogate, wobei alle Surrogate über die gleiche Speicherkapazität verfügen. Die Speicherkapazität wird in den Experimenten im Intervall zwischen 5 KB und 500 KB entsprechend der Stufen in Tabelle 5.1 variiert. Eine weitere Serie von Experimenten wird mit unbegrenzter Speicherkapazität durchgeführt. Die angenommenen Speicherkapazitäten von 5 KB bis 500 KB pro Surrogat sind sehr gering, verglichen mit der Festplattenkapazität verfügbarer Rechnerysteme von mehreren hundert Gigabyte. So geringe Speicherkapazitäten werden gewählt, weil in den Experimenten nur eine begrenzte Zahl von 100 Objekten berücksichtigt wird und JPEG-Bilder

¹Im Rahmen einer Sensitivitätsanalyse werden auch Experimente mit weniger als 15 Zugangspunkten (3,6,9,12) durchgeführt (Abs. 6.6.10). Systeme mit mehr als 15 Zugangspunkten können aufgrund der hohen Rechenzeiten für die Simulationen im Rahmen dieser Arbeit nicht untersucht werden.

Tabelle 5.1: Speicherkapazität pro Surrogat in den Simulationsexperimenten

Nr.	1	2	3	4	5	6	7	8	9
$sc(h)$ /KB	5	10	15	20	25	30	35	40	50
Nr.	10	11	12	13	14	15	16	17	18
$sc(h)$ /KB	75	100	150	200	250	300	350	500	∞

Tabelle 5.2: Eigenschaften der AS-Topologien

Topologie	0	1	2	3	4
Anzahl der Knoten	25	25	25	25	25
Anzahl der Links	300	300	300	300	300
Zahl der AS-Netzwerk hops pro Link					
Mittelwert	3,36	3,54	3,92	3,49	4,00
Median	3	4	4	3	4
Min	2	2	2	2	2
Max	6	6	6	6	6

im Internet nur einen geringen Speicherplatzbedarf haben (vgl. Abschnitt 5.5). Eine Kapazität von 814 KB pro Surrogat genügt, um in allen Experimenten² alle Bilder in allen Surrogaten in allen benötigten Repräsentationen zu replizieren. Im Mittel genügen dafür bereits ca. 488 KB. Die Gesamtgröße aller Originalreplikat beträgt im Mittel 610 KB.

Die Netzwerkprotokolle im Internet erlauben Anwendungen eine transparente Punkt-zu-Punkt-Kommunikation zwischen beliebigen Netzwerkknoten. Deshalb ist die eigentliche Kommunikationsstruktur auf Anwendungsebene, auf der CDNs arbeiten, die transitive Hülle der AS-Topologie. Aus diesem Grund wird die transitive Hülle der AS-Topologie ermittelt, bei der jede Kante mit der Zahl der Netzwerk hops auf AS-Ebene gewichtet wird. Auf diese Weise können auch alle Knoten, die weder Standort eines Originalservers, eines Surrogates noch eines Zugangspunktes sind, vernachlässigt werden. Die resultierenden AS-Topologien sind vollständige Graphen, die nur noch 25 Knoten und 300 Kanten umfassen. Die maximale Länge einer Kante beträgt in allen fünf Topologien 6 AS-Hops, die mittlere Länge einer Kante im Durchschnitt ca. 3,7 Hops. Die Eigenschaften der ermittelten Topologien sind in Tabelle 5.2 zusammengefasst³.

Da die generierten AS-Topologien keine Informationen bezüglich Latenz oder Datenrate enthalten, können in den Simulationen nur grobe Abschätzungen dieser Werte berücksichtigt werden. Im Beispielszenario wird eine Latenz von 50 ms pro AS-Hop zugrunde gelegt. Dieser Wert entspricht der durchschnittlichen Latenz eines AS-Hops im Internet [KM02, KK03] entsprechend Messungen durch die HP Laboratories [Kar04]. Als effektive Datenrate werden 400 kbit/s

²Gemeint sind alle Experimente entsprechend dem Simulationsmodell, welches in diesem Kapitel beschrieben wird. Bei den Sensitivitätsanalysen (Abs. 6.6.3 bis 6.6.11) weicht der Speicherplatzbedarf für eine vollständige Replikation (d.h. Replikation in allen Surrogaten) aller benötigten Repräsentationen teilweise davon ab. Bei den Experimenten mit deutlich größeren Primärkopien in Abschnitt 6.6.7 beträgt er bis zu 2,7 GB.

³Da die Netzwerktopologie keinen signifikanten Einfluss auf die Leistungsfähigkeit der Platzierungsmechanismen hat, wurden für die Untersuchungen in den Abschnitten 6.6.3 – 6.6.11, 6.7 und 6.8 nur Experimente mit 3 verschiedenen Topologien (Topologie 0 – 2 entspr. Tab. 5.2) durchgeführt.

Tabelle 5.3: Endgeräte und Netzwerkanbindungen

Zugangspunkt	Mobilfunk-Basisstation	WLAN-Hotspot	Festnetz-Einwahlpunkt	
Netzwerk- verbindung	GPRS	IEEE 802.11b	ISDN	DSL
Datenrate	53,6 kbit/s ^a	500 kbit/s ^b	128 kbit/s ^c	768 kbit/s
Verbindungs- kosten	2,93E-6 EUR/Byte ^d	1,98E-7 EUR/Byte ^e	1,85E-7 EUR/Byte ^f	1,52E-8 EUR/Byte ^g

Endgerät	Display- auflösung		↓	↓	↓	↓
Smart Phone, niedrige Auflösung (z.B. Siemens MC60)	101x80	→	✓			
Smart Phone, mittlere Auflösung (z.B. Samsung SGH-E700)	128x160	→	✓			
Smart Phone, hohe Auflösung (z.B. Nokia 6600)	176x208	→	✓			
Palm PDA, niedrige Auflösung (z.B. Palm Zire m150)	160x160	→	✓	✓		
Palm PDA, hohe Auflösung (z.B. Palm Tungsten E)	320x320	→	✓	✓		
Pocket PC PDA (z.B. HP iPAQ 4355)	240x320	→	✓	✓		
Handheld PC PDA (z.B. NEC MobilePro 900c)	640x240	→	✓	✓		
Sub-Notebook (z.B. Casio FIVA MPC-205E)	800x600	→	✓	✓		
Laptop	1024x768 ^h	→	✓	✓		
Arbeitsplatzcomputer	1024x768 ^h	→			✓	✓

^a bei Nutzung von 4 Kanälen (TDMA-Zeitschlitz) und Kodierungsschema CS-2

^b Da die Bruttodatenrate von 11 Mbit/s auf der Luftschnittstelle von mehreren Teilnehmern konkurrierend genutzt wird, wird angenommen, dass jedem Teilnehmer eine Nettodatenrate von 500 kbit/s zur Verfügung gestellt werden kann.

^c bei Nutzung von 2 B-Kanälen

^d Vodafone Tarif MobilData L, Stand März 2004 (0,09 EUR / 30KB) [vodafone.de]

^e Vodafone W-LAN - Internetzeit, Stand März 2004 (3,95 EUR / 30 min) bei 18% Ausnutzungⁱ [vodafone.de]

^f T-Online Tarif T-Online eco, Stand Aug. 2004 (0,016 EUR / 1 min) bei 18% Ausnutzungⁱ [t-online.de]

^g T-Online Tarif dsl 1500 MB, Preis je MB über dem Inklusivvolumen, Stand Aug. 2004 (0,016 EUR / 1 MB) [t-online.de]

^h Moderne Laptops und Arbeitsplatzcomputer können deutlich höhere Auflösungen als 1024x768 unterstützen. Für die Simulationen ist das jedoch wenig relevant, da die primäre Repräsentation fast aller berücksichtigten Bilder eine geringere Auflösung haben, die Bilder somit auf den Geräten mit maximaler Auflösung dargestellt werden können.

ⁱ WLAN und ISDN werden nur zu Zeittarifen, nicht zu Volumentarifen angeboten. Deshalb wird hier der Zeittarif auf einen Volumenpreis abgebildet. Dabei wird angenommen, dass die Verbindung im Mittel zu 18% ausgenutzt wird, also 18% der bezahlten Nutzungszeit Daten mit der maximal möglichen Übertragungsrate übertragen werden. Die Ausnutzung von 18% wird zugrunde gelegt, weil T-Online für die Umrechnung von DSL-Zeittarifen auf DSL-Volumentarife von einer Ausnutzung von 18% ausgeht (T-Online dsl surfzeit 30 0,016 EUR / 1 min; T-Online dsl 1500 MB 0,016 EUR / 1 MB).

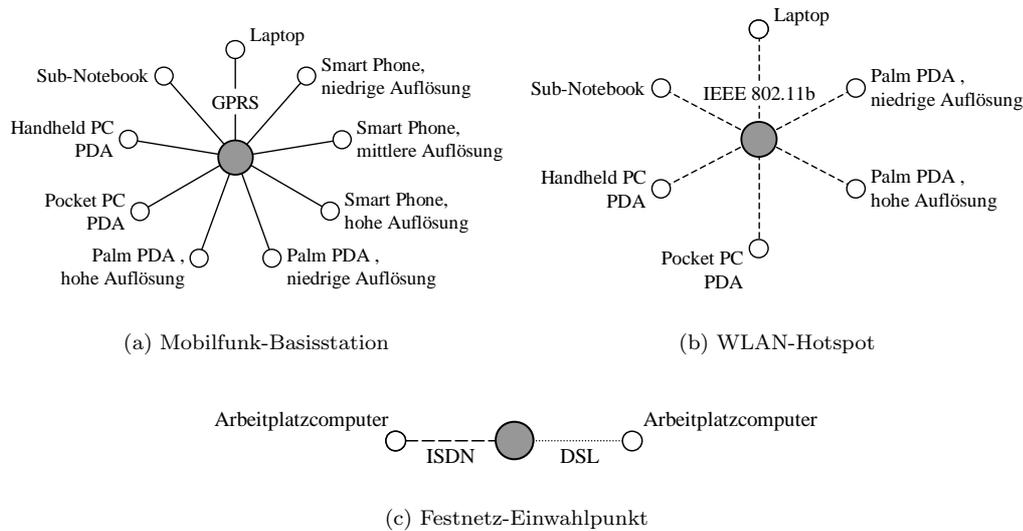


Abbildung 5.1: Zuordnung von Clientknoten zu Zugangspunkten

für alle Verbindungen im Internet-Backbone angenommen. Die Datenrate von 400 kbit/s wurde durch Messungen der effektiven Datenrate der Verbindung zu WWW-Servern in entfernten autonomen Systemen im Rahmen dieser Arbeit abgeschätzt⁴. Die Kosten für die Datenübertragung zwischen autonomen Systemen betragen 1 EUR/GB, unabhängig davon, wieviele AS-Systeme auf dem Pfad zwischen dem Start- und dem Zielknoten traversiert werden.

Die Zugangspunkte bieten den Clients Anschluss an das Internet mit unterschiedlichen Netzwerkverbindungen. Es werden drei Typen von Zugangspunkten berücksichtigt: Mobilfunk-Basisstationen, WLAN-Hotspots und Festnetz-Einwahlpunkte, jeweils $N_d = 5$ Zugangspunkte jeden Typs. Die Mobilfunk-Basisstationen im Modell bieten den Clients Internetzugang über GPRS an. Dabei wird von einer konstanten verfügbaren Datenrate von 53,6 kbit/s ausgegangen. Selbstverständlich können Mobilfunk-Basisstationen in der Praxis auch weitere Protokolle, z.B. HSCSD oder UMTS, unterstützen. Ebenso kann GPRS auch mit anderen Datenraten arbeiten und die Datenrate muss nicht konstant garantiert werden. Beim Simulationsmodell wird davon jedoch vereinfachend abgesehen, da die Auswahl von Netzwerkanbindungen im Simulationsmodell nur das Ziel hat, eine plausible Mannigfaltigkeit von verfügbaren Datenraten und Netzwerkkosten zu modellieren, nicht aber ein konkretes Szenario aus der realen Welt nachzubilden. Die WLAN-Hotspots im Simulationsmodell bieten Internetzugang über IEEE 802.11b. Da die Bruttodatenrate von 11 Mbit/s auf der Luftschnittstelle von mehreren Teilnehmern genutzt wird, wird im Modell vereinfachend angenommen, dass jedem Teilnehmer eine konstante Nettodatenrate von 500 kbit/s zur Verfügung gestellt wird. Auch WLAN-Hotspots können in der Praxis verschiedene Protokolle aus der IEEE 802.11-Protokollfamilie mit unterschiedlichen Datenraten unterstützen. Davon wird im Simulationsmodell jedoch ebenso abgesehen, wie von der Modellierung verschiedener Mobilfunk-Protokolle. Die Festnetz-Einwahlpunkte im Simulationsmodell bieten Festnetzrechnern zwei Formen des Zugangs an, ISDN und DSL. Auch hier werden andere Festnetzanbindungen, z.B. T1 oder Kabelmodem, unberücksichtigt gelassen, da kein konkretes Realweltszenario nachgebildet und die Komplexität des Simulationsmodells beschränkt werden soll. Die Eigenschaften der Netzwerkverbindungen werden in Tabelle 5.3 zusammengefasst.

⁴Da durch die durchgeführten Messungen die Datenrate im Internet nur grob abgeschätzt werden kann, wird bei den Simulationsexperimenten die Sensitivität der Ergebnisse gegenüber der effektiven Datenraten evaluiert (Abs. 6.6.6).

Im Beispielszenario werden zehn verschiedene Typen von Endgeräten berücksichtigt (Tab. 5.3 auf Seite 104). Alle diese Geräte sind in der Lage, JPEG-Bilder farbig darzustellen. Allerdings wird die räumliche Auflösung darstellbarer Bilder durch die Displayauflösung der Geräte eingeschränkt. Somit definiert die Displayauflösung $x_c \times y_c$ jedes Gerätes (wobei c einen Client vom entsprechenden Gerätetyp repräsentiert) die Menge der darstellbaren Repräsentationen der Bilder:

$$mhc(c) = \{((\text{image}, \text{jpeg}), (\text{pix-x}, \text{pix-y}, \text{q-factor}, \text{size}), (\text{delay})) \\ | (\text{pix-x} \leq x_c) \wedge (\text{pix-y} \leq y_c)\}$$

Die Menge der darstellbaren Repräsentationen kann vom Client direkt in der Syntax der IEEE Media Feature Sets [Kly99] beschrieben werden ($(\& (\text{pix-x} \leq x_c) (\text{pix-y} \leq y_c))$) und im HTTP-Protokollkopf [FGM⁺99] der Anfrage bereitgestellt werden. Alternativ kann $mhc(c)$ aber auch aus den Geräteeigenschaften ermittelt werden, die z.B. in Form eines CC/PP-Profiles [KRW⁺04] bereitgestellt werden.

Wie in Abschnitt 3.2.1 dargestellt, werden alle Clients im Systemmodell durch Clientknoten $c \in C$ repräsentiert, wobei ein Clientknoten c nicht für einen individuellen Client steht, sondern eine Gruppe von Clients vertritt, die die gleichen Repräsentationen von Objekten darstellen können, die gleichen Bewertungsmaßstäbe für Qualität haben und über Netzwerkverbindungen mit gleichen Eigenschaften mit dem gleichen Zugangspunkt verbunden sind. Dementsprechend wird jedem Zugangspunkt jeweils maximal ein Clientknoten für jede Kombination von Endgerätetyp und Netzwerkverbindung zugeordnet. Da nicht alle Endgeräte über Netzwerkschnittstellen für alle Netzwerke verfügen und die Zugangspunkte nur ausgewählte Netzwerkverbindungen unterstützen (vgl. Tab. 5.3 auf Seite 104), werden jedem Zugangspunkt Clientknoten entsprechend Abbildung 5.1 zugeordnet.

5.3 Qualitätsmodell

Im Beispielszenario wird die Qualität bei der Auslieferung eines JPEG-Bildes nach drei Kriterien beurteilt:

- der räumlichen Auflösung des Bildes,
- der subjektiv wahrgenommenen Degeneration des Bildes (Detailtreue, Rauschen, Kompressionsartefakte) und
- der Antwortzeit.

Die Definition einer Qualitätsfunktion qf (vgl. Abschnitt 3.2.1), die alle diese Kriterien berücksichtigt, ist nicht trivial. In der Literatur werden zwar eine Vielzahl von Qualitätsmetriken diskutiert⁵, diese decken aber immer nur einzelne Qualitätskriterien ab (z.B. wahrnehmbare Degeneration des Bildes). Um dieses Problem zu adressieren, schlagen Walpole et al. [WKL⁺99] ein mehrdimensionales Qualitätsmodell vor. In diesem Modell werden einzelne Qualitätsdimensionen d identifiziert, für die jeweils separate Qualitätswerte $qf_d(c, k, (\mathbf{m}, \mathbf{s})) \in \mathbb{R} [0, 1]$ ermittelt werden. Eine Qualitätsdimension d ist dabei nicht mit einer Dimension des Parameterraumes $((M_1 \times \dots \times M_\mu) \times (S_1 \times \dots \times S_\sigma))$ der Medien- und Servicequalitätsmerkmale gleichzusetzen, sondern kann durchaus auf Basis mehrerer Medien- oder Servicequalitätsmerkmale definiert sein. Die Qualitätswerte der einzelnen Qualitätsdimensionen werden schließlich zu einem Gesamtqualitätswert $qf(c, k, (\mathbf{m}, \mathbf{s}))$, der alle Dimensionen umspannt, kombiniert. Dabei haben die Qualitätsdimensionen d unterschiedliche Wichtungen w_d (mit $\sum_d w_d = 1$), um die unterschiedliche Auswirkung der verschiedenen Qualitätsdimensionen auf die Gesamtqualität zu berücksichtigen.

Entsprechend der Qualitätskriterien räumliche Auflösung, subjektiv wahrgenommene Degeneration und Antwortzeit werden hier die drei Qualitätsdimensionen res , sqd und del unterschieden. Diese sind entsprechend mit $w_{res} = 0,4$, $w_{sqd} = 0,2$ und $w_{del} = 0,4$ gewichtet. Diese

⁵Ein Überblick von Qualitätsmetriken für Bilder findet sich in [EB98].

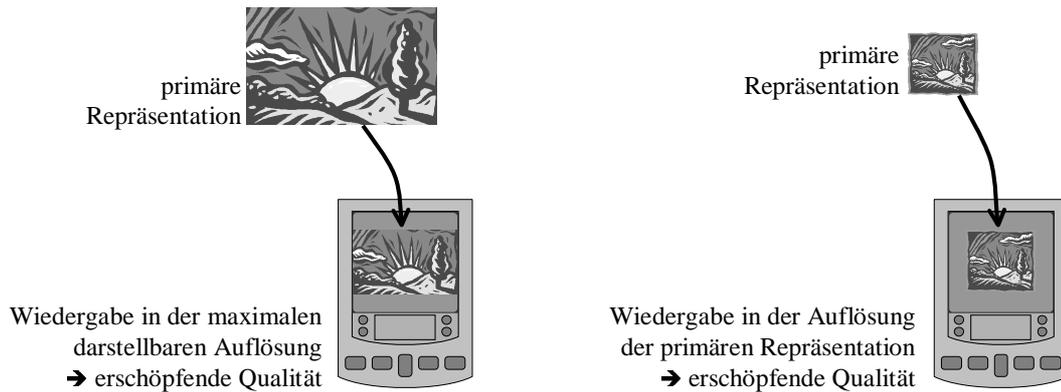


Abbildung 5.2: Qualität von JPEG-Bildern in der Dimension der räumlichen Auflösung

Wichtungen werden gewählt, um die vergleichsweise höhere Toleranz der Nutzer gegenüber Komprimierungsartefakten im Vergleich zu Abstrichen bei der räumlichen Auflösung oder Antwortzeit darzustellen. Sie beruhen jedoch nicht auf empirischen Untersuchungen der subjektiven Bedeutung der verschiedenen Qualitätsdimensionen aus Sicht einer Gruppe von Versuchspersonen. Die Gesamtqualität ist als das gewichtete geometrische Mittel der Qualitätswerte der einzelnen Qualitätsdimensionen definiert:

$$\begin{aligned} qf(c, k, (\mathbf{m}, \mathbf{s})) &= \prod_d qf_d(c, k, (\mathbf{m}, \mathbf{s}))^{w_d} \\ &= qf_{res}(c, k, (\mathbf{m}, \mathbf{s}))^{w_{res}} \cdot qf_{sqd}(c, k, (\mathbf{m}, \mathbf{s}))^{w_{sqd}} \cdot qf_{del}(c, k, (\mathbf{m}, \mathbf{s}))^{w_{del}} \end{aligned} \quad (5.1)$$

5.3.1 Räumliche Auflösung

Die Qualität in der Dimension der räumlichen Auflösung wird durch die Zahl der Bildpunkte $pix = \text{pix-x} \cdot \text{pix-y}$ bestimmt. Die Qualität wird als erschöpfend angesehen ($qf_{res}(c, k, (\mathbf{m}, \mathbf{s})) = 1$), wenn das Bild die maximale Auflösung hat, die unter Einhaltung des Seitenverhältnisses des Originals bzw. der primären Repräsentation vollständig auf dem Display des Endgerätes dargestellt werden kann. Ist jedoch die Auflösung der primären Repräsentation des Bildes in der x- und y-Dimension kleiner als die des Endgerätes, so ist auch die Qualität einer Repräsentation in der Auflösung der primären Repräsentation als erschöpfend in der Qualitätsdimension res zu betrachten (vgl. Abb. 5.2). Somit lässt sich der Schwellwert für erschöpfende Qualität in der Dimension der räumlichen Auflösung folgendermaßen definieren:

$$pix_{exh} = \begin{cases} \min\{x_c, x_{0,k}\}^2 \frac{y_{0,k}}{x_{0,k}} & \text{falls } \frac{x_{0,k}}{y_{0,k}} \geq \frac{x_c}{y_c} \\ \min\{y_c, y_{0,k}\}^2 \frac{x_{0,k}}{y_{0,k}} & \text{sonst} \end{cases} \quad (5.2)$$

Dabei stehen x_c und y_c für die Displayauflösung des Endgerätes und $x_{0,k}$ und $y_{0,k}$ für die Auflösung der primären Repräsentation des Bildes k jeweils in x- bzw. y-Dimension. Für alle Repräsentationen, deren Auflösung größer oder gleich diesem Schwellwert ist, ist der Qualitätswert in der Dimension der räumlichen Auflösung $qf_{res}(c, k, (\mathbf{m}, \mathbf{s})) = 1$. Die Werte x_c und y_c müssen als Metadaten über das Endgerät bereitgestellt werden. Ebenso müssen $x_{0,k}$ und $y_{0,k}$ als Metadaten über das Bild k bekannt sein.

Die Qualität in der Dimension der räumlichen Auflösung ist unzureichend, wenn der Bildinhalt aufgrund der geringen Auflösung nicht mehr erkennbar ist, der Nutzer aus dem Bild also keine Informationen entnehmen kann. Da der Schwellwert für unzureichende Qualität abhängig vom Bildinhalt ist, variiert er bei verschiedenen Bildern. Deshalb wird vorausgesetzt, dass die Inhaltsanbieter für jedes Bild Ausgabebedingungen (vgl. [Hei04]) in Form der Parameter $x_{\min,k}$

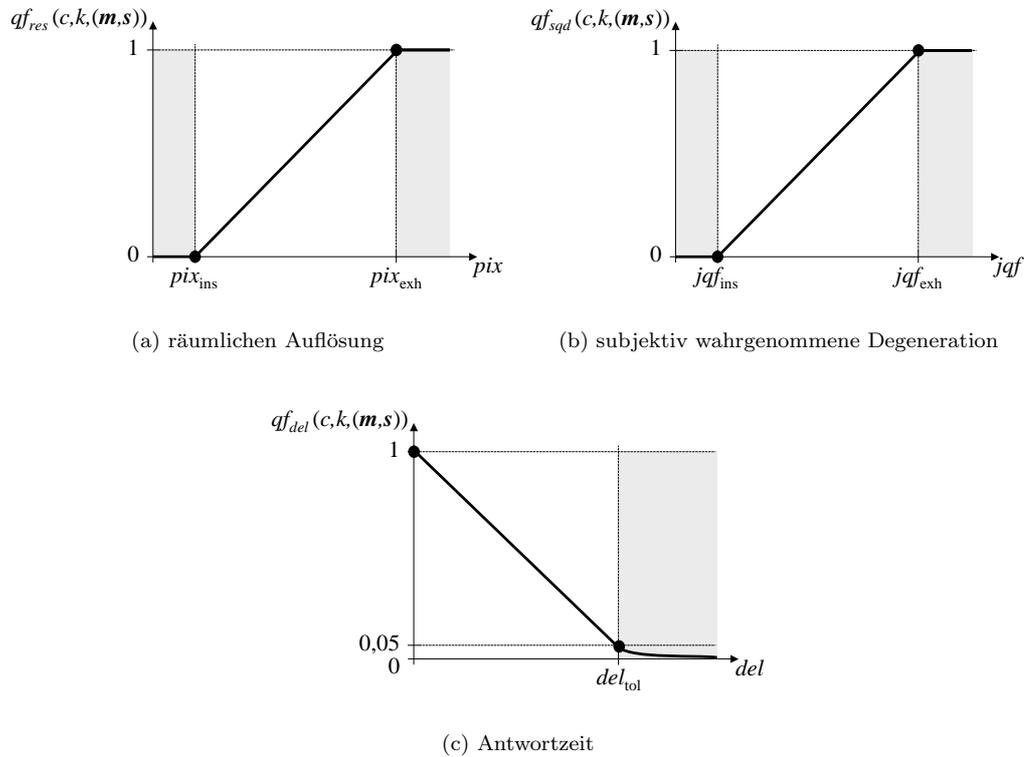


Abbildung 5.3: Qualitätsfunktionen in den verschiedenen Qualitätsdimensionen

und $y_{\min,k}$ definieren und als Metadaten zur Verfügung stellen. Die Bildparameter $x_{\min,k}$ und $y_{\min,k}$ drücken für ein Bild k die kleinste Auflösung in x- bzw. y-Dimension aus, bei der das Bild nutzbar ist. Bilder mit geringerer Auflösung sind für den Nutzer unbrauchbar. Es wird davon ausgegangen, dass der Schwellwert für unzureichende Qualität nur vom Bildinhalt, nicht jedoch vom Nutzer oder Endgerät abhängig ist⁶. Somit ist der Schwellwert für unzureichende Qualität in der Dimension der räumlichen Auflösung folgendermaßen definiert:

$$pix_{\text{ins}} = x_{\min,k} \cdot y_{\min,k} - 1 \quad (5.3)$$

Für alle Repräsentationen, deren Auflösung kleiner oder gleich diesem Schwellwert ist, ist der Qualitätswert in der Dimension der räumlichen Auflösung $qf_{\text{res}}(c, k, (\mathbf{m}, \mathbf{s})) = 0$. Die Qualität zwischen den Schwellwerten pix_{exh} und pix_{ins} für erschöpfende bzw. unzureichende Qualität in der Dimension der räumlichen Auflösung wird im Modell linear angenähert (Abb. 5.3(a)):

$$qf_{\text{res}}(c, k, (\mathbf{m}, \mathbf{s})) = \begin{cases} 0 & \text{falls } pix \leq pix_{\text{ins}} \\ \frac{pix - pix_{\text{ins}}}{pix_{\text{exh}} - pix_{\text{ins}}} & \text{falls } pix_{\text{ins}} < pix < pix_{\text{exh}} \\ 1 & \text{falls } pix_{\text{exh}} \leq pix \end{cases} \quad (5.4)$$

Eine detailliertere Modellierung würde aufwendiger empirischer Studien der subjektiven Qualitätswahrnehmung einer Gruppe von Versuchspersonen bedürfen, die im Rahmen dieser Arbeit nicht durchführbar waren.

⁶Diese Definition des Schwellwertes für unzureichende Qualität, die unabhängig vom Endgerät und vom Nutzer ist, ist für WWW-Szenarien im Allgemeinen ausreichend. Für Anwendungen, in denen die Nutzbarkeit der Inhalte auch vom Nutzer oder vom verwendeten Endgerät abhängig ist, lässt sich das Modell aber ohne weiteres um endgeräte- oder nutzerspezifische Schwellwerte erweitern.

5.3.2 Subjektiv wahrgenommene Degeneration des Bildes

Die Qualitätsdimension sqd der subjektiv wahrgenommenen Degeneration des Bildes beschreibt die Qualitätseinbußen infolge der verlustbehafteten JPEG-Kompression. In der vorliegenden Arbeit wird als Metrik für die Qualität in der sqd -Dimension der JPEG-Qualitätsfaktor entsprechend der Definition durch die Independent JPEG Group (IJG) [LGB⁺98] verwendet. Gemäß dieser Definition ist der JPEG-Qualitätsfaktor ein Wert auf einer Skala von 0 bis 100, wobei 100 für die beste Qualität steht, während bei einem JPEG-Qualitätsfaktor von 0 die Kompression maximal ist. Auf Grundlage des JPEG-Qualitätsfaktors wird die Quantisierungsmatrix ermittelt, die bei der Quantisierung der DCT-transformierten 8×8 -Blöcke des Ausgangsbildes bei der JPEG-Kodierung [PM93] verwendet wird (siehe auch Anhang C.2).

Der JPEG-Qualitätsfaktor nach Definition der IJG ist gemäß [CSE99] ein guter Indikator zur Vorhersage der subjektiv wahrgenommenen Bildqualität. Deshalb wird er in dieser Arbeit als Metrik für die Qualität in der Dimension der subjektiv wahrgenommenen Bilddegeneration verwendet. Er ist explizit durch das Medienqualitätsmerkmal **q-factor** gegeben. Jedoch muss ein JPEG-Qualitätsfaktor im Bereich zwischen 0 und 100 auf einen Qualitätswert $qf_{sqd}(c, k, (\mathbf{m}, \mathbf{s})) \in \mathbb{R}[0, 1]$ gemäß der Definition aus Abschnitt 3.2.1 abgebildet werden. Hierzu wird angenommen, dass für jedes Bild je ein Schwellwert jqf_{exh} für erschöpfende und jqf_{ins} für unzureichende Qualität als Ausgabebedingungen definiert sind und in Form von Metadaten zur Verfügung gestellt werden. Der Qualitätswert für einen JPEG-Qualitätsfaktor von jqf_{exh} oder größer ist 1. Bildrepräsentationen mit einem JPEG-Qualitätsfaktor von jqf_{ins} oder kleiner haben einen Qualitätswert von 0. Die Qualität zwischen den Schwellwerten wird auch in der Dimension der subjektiv wahrgenommenen Degeneration linear angenähert (Abb. 5.3(b)):

$$qf_{sqd}(c, k, (\mathbf{m}, \mathbf{s})) = \begin{cases} 0 & \text{falls } jqf \leq jqf_{\text{ins}} \\ \frac{jqf - jqf_{\text{ins}}}{jqf_{\text{exh}} - jqf_{\text{ins}}} & \text{falls } jqf_{\text{ins}} < jqf < jqf_{\text{exh}} \\ 1 & \text{falls } jqf_{\text{exh}} \leq jqf \end{cases} \quad (5.5)$$

5.3.3 Antwortzeit

Für die Antwortzeit, definiert durch das Servicequalitätsmerkmal **delay**, werden keine Schwellwerte für erschöpfende und unzureichende Qualität definiert. Anstelle dessen wird angenommen, dass die Qualität nur dann optimal ($qf_{\text{del}}(c, k, (\mathbf{m}, \mathbf{s})) = 1$) ist, wenn das Objekt ohne Verzögerungen ausgeliefert werden kann. In der Praxis werden sich Antwortzeitverzögerungen natürlich nicht ausschließen lassen, so dass die Qualität grundsätzlich eingeschränkt sein wird ($qf_{\text{del}}(c, k, (\mathbf{m}, \mathbf{s})) < 1$). Eine obere Schranke der Antwortzeit wird durch die Geduld des Nutzers bestimmt. Es ist davon auszugehen, dass die Toleranz des Nutzers bzgl. der Antwortzeit bei der Nutzung verschiedener Endgeräte bzw. Netzwerkanbindungen variiert. Während ein Festnetznutzer nur minimale Antwortzeiten tolerieren wird, ist von einem Nutzer mit Netzanbindung über ein zellulares Mobilfunknetz mehr Toleranz bzgl. der Antwortzeiten zu erwarten. Deshalb wird im Beispielszenario angenommen, dass für jeden Clientknoten eine Toleranzschwelle del_{tol} bzgl. der Wartezeit als Metainformation verfügbar ist (Tab. 5.4).

Tabelle 5.4: Toleranzschwelle del_{tol} in Abhängigkeit von der Netzwerkanbindung der Clients

Netzwerkanbindung	del_{tol} [ms]
GPRS	2000
IEEE 802.11b	500
ISDN	1000
DSL	500

Bei Testläufen des Simulators hat sich gezeigt, dass die Festlegung der Toleranzschwelle del_{tol} als festen Schwellwert für unzureichende Qualität (d.h. $qf_{del}(c, k, (\mathbf{m}, \mathbf{s})) = 0$ für $del > del_{tol}$) nicht praktikabel ist. Andernfalls können einige Bildobjekte zu bestimmten Clients nie mit hinreichender Qualität ($qf_{del}(c, k, (\mathbf{m}, \mathbf{s})) > 0$) ausgeliefert werden, obwohl das Bildobjekt trotz der Verzögerung objektiv einen Nutzen hat. Deshalb wird in den Simulationen eine Qualitätsfunktion $qf_{del}(c, k, (\mathbf{m}, \mathbf{s}))$ verwendet, die an der Stelle $del = del_{tol}$ einen sehr kleinen, aber positiven Wert (hier: 0,05) annimmt und sich für $del \rightarrow \infty$ asymptotisch 0 annähert. Für Verzögerungen unterhalb der Toleranzschwelle del_{tol} wird eine lineare Näherung für $qf_{del}(c, k, (\mathbf{m}, \mathbf{s}))$ verwendet (Abb. 5.3(c)). Damit ergibt sich folgende Qualitätsfunktion in der Qualitätsdimension der Antwortzeit:

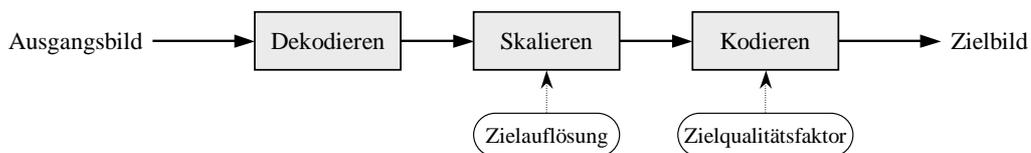
$$qf_{del}(c, k, (\mathbf{m}, \mathbf{s})) = \begin{cases} 1 - 0,95 \frac{delay}{del_{tol}} & \text{falls } delay \leq del_{tol} \\ 0,05 \cdot e^{-19 \frac{delay - del_{tol}}{del_{tol}}} & \text{sonst} \end{cases} \quad (5.6)$$

5.4 Adaptionsoperationen

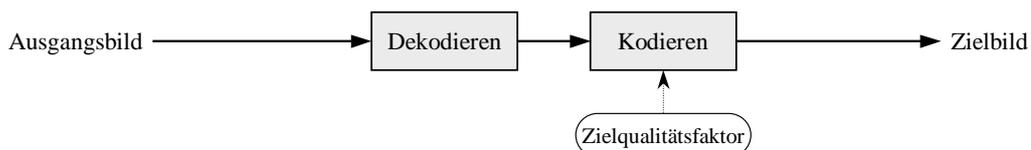
Wie bereits in der Einleitung zu diesem Kapitel erwähnt, werden im Simulationsszenario zwei Formen der Adaption von JPEG-Bildern berücksichtigt:

- Skalierung (genauer gesagt Verringerung) der räumlichen Auflösung unter Beibehaltung des Seitenverhältnisses
- Erhöhung der Komprimierung (verlustbehaftet) durch Requantisierung mit vermindertem JPEG-Qualitätsfaktor

Die Skalierung in der räumlichen Auflösung dient in erster Linie der Adaption der Bilder an die Auflösung der verwendeten Endgeräte. Deshalb werden Adaptionsoperationen zur Skalierung an die Displayauflösung jedes Endgerätes im Simulationsszenario berücksichtigt. Die Displayauflösungen der verschiedenen Endgeräte bilden somit die Zielauflösungen der Skalierungsoperationen (Tab. 5.5(a)). Bei allen Skalierungsoperationen wird das Seitenverhältnis des Ausgangsbildes beibehalten, so dass sich die Auflösung $\text{pix-x} \times \text{pix-y}$ der skalierten Repräsentation aus der



(a) Skalierung der räumlichen Auflösung (inkl. Requantisierung)



(b) Erhöhung der Komprimierung durch Requantisierung

Abbildung 5.4: Adaptionsoperationen

Tabelle 5.5: Zielparameter der Adaptionsoptionen

Nr.	1	2	3	4	5	6	7	8	9
$x_{t,o}$	101	128	160	176	240	320	640	800	1024
$y_{t,o}$	80	160	160	208	320	320	240	600	768

(a) Zielauflösungen

Nr.	1	2	3
$jqt_{t,o}$	30	60	90

(b) Zielqualitätsfaktoren

Auflösung $x_{0,k} \times y_{0,k}$ der primären Repräsentation des Bildes k und der Zielaufösung $x_{t,o} \times y_{t,o}$ der Skalierungsoperation o ergibt:

$$\text{pix-x} := \min \left\{ x_{t,o}, y_{t,o} \frac{x_{0,k}}{y_{0,k}} \right\} \quad (5.7)$$

$$\text{pix-y} := \min \left\{ x_{t,o} \frac{y_{0,k}}{x_{0,k}}, y_{t,o} \right\} \quad (5.8)$$

Im Beispielszenario für die Simulationsexperimente werden keine Adaptionsoptionen auf kodierten Medienobjekten (vgl. Abs. 2.3) berücksichtigt. Folglich erfordert jede Skalierung von JPEG-Bildern, dass das Ausgangsbild vor der Skalierung dekodiert und nach erfolgter Skalierung erneut kodiert wird (Abb. 5.4(a)). Bei der erneuten Kodierung kann die gleiche Quantisierungsmatrix wie beim Ausgangsbild verwendet werden oder aber eine Quantisierung mit vermindertem JPEG-Qualitätsfaktor erfolgen. Somit können Skalierung der räumlichen Auflösung und Erhöhung der Kompression in einer Adaptionsoption kombiniert werden. Als mögliche JPEG-Qualitätsfaktoren werden im Simulationsmodell drei Zielqualitätsfaktoren $jqt_{t,o}$ berücksichtigt (Tab. 5.5(b)). Das Medienqualitätsmerkmal **q-factor** des adaptierten Bildes ergibt sich aus dem Zielqualitätsfaktor bei der JPEG-Kodierung:

$$\text{q-factor} := jqt_{t,o} \quad (5.9)$$

Durch die Kombination von Zielauflösungen und Zielqualitätsfaktoren ergeben sich $9 \times 3 = 27$ verschiedene Ausprägungen von kombinierten Skalierungs- und Komprimierungsoperationen.

Neben den kombinierten Skalierungs- und Komprimierungsoperationen werden im Simulationsmodell auch reine Komprimierungsoperationen (ohne Skalierung, Abb. 5.4(b)) berücksichtigt, die in jeweils einer Ausprägung für jede der drei Zielqualitätsfaktoren in Tabelle 5.5(b) vorliegen.

In den Simulationen wird angenommen, dass alle 30 Adaptionsoptionen in allen Surrogaten installiert sind. Die Kosten der Operationen sowie deren Auswirkungen auf Medien- und Servicequalitätsmerkmale der Objekte (z.B. Größe bzw. Verzögerung) seien in allen Surrogaten gleich. Bezüglich der Verfügbarkeit von Adaptionsoptionen in den Originalservern werden in den Experimenten zwei Fälle unterschieden:

1. Originalserver verfügen nicht über Adaptionmöglichkeiten.
2. In den Originalserver sind die gleichen Adaptionsoptionen wie in den Surrogaten des A-CDN installiert. Die Kosten und die Auswirkungen auf Medien- und Servicequalitätsmerkmale der Objekte sind zu denen in den Surrogaten identisch.

Tabelle 5.6: Ausführungszeiten der Teiloperationen bei der Adaption

Teiloperation	Ausführungszeit [ms]
Dekodieren	$t_{\text{dec}} = 1,54 + 0,015 \cdot blk^{1.07}$
Skalieren	$t_{\text{scl}} = 0,80 + 0,036 \cdot blk^{0.92}$
Kodieren	$t_{\text{enc}} = 0,68 + 0,015 \cdot blk^{1.07}$

blk = Zahl der Blöcke von 8×8 Bildpunkten (im Zielbild)

Die Adaptionkosten und die Auswirkungen auf Medien- und Servicequalitätsmerkmale der Objekte werden folgendermaßen modelliert. Die Größe des adaptierten Bildes (Medienqualitätsmerkmal **size**) wird für alle Adaptionoperationen nach folgender Formel ermittelt:

$$\mathbf{size} := \lfloor p_{1k} + p_{2k} \cdot blk^{p_{3k}} + (e^{2.53 - 0.0127 \cdot scl_{t,o}} - 1) \cdot (p_{4k} - p_{5k} \cdot e^{-p_{6k} \cdot blk}) \rfloor \quad (5.10)$$

Dieser Zusammenhang wurde durch eine Analyse der Transcoding-Eigenschaften von JPEG-Bildern gewonnen, die im Rahmen dieser Arbeit durchgeführt wurde. Eine Beschreibung der Untersuchungen ist in Anhang C.2 gegeben. In obiger Formel ist

$$blk = \left\lceil \frac{\mathbf{pix-x}}{8} \right\rceil \cdot \left\lceil \frac{\mathbf{pix-y}}{8} \right\rceil \quad (5.11)$$

die Zahl der Blöcke von 8×8 Punkten im Zielbild und

$$scl_{t,o} = \begin{cases} \left\lceil \frac{5000}{jqf_{t,o}} \right\rceil & \text{falls } jqf_{t,o} \geq 50 \\ 200 - 2 \cdot jqf_{t,o} & \text{sonst} \end{cases} \quad (5.12)$$

der Skalierungsfaktor zur Ermittlung der Quantisierungsmatrix zur JPEG-Kodierung, der sich aus dem Zielqualitätsfaktor $jqf_{t,o}$ ergibt. Die Parameter $p_{1k}, p_{2k}, p_{3k}, p_{4k}, p_{5k}$ und p_{6k} sind bildspezifische Kodierungsparameter, die die Inhaltsabhängigkeit der Bildgröße ausdrücken. Sie müssen als Metadaten über die Bilder bereitgestellt werden (zur Zuweisung der bildspezifischen Kodierungsparameter zu Bildern im Simulationsmodell siehe Abschnitt 5.5).

Die Verzögerungen, die durch die Adaption entstehen, wurden ebenfalls im Rahmen der Transcoding-Analyse (Anhang C.2) bestimmt. Die Ausführungszeiten für die einzelnen Teiloperationen sind in Tabelle 5.6 zusammengefasst. Die Verzögerung für eine komplexe Adaptionoperation ergibt sich als Summe der Ausführungszeiten der Teiloperationen. Somit lässt sich das Servicequalitätsmerkmal **delay** nach Ausführung einer kombinierten Skalierungs- und Komprimierungsoperation folgendermaßen bestimmen⁷:

$$\mathbf{delay} := \mathbf{delay} + t_{\text{dec}} + t_{\text{scl}} + t_{\text{enc}} \quad (5.13)$$

Analog ergibt sich für das Servicequalitätsmerkmal **delay** nach Ausführung einer reinen Komprimierungsoperation:

$$\mathbf{delay} := \mathbf{delay} + t_{\text{dec}} + t_{\text{enc}} \quad (5.14)$$

Die Kosten für die Ausführung der Adaptionoperationen o werden direkt aus den Ausführungszeiten $t = t_{\text{dec}} + t_{\text{scl}} + t_{\text{enc}}$ bzw. $t = t_{\text{dec}} + t_{\text{enc}}$ abgeleitet:

$$cf_o(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s})) = c_{\text{cpu}} \cdot t \quad (5.15)$$

Die Basiskosten c_{cpu} pro Zeiteinheit CPU-Nutzung werden auf 5 EUR/h festgesetzt. Diese Kosten liegen in der Größenordnung der CPU-Nutzungsgebühren, die auf den WWW-Seiten verschiedener Rechenzentren ausgeschrieben sind (vgl. Anhang D).

⁷Entsprechend Abschnitt 3.2.1 werden die Verzögerungszeiten mit **delay** = 0s initialisiert.

5.5 Objekte und Anfragen

Für die Simulationen in dieser Arbeit werden synthetisch generierte Modelle der Eigenschaften der Objekte und der Anfragelast zugrunde gelegt. Das bedeutet, dass keine Anfrageprotokolle von realen Systemen verwendet werden, sondern die Anfragelasten basierend auf vorgegebenen Eigenschaften zufällig generiert werden. Ebenso werden die Eigenschaften der Objekte entsprechend vorgegebener Verteilungen zufällig generiert.

Die Verwendung synthetischer Lastmodelle hat den Vorteil, dass deren Eigenschaften variiert werden können, und somit der Einfluss verschiedener Parameter bestimmt werden kann. Für das anvisierte Anwendungsszenario sind ohnehin keine geeigneten Anfrageprotokolle realer Systeme verfügbar. Heutzutage ist die Zahl mobiler Internetnutzer noch sehr begrenzt, und die Nutzer greifen vornehmlich auf Angebote zu, die durch den Inhaltsanbieter speziell für mobile Clients aufbereitet werden. Zudem werden Informationen über das Anfrageverhalten der mobilen Nutzer durch die Dienstanbieter erhoben und sind nicht öffentlich verfügbar.

5.5.1 Objekteigenschaften

Relevante Eigenschaften eines Objektes, die im Simulationsmodell nachgebildet werden müssen, sind:

- die räumliche Auflösung der primären Repräsentation
- die kleinste räumliche Auflösung, bei der das Bild nutzbar ist
- der JPEG-Qualitätsfaktor der primären Repräsentation
- die Schwellwerte des JPEG-Qualitätsfaktors für ungenügende und erschöpfende Qualität
- die bildspezifischen JPEG-Kodierungsparameter

Darüber hinaus sind die Eigenschaften und die Platzierung von Originalreplikaten auf den Originalservern festzulegen.

Räumliche Auflösung der primären Repräsentation

Die Modellierung der Verteilung der räumlichen Auflösung der primären Repräsentationen basiert auf einer empirischen Untersuchung der räumlichen Auflösung der Bilder, die im Anfrageprotokoll eines WWW-Proxy für Festnetznutzer referenziert werden. Die Details dieser Analyse sind in Anhang C.1 dargestellt. Die räumliche Auflösung wird dabei durch die Verteilung des Seitenverhältnisses a und der Zahl der Bildpunkte pix der Bilder modelliert. Aus diesen Größen lässt sich die räumliche Auflösung in x- ($pix-x$) und y-Dimension ($pix-y$) folgendermaßen bestimmen:

$$pix-x = \sqrt{pix \cdot a} \quad (5.16)$$

$$pix-y = \sqrt{pix \cdot \frac{1}{a}} \quad (5.17)$$

Entsprechend der Analyse ist die Verteilung des Seitenverhältnisses durch eine Überlagerung von drei logarithmischen Normalverteilungen mit zwei zusätzlichen Sprüngen bei den Seitenverhältnissen 1:1 und 4:3 modelliert. Die Zahl der Bildpunkte folgt einer hybriden, heavy-tailed Verteilung, deren Rumpf einer logarithmischen Normalverteilung und deren Schwanz einer Pareto-Verteilung entspricht (vgl. Anhang C.1). Der Übergangspunkt zwischen Rumpf und Schwanz der Verteilung liegt bei 84 009 Bildpunkten.

Kleinste räumliche Auflösung, bei der ein Bild nutzbar ist

In der Praxis wird die kleinste räumliche Auflösung $(x_{\min,k}, y_{\min,k})$, bei der ein Bild k nutzbar ist, durch den Autor, der das Bild zur Verfügung stellt, festgelegt. Für die Modellierung ist ein solches Verfahren allerdings nicht praktikabel. Deshalb wird im Simulationsszenario angenommen, dass für jedes Bild die kleinste nutzbare Auflösung derjenigen entspricht, bei der die längere Kante des Bildes eine Ausdehnung von 50 Bildpunkten hat, es sei denn, die primäre Repräsentation hat bereits eine geringere Auflösung. In diesem Fall hat die primäre Repräsentation die kleinste nutzbare Auflösung. Somit gilt:

$$x_{\min,k} = \min \left\{ 50, 50 \cdot \frac{x_{0,k}}{y_{0,k}}, x_{0,k} \right\} \quad (5.18)$$

$$y_{\min,k} = \min \left\{ 50 \cdot \frac{y_{0,k}}{x_{0,k}}, 50, y_{0,k} \right\} \quad (5.19)$$

JPEG-Qualitätsfaktor der primären Repräsentation

Laut der empirischen Untersuchung in Anhang C.1 variieren die JPEG-Qualitätsfaktoren von JPEG-Bildern im heutigen WWW sehr stark. Es wurden Werte zwischen 5 und 97 festgestellt. Hohe Qualitätsfaktoren herrschen jedoch vor: Über 60% der Bilder haben einen Qualitätsfaktor von mehr als 80, nur 8,4% einen Qualitätsfaktor unter 50. Der Mittelwert liegt bei 77,7.

Allerdings ist auch hier davon auszugehen, dass einige Bilder für die Übertragung im Internet voradaptiert sind, d.h. in diesem Fall mit Qualitätsverlusten komprimiert. Im Einsatzszenario eines A-CDN werden Inhaltsanbieter die Bilder aber ohne Qualitätsverluste bereitstellen. Deshalb wird im Simulationsmodell angenommen, dass der JPEG-Qualitätsfaktor aller primären Repräsentationen dem Schwellwert jqf_{exh} für erschöpfende Qualität entspricht (hier: $jqf_{exh} = 90$ für alle Bilder, s. unten).

Schwellwerte des JPEG-Qualitätsfaktors

In der Praxis sind die Schwellwerte für ungenügende und erschöpfende Qualität vom Bildinhalt abhängig. Sie müssen deshalb durch den Autor, der die Bilder bereitstellt, spezifiziert werden. Da ein solches Vorgehen für die Entwicklung des Simulationsmodells nicht praktikabel ist, wird hier vereinfachend angenommen, dass die Schwellwerte für alle Bilder gleich sind. Der Schwellwert für erschöpfende Qualität jqf_{exh} wird einheitlich auf 90 festgelegt. Der Schwellwert jqf_{ins} für ungenügende Qualität sei 20 für alle Bilder.

JPEG-Kodierungsparameter

Die JPEG-Kodierungsparameter $(p_{1k}, p_{2k}, p_{3k}, p_{4k}, p_{5k}, p_{6k})$ bestimmen die Größe eines Bildes bei einer bestimmten Auflösung und mit einem bestimmten JPEG-Qualitätsfaktor (vgl. Formel 5.10 auf Seite 112). Im Simulationsmodell wird jedem Bild zufällig gleichverteilt eines der 83 Tupel von JPEG-Kodierungsparametern zugewiesen, die bei der Transcoding-Analyse (Anhang C.2) ermittelt wurde. Auf diese Weise werden sowohl die Verteilungen als auch die Korrelationen der einzelnen Parameter erfasst. Damit ergeben sich für die Originalreplikate im Simulationsmodell Dateigrößen zwischen 0,5 KB und 282 KB mit einem Mittelwert von 6 KB.

Eigenschaften und Platzierung von Originalreplikaten

Die Medienqualitätsmerkmale der Originalreplikate im Simulationsmodell entsprechen denen der primären Repräsentationen. Es wird pro Objekt genau ein Originalreplikat zur Verfügung gestellt. Diese werden zufällig gleichverteilt einem Originalserver im Systemmodell zugeordnet.

5.5.2 Lastmodell

Das Lastmodell beschreibt die Anfragen der Clients in Form der Rate $\lambda_{c,k}$ von Anfragen für jedes Objekt k von jedem Clientknoten c . In den Simulationsexperimenten werden nur Anfragen

nach den 100 populärsten Objekten berücksichtigt. Folglich wird die Menge K der Objekte im Systemmodell auf diese 100 Objekte beschränkt. Experimente mit mehr als 100 Objekten konnten im Rahmen dieser Arbeit aufgrund der hohen Rechenzeiten der Simulationen nicht durchgeführt werden⁸

Die Popularität der Objekte $k \in K$, d.h. die aggregierten Anfrageraten $\lambda_k = \sum_{c \in C} \lambda_{c,k}$ aller Clients, wird in Übereinstimmung mit der wissenschaftlichen Literatur zur Modellierung von WWW-Anfrageraten [Gla94, CBC95, BCF⁺99, BBBC99, JB00, RMV99, CWVL01, LMP01, BJS02] als Zipf-ähnliche Verteilung modelliert:

$$\lambda_k = \Omega \cdot r_k^{-z} \quad \text{mit} \quad \Omega = \frac{1}{\sum_{r=1}^{|K|} r^{-z}} \quad (5.20)$$

Dabei steht r_k für den Rang des Objektes k in einer Liste, in der alle Objekte $k \in K$ entsprechend ihrer Popularität sortiert sind. Entsprechend Formel 5.20 ist die Summe $\sum_{(c,k) \in C \times K} \lambda_{c,k}$ der Anfrageraten aller Anfragen auf 1 normiert. Entsprechend vorliegenden Studien zu den Eigenschaften von WWW-Anfrageraten [Gla94, CBC95, BCF⁺99, BBBC99, JB00, RMV99, BJS02] variiert der Zipf-Parameter z zwischen 0,5 und 1,0. Für Internet-Videodienste wurden Werte zwischen 0,47 [CWVL01] und 0,73 [LMP01] ermittelt. Bei der empirischen Untersuchung von Anfragen nach JPEG-Bildern in der Protokolldatei eines WWW-Proxy im Rahmen dieser Arbeit (Anhang C.1) wurde eine Zipf-ähnliche Verteilung mit $z = 0,6$ festgestellt. Da keine Erkenntnisse über die Verteilung der Objektpopularitäten in einem zukünftigen A-CDN vorliegen, wird in den Experimenten der Zipf-Parameter im Intervall zwischen 0,6 und 1,0 in Schritten von 0,2 variiert.

In Bezug auf die Verteilung von Anfragen auf Clientknoten ist zu erwarten, dass die Leistungsfähigkeit eines A-CDN davon abhängig ist, ob die Objekte von verschiedenen Typen von Clients mit unterschiedlichen Netzanbindungen angefordert werden oder ob jedes Objekt nur von einer bestimmten Gruppe von Clients benötigt wird, die ähnliche Eigenschaften haben. Die Mannigfaltigkeit der Clients, die ein bestimmtes Objekt benötigen, bestimmt die Zahl der Repräsentationen, die gegebenenfalls vorgehalten und ausgeliefert werden müssen. Jedoch ist die Überlappung der Anfrageraten von Objekten verschiedener Typen von Clients in zukünftigen Anwendungsszenarien von A-CDNs kaum abzuschätzen. Deshalb werden in den Experimenten verschiedene Anfrageraten berücksichtigt, bei denen der Grad der Überlappung variiert.

Zu diesem Zweck werden die Clients entsprechend der verwendeten Endgeräte und Netzanbindungen in Klassen eingeteilt (Tabelle 5.7), wobei angenommen wird, dass sich das Anfrageverhalten der Clients innerhalb einer Klasse nicht unterscheidet, während sich das Anfrageverhalten von Clients in unterschiedlichen Klassen unterscheiden kann.

Der Grad der Überlappung der Anfrageraten der verschiedenen Klassen von Clients wird durch einen Parameter $s \in \mathbb{R}[0,1]$ gesteuert. Der Parameter s gibt an, welcher Anteil der Anfragen Objekte k_s aus einer Teilmenge $K_s \subseteq K$ anfordern, die von Clients aller Klassen angefordert werden:

$$s \approx \frac{\sum_{k_s \in K_s} \lambda_{k_s}}{\sum_{k \in K} \lambda_k} \quad (\text{siehe Fußnote } ^9) \quad (5.21)$$

Die übrigen Objekte $k \in \bar{K}_s = (K - K_s)$ werden jeweils nur von Clients einer Klasse angefordert. Folglich lässt sich \bar{K}_s in die disjunkten Teilmengen $K_{c1}, K_{c2}, K_{c3}, K_{c4}, K_{c5}$ und K_{c6} mit $K_{c1} \cup K_{c2} \cup K_{c3} \cup K_{c4} \cup K_{c5} \cup K_{c6} = \bar{K}_s$ zerlegen, wobei eine Menge K_{ci} die Objekte enthält, die von den Clients der Klasse i angefordert werden. Die Zerlegung von \bar{K}_s wird zufällig gewählt, wobei die kumulierte Anfragerate $\sum_{k_{ci} \in K_{ci}} \lambda_{k_{ci}}$ der Objekte $k_{ci} \in K_{ci}$ in jeder Klasse i

⁸Um den Einfluss der Einschränkung auf 100 Objekte auf die Simulationsergebnisse abzuschätzen, wird die Sensitivität der Ergebnisse gegenüber dem Parameter $|K|$ in Experimenten mit geringeren Zahlen von Objekten ($|K| \in \{10, 20, 40, 60, 80\}$) untersucht (vgl. Abs. 6.6.11).

⁹Da jedes Objekt k mit einer Anfragerate von λ_k nur genau einer der Teilmengen K_s oder $\bar{K}_s = (K - K_s)$ zugeordnet werden kann (λ_k nicht anteilig für verschiedene Teilmengen berücksichtigt werden kann), kann die Bedingung in Formel 5.21 bei vorgegebenem s nur näherungsweise eingehalten werden.

Tabelle 5.7: Klasseneinteilung der Clients

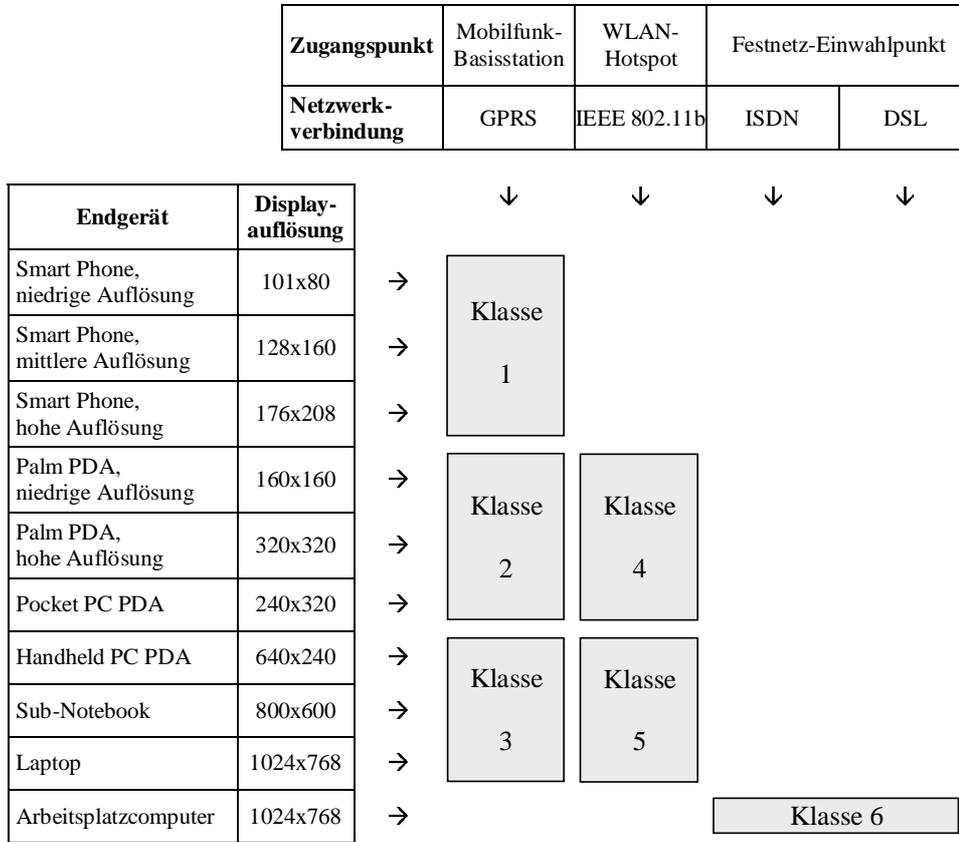


Tabelle 5.8: Verteilung der Anfragen (Abschätzung eines möglichen Anfrageverhaltens)

1 Endgerät (Displayauflösung)	2 Zugangspunkt	3 Netzwerk- verbindung	4	5, 6, 7 Anteil am Anfragevolumen		
			Client- klasse <i>i</i>	σ_i	τ_d	
Smart Phone, niedrige Auflösung (101x80)	Mobilfunk-Basisstation	GPRS	1	40%	20%	6,7%
Smart Phone, mittlere Auflösung (128x160)						6,7%
Smart Phone, hohe Auflösung (176x208)						6,7%
Palm PDA, niedrige Auflösung (160x160)			2		5,0%	
Palm PDA, hohe Auflösung (320x320)					2,5%	
Pocket PC PDA (240x320)					8,5%	
Handheld PC PDA (640x240)					0,8%	
Sub-Notebook (800x600)	3	1,2%				
Laptop (1024x768)		2,0%				
Palm PDA, niedrige Auflösung (160x160)	WLAN-Hotspot	IEEE 802.11b	4	20%	10%	3,1%
Palm PDA, hohe Auflösung (320x320)						1,6%
Pocket PC PDA (240x320)			5		5,3%	
Handheld PC PDA (640x240)					10%	2,0%
Sub-Notebook (800x600)						3,0%
Laptop (1024x768)					5,0%	
Arbeitsplatzcomputer (1024x768)			Festnetz-Einwahlpunkt		ISDN	6
Arbeitsplatzcomputer (1024x768)	DSL	20,0%				

folgende Bedingung erfüllt:

$$\sum_{k_{ci} \in K_{ci}} \lambda_{k_{ci}} \approx \sigma_i \sum_{\bar{k}_s \in \bar{K}_s} \lambda_{\bar{k}_s} \quad (\text{siehe Fußnote }^{10}) \quad (5.22)$$

Dabei gibt σ_i die relative Anfragehäufigkeit der Clients der Klasse i an. Die Werte σ_i der Klassen i sind in Tabelle 5.8, Spalte 6 zusammengefasst. Diese Einteilung beruht auf einer groben Abschätzung eines möglichen Anfrageverhaltens¹¹. Die relativen Anfragehäufigkeiten τ_d der verschiedenen Clienttypen d wurden in gleicher Weise geschätzt und sind in Tabelle 5.8, Spalte 7 dargestellt.

Die kumulierte Häufigkeit der Anfragen von Clients des gleichen Clienttyps d an verschiedenen Zugangspunkten (jeweils durch einen Clientknoten c repräsentiert) wird als gleichverteilt angenommen. Somit ergibt sich die Anfragerate einer Anfrage (c, \bar{k}_s) nach einem Objekt $\bar{k}_s \in \bar{K}_s$ folgendermaßen:

$$\lambda_{c, \bar{k}_s} = \begin{cases} \lambda_{\bar{k}_s} \frac{\tau_d}{\sigma_i} \frac{1}{N_d} & \text{falls } \bar{k}_s \in K_{ci} \\ 0 & \text{sonst} \end{cases} \quad (5.23)$$

Dabei ist d der Clienttyp des Clientknoten c und i dessen Clientklasse. N_d ist die Zahl der Zugangspunkte für Clients des Typs d (entsprechend Abschnitt 5.2 ist N_d im Simulationsmodell für alle Clienttypen gleich). Die Verteilung der Anfragen nach Objekten $k_s \in K_s$, die von Clients aller Klassen angefordert werden, auf die verschiedenen Clienttypen folgt ebenfalls der Verteilung in Tabelle 5.8, Spalte 7. Folglich ergibt sich die Rate der Anfragen (c, k) entsprechend:

$$\lambda_{c, k_s} = \lambda_{k_s} \tau_d \frac{1}{N_d} \quad (5.24)$$

In den Simulationsexperimenten wird der Parameter s zwischen 0,0, 0,5 und 1,0 variiert. Somit ergeben sich 3×3 Tupel (z, s) von Lasteigenschaften. Es werden für jedes Tupel jeweils 4 Experimente mit zufällig generierten Lasten mit den Eigenschaften (z, s) durchgeführt. Das ergibt insgesamt 36 verschiedene Lastmodelle.

Modelle des Anfrageverhaltens von WWW-Clients, die die Basis für die Entwicklung des hier vorgestellten Lastmodells bilden, modellieren nicht individuelle Clients, sondern nur aggregierte Anfragelasten. Folglich lässt sich die Änderung der Anfragelast eines Clientknoten infolge der individuellen Mobilität eines Client (vgl. Abs. 3.2.1) auf der Grundlage dieser Lastmodellierung nicht bestimmen. Deshalb wird im Simulationsmodell die individuelle Mobilität der Clients vernachlässigt. Bei voneinander unabhängigen Mobilitätsverhalten der verschiedenen Clients und einer hinreichend großen Zahl von Clients, die durch jeden Clientknoten c repräsentiert werden, ist allerdings davon auszugehen, dass sich die Effekte der Mobilität individueller Clients auf die aggregierten Anfragelasten der Clientknoten ausgleichen.

5.6 Preissystem

Das Preissystem definiert die nominellen Preise $price_{c,k}$ für die Beantwortung aller Anfragen $(c, k) \in C \times K$. Nominelle Preise bestimmen den Erlös, den ein A-CDN-Anbieter durch Auslieferung von Objekten an Clients erzielt (vgl. Formel 3.37 auf Seite 69). Sie wichten darüber hinaus Kosten und Qualität in der Zielfunktion. In der Praxis können A-CDN-Anbieter und deren Kunden beliebige nominelle Preise aushandeln. Dabei können verschiedene Kriterien berücksichtigt werden, z.B. Größe der Objekte, Bedeutung der Objekte, Kosten für die Auslieferung

¹⁰Da jedes Objekte k mit einer Anfragerate von λ_k nur genau einer Teilmenge K_{ci} zugeordnet werden kann (λ_k nicht anteilig für verschiedene Teilmengen berücksichtigt werden kann), kann die Bedingung in Formel 5.22 bei vorgegebenen σ_i nur näherungsweise eingehalten werden.

¹¹Empirische Daten hierzu sind leider nicht verfügbar. Um die Sensitivität der Ergebnisse bzgl. der konkreten Verteilung der Anfragen einschätzen zu können, werden im Rahmen einer Sensitivitätsanalyse (Abs. 6.6.8) auch Experimente mit alternativen Anfrageverteilungen durchgeführt.

der Objekte, Anforderungen bestimmter Nutzer, usw. Auf jeden Fall werden die ausgehandelten Preise im Allgemeinen in einer Größenordnung liegen, die dem A-CDN-Anbieter erlaubt, bei der Erbringung seines Dienstes Gewinne zu erwirtschaften, und bei der die Kosten für den Inhaltsanbieter nicht signifikant höher sind, als würde er selbst hinreichend Serverkapazitäten aufbauen, um alle Anfragen seiner Clients mit befriedigender Qualität beantworten zu können.

Im Simulationsmodell werden Kriterien, wie die Bedeutung der Objekte oder Anforderungen bestimmter Nutzer, zur Definition der nominellen Preise nicht berücksichtigt. Anstelle dessen werden nominelle Preise ermittelt, die die Größe der Objekte und Kosten für deren Auslieferung in der Form berücksichtigen, dass der A-CDN-Anbieter bei der Erbringung des Dienstes Gewinne erwirtschaften kann und die Kosten für den Inhaltsanbieter nicht höher sind, als wenn er selbst den Dienst erbringen würde. Hierzu werden die nominellen Preise dergestalt festgelegt, dass sie genau kostendeckend sind, wenn die Objekte direkt von den Originalservern mit maximal möglicher Qualität ausgeliefert werden, wobei Inhaltsadaption, wenn nötig, auf dem Originalserver ausgeführt wird¹².

Zur Ermittlung solcher Preise wird der Algorithmus zur Bestimmung optimaler Adaptionspfade verwendet. In der Zielfunktion werden dabei die Kosten vernachlässigt, so dass eine Optimierung ausschließlich nach Qualitätskriterien erfolgt:

$$qual_{\pi,c} = \max! \quad (5.25)$$

Auf diese Weise werden Adaptionspfade ermittelt, die die Objekte mit maximal möglicher Qualität ausliefern. Als Ausgangspunkt für mögliche Adaptionspfade werden nur die Originalreplikate auf den Originalservern berücksichtigt, keine Replikate in Surrogaten des A-CDN. Außerdem werden nur Adaptionoperationen in den Originalservern berücksichtigt. Die Kosten $cost_{\pi}$ und die Qualität $qual_{\pi,c}$ der dabei ermittelten Adaptionspfade π bilden die Grundlage für die Definition der nominellen Preise. Die nominellen Preise $price_{c,k}$ sind so zu bestimmen, dass der Erlös die Kosten deckt, somit der Profit $profit_c(\pi) = 0$ ist:

$$\begin{aligned} rev_{\pi,c} &= cost_{\pi} \\ price_{c,k} \cdot qual_{\pi,c} &= cost_{\pi} \\ price_{c,k} &= \frac{cost_{\pi}}{qual_{\pi,c}} \end{aligned} \quad (5.26)$$

Auf diese Weise ergeben sich für das Simulationsszenario nominelle Preise in Bereich zwischen 0,0001 EUR und 9,36 EUR mit einem Mittelwert von 0,026 EUR bei einer Standardabweichung von 0,034 EUR¹³.

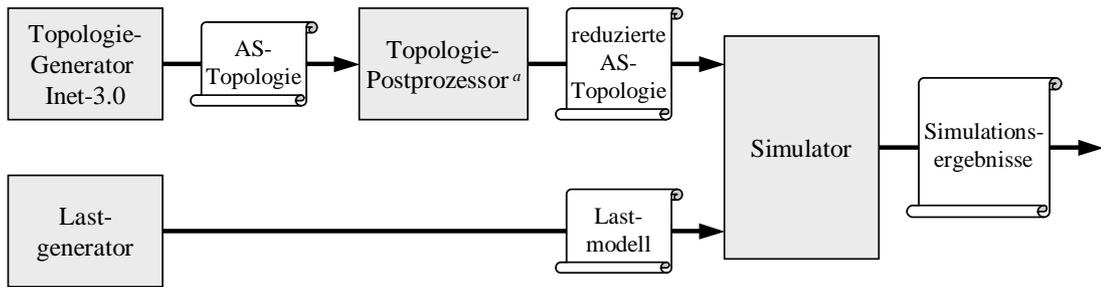
5.7 Implementierung

Die Implementierung des Simulationsmodells erfolgt in der Programmiersprache Java. Lastmodell und Netzwerktopologie für die Simulationsexperimente werden durch externe Werkzeuge generiert und dem Simulator in Form von Eingabedateien zur Verfügung gestellt (Abb. 5.5). Somit können in den Experimenten mit unterschiedlichen Platzierungsmechanismen die selben System- und Lastmodelle verwendet werden, wodurch die Vergleichbarkeit der Ergebnisse sichergestellt ist.

Im Simulator wird für das gegebene System- und Lastmodell eine Platzierung P_R von Replikaten bestimmt und für jede Anfrage der optimale Adaptionspfad zur Beantwortung der

¹²Man beachte, dass die auf diese Weise bestimmten nominelle Preise nicht die Eigenschaft erfüllen, dass die Äquivalenz zweier Clients bzgl. der Qualitätsfunktion (Äquivalenzrelation \equiv_{qf} , vgl. Abs. 4.1.4) auch Äquivalenz bzgl. des Preissystems ($\equiv_{(price_{c,k})}$) impliziert.

¹³Obwohl in der Praxis nominelle Preise in der Größenordnung entsprechend Formel 5.26 zu erwarten sind, können die konkreten Werte davon abweichen. Um den Einfluss der nominellen Preise auf die Simulationsergebnisse abschätzen zu können, werden deshalb im Rahmen einer Sensitivitätsanalyse auch Experimente mit geringeren oder höheren nominellen Preisen durchgeführt (vgl. Abs. 6.6.3).



^a Durch den Topologie-Postprozessor werden zufällig Knoten als Standorte für Originalserver, Surrogate und Zugangspunkte ausgewählt, die transitive Hülle der AS-Topologie gebildet und diese um alle Knoten reduziert, die nicht Standort eines Originalservers, Surrogates und Zugangspunktes sind.

Abbildung 5.5: Implementierung des Simulationsmodells

Anfrage bei dieser Platzierung ermittelt. Aus den ermittelten Adaptionspfaden wird der Gesamtgewinn $Profit_{\Sigma,(c,k)}(P_R)$ berechnet und als Simulationsergebnis in eine Ausgabedatei geschrieben. Darüber hinaus wird auch

- die Summe der Anfrageraten aller Anfragen,
- die Summe der Anfrageraten aller Anfragen, die durch Replikate in den Surrogaten des A-CDN beantwortet werden,
- das entsprechend den Anfrageraten gewichtete Mittel der Qualität der Adaptionspfade und
- für jedes Surrogat des A-CDN der Speicherplatz (in Bytes), der durch Replikate belegt ist,

erfasst und in der Ausgabedatei protokolliert. Die Auswertung der Simulationsergebnisse in der Ausgabedatei erfolgt mittels *MATHEMATICA5* [oV04g].

Kapitel 6

Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen

In diesem Kapitel werden die im Rahmen dieser Arbeit eingeführten adaptivitätssensitiven Platzierungsmechanismen evaluiert und deren Leistungsfähigkeit mit der von A-CDN-Ansätzen verglichen, bei denen herkömmliche Verfahren zur Platzierung von Replikaten verwendet werden. Die Evaluation erfolgt durch Simulation auf Basis des in Kapitel 5 eingeführten Simulationsmodells. Bei den durchgeführten Experimenten handelt es sich um sogenannte Offline-Analysen, das heißt, es wird vorausgesetzt, dass die Anfragelast bereits im Voraus bekannt ist und nicht auf Grundlage der Anfragehistorie abgeschätzt wird.

Zunächst werden Metriken zur Bewertung der Leistungsfähigkeit der verschiedenen A-CDN-Ansätze diskutiert (Abs. 6.1). Anschließend werden die A-CDN-Ansätze mit adaptivitätssensitiven und herkömmlichen Platzierungsmechanismen vorgestellt, deren Leistungsfähigkeit in Simulationsexperimenten verglichen werden soll (Abs. 6.2). In den Abschnitten 6.3 und 6.4 wird die Güte der in den Platzierungsalgorithmen verwendeten Greedy-Heuristik und deren Robustheit gegenüber ungenauen Lastinformationen evaluiert. Die Ergebnisse der vergleichenden Analyse der verschiedenen Ansätze werden schließlich in Abschnitt 6.5 vorgestellt und diskutiert. In Abschnitt 6.6 wird die Abhängigkeit der Ergebnisse von den verschiedenen Parametern des System- und Lastmodells untersucht. Die Quellen der Vorteile des adaptivitätssensitiven Ansatzes zur Platzierung von Replikaten gegenüber dem Einsatz herkömmlicher Mechanismen werden in Abschnitt 6.7 diskutiert. Während in den Abschnitten 6.3 bis 6.7 ausschließlich relaying A-CDNs betrachtet werden, werden in Abschnitt 6.8 die Vorteile des adaptivitätssensitiven Ansatzes in hosting A-CDNs evaluiert. Das Kapitel wird in Abschnitt 6.9 mit einer Zusammenfassung der Ergebnisse abgeschlossen.

6.1 Bewertungsmetriken

Wie in Abschnitt 3.2.3 gezeigt, ist das Ziel beim Betrieb eines A-CDN, den Wert der Zielfunktion $Profit_{\Sigma, \Lambda}(P_R)$ zu maximieren. Dementsprechend kann die Leistungsfähigkeit eines Mechanismus zur Platzierung von Replikaten in einem A-CDN durch den Zuwachs des Wertes der Zielfunktion infolge der Replikation gegenüber der Beantwortung aller Anfragen ohne die Nutzung von Replikaten ausgedrückt werden. Um Experimente mit unterschiedlichen Lastmodellen vergleichbar zu machen, wird der Zuwachs des Wertes der Zielfunktion zur Summe der Anfrageraten aller Anfragen ins Verhältnis gesetzt. Damit ergibt sich als Metrik zur Bewertung

der Leistungsfähigkeit eines A-CDN-Ansatzes der mittlere Zuwachs an Gewinn pro Anfrage:

$$\Delta Profit_{\Sigma, \Lambda} = \frac{Profit_{\Sigma, \Lambda}(P_R) - Profit_{\Sigma, \Lambda}(\emptyset)}{\sum_{(c, k) \in C \times K} \lambda_{c, k}} \quad (6.1)$$

Dabei steht $Profit_{\Sigma, \Lambda}(P_R)$ für den Gewinn aus der Beantwortung aller Anfragen bei der optimalen Platzierung P_R und $Profit_{\Sigma, \Lambda}(\emptyset)$ für den Gewinn aus der Beantwortung aller Anfragen ohne die Nutzung von Replikation im A-CDN.

Zum Vergleich der Leistungsfähigkeit verschiedener A-CDN-Ansätze wird die relative Leistungsfähigkeit

$$A_{\text{Ansatz}_1 | \text{Ansatz}_2} = \frac{\Delta Profit_{\Sigma, \Lambda}^{\text{Ansatz}_1}}{\Delta Profit_{\Sigma, \Lambda}^{\text{Ansatz}_2}} \quad (6.2)$$

herangezogen. $A_{\text{Ansatz}_1 | \text{Ansatz}_2} - 1$ drückt den relativen Vorteil eines A-CDN gemäß Ansatz_1 gegenüber einem A-CDN, das dem Ansatz_2 folgt, aus.

Bei der Auswertung der Simulationsexperimente wird mit Mittelwerten der Bewertungsmetriken $\Delta Profit_{\Sigma, \Lambda}$ und $A_{\text{Ansatz}_1 | \text{Ansatz}_2}$ gearbeitet. Für die Mittelwerte wird jeweils das Konfidenzintervall

$$\bar{\mathcal{X}} \pm \Phi^{-1}\left(\frac{\alpha}{2}\right) \frac{\mathcal{S}}{\sqrt{\mathcal{N}}} \quad (6.3)$$

mit einem Konfidenzniveau von $1 - \alpha = 95\%$ abgeschätzt und in den Diagrammen dargestellt. In Formel 6.3 steht $\bar{\mathcal{X}}$ für den Mittelwert, \mathcal{S} für die Standardabweichung und \mathcal{N} für den Umfang (Zahl der Experimente) der Stichprobe von Simulationsergebnissen. Der Mittelwert $\bar{\mathcal{X}}$ und die Standardabweichung \mathcal{S} werden aus der Stichprobe geschätzt. $\Phi^{-1}\left(\frac{\alpha}{2}\right)$ ist das Quantil $\frac{\alpha}{2}$ -ter Ordnung der normierten Normalverteilung. Die Verwendung der Quantile der normierten Normalverteilung zur Abschätzung der Konfidenzintervalle ist aufgrund des Stichprobenumfangs von jeweils mindestens 36 Messwerten gerechtfertigt.

6.2 A-CDN-Ansätze im Vergleich

In der Evaluation sollen A-CDN-Ansätze mit adaptivitätssensitiven Mechanismen zur Platzierung von Replikaten mit A-CDN-Ansätzen, die herkömmliche Platzierungsmechanismen nutzen, verglichen werden. Entsprechend Abschnitt 1.3 gibt es zwei Varianten, wie herkömmliche Platzierungsmechanismen in A-CDNs eingesetzt werden können:

1. Die Surrogate des A-CDN replizieren ausschließlich statisch voradaptierte Repräsentationen. Somit werden die Replikate ausschließlich bei der Verteilung, jedoch nicht bei der Auslieferung adaptiert. Nur wenn eine Anfrage durch ein generisches Originalreplikat beantwortet wird, findet dynamische Inhaltsadaption statt. Bei dieser Variante liegt den Platzierungsmechanismen das Ebenenmodell aus Abbildung 1.13(b) auf Seite 17 zugrunde. A-CDN-Ansätze entsprechend dieser Variante können nur konkrete Anfragen berücksichtigen, bei denen in der Anfrage eine konkrete Repräsentation spezifiziert wird. Gemäß dem Benennungsschema aus Formel 1.1 auf Seite 19 wird die Klasse der A-CDNs, die dieser Variante folgen, mit $K/V/*$ bezeichnet.
2. Es werden nur generische Repräsentationen im A-CDN repliziert. Die Inhaltsadaption findet ausschließlich bei der Auslieferung der Inhalte statt. Bei dieser Variante liegt den Platzierungsmechanismen das Ebenenmodell aus Abbildung 1.13(a) auf Seite 17 zugrunde. Aufgrund der 1:1-Beziehung zwischen den logischen Inhaltsobjekten und den Repräsentationen der Replikate sind die Platzierungsmechanismen in der Lage, generische Anfragen zu berücksichtigen. Die Klasse der A-CDN-Ansätze, die dieser Variante folgen, wird gemäß dem Benennungsschema aus Formel 1.1 auf Seite 19 mit $G/A/*$ bezeichnet.

Bei A-CDNs mit adaptivitätssensitiven Platzierungsmechanismen kann die Inhaltsadaption sowohl bei der Verteilung als auch bei der Auslieferung der Inhalte erfolgen und es können

generische Anfragen berücksichtigt werden. Dementsprechend werden in der Evaluation die A-CDN-Ansätze der Klasse $G/VA/*$ betrachtet.

In Bezug auf die Verfügbarkeit von Adaptionmöglichkeiten werden in den Analysen zwei verschiedene Konfigurationen betrachtet:

1. Alle Surrogate verfügen über die gleichen Fähigkeiten zur Inhaltsadaption. Die Kosten für die Adaption und die Auswirkungen der Adaptionoperationen auf Medien- und Servicequalitätsmerkmale der Objekte unterscheiden sich in den verschiedenen Surrogaten nicht. Originalserver verfügen nicht über Adaptionmöglichkeiten ($*/*/C$).
2. Sowohl Originalserver als auch die Surrogate des A-CDN verfügen über Möglichkeiten zur Inhaltsadaption ($*/*/OC$). In allen Surrogaten und Originalservern sind die gleichen Adaptionoperationen installiert und können an jedem Ort zu den gleichen Adaptionkosten und mit den gleichen Auswirkungen auf Medien- und Servicequalitätsmerkmale der Objekte ausgeführt werden.

Durch Kombination der verschiedenen Platzierungsmechanismen mit den verschiedenen Konfigurationen der Adaptionmöglichkeiten ergeben sich schließlich die folgenden sechs A-CDN-Ansätze, die in der Evaluation verglichen werden:

- $G/VA/C$ mit $K/V/C$ und $G/A/C$
- $G/VA/OC$ mit $K/V/OC$ und $G/A/OC$

Implementierung im Simulator

In den $G/VA/*$ -Ansätzen wird die Implementierung der Greedy-Heuristik entsprechend Algorithmus 4.6 auf Seite 95 zur adaptivitätssensitiven Platzierung von Replikaten verwendet. Zur Komposition optimaler Adaptionspfade wird der Dijkstra-Algorithmus 4.1 auf Seite 80 in Problemgraphen entsprechend Abschnitt 4.1.1 benutzt. Bei der Konstruktion der Problemgraphen werden die heuristischen Einschränkungen des Lösungsraums gemäß Abschnitt 4.1.3 berücksichtigt. Da das Preissystem im Simulationsmodell nicht die Eigenschaft erfüllt, dass Clients mit den gleichen Endgeräteeigenschaften und mit Netzwerkanbindungen des selben Typs äquivalent bezüglich des Preissystems sind, werden im Simulator keine Problemgraphen entsprechend Abschnitt 4.1.4 zur Bestimmung optimaler Adaptionspfade zur Beantwortung einer Menge von Anfragen genutzt.

Zur Platzierung von Replikaten in A-CDNs der Klassen $K/V/*$ und $G/A/*$ wird ebenfalls eine Greedy-Heuristik verwendet. Vorausgegangene Forschungsarbeiten [KRR01, KM02, KKM02] haben gezeigt, dass Greedy-Heuristiken gut geeignet sind, im nicht-adaptivitätssensitiven Fall eine Näherung der optimalen Platzierung zu bestimmen. Um die Ergebnisse für die verschiedenen Ansätze zur Platzierung von Replikaten vergleichen zu können, wird auch für die Ansätze mit herkömmlichen Platzierungsmechanismen die Maximierung des Gesamtgewinns $Profit_{\Sigma, \Lambda}(P_R)$ (vgl. Problem 3.1 auf Seite 70) als Optimierungsziel betrachtet. Auch hier wird die Greedy-Heuristik entsprechend Algorithmus 4.6 auf Seite 95 implementiert. Bei den Ansätzen der Klasse $G/A/*$ ist jedoch die Menge PR_{P_0} der potentiellen Replikate auf Platzierungen der Repräsentationen der Originalreplikate beschränkt:

$$PR_{P_0} = \{(k, h, (\mathbf{t}, \mathbf{m})) \mid (h \in H) \wedge ((k, b, (\mathbf{t}, \mathbf{m})) \in P_0)\} \quad (6.4)$$

Damit liegt die obere Schranke der Zeitkomplexität mit

$$O(|C| \cdot |K| \log|K| \cdot |B| \log|B| \cdot |H|^2 \log|H| \cdot |T| \log|T| \cdot |M| \log|M| \cdot |S| \log|S| \cdot |O|) \quad (6.5)$$

etwas unterhalb der Komplexität bei den $G/VA/*$ -Ansätzen (vgl. Tab. 4.1(b) auf Seite 96).

Bei den Ansätzen der Klasse $K/V/*$ entspricht die Menge der potentiellen Replikate der im adaptivitätssensitiven Fall (Formel 3.30 auf Seite 67). Allerdings muss bei den $K/V/*$ -Ansätzen nur beim Zugriff auf ein Originalreplikat ein Adaptionspfad zur dynamischen Inhaltsadaption ermittelt werden. Wenn eine Anfrage durch ein Replikate in einem Surrogat beantwortet

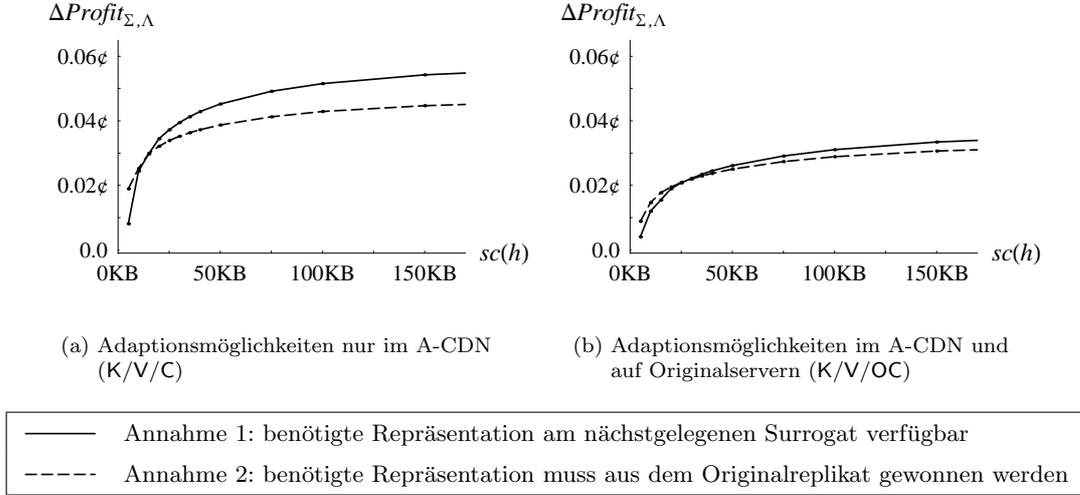


Abbildung 6.1: Vergleich der Möglichkeiten zur Bestimmung einer konkreten Repräsentation für konkrete Anfragen (anhand des mittleren Zuwachses an Gewinn pro Anfrage in Eurocent)

wird, findet keine Inhaltsadaption statt. Um das bei der Berechnung von $Profit_{\Sigma,(c,k)}(P_R)$ im Greedy-Algorithmus zu berücksichtigen, wird analog zum Vorgehen, welches in Abschnitt 4.1.3 beschrieben wird, bei der Konstruktion eines Problemgraphen $\Gamma_{c,k}$ ein zusätzliches Servicequalitätsmerkmal berücksichtigt. Dieses drückt aus, ob das Replikat, von welchem der Adaptionspfad ausgeht, ein Originalreplikat ist. Vorbedingung für die Anwendbarkeit der Adaptionsoptionen ist, dass der Adaptionspfad von einem Originalreplikat ausgeht. Folglich wird bei der Auslieferung eines Replikates aus einem Surrogat der Dijkstra-Algorithmus ausschließlich zur Ermittlung des optimalen physischen Pfades verwendet. Da bei den K/V/*-Ansätzen eine Anfrage (c, k) nur durch eine konkrete Repräsentation $(\mathbf{t}_{c,k}, \mathbf{m}_{c,k})$ beantwortet werden kann, sind darüber hinaus im Problemgraphen $\Gamma_{c,k}$ nur Knoten $(k, n, (\mathbf{t}, \mathbf{m}, \mathbf{s}))$ mit $\mathbf{t} = \mathbf{t}_{c,k}$, $\mathbf{m} = \mathbf{m}_{c,k}$ und $n = c$ mit dem virtuellen Zielknoten $sink_{c,k}$ verbunden.

Obwohl bei den K/V/*-Ansätzen nur beim Zugriff auf Originalreplikate Adaptionspfade zur dynamischen Inhaltsadaption ermittelt werden müssen, ist die obere Schranke der Zeitkomplexität identisch der bei den G/VA/*-Ansätzen. Die tatsächlichen Ausführungszeiten des Platzierungsalgorithmus lagen in den Simulationen für die K/V/*-Ansätze aber um Faktor 5 bis 6 unterhalb den Ausführungszeiten bei den Experimenten für die G/VA/*-Ansätze.

Die Bestimmung der konkreten Repräsentation $(\mathbf{t}_{c,k}, \mathbf{m}_{c,k})$ zur Beantwortung einer Anfrage (c, k) durch ein A-CDN der Klasse K/V/* erfolgt auf Basis von Annahmen über die Verfügbarkeit von Replikaten. In dieser Arbeit wurde mit zwei verschiedenen Annahmen experimentiert:

1. Ein Replikat in der benötigten Repräsentation ist an dem Surrogat verfügbar, das dem anfragendem Client am nächsten liegt.
2. Es ist kein Replikat in der benötigten Repräsentation verfügbar. Die benötigte Repräsentation muss durch dynamische Inhaltsadaption aus dem Originalreplikat gewonnen werden.

In Abbildung 6.1 wird die Auswirkung der Annahmen auf die Leistungsfähigkeit des K/V/C- bzw. K/V/OC-Ansatzes illustriert. Es zeigt sich, dass mit Ausnahme sehr geringer Speicherkapazitäten von ca. 25 KB oder weniger die Ergebnisse bei Annahme 1 denen bei Annahme 2 überlegen sind. Deshalb wird in den folgenden Analysen die Annahme 1 zur Bestimmung der angeforderten Repräsentation bei den K/V/*-Ansätzen zugrunde gelegt.

6.3 Güte der Greedy-Heuristik

In Abschnitt 4.2 wurde gezeigt, dass das Problem der adaptivitätssensitiven Platzierung von Replikaten in einem A-CDN NP-schwer ist, eine exakte Lösung also nicht mit polynomialem Aufwand ermittelt werden kann. Deshalb wurde eine Greedy-Heuristik zur Bestimmung einer Näherungslösung vorgeschlagen. Zur Bewertung der Güte der Greedy-Heuristik sollen in diesem Abschnitt die Näherungslösungen für ausgewählte Beispielszenarien jeweils mit der entsprechenden exakten Lösung verglichen werden.

Die exakte Lösung soll hierzu mittels erschöpfender Suche im Lösungsraum (Alg. 4.4 auf Seite 90) ermittelt werden. Aufgrund der hohen Komplexität der erschöpfenden Suche kann ein solcher Vergleich allerdings nur für wenige Beispielszenarien durchgeführt werden, bei denen aufgrund einer geringen Zahl von Surrogaten, Objekten und Repräsentationen sowie durch eine begrenzte Speicherkapazität in den Surrogaten der Lösungsraum stark eingeschränkt ist. Hierfür wird das in Kapitel 5 vorgestellte Simulationsmodell folgendermaßen eingeschränkt:

1. Die Zahl $|K|$ der betrachteten Inhaltsobjekte wird auf 10 beschränkt.
2. Es werden nur 4 verschiedenen Typen von Endgeräten mit 3 verschiedenen Displayauflösungen betrachtet (Tabelle E.5 auf Seite 190). Dementsprechend werden auch nur die Auflösungen von 128×160 , 240×320 und 1024×768 als Zielaufösungen von Skalierungsoperationen berücksichtigt (vgl. Abs. 5.4).
3. Als Zielqualitätsfaktoren der Adaptionoperationen werden nur die Qualitätsfaktoren 30 und 90 berücksichtigt.
4. Es werden nur 3 Zugangspunkte berücksichtigt, eine Mobilfunk-Basisstation, ein WLAN-Hotspot und ein Festnetz-Einwahlpunkt. Somit umfassen das A-CDN in den Beispielszenarien auch nur 3 Surrogate, jeweils eines im AS jedes Zugangspunktes.
5. Die Speicherkapazität der Surrogate wird im Intervall zwischen 5 KB und 15 KB variiert.

Tabelle 6.1: Güte der Greedy-Heuristik

$A_{\text{Greedy-Heuristik erschöpfende Suche}}$	relative Häufigkeit		
	G/VA/C	G/VA/OC	gesamt
= 100%	42,1%	41,7%	41,9%
≥ 99%	52,6%	58,3%	54,8%
≥ 95%	78,9%	83,3%	80,6%
≥ 90%	78,9%	83,3%	80,6%
≥ 75%	100,0%	100,0%	100,0%

(a) relative Häufigkeit der Experimente, bei denen die relative Leistungsfähigkeit $A_{\text{Greedy-Heuristik|erschöpfende Suche}}$ bestimmte untere Grenzen einhält

$A_{\text{Greedy-Heuristik erschöpfende Suche}}$	G/VA/C	G/VA/OC	gesamt
Mittelwert	95,7%	96,7%	96,1%
Median	99,6%	99,7%	99,6%

(b) Mittelwert und Median der relativen Leistungsfähigkeit $A_{\text{Greedy-Heuristik|erschöpfende Suche}}$

Bei den Analysen werden insgesamt 31 Beispielszenarien betrachtet. Bei 19 Szenarien findet die Inhaltsadaption ausschließlich in den Surrogaten des A-CDN statt (G/VA/C). In den übrigen 12 Szenarien verfügen auch die Originalserver über Adaptionmöglichkeiten (G/VA/OC)¹.

Zur Bewertung der Güte der Greedy-Heuristik wird die relative Leistungsfähigkeit der Greedy-Heuristik bezüglich der erschöpfenden Suche herangezogen. Die Ergebnisse der Analyse sind in Form der relativen Häufigkeiten der Experimente, bei denen die relative Leistungsfähigkeit der Greedy-Heuristik bestimmte untere Grenzen einhält, in Tabelle 6.1(a) dargestellt. Mittelwert und Median der relativen Leistungsfähigkeit sind in Tabelle 6.1(b) zusammengefasst. Aus Tabelle 6.1 sieht man, dass in über 80% der Experimente die Greedy-Heuristik höchstens 5% schlechter abschneidet als die exakte Lösung mittels erschöpfender Suche. In keinem der Experimente ist die relative Leistungsfähigkeit der Greedy-Heuristik schlechter als 75%. Das zeigt, dass die Greedy-Heuristik zur Bestimmung einer Näherungslösung einer optimalen Platzierung von Replikaten in einem A-CDN geeignet ist.

6.4 Robustheit gegenüber ungenauen Lastinformationen

In den experimentellen Untersuchungen in dieser Arbeit werden Offline-Analysen durchgeführt, das heißt, es wird vorausgesetzt, dass die Anfragelast bereits im Voraus bekannt ist. In der Praxis ändert sich die Anfragelast jedoch dynamisch und kann auf Basis der Anfragehistorie nur grob abgeschätzt werden. Fehler in der Abschätzung der Anfragelast äußern sich als Ungenauigkeiten in den Anfrageraten $\lambda_{c,k}$, die bei der Bestimmung einer Platzierung von Replikaten entsprechend den Algorithmen in Kapitel 4 zugrunde gelegt werden. Um die Auswirkungen dieser Ungenauigkeiten auf die Leistungsfähigkeit der Platzierungsmechanismen abschätzen zu können, soll die Robustheit der Ergebnisse gegenüber verfälschten Anfrageraten experimentell untersucht werden.

¹Diesen Experimenten lag das Qualitätsmodell mit der alternativen Qualitätsfunktion in der Dimension der räumlichen Auflösung entsprechend Formel 6.7 auf Seite 136 zugrunde

Tabelle 6.2: Robustheit gegenüber ungenauen Lastinformationen

$A_{\text{verfälschtes Lastmodell exaktes Lastmodell}}$	relative Häufigkeit					
	G/VA/C		G/VA/OC		gesamt	
	<i>bias</i> =2	<i>bias</i> =4	<i>bias</i> =2	<i>bias</i> =4	<i>bias</i> =2	<i>bias</i> =4
= 100%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
≥ 99%	34,4%	29,1%	13,7%	17,6%	24,1%	23,3%
≥ 95%	96,3%	98,3%	85,4%	77,2%	90,8%	87,8%
≥ 90%	100,0%	100,0%	97,8%	97,6%	98,9%	98,8%
≥ 75%	100,0%	100,0%	100,0%	100,0%	100%	100,0%

(a) relative Häufigkeit der Experimente, bei denen die relative Leistungsfähigkeit $A_{\text{verfälschtes Lastmodell|exaktes Lastmodell}}$ bestimmte untere Grenzen einhält

$A_{\text{verfälschtes Lastmodell exaktes Lastmodell}}$	G/VA/C		G/VA/OC		gesamt	
	<i>bias</i> =2	<i>bias</i> =4	<i>bias</i> =2	<i>bias</i> =4	<i>bias</i> =2	<i>bias</i> =4
Mittelwert	98,4%	98,1%	96,9%	96,4%	97,6%	97,3%
Median	98,7%	98,3%	97,6%	96,9%	98,3%	97,8%

(b) Mittelwert und Median der relativen Leistungsfähigkeit $A_{\text{verfälschtes Lastmodell|exaktes Lastmodell}}$

Hierzu wird die Güte von Platzierungen, die auf Grundlage verfälschter Lastinformationen ermittelt werden², mit der von Platzierungen, die auf Grundlage der tatsächlichen Anfrageraten bestimmt werden, verglichen. Die verfälschten Anfrageraten $\lambda_{c,k}^{\text{verfälscht}}$ werden aus den tatsächlichen Anfrageraten $\lambda_{c,k}$ durch Multiplikation mit zufälligen Faktoren $e_{c,k} \in \mathbb{R}[\frac{1}{bias}, bias]$ erzeugt:

$$\lambda_{c,k}^{\text{verfälscht}} = e_{c,k} \cdot \lambda_{c,k} \quad (6.6)$$

Jeder Faktor $e_{c,k}$ wird mit einer Wahrscheinlichkeit von 50% gleichverteilt zufällig aus $\mathbb{R}[1, bias]$ gezogen. Im anderen Fall ist $e_{c,k}$ das Reziproke einer gleichverteilten Zufallszahl aus dem Intervall $\mathbb{R}[1, bias]$.

Es werden Experimente mit den exakten Anfrageraten, verfälschten Anfrageraten mit $bias = 2$ und solchen mit $bias = 4$ durchgeführt. Die Speicherkapazität der Surrogate wird zwischen 50 KB, 100 KB und 300 KB variiert. Die Ergebnisse der Analyse sind in Tabelle 6.2 dargestellt. Tabelle 6.2(a) zeigt, dass nur in ca. 10% der Experimente die Platzierungen die auf Basis von Anfrageraten ermittelt wurden, die um einen Faktor zwischen $\frac{1}{2}$ und 2 verfälscht sind, mehr als 5% schlechter sind, als die, die auf dem exakten Lastmodell beruhen. Selbst wenn die Anfrageraten, die bei der Bestimmung der Platzierung zugrunde gelegt werden, um einen Faktor zwischen $\frac{1}{4}$ und 4 von den tatsächlichen Werten abweichen, ist die relative Häufigkeit von Platzierungen, die mehr als 5% schlechter sind als bei Verwendung der exakten Werte, nur 13%. In über 98% der Experimente mit $bias = 4$ beträgt die relative Leistungsfähigkeit mindestens 90%.

Die Untersuchungen zeigen, dass die Güte der Platzierungen, die in der Praxis auf Grundlage geschätzter Lastinformationen ermittelt werden können, im Allgemeinen nicht signifikant von der Güte der Platzierungen abweicht, die in Offline-Analysen auf Grundlage eines exakten Lastmodells bestimmt werden. Es ist deshalb davon auszugehen, dass sich die Ergebnisse, die in dieser Arbeit in Offline-Analysen gewonnen werden, in der Praxis reproduzieren lassen.

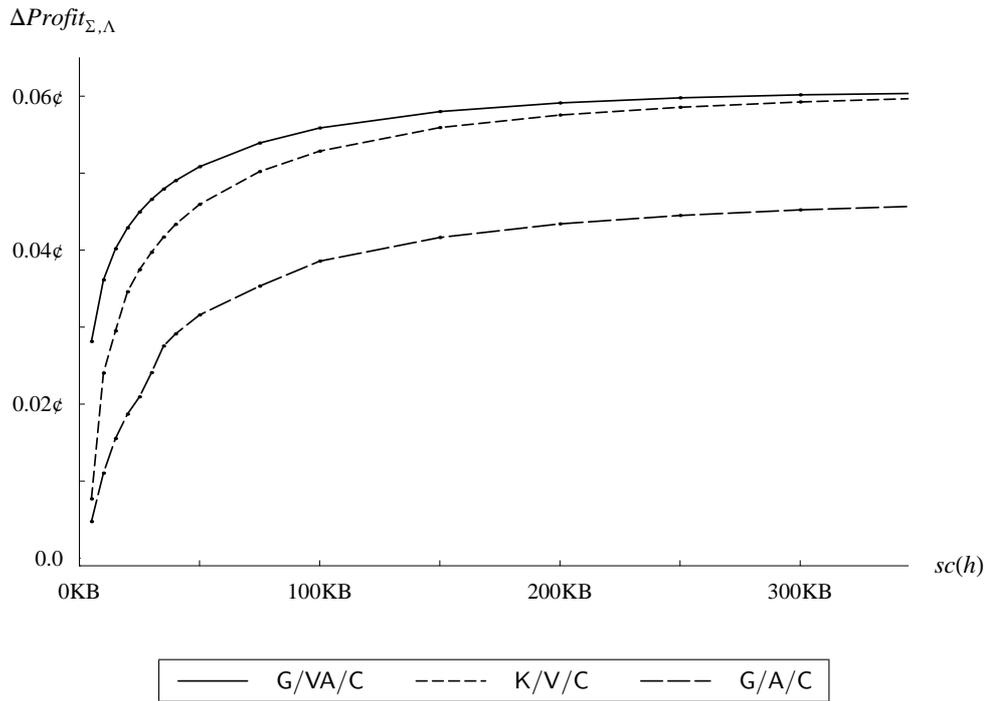
6.5 Leistungsfähigkeit der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs im Vergleich zum Einsatz herkömmlicher Platzierungsmechanismen

In diesem Abschnitt werden die Ergebnisse des Vergleichs der verschiedenen adaptivitätssensitiven und herkömmlichen Ansätze zur Platzierung von Replikaten in A-CDNs vorgestellt. Abbildung 6.2 zeigt den mittleren absoluten Zuwachs $\Delta Profit_{\Sigma, \Lambda}$ an Gewinn pro Anfrage in Abhängigkeit von der Speicherkapazität pro Surrogat $sc(h)$.

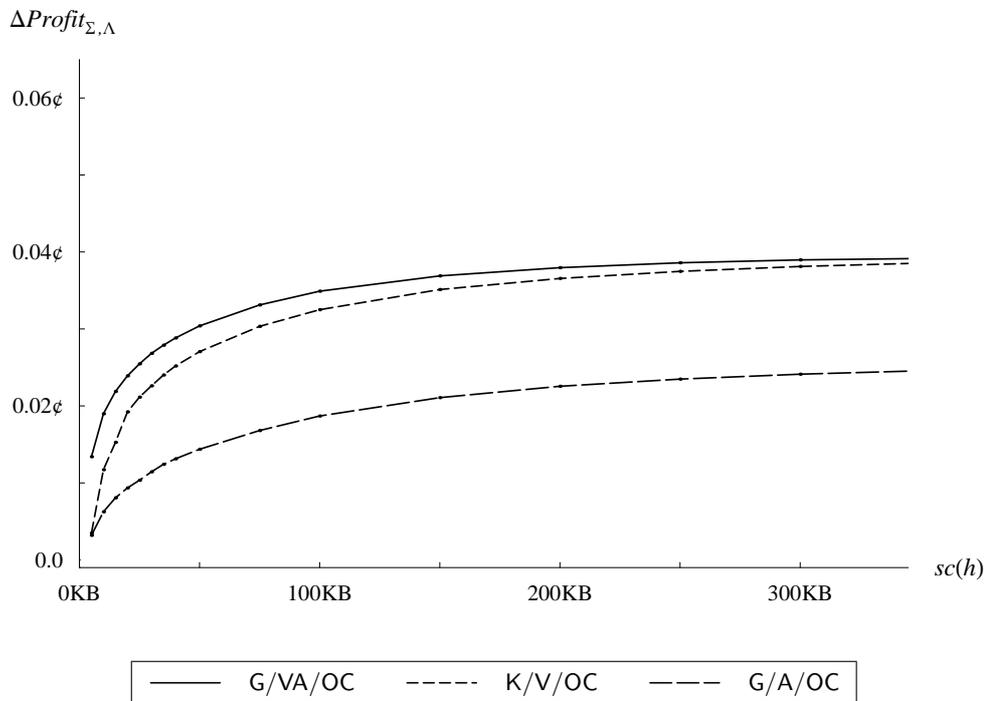
Aus den Diagrammen ist ersichtlich, dass der Zuwachs des Wertes der Zielfunktion bezüglich eines Systems ohne Replikation für alle Ansätze bei Speicherkapazitäten unter 50 KB einen steilen Anstieg zeigt. Der Anstieg der Kurven nimmt jedoch mit wachsender Speicherkapazität ab, bis der absolute Zuwachs an Gewinn bei allen Ansätzen schließlich einen Grenzwert erreicht. Auch bei unendlicher Speicherkapazität der Surrogate sind durch Replikation keine höheren Gewinne realisierbar. Es fällt auf, dass der G/VA/C- und K/V/C- bzw. G/VA/OC- und K/V/OC-Ansatz jeweils den selben Grenzwert erreichen, während der G/A/C- bzw. G/A/OC-Ansatz jeweils gegen einen niedrigeren Grenzwert strebt. Für die G/VA/*- und K/V/*-Ansätze entspricht dieser Grenzwert dem Zugewinn an Profit, der durch vollständige Replikation aller benötigten Repräsentationen erreicht wird. Für die G/A/*-Ansätze ist der Grenzwert der Zugewinn bei vollständiger Replikation aller generischen Repräsentationen. Die mittlere Speicherkapazität $sc(h)$, bei der der jeweilige Grenzwert erreicht wird, ist für die verschiedenen Ansätze in Tabelle 6.3 auf Seite 129 zusammengefasst.

In Abbildung 6.3 auf Seite 130 sind die Zusammenhänge aus Abbildung 6.2 in Form der relativen Leistungsfähigkeit des G/VA/C-Ansatzes bezüglich des K/V/C- und G/A/C-Ansatzes bzw. des G/VA/OC-Ansatzes bezüglich des K/V/OC- und G/A/OC-Ansatzes dargestellt. Dabei

²Die Berechnung der Metrik $\Delta Profit_{\Sigma, \Lambda}$ zur Bewertung einer Platzierung, die auf Grundlage verfälschter Lastinformationen ermittelt wurde, muss natürlich auf Basis der tatsächlichen Anfrageraten erfolgen.



(a) Adaptionsmöglichkeiten nur im A-CDN (*/*/C)



(b) Adaptionsmöglichkeiten im A-CDN und in den Originalservern (*/*/OC)

Abbildung 6.2: Mittlerer Zuwachs an Gewinn pro Anfrage (in Eurocent)

Tabelle 6.3: Mittlere untere Grenze der Speicherkapazität $sc(h)$, bei der $\Delta Profit_{\Sigma, \Lambda}$ maximal ist (inkl. 95%-Konfidenzintervall)

	//C	*/*/OC
G/VA/*	553 KB \pm 42 KB	550 KB \pm 43 KB
K/V/*	489 KB \pm 36 KB	487 KB \pm 36 KB
G/A/*	462 KB \pm 20 KB	458 KB \pm 21 KB

werden jedoch nur die Ergebnisse für Speicherkapazitäten von $sc(h) \geq 25$ KB dargestellt. Bei geringeren Speicherkapazitäten können mehr als 5% der primären Repräsentationen aufgrund ihrer Größe überhaupt nicht repliziert werden und die Zahl der materialisierbaren Replikate pro Surrogat ist so gering, dass die relative Leistungsfähigkeit zwischen den verschiedenen Experimenten starken Schwankungen unterliegt. Somit ist bei geringeren Speicherkapazitäten kein zuverlässiger Mittelwert der relativen Leistungsfähigkeit bestimmbar.

Abbildung 6.2 zeigt, dass A-CDN-Ansätze mit adaptivitätssensitiven Platzierungsmechanismen solchen mit herkömmlichen Platzierungsmechanismen in der Mehrzahl der Experimente überlegen sind. Der relative Vorteil der adaptivitätssensitiven Platzierung nimmt jedoch mit zunehmender Speicherkapazität der Surrogate ab.

Während bei einer Speicherkapazität von 50 KB (ca. $\frac{1}{12}$ der durchschnittlichen Gesamtgröße aller primären Repräsentationen) der G/VA/C-Ansatz im Mittel einen Vorteil von 12% gegenüber dem K/V/C-Ansatz mit sich bringt, beträgt dieser bei einer Speicherkapazität von 100 KB nur noch 6%. Bei einer Speicherkapazitäten von 300 KB oder mehr liegt der Vorteil unter 2% und verschwindet bei durchschnittlich 489 KB (vgl. Tab. 6.3) komplett. In ähnlicher Weise sinkt auch der Vorteil des G/VA/OC-Ansatzes gegenüber dem K/V/OC-Ansatz von 14% bei einer Speicherkapazität von 50 KB über 9% bei 100 KB bis auf 2% bei 300 KB. Bei Speicherkapazitäten von durchschnittlich über 487 KB bringt der G/VA/OC-Ansatz keine Vorteile gegenüber dem K/V/OC-Ansatz mit sich.

Der Rückgang der relativen Leistungsfähigkeit eines adaptivitätssensitiven Ansatzes gegenüber einem herkömmlichen Ansatz, bei dem die vollständig adaptierten Repräsentationen repliziert werden (K/V/C und K/V/OC), ist damit zu erklären, dass bei hinreichender Speicherkapazität die vollständige Replikation aller benötigten Repräsentationen aller Objekte optimal ist. Dadurch können alle Anfragen mittels statisch voradaptierter Replikate beantwortet werden. Vollständige Replikation aller benötigten Repräsentationen in statisch voradaptierter Form ist aber genau das, was auch in einem A-CDN der Klasse K/V/* bei hinreichender Speicherkapazität realisiert wird. Darüber hinaus ist bei vollständiger Replikation der benötigten Repräsentationen auch die Annahme erfüllt, die bei der Bestimmung der anzufordernden Repräsentationen in konkreten Anfragen zugrunde gelegt wird (vgl. Abs. 6.2, Annahme 1), so dass aus der Verwendung konkreter Anfragen in den K/V/*-Ansätzen kein Nachteil erwächst. Folglich lässt sich bei hinreichend großer Speicherkapazität durch adaptivitätssensitive Platzierung kein Vorteil gegenüber den K/V/*-Ansätzen realisieren.

Wie aus Tabelle 6.3 hervorgeht, wird der Grenzwert des absoluten Zuwachses an Gewinn pro Anfrage bei den K/V/*-Ansätzen bereits bei einer Speicherkapazität von 489 KB (K/V/C) bzw. 487 KB (K/V/OC) erreicht, während die untere Grenze der Speicherkapazität für einen maximalen Zuwachs an Gewinn bei den G/VA/*-Ansätzen mit 553 KB (G/A/C) bzw. 550 KB (G/A/OC) etwas darüber liegt. Dieser Unterschied lässt sich dadurch erklären, dass die Platzierungen mittels einer Greedy-Heuristik bestimmt werden. Könnten in den Experimenten optimale Platzierungen der Replikate berücksichtigt werden, wären die adaptivitätssensitiven Platzierungsmechanismen unter allen Umständen mindestens genauso gut wie die herkömmlichen Platzierungsmechanismen, da jede mögliche Platzierung nach herkömmlichen Verfahren auch eine mögliche Platzierung im adaptivitätssensitiven Fall ist. Aufgrund des heuristischen Vorgehens gab es in den Experimenten wenige Fälle, bei denen ein herkömmlicher Platzierungs-

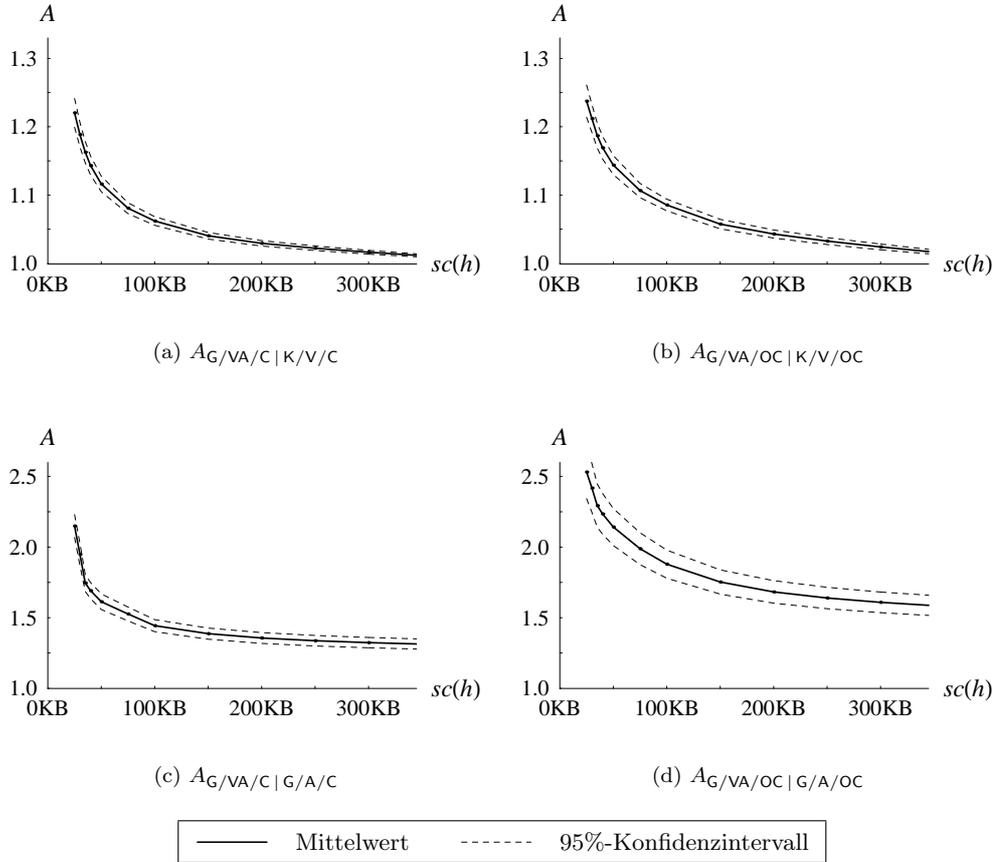


Abbildung 6.3: Relative Leistungsfähigkeit

mechanismus Vorteile gegenüber einem adaptivitätssensitiven Platzierungsmechanismus hatte. So war in ca. 2,1% der Experimente $A_{G/VA/C|K/V/C} < 1$ und in ca. 1,7% der Experimente $A_{G/VA/OC|K/V/OC} < 1$. Der Vorteil des herkömmlichen Platzierungsmechanismus war aber in keinem Fall größer als 0,2%.

Der absolute Vorteil der adaptivitätssensitiven Platzierung von Replikaten gegenüber dem $K/V/C$ -Ansatz ist größer als der gegenüber dem $K/V/OC$ -Ansatz (Abb. 6.2 auf Seite 128). Das liegt daran, dass die Beantwortung von Anfragen durch Originalreplikate günstiger in Bezug auf Kosten und Verzögerungen ist, wenn auch Originalserver über Adaptionmöglichkeiten verfügen, und bei den $K/V/*$ -Ansätzen mehr Anfragen durch Originalreplikate beantwortet werden müssen (Abb. 6.4). Da im Fall $P_R = \emptyset$ alle Anfragen mittels Originalreplikaten beantwortet werden, steigt aber auch $Profit_{\Sigma, \Lambda}(\emptyset)$, wenn die Originalserver über Adaptionmöglichkeiten verfügen. Deshalb fällt der absolute Zuwachs $\Delta Profit_{\Sigma, \Lambda}$ an Gewinn pro Anfrage bei den $*/*/OC$ -Ansätzen insgesamt geringer aus als bei den Konfigurationen ohne Adaptionmöglichkeiten in den Originalservern. Infolge dessen liegt der relative Vorteil des $G/VA/C$ -Ansatzes gegenüber dem $K/V/C$ -Ansatz trotz höheren absoluten Vorteils etwas unter dem relativen Vorteil des $G/VA/OC$ -Ansatzes gegenüber dem $K/V/OC$ -Ansatz.

Auch der Vorteil der $G/VA/*$ -Ansätze gegenüber den $G/A/*$ -Ansätzen nimmt mit zunehmender Speicherkapazität der Surrogate ab, liegt aber mit $A_{G/VA/C|G/A/C} - 1 = 44\%$ bzw. $A_{G/VA/OC|G/A/OC} - 1 = 88\%$ an der Stelle $sc(h) = 100\text{KB}$ deutlich höher als der Vorteil gegenüber den $K/V/*$ -Ansätzen (Abb. 6.3(c) und 6.3(d)). Darüber hinaus bringt adaptivitätssensitive Platzierung auch bei $sc(h) \rightarrow \infty$ im Mittel einen relativen Vorteil von 29% gegenüber dem $G/A/C$ -Ansatz bzw. 54% gegenüber dem $G/A/OC$ -Ansatz mit sich.

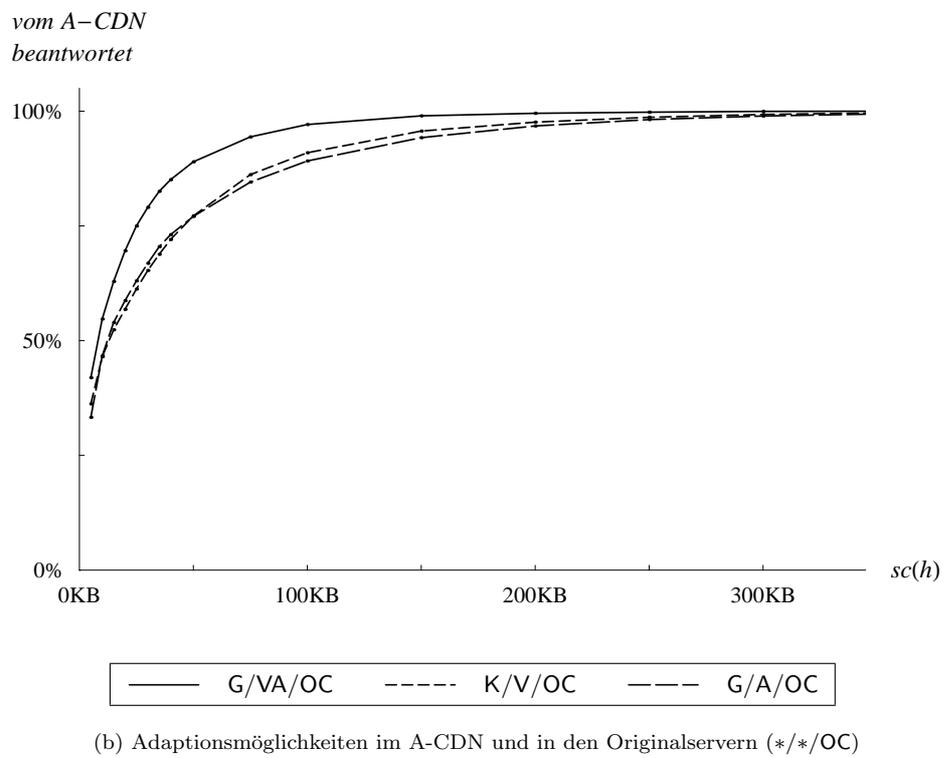
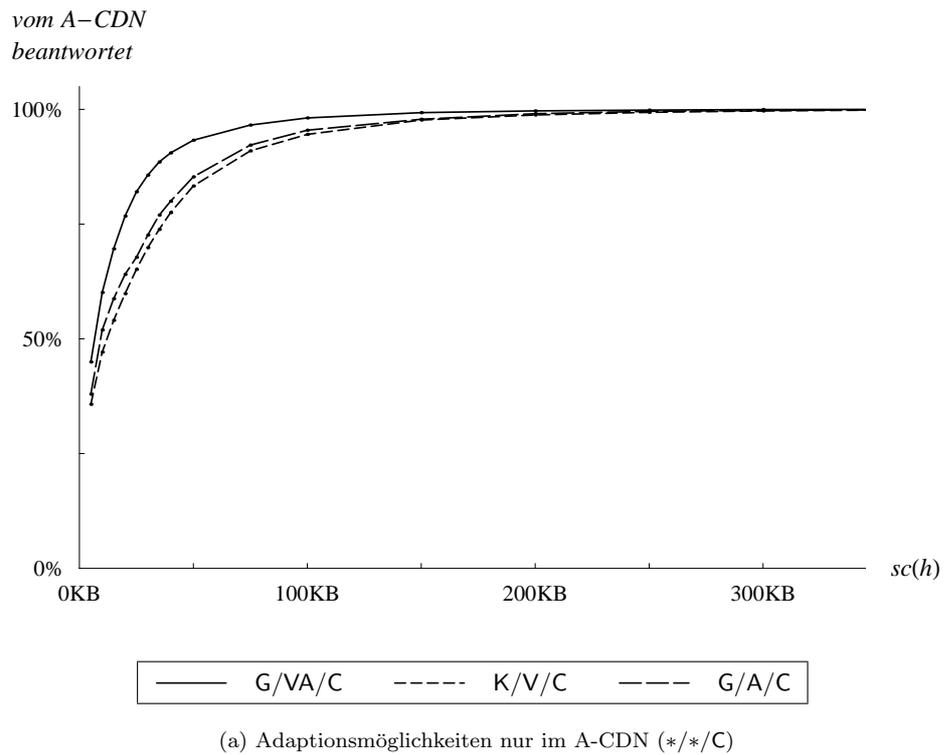


Abbildung 6.4: Anteil der Anfragen, die durch Replikate in den Surrogaten des A-CDN beantwortet werden

Der absolute Vorteil adaptivitätssensitiver Platzierung gegenüber den G/A/*-Ansätzen hängt kaum von der Verfügbarkeit von Adaptionmöglichkeiten in den Originalservern ab. Da jedoch infolge des höheren $Profit_{\Sigma,\Lambda}(\emptyset)$ der absolute Zuwachs $\Delta Profit_{\Sigma,\Lambda}$ an Gewinn pro Anfrage aller Ansätze bei Adaptionmöglichkeiten in den Originalservern geringer ausfällt, ist der relative Vorteil des G/VA/OC-Ansatzes gegenüber dem G/A/OC-Ansatz deutlich größer als der des G/VA/C-Ansatzes gegenüber dem G/A/C-Ansatz.

6.6 Untersuchung der Einflussfaktoren auf die Leistungsfähigkeit der Verfahren

Die Ergebnisse in Abschnitt 6.5 stellen nur Mittelwerte über eine Vielzahl von Experimenten mit unterschiedlichen Lastmodellen entsprechend Abschnitt 5.5 in unterschiedlichen Systemmodellen entsprechend Abschnitt 5.2 dar. Es zeigt sich jedoch, dass sich die Ergebnisse bei unterschiedlichen Lastmodellen teilweise signifikant unterscheiden. Deshalb soll in diesem Abschnitt der Einfluss unterschiedlicher Parameter des Lastmodells auf die Leistungsfähigkeit der unterschiedlichen Verfahren zur Platzierung von Replikaten in A-CDNs untersucht werden.

Weiterhin ist festzustellen, dass das Simulationsmodell, für welches die Ergebnisse in Abschnitt 6.5 ermittelt wurden, auf Annahmen über mögliche Anwendungsszenarien beruht und durch Randbedingungen der Simulation eingeschränkt ist (z.B. bzgl. der Systemgröße). Aus diesem Grund soll in diesem Abschnitt auch die Abhängigkeit der Ergebnisse von den getroffenen Annahmen und der Einfluss der Restriktionen auf die Leistungsfähigkeit der verschiedenen A-CDN-Ansätze evaluiert werden.

6.6.1 Überlappung der Anfragemlasten verschiedener Clients

Die Überlappung der Anfragemlasten verschiedener Clients wird im Simulationsmodell durch den Parameter s gesteuert. Der Parameter s gibt an, welcher Anteil der Anfragen nach Objekten sind, die von allen Clients unabhängig von ihren Geräteeigenschaften und Netzanbindungen angefordert werden. Objekte, die nicht von allen Clients angefordert werden, werden nur von Clients mit ähnlichen Geräteeigenschaften und Netzerkanbindungen angefordert (vgl. Abs. 5.5.2).

Abbildung 6.5 stellt die Ergebnisse aus Abbildung 6.3 auf Seite 130 für eine Konfiguration ohne Adaptionmöglichkeiten in den Originalservern (*/*/*C) in Abhängigkeit vom Parameter s für eine Speicherkapazität von 100 KB pro Surrogat dar. Neben den Mittelwerten der relativen Leistungsfähigkeit und den Grenzen des Konfidenzintervalls zum Konfidenzniveau von 95% ist in den Diagrammen die Regressionsgerade der linearen Regression zwischen dem Parameter s und der relativen Leistungsfähigkeit, der Korrelationskoeffizient r und der Anstieg a der Regressionsgeraden (Regressionskoeffizient) angegeben. In Abbildung 6.5 werden die Kurven der Mittelwerte fast vollständig durch die jeweilige Regressionsgerade überdeckt. Das zeigt, dass die lineare Regressionskurve eine gute Näherung des tatsächlichen Verlaufs darstellt.

Abbildung 6.5(a) zeigt, dass der Vorteil der adaptivitätssensitiven Platzierung von Replikaten gegenüber dem K/V/C-Ansatz stark von der Überlappung der Anfragemlasten der verschiedenen Klassen von Clients abhängt. Wenn die Clients in allen Klassen das selbe Anfrageverhalten zeigen ($s = 1,0$), so bringt adaptivitätssensitive Platzierung an der Stelle $sc(h) = 100$ KB einen relativen Vorteil von 11% mit sich. Werden Inhaltsobjekte jedoch grundsätzlich nur von Clients innerhalb einer Klasse angefordert ($s = 0,0$), ist bei gleicher Speicherkapazität der Vorteil der adaptivitätssensitiven Platzierungsmechanismen vernachlässigbar (ca. 2%).

Die Ergebnisse für die relative Leistungsfähigkeit eines A-CDN gemäß dem G/VA/OC-Ansatz bezüglich eines A-CDN mit einem herkömmlichen Platzierungsmechanismus entsprechend dem K/V/OC-Ansatz sind mit den in Abbildung 6.5(a) gezeigten Zusammenhängen vergleichbar. Da die relative Leistungsfähigkeit $A_{G/VA/OC|K/V/OC}$ etwas höher liegt als $A_{G/VA/C|K/V/C}$, wird für $s = 1,0$ an der Stelle $sc(h) = 100$ KB ein Vorteil von ca. 14% erreicht. An der Stelle $sc(h) = 50$ KB beträgt der Vorteil des adaptivitätssensitiven G/VA/OC-Ansatzes sogar mehr als 23%, wenn $s = 1,0$.

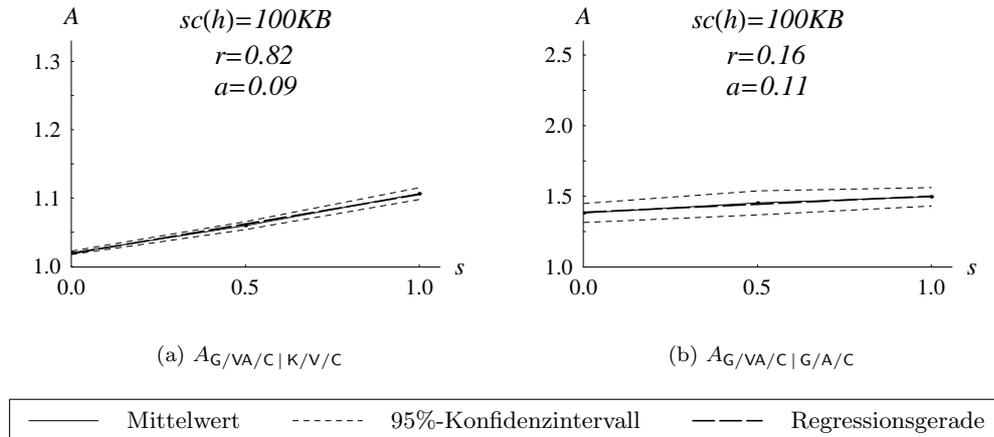


Abbildung 6.5: Korrelation zwischen der Überlappung der Anfragelasten der verschiedenen Clientklassen (Parameter s) und der relativen Leistungsfähigkeit

Dieses Verhalten ist dadurch zu erklären, dass bei kompletter Überlappung der Anfragelasten der verschiedenen Klassen von Clients alle Objekte potentiell in allen Repräsentationen benötigt werden. Wenn die Anfragelasten der verschiedenen Klassen von Clients hingegen keine Inhaltsobjekte teilen, so muss jedes Inhaltsobjekt nur in wenigen Repräsentationen (maximal 3, denn es sind maximal 3 Typen von Clients in einer Klasse zusammengefasst) ausgeliefert werden. In diesem Fall entsteht durch die Replikation statisch voradaptierter Repräsentationen kein Mehrbedarf an Speicherplatz bzw. der Mehrbedarf ist nicht signifikant.

In der Praxis ist zu erwarten, dass bei Diensten, die für mobile und stationäre Clients entwickelt werden, Clients mit unterschiedlichen Eigenschaften zum großen Teil auf die selben Inhaltsobjekte zugreifen, auch wenn ausgewählte Inhaltsobjekte vornehmlich von Clients einer bestimmten Klasse, z.B. nur von stationären Clients, genutzt werden. Somit ist in der Praxis von einer Überlappung zwischen $s = 0,5$ und $s = 1,0$ auszugehen. In diesem Fall beträgt der Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber herkömmlichen Platzierungsmechanismen der Klasse $K/V/*$ bei einer Speicherkapazität von $sc(h) = 100$ KB zwischen 6% und 14%.

Anders als die relative Leistungsfähigkeit der adaptivitätssensitiven Platzierungsmechanismen gegenüber den Ansätzen der Klasse $K/V/*$ ist die relative Leistungsfähigkeit gegenüber den Ansätzen der Klasse $G/A/*$ nicht signifikant mit der Überlappung der Anfragelasten verschiedener Typen von Clients korreliert (vgl. Abb. 6.5(b)).

6.6.2 Verteilung der Objektpopularitäten

Die Popularität der einzelnen Objekte wird im Simulationsmodell durch den Parameter z der Zipf-ähnlichen Verteilung der Objektpopularitäten gesteuert (vgl. Abs. 5.5.2). Der Parameter z beschreibt die Schiefe der Zipf-ähnlichen Verteilung, das heißt, wie stark die Anfragen auf die populärsten Inhaltsobjekte konzentriert sind. Ein Zipf-Parameter von $z = 1$ zum Beispiel bedeutet, dass das populärste Inhaltsobjekt im Lastmodell doppelt so oft angefordert wird wie das zweitpopulärste und dreimal so oft wie das drittpopulärste. Bei $z = 0,6$ ist die Popularität gleichmäßiger über die verschiedenen Inhaltsobjekte verteilt. In diesem Fall wird das populärste Inhaltsobjekte nur 1,5-mal so oft wie das zweitpopulärste und 1,9-mal so oft wie das drittpopulärste Inhaltsobjekt angefordert. Der Fall $z = 0$ entspräche einer Gleichverteilung.

Der Zusammenhang zwischen dem Zipf-Parameter z und der relativen Leistungsfähigkeit des adaptivitätssensitiven $G/VA/C$ -Ansatzes gegenüber den A-CDN-Ansätzen mit herkömmlichen Platzierungsmechanismen ist in Abbildung 6.6 dargestellt.

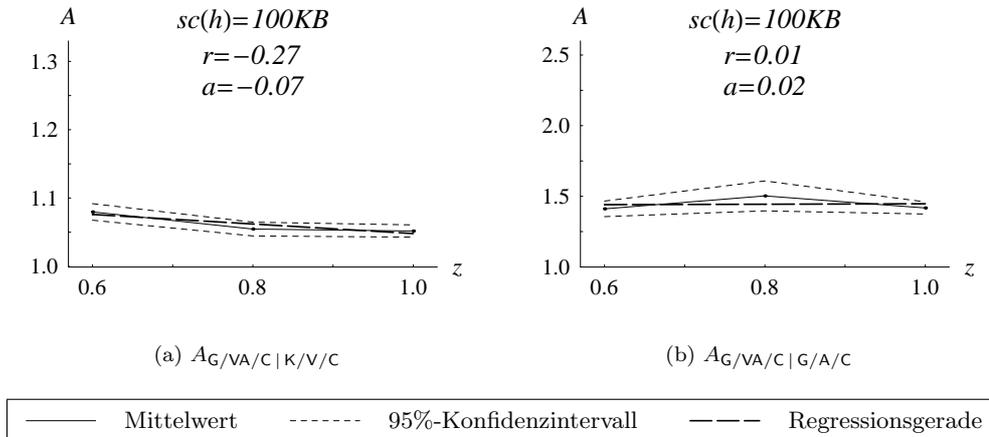


Abbildung 6.6: Korrelation zwischen der Verteilung der Objektpopularitäten (Zipf-Parameter z) und der relativen Leistungsfähigkeit

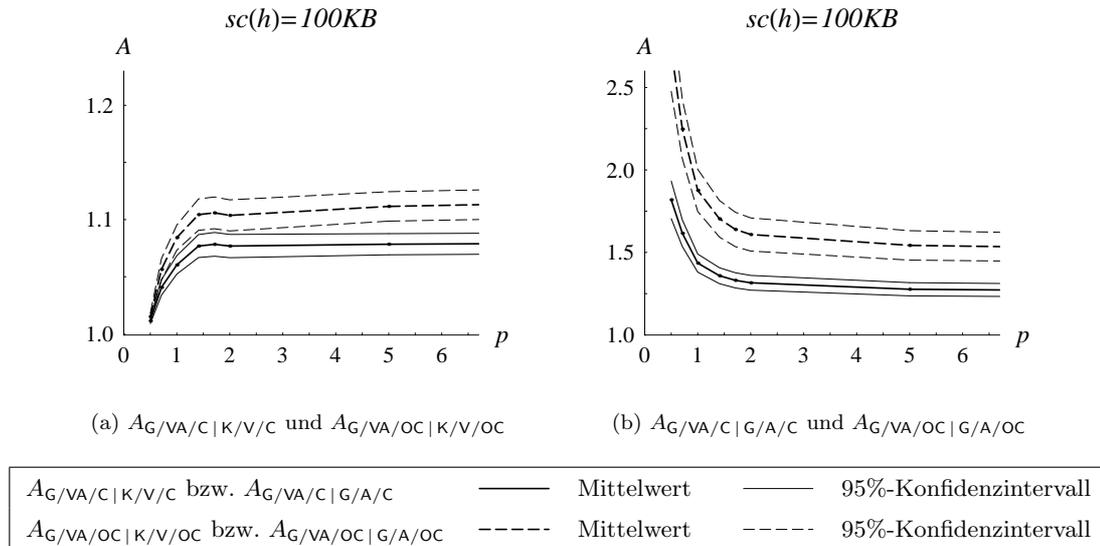
Abbildung 6.6(a) zeigt eine schwache negative Korrelation zwischen dem Zipf-Parameter und dem Vorteil der adaptivitätssensitiven Platzierungsmechanismen gegenüber dem K/V/C-Ansatz. Ein ähnliches Ergebnis mit einer etwas stärker ausgeprägten Korrelation ($r = -0,36$) lässt sich auch für den Vorteil des adaptivitätssensitiven G/VA/OC-Ansatzes gegenüber dem K/V/OC-Ansatz beobachten. Die negative Korrelation lässt sich dadurch erklären, dass bei einer eher ausgeglichenen Verteilung der Objektpopularitäten (kleine z) bei einer optimalen adaptivitätssensitiven Platzierung viele Inhaltsobjekte in einer generischen oder einer teilweise adaptierten Repräsentation repliziert werden, wobei nicht replizierte Repräsentationen bei Bedarf durch dynamische Adaption gewonnen werden. Sind hingegen die Anfragen sehr stark auf wenige populäre Objekte konzentriert (große z), sieht eine optimale adaptivitätssensitive Platzierung so aus, dass die populärsten Objekte in allen benötigten Repräsentationen repliziert werden und die weniger populären Objekte von den Originalservern ausgeliefert und dynamisch adaptiert werden. Während letzteres dem Verfahren bei den K/V/*-Ansätzen entspricht, können die K/V/*-Ansätze bei einer eher ausgeglichenen Verteilung der Objektpopularität nicht von der Replikation der generischen oder nur teilweise adaptierten Repräsentationen profitieren. Deshalb ist der Vorteil der adaptivitätssensitiven Platzierung bei kleineren z stärker ausgeprägt.

Der Vorteil der adaptivitätssensitiven Platzierung gegenüber herkömmlichen Verfahren der Klasse G/A/* ist nicht mit dem Zipf-Parameter z der Verteilung der Objektpopularitäten korreliert. Sowohl für $A_{G/VA/C|G/A/C}$ (vgl. Abb. 6.6(b)) als auch für $A_{G/VA/OC|G/A/OC}$ wurden Korrelationskoeffizienten $r \leq 0,05$ ermittelt.

6.6.3 Nominelle Preise

Die nominellen Preise bestimmen den Erlös, den ein A-CDN-Anbieter durch die Beantwortung der Anfragen der Clients erzielt (vgl. Formel 3.37 auf Seite 69). Darüber hinaus wichten sie Kosten und Qualität in der Zielfunktion (vgl. Fußnote 8 auf Seite 70). Im Simulationsmodell, welches den Ergebnissen in Abschnitt 6.5 zugrunde liegt, wird angenommen, dass die nominellen Preise derart gestaltet sind, dass sie genau kostendeckend sind, wenn alle Anfragen durch die Originalserver der Inhaltsanbieter mit maximal möglicher Qualität beantwortet werden. Dabei wird Inhaltsadaption, wenn nötig, in den Originalservern ausgeführt. Obwohl in der Praxis nominelle Preise in dieser Größenordnung zu erwarten sind, können die konkreten Werte natürlich davon abweichen.

Um den Einfluss von abweichenden nominellen Preisen auf die Simulationsergebnisse abschätzen zu können, wurde eine Serie von Experimente mit geringeren und höheren nominellen

Abbildung 6.7: Relative Leistungsfähigkeit in Abhängigkeit vom Preisniveau p

Preisen durchgeführt, deren Ergebnisse in diesem Abschnitt dargestellt werden. Hierzu wurden die entsprechend Formel 5.26 auf Seite 118 ermittelten nominellen Preise um einen konstanten Faktor p variiert. Der Faktor p wird im Folgenden als Preisniveau bezeichnet. Das Preisniveau p wurde in den Experimenten innerhalb $\{0,5, 0,71, 1,0, 1,41, 1,71, 2,0, 5,0, 10,0\}$ variiert. Der Zusammenhang zwischen dem Preisniveau und der relativen Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen bezüglich herkömmlicher Platzierungsmechanismen ist für $sc(h) = 100\text{KB}$ in Abbildung 6.7 dargestellt.

Abbildung 6.7(a) zeigt, dass die relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen bezüglich herkömmlicher Platzierungsmechanismen der Klasse K/V/* bei einem Preisniveau unter $p = 1,41$ sehr deutlich vom Preisniveau abhängt. So beträgt der mittlere Vorteil an der Stelle $p = 0,5$ bei einer Speicherkapazität von $sc(h) = 100\text{KB}$ lediglich 1% (Inhaltsadaption nur im A-CDN) bis 2% (auch Adaptionmöglichkeiten in den Originalservern) im Vergleich zu 8% bzw. 10% an der Stelle $p = 1,41$. Über $p = 1,41$ steigt die relative Leistungsfähigkeit des adaptivitätssensitiven Ansatzes aber nur noch mit flachem Anstieg, so dass selbst bei $p = 10,0$ der mittlere Vorteil nicht mehr als 8% bzw. 12% beträgt.

Die relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen gegenüber herkömmlichen Mechanismen der Klasse G/A/* ist in Abbildung 6.7(b) illustriert. Auch hier ist ein deutlicher Zusammenhang zwischen dem Preisniveau und der relativen Leistungsfähigkeit erkennbar. Der Vorteil der adaptivitätssensitiven Ansätze gegenüber den G/A/*-Ansätzen fällt monoton mit zunehmendem Preisniveau. Allerdings beträgt selbst bei einem Preisniveau von $p = 10,0$ der Vorteil adaptivitätssensitiver Platzierung zum Beispiel bei $sc(h) = 100\text{KB}$ im Mittel immer noch 26% bzw. 52%, je nachdem, ob nur die Surrogate oder auch die Originalserver über Adaptionmöglichkeiten verfügen (im Vergleich zu 44% bzw. 88% an der Stelle $p = 1,0$).

In der Praxis ist nicht zu erwarten, dass ein A-CDN-Anbieter mit seinen Kunden ein Preisniveau deutlich unter $p = 1,0$ aushandelt. Bereits bei einem Preisniveau unter $p = 0,8$ übersteigen die durchschnittlichen Betriebskosten $Cost_{\Sigma,\Lambda}(P_R)$ den durchschnittlichen Erlös $Rev_{\Sigma,\Lambda}(P_R)$. Bei einem Preisniveau über $p = 0,8$ beträgt der mittlere Vorteil adaptivitätssensitiver Platzierung mindestens 5%. Auch Preise deutlich über $p = 1,0$ sind in der Praxis nicht zu erwarten. In diesem Fall würde ein Inhaltsanbieter als Kunde des A-CDN-Anbieters deutlich mehr für die Auslieferung der Inhalte zahlen, als wenn er den Dienst selbst erbringen würde. Dabei ist auch bei einem Preisniveau von $p = 10,0$ die Qualität der ausgelieferten Inhalte nicht nennenswert höher als bei einem Preisniveau von $p = 2,0$ (um weniger als 0,01).

6.6.4 Qualitätsmodell

Den Experimenten in Abschnitt 6.5 liegt das Qualitätsmodell aus Abschnitt 5.3 zugrunde, bei dem die Clients tolerant gegenüber unterschiedlichen räumlichen Auflösungen der Bilder sind. Es gibt jedoch Anwendungsszenarien, in denen nicht davon ausgegangen werden kann, dass auch abweichende räumliche Auflösungen akzeptabel sind, zum Beispiel, wenn die Bilder Teil einer WWW-Seite sind, deren Elemente zueinander proportional sein müssen. Deshalb werden neben den Experimenten auf Basis der Qualitätsfunktion in Formel 5.4 auf Seite 108 auch Experimente mit der folgenden Qualitätsfunktion in der Qualitätsdimension der räumlichen Auflösung durchgeführt, bei der jeder Client nur genau eine Bildauflösung akzeptiert:

$$qf_{res}(c, k, (\mathbf{m}, \mathbf{s})) = \begin{cases} 1 & \text{falls } pix_{exh} \leq pix \\ 0 & \text{sonst} \end{cases} \quad (6.7)$$

Diese Qualitätsfunktion weist allen Repräsentationen, die eine geringere als die geforderte räumliche Auflösung aufweisen, einen Qualitätswert von 0 zu. Repräsentationen, die sich in der Qualitätsdimension der subjektiv wahrgenommenen Degeneration des Bildes oder der Antwortzeit unterscheiden, werden jedoch akzeptiert (mit unterschiedlichen Qualitätswerten).

Wenn die Toleranz der Clients gegenüber verschiedenen Repräsentationen gering ist, können A-CDNs, die generische Anfragen unterstützen, weniger von der Möglichkeit profitieren, eine Repräsentation für Clients mit unterschiedlichen Eigenschaften wiederzuverwenden. Folglich ist

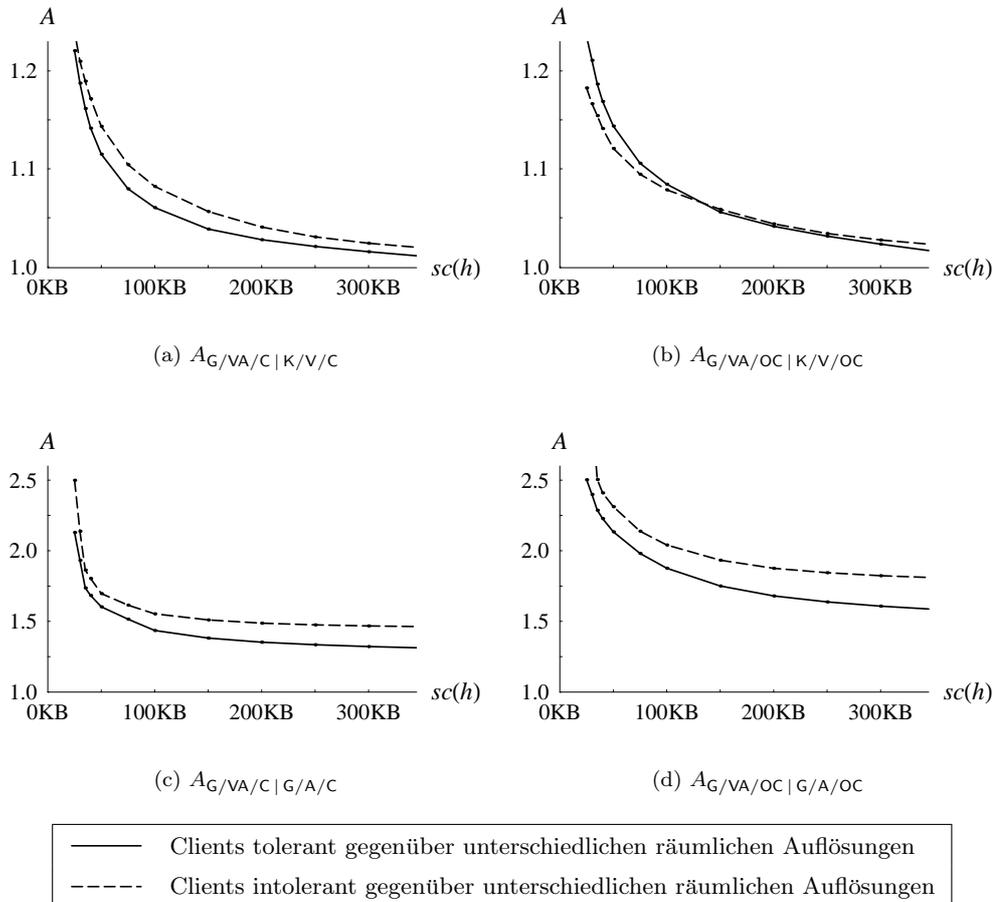


Abbildung 6.8: Abhängigkeit der relativen Leistungsfähigkeit vom Qualitätsmodell

Tabelle 6.4: Mittlere untere Grenze der Speicherkapazität $sc(h)$, bei der beim alternativen Qualitätsmodell gemäß Formel 6.7 $\Delta Profit_{\Sigma,\Lambda}$ maximal ist (inkl. 95%-Konfidenzintervall)

	//C	*/*/OC
G/VA/*	837 KB \pm 88 KB	831 KB \pm 88 KB
K/V/*	803 KB \pm 85 KB	802 KB \pm 85 KB
G/A/*	458 KB \pm 27 KB	455 KB \pm 27 KB

zu erwarten, dass A-CDNs der Klasse G/VA/* beim alternativen Qualitätsmodell einen geringeren Vorteil gegenüber A-CDNs der Klasse K/V/* mit sich bringen. Die Abbildungen 6.8(a) und 6.8(b) zeigen jedoch, dass das nur für A-CDNs entsprechend dem G/VA/OC-Ansatz gilt und auch nur dann, wenn die Speicherkapazität der Surrogate weniger als ca. 140 KB beträgt. Das ist damit zu erklären, dass Clients, die keine unterschiedlich aufgelösten Repräsentationen akzeptieren, größere und auch mehr verschiedene Repräsentationen anfordern als Clients, die zugunsten niedrigerer Kosten und Antwortzeiten geringer aufgelöste Repräsentationen tolerieren. Das zeigt sich auch darin, dass der absolute Zuwachs $\Delta Profit_{\Sigma,\Lambda}$ an Gewinn pro Anfrage erst bei deutlich höheren Speicherkapazitäten seinen Grenzwert erreicht (vgl. Tab. 6.3 auf Seite 129 und Tab. 6.4). Wenn man zum Vergleich der Qualitätsmodelle nicht die relative Leistungsfähigkeit bei gleichen absoluten Speicherkapazitäten heranzieht, sondern den Vergleich bei gleichen relativen Speicherkapazitäten durchführt (die relative Speicherkapazität ist das Verhältnis der absoluten Speicherkapazität $sc(h)$ und des maximalen Speicherplatzbedarfs pro Surrogat nach Tab. 6.3 bzw. Tab. 6.4), so ist die relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen beim alternativen Qualitätsmodell erwartungsgemäß geringer als bei toleranteren Clients.

Der Vorteil adaptivitätssensitiver Platzierung von Replikaten in A-CDNs der Klasse G/VA/* gegenüber A-CDNs der Klasse G/A/* für die verschiedenen Qualitätsmodelle ist in Abbildung 6.8(c) und 6.8(d) dargestellt. Wenn die Clients eine größere Toleranz gegenüber unterschiedlichen Repräsentationen zeigen, so sind A-CDNs der Klasse G/A/* in der Lage, die erhöhten Kosten und Verzögerungen infolge der dynamischen Inhaltsadaption bei jeder Anfrage etwas zu kompensieren, indem Repräsentationen mit einer geringeren Auflösung ausgeliefert werden, die zu niedrigeren Kosten und mit geringeren Verzögerungen generiert werden können. Deshalb ist der Vorteil adaptivitätssensitiver Platzierungsmechanismen bei intoleranten Clients höher, als wenn die Clients unterschiedliche räumliche Auflösungen akzeptieren.

6.6.5 Kosten und Verzögerungen infolge der Inhaltsadaption

Bei dynamischer Inhaltsadaption führen die Ausführungszeiten der Adaptionsoptionen zu Verzögerungen bei der Beantwortung von Anfragen der Clients. Fernerhin sind die Kosten der Adaptionsoptionen über den Faktor c_{cpu} (Basiskosten pro Zeiteinheit CPU-Nutzung) an die Ausführungszeiten gebunden (vgl. Formel 5.15 auf Seite 112). Die Ausführungszeiten der Adaptionsoptionen im Simulationsmodell entsprechen empirisch ermittelten Ausführungszeiten von Implementierungen der Operationen auf Basis des Java Advanced Imaging Toolkit JAI 1.1.2 (vgl. Anhang C.2). Durch effektive Implementierungen, zum Beispiel durch die Nutzung von Möglichkeiten zur Adaption kodierter Medienobjekte (vgl. Abs. 2.3), leistungsfähige Hardware und die Nutzung von Spezialhardware zur Adaption können die Adaptionzeiten in der Praxis gegenüber den empirisch ermittelten Werten deutlich reduziert werden. Da jedoch in der Praxis parallel ausgeführte Adaptionsoptionen um Ressourcen (z.B. CPU) konkurrieren, sind bei einer Implementierung und Hardwareplattform, die mit den in der Transcoding-Analyse in Anhang C.2 verwendeten vergleichbar sind, effektive Adaptionzeiten zu erwarten, die höher als die bei der Transcoding-Analyse ermittelten sind.

Tabelle 6.5: Korrelation zwischen den Ausführungszeiten der Adaptionsoptionen (Faktor tf) und der relativen Leistungsfähigkeit

	Korrelationskoeffizient r Anstieg der Regressionsgeraden a		
	$sc(h) = 50$ KB	$sc(h) = 100$ KB	$sc(h) = 300$ KB
$A_{G/VA/C K/V/C}$	$r = -0,04$ $a = -0,01$	$r = -0,12$ $a = -0,01$	$r = -0,11$ $a = -0,01$
$A_{G/VA/OC K/V/OC}$	$r = -0,09$ $a = -0,02$	$r = -0,20$ $a = -0,03$	$r = -0,15$ $a = -0,01$
$A_{G/VA/C G/A/C}$	$r = 0,61$ $a = 0,43$	$r = 0,65$ $a = 0,34$	$r = 0,65$ $a = 0,30$
$A_{G/VA/OC G/A/OC}$	$r = 0,57$ $a = 0,95$	$r = 0,62$ $a = 0,73$	$r = 0,64$ $a = 0,56$

Um die Sensitivität der Simulationsergebnisse bezüglich der Adaptionszeiten abschätzen zu können, wurde in einer Serie von Experimenten mit unterschiedlichen Adaptionszeiten die Korrelation zwischen den Adaptionszeiten und der relativen Leistungsfähigkeit der Ansätze mit adaptivitätssensitiver Platzierung bezüglich der Ansätze mit herkömmlichen Platzierungsmechanismen bestimmt. Dazu werden die Adaptionszeiten gemäß dem empirisch ermittelten Modell in Tabelle 5.6 auf Seite 112 um einen konstanten Faktor $tf \in \{0, 0,1, 0,5\}$ variiert. Die Ergebnisse der Analyse sind in Tabelle 6.5 in Form des Korrelationskoeffizienten und des Anstiegs der Regressionsgeraden dargestellt.

Tabelle 6.5 zeigt, dass der Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber den Ansätzen der Klasse $K/V/*$ mit sinkenden Adaptionszeiten geringfügig ansteigt. Die Ursache dafür ist darin zu sehen, dass bei geringeren Adaptionszeiten die Möglichkeit der dynamischen Inhaltsadaption extensiver ausgenutzt werden kann. Die Korrelation ist aber sehr schwach und der Anstieg der Regressionsgeraden flach, so dass auch, wenn die Adaptionszeiten gegen 0 gehen ($tf \rightarrow 0$), der mittlere Vorteil des adaptivitätssensitiven Ansatzes nur bis zu 3 Prozentpunkten über dem bei $tf = 1$ liegt (vgl.

Adaptivitätssensitive Platzierungsmechanismen realisieren einen Vorteil gegenüber den $G/A/*$ -Ansätzen, indem sie durch die Wiederverwendung von adaptierten Repräsentationen Adaptionszeiten und -kosten bei der Auslieferung einsparen. Folglich ist bei sinkenden Adaptionszeiten und -kosten eine Abschwächung des Vorteils zu beobachten. Das wird durch die positive Korrelation zwischen dem Faktor tf und der relativen Leistungsfähigkeit $A_{G/VA/C|G/A/C}$ bzw. $A_{G/VA/OC|G/A/OC}$ mit steilem Anstieg der Regressionsgeraden belegt (vgl. Tab. 6.5). Wenn die Adaptionszeiten gegen 0 gehen, ist bei hinreichender Speicherkapazität der Surrogate ($sc(h) \geq 462$ KB) überhaupt kein Vorteil der adaptivitätssensitiven Platzierung zu beobachten.

Tabelle 6.6: Relative Leistungsfähigkeit für $tf \rightarrow 0$ (inkl. 95%-Konfidenzintervall)

	$sc(h) = 50$ KB	$sc(h) = 100$ KB	$sc(h) = 300$ KB
$A_{G/VA/C K/V/C}$	$1,12 \pm 0,01$	$1,07 \pm 0,01$	$1,02 \pm 0,00$
$A_{G/VA/OC K/V/OC}$	$1,16 \pm 0,02$	$1,11 \pm 0,01$	$1,04 \pm 0,01$
$A_{G/VA/C G/A/C}$	$1,18 \pm 0,01$	$1,09 \pm 0,00$	$1,03 \pm 0,00$
$A_{G/VA/OC G/A/OC}$	$1,20 \pm 0,02$	$1,15 \pm 0,01$	$1,05 \pm 0,01$

Bei geringeren Speicherkapazitäten strebt die relative Leistungsfähigkeit bei $tf \rightarrow 0$ aber nicht gegen 1 (Tab. 6.6), da der Speicherplatzbedarf bei der Replikation teilweise adaptierter Repräsentationen geringer als bei der Replikation der generischen Repräsentationen ist und somit bei adaptivitätssensitiver Platzierung mehr Inhaltsobjekte repliziert werden können.

6.6.6 Übertragungskosten und -verzögerungen im Internet

Übertragungskosten

Im Simulationsmodell wird von Übertragungskosten von 1 EUR/GB zwischen autonomen Systemen im Internet ausgegangen. Das entspricht in etwa der Größenordnung der Übertragungskosten, die in heutiger Zeit in Peering-Abkommen zwischen ISPs ausgehandelt werden. Endkundenpreise liegen heute mit 3 bis 6 EUR/GB [oV04d] etwas darüber. In der Zukunft ist jedoch bei steigenden Übertragungskapazitäten mit sinkenden Netzwerkkosten zu rechnen.

Um den Einfluss der Übertragungskosten auf die Simulationsergebnisse abzuschätzen, wurden Experimente mit Übertragungskosten von 0,1 EUR/GB und 10 EUR/GB durchgeführt und den Ergebnissen für 1 EUR/GB aus Abschnitt 6.5 gegenübergestellt. Diese Untersuchung zeigt, dass die Wahl der Übertragungskosten zwischen autonomen Systemen keinen nennenswerten Einfluss auf die relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen bezüglich herkömmlicher Platzierungsmechanismen hat. Die Korrelation zwischen Übertragungskosten und relativer Leistungsfähigkeit hat einen Betrag von weniger als 0,07. Die Differenz der Mittelwerte der relativen Leistungsfähigkeit bei 0,1 EUR/GB und 10 EUR/GB beträgt weniger als die halbe Breite der Konfidenzintervalle der Mittelwerte.

Der vernachlässigbare Einfluss der Übertragungskosten im Internet ist damit zu erklären, dass diese Kosten gering und folglich gegenüber den Übertragungskosten für die Verbindung zwischen den Clients und ihren Zugangspunkten nicht signifikant sind.

Übertragungsverzögerungen

Verzögerungen bei der Übertragung von Inhaltsobjekten im Internet werden im Simulationsmodell durch die Latenz und die Datenrate der Netzwerkverbindungen zwischen den autonomen Systemen bestimmt. Den Simulationsergebnissen in Abschnitt 6.5 liegt die Annahme einer Latenz von 50 ms und einer Datenrate von 400 kbit/s zugrunde. Diese Werte beruhen auf Messungen im Internet. Sie sind jedoch nur als grobe Abschätzung zu verstehen.

Um den Einfluss der Übertragungsverzögerungen im Internet auf die Ergebnisse abschätzen zu können, wird die Sensitivität gegenüber der Latenz und der Datenrate in Experimenten, bei denen die Latenz zwischen 30 ms und 70 ms und die effektive Datenrate zwischen 400 kbit/s und 2 Mbit/s variiert wird, analysiert. Die Ergebnisse sind in Tabelle 6.7 in Form der relativen Leistungsfähigkeit der Ansätze mit adaptivitätssensitiven Platzierungsmechanismen bezüglich der Ansätze mit herkömmlichen Platzierungsmechanismen bei $sc(h) = 100$ KB zusammengefasst.

Tabelle 6.7: Relative Leistungsfähigkeit in Abhängigkeit von den Übertragungsverzögerungen im Internet (für $sc(h) = 100$ KB, inkl. 95%-Konfidenzintervall)

Latenz	30 ms	70 ms	50 ms	
effektive Datenrate	400 kbit/s		400 kbit/s	2 Mbit/s
$A_{G/VA/C K/V/C}$	$1,03 \pm 0,01$	$1,09 \pm 0,01$	$1,06 \pm 0,01$	$1,07 \pm 0,01$
$A_{G/VA/OC K/V/OC}$	$1,05 \pm 0,01$	$1,11 \pm 0,01$	$1,08 \pm 0,01$	$1,08 \pm 0,01$
$A_{G/VA/C G/A/C}$	$1,49 \pm 0,07$	$1,40 \pm 0,05$	$1,44 \pm 0,05$	$1,74 \pm 0,10$
$A_{G/VA/OC G/A/OC}$	$2,40 \pm 0,22$	$1,65 \pm 0,09$	$1,88 \pm 0,13$	$1,89 \pm 0,13$

Die Untersuchungen zeigen, dass die Latenz im betrachteten Intervall positiv mit der relativen Leistungsfähigkeit bezüglich der K/V/*-Ansätze korreliert ist, während die Korrelation mit der relativen Leistungsfähigkeit bezüglich der G/A/*-Ansätze negativ ist. Sowohl der sinkende Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber den K/V/*-Ansätzen als auch der steigende Vorteil gegenüber den G/A/*-Ansätzen bei sinkender Latenz ist mit den geringeren Verzögerungen beim Zugriff auf Originalserver und entfernte Surrogate erklärbar. Da bei den adaptivitätssensitiven Ansätzen Replikate öfter wiederverwendet werden können als bei den K/V/*-Ansätzen, können mehr Anfragen durch das nächstgelegene Surrogat beantwortet werden und es sind weniger Zugriffe auf entfernte Surrogate oder Originalserver erforderlich. Bei den G/A/*-Ansätzen werden ausschließlich generische Repräsentationen repliziert, so dass jedes Replikat eines Inhaltsobjektes zur Beantwortung aller Anfragen nach dem Objekt verwendet werden kann. Aufgrund der Adaptionkosten und adaptionbedingten Verzögerungen ist dieses Verhalten zwar nicht optimal (daher der deutliche Nachteil gegenüber adaptivitätssensitiven Verfahren), der Nachteil gegenüber der Replikation adaptierter Repräsentationen durch adaptivitätssensitive Platzierungsmechanismen nimmt aber mit steigenden Verzögerungen ab.

Im Gegensatz zum Einfluss der Latenz auf die Ergebnisse ist die Auswirkung steigender effektiver Datenraten – mit Ausnahme der Auswirkung auf $A_{G/VA/C|G/A/C}$ – vernachlässigbar. Das lässt sich damit erklären, dass bei der Übertragung adaptierter Repräsentationen die Verzögerungen infolge begrenzter Datenrate gegenüber den Latenzen vernachlässigbar klein sind. $A_{G/VA/C|G/A/C}$ hingegen wird signifikant durch die effektive Datenrate beeinflusst. Je höher die Datenrate, desto größer ist der Vorteil der adaptivitätssensitiven Platzierung. Das ist darauf zurückzuführen, dass infolge mangelnder Adaptionmöglichkeiten in den Originalservern für jede Anfrage, die nicht durch ein Surrogat beantwortet werden kann, die generische Repräsentation vom Originalserver an das A-CDN übertragen werden muss, um sie dort zu adaptieren. Da die generischen Repräsentationen im Mittel deutlich größer als die adaptierten sind, wirkt sich bei deren Übertragung neben der Latenz auch die Datenrate der Netzwerkverbindung deutlich auf die Antwortzeit aus.

6.6.7 Räumliche Auflösung der primären Repräsentationen

Die Verteilung der räumlichen Auflösung der primären Repräsentationen im Simulationsmodell basiert auf einer empirischen Untersuchung der Eigenschaften der Bilder im heutigen Internet. Allerdings ist anzunehmen, dass Bilder im heutigen Internet häufig bereits voradaptiert (d.h. in diesem Fall verkleinert) werden, bevor sie auf den Originalservern hinterlegt werden, um auf diese Weise die Antwortzeiten beim Zugriff zu reduzieren. Eine solche Voradaptation ist im Einsatzszenario eines A-CDN natürlich nicht notwendig und auch nicht sinnvoll. Hier sollen Inhalte durch den Inhaltsanbieter ja möglichst verlustfrei bereitgestellt werden. Die Inhaltsadaptation erfolgt erst bei der Verteilung und Auslieferung durch das A-CDN. Allerdings lässt sich aus empirischen Untersuchungen des WWW-Verkehrs im heutigen Internet nicht ermitteln, welche Bilder von einer Voradaptation betroffen sind und wie deren Originalauflösung ausgesehen hätte, wenn auf Voradaptation verzichtet worden wäre.

Um den Einfluss der Verteilung der räumlichen Auflösung der primären Repräsentationen auf die Ergebnisse abzuschätzen, wurde eine Serie von Simulationsexperimenten mit deutlich größeren primären Repräsentationen durchgeführt. Dabei wurde die Zahl der Bildpunkte nicht wie bei den Experimenten in Abschnitt 6.5 zufällig aus der hybriden Verteilung gemäß Anhang C.1 gezogen, sondern nur aus dem Pareto-verteilten Schwanz der empirisch ermittelten Verteilung. Somit umfassen die primären Repräsentationen mindestens 84 010 Bildpunkte. Das entspricht zum Beispiel einer Auflösung von ca. 335×251 Bildpunkten. Die mittlere Größe eines Originalreplikates beträgt damit 26 KB.

Die Ergebnisse dieser Experimente sind für eine Konfiguration ohne Adaptionmöglichkeiten in den Originalservern (*/*/C) in Abbildung 6.9 den Ergebnissen mit der hybriden Verteilung der Zahl der Bildpunkte nach Anhang C.1 gegenübergestellt. Wie in den Diagrammen zu sehen ist, führt die größere Auflösung der primären Repräsentationen zu einem deutlich größeren Vor-

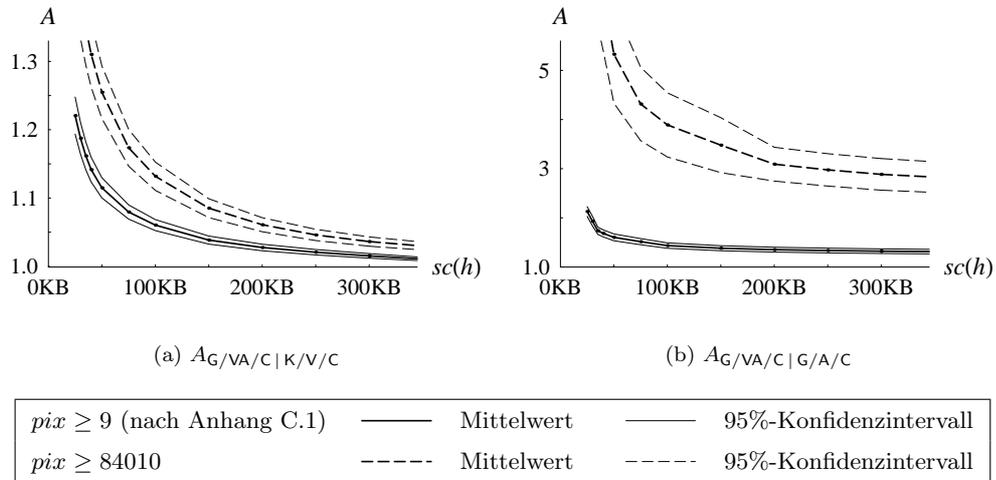


Abbildung 6.9: Relative Leistungsfähigkeit bei unterschiedlichen räumlichen Auflösungen der primären Repräsentationen

teil des adaptivitätssensitiven Ansatzes gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen. Zum Beispiel bei einer Speicherkapazität von $sc(h) = 100$ KB beträgt der mittlere relative Vorteil gegenüber dem K/V/C-Ansatz 13% im Gegensatz zu 6%, wenn die Verteilung der räumlichen Auflösung der empirisch ermittelten entspricht (Abb. 6.9(a)). Dementsprechend sinkt die relative Leistungsfähigkeit auch erst bei einer Speicherkapazität von 1,5 GB auf 1. Ähnliche Ergebnisse können auch für die relative Leistungsfähigkeit des adaptivitätssensitiven G/VA/OC-Ansatzes bezüglich des K/V/OC-Ansatzes beobachtet werden (nicht im Diagramm dargestellt).

Der deutlichere Vorteil der adaptivitätssensitiven Platzierung gegenüber den herkömmlichen Platzierungsmechanismen in Ansätzen der Klasse K/V/* bei größeren primären Repräsentationen ist auf zwei Faktoren zurückzuführen:

1. Eine höhere durchschnittliche räumliche Auflösung der primären Repräsentationen bedeutet, dass im Mittel mehr verschiedene Repräsentationen pro Inhaltsobjekt benötigt werden. Das liegt darin begründet, dass entsprechend dem Qualitätsmodell (Abs. 5.3.1) kein Client eine Repräsentation benötigt, die höher als die primäre Repräsentation aufgelöst ist. Infolge dessen wächst der Speicherplatzbedarf für die Replikation aller benötigten Repräsentationen in statisch voradaptierter Form mit zunehmender Größe der primären Repräsentationen. So wird zur vollständigen Replikation aller benötigten Repräsentationen bei den Experimenten mit den größeren primären Repräsentationen mehr als dreimal so viel Speicherplatz benötigt.
2. Darüber hinaus nehmen mit der Größe der primären Repräsentationen auch die durch die Adaption der Originalreplikate bedingten Kosten und Verzögerungen zu. Hinzu kommen erhöhte Kosten und Verzögerungen für die Übertragung von Originalreplikaten an Surrogate, falls die Originalserver nicht über Adaptionmöglichkeiten verfügen und deshalb die Inhaltsadaption im A-CDN erfolgen muss (*/*/*). Diese Probleme betreffen sowohl A-CDNs mit adaptivitätssensitiven als auch solche mit herkömmlichen Platzierungsmechanismen. Adaptivitätssensitive Verfahren sind jedoch in der Lage, diese Effekte durch dynamische Inhaltsadaption und durch die Ausnutzung der Toleranz der Clients gegenüber unterschiedlichen Repräsentationen teilweise zu kompensieren.

Auch der Vorteil der adaptivitätssensitiven Platzierung gegenüber dem G/A/C-Ansatz nimmt mit zunehmender Größe der primären Repräsentationen zu (Abb. 6.9(b)). Die relative Leistungsfähigkeit $A_{G/VA/C|G/A/C}$ beträgt zum Beispiel bei einer Speicherkapazität von $sc(h) = 100$ KB durchschnittlich ca. 3,89 im Vergleich zu 1,44, wenn die räumliche Auflösung

der primären Repräsentationen der empirisch ermittelten Verteilung entspricht. In ähnlicher Weise liegt auch die relative Leistungsfähigkeit $A_{G/VA/OC|G/A/OC}$ für eine Konfiguration, in der auch die Originalserver über Adaptionmöglichkeiten verfügen, bei größeren primären Repräsentationen deutlich höher als bei der Größenverteilung nach Anhang C.1 (nicht im Diagramm dargestellt).

Zurückzuführen ist die höhere relative Leistungsfähigkeit der adaptivitätssensitiven Ansätze bezüglich der G/A/*-Ansätze auf die höheren Kosten und Verzögerungen durch die Adaption der Originalreplika und den höheren Speicherplatzbedarf für die Replikation der primären Repräsentationen bei den Ansätzen der Klasse G/A/*.

6.6.8 Anfrageverteilung

Die Verteilung der Anfragen auf die verschiedenen Clients im Simulationsmodell (vgl. Tab. 5.8 auf Seite 116) basiert auf einer groben Abschätzung eines möglichen Anfrageverhaltens in einem praktischen Einsatzszenario. Empirische Daten dazu sind leider nicht verfügbar. Um die Abhängigkeit der Ergebnisse von der Anfrageverteilung zu untersuchen, wurden Experimente mit den alternativen Verteilungen \mathcal{V}_1 , \mathcal{V}_2 , \mathcal{V}_3 und \mathcal{V}_4 in Anhang E (die Ergebnisse für \mathcal{V}_5 werden in Abschnitt 6.6.9 diskutiert) durchgeführt. Die Ergebnisse werden in Tabelle 6.8 denen für die Anfrageverteilung in Tabelle 5.8 auf Seite 116, welche den Experimenten in den vorangegangenen Abschnitten zugrunde lag, gegenübergestellt. Das Modell der Anfrageverteilung in Tabelle 5.8 wird in Tabelle 6.8 als \mathcal{V}_0 bezeichnet.

Tabelle 6.8 zeigt, dass die relative Leistungsfähigkeit der adaptivitätssensitiven G/VA/*-Ansätze bezüglich der Ansätze mit herkömmlichen Platzierungsmechanismen bei unterschiedlichen Anfrageverteilungen variiert. Der mittlere Vorteil der G/VA/*-Ansätze gegenüber den K/V/*-Ansätzen liegt bei den betrachteten alternativen Anfrageverteilungen \mathcal{V}_1 , \mathcal{V}_2 , \mathcal{V}_3 und \mathcal{V}_4 bei Speicherkapazitäten zwischen 50 KB und 300 KB bis zu 2 Prozentpunkten unter dem bei \mathcal{V}_0 , aber auch bis zu 2 Prozentpunkten darüber. Während eine Zunahme der Anfragen von stationären Clients im Festnetz (\mathcal{V}_1) den Vorteil adaptivitätssensitiver Platzierung gegenüber den K/V/*-Ansätzen drückt, fällt der Vorteil bei einem höheren Anteil von mobilen Clients in WLAN-Hotspots (\mathcal{V}_2) größer aus. Bei der Anfrageverteilung \mathcal{V}_3 , bei der der Anteil der Anfragen von Clients in zellularen Mobilfunknetzen auf 60% erhöht ist, ist der Vorteil adaptivitätssensitiver Platzierungsmechanismen bei Speicherkapazitäten von 50 KB und 100 KB geringer als bei \mathcal{V}_0 . Bei Speicherkapazitäten von 300 KB ist für ein Szenario, in dem Inhaltsadaption ausschließlich im A-CDN erfolgt, kaum ein Unterschied festzustellen. Wenn auch die Originalserver über Adaptionmöglichkeiten verfügen, ist der Vorteil an der Stelle $sc(h) = 300$ KB sogar geringfügig höher als bei \mathcal{V}_0 .

Die unterschiedliche relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen bei verschiedenen Verteilungen der Anfragen auf die verschiedenen Clienttypen lässt sich aus der unterschiedlich starken Konzentration der Anfragen auf wenige Typen von Clients erklären. Je stärker die Anfragen auf bestimmte Clienttypen konzentriert sind, desto weniger Repräsentationen werden benötigt, um einen Großteil der Anfragen mit einer optimalen Repräsentation zu beantworten. Folglich kann sich ein A-CDN der Klasse K/V/* bei einer starken Konzentration der Anfragen auf wenige Clienttypen darauf beschränken, nur geeignete Repräsentationen für die Clients in statisch voradaptierter Form zu replizieren, die sehr viele Anfragen initiieren. Das erklärt den geringeren Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber den K/V/*-Ansätzen bei \mathcal{V}_1 , wo 60% aller Anfragen von Arbeitsplatzcomputern im Festnetz initiiert werden. Sind die Anfragen hingegen über alle Typen von Clients ungefähr gleichverteilt, so ist der Gewinn aus der Replikation ausgewählter Repräsentationen geringer. Folglich können adaptivitätssensitive Platzierungsmechanismen in diesem Fall mehr von der Unterstützung generischer Anfragen und der Möglichkeit der dynamischen Inhaltsadaption profitieren. Diese Argumentation wird durch die Ergebnisse für \mathcal{V}_4 gestützt. Bei \mathcal{V}_4 sind die Anfragen über alle Typen von Endgeräten gleichverteilt. Infolge dessen ist die relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen bezüglich der K/V/*-Ansätze bei \mathcal{V}_4 größer als bei \mathcal{V}_0 .

Tabelle 6.8: Relative Leistungsfähigkeit bei alternativen Verteilungen der Anfragen auf die Clients ($\mathcal{V}_1 - \mathcal{V}_4$) und bei reduzierter Heterogenität der Clientpopulation (\mathcal{V}_5)

relative Leistungsfähigkeit	sc(h)	\mathcal{V}_0	Differenz zu \mathcal{V}_0				
			\mathcal{V}_1	\mathcal{V}_2	\mathcal{V}_3	\mathcal{V}_4	\mathcal{V}_5
$A_{G/VA/C K/V/C}$	50 KB	1, 11	(-0, 011)	(+0, 023)	(-0, 017)	(+0, 004)	-0, 055
	100 KB	1, 06	(-0, 006)	(+0, 007)	(-0, 005)	(+0, 003)	-0, 036
	300 KB	1, 02	(-0, 002)	(+0, 005)	(-0, 000)	(+0, 002)	-0, 011
$A_{G/VA/OC K/V/OC}$	50 KB	1, 14	(-0, 017)	(+0, 019)	(-0, 015)	(+0, 006)	-0, 067
	100 KB	1, 08	(-0, 012)	(+0, 005)	(-0, 003)	(+0, 005)	-0, 049
	300 KB	1, 02	(-0, 003)	(+0, 000)	(+0, 001)	(+0, 003)	-0, 015
$A_{G/VA/C G/A/C}$	50 KB	1, 60	(+0, 120)	(+0, 086)	-0, 115	(-0, 087)	(+0, 009)
	100 KB	1, 44	(+0, 089)	(+0, 067)	-0, 090	(-0, 068)	(+0, 008)
	300 KB	1, 32	(+0, 068)	(+0, 056)	(-0, 073)	(-0, 045)	(+0, 003)
$A_{G/VA/OC G/A/OC}$	50 KB	2, 13	(+0, 249)	(+0, 131)	(-0, 237)	(-0, 202)	(+0, 030)
	100 KB	1, 88	(+0, 186)	(+0, 094)	(-0, 174)	(-0, 149)	(+0, 021)
	300 KB	1, 61	(+0, 122)	(+0, 083)	(-0, 126)	(-0, 086)	(+0, 006)

(Werte in Klammern bedeuten, dass sich die 95%-Konfidenzintervalle der Mittelwerte überschneiden)

Über den beschriebenen Effekt hinaus spielt aber auch die Größe der benötigten Repräsentationen eine wichtige Rolle. Während zum Beispiel die Anfragen bei \mathcal{V}_0 stärker konzentriert sind als bei \mathcal{V}_3 , ist der Anteil von Anfragen nach großen Repräsentationen, die für Arbeitsplatzcomputer im Festnetz benötigt werden, bei \mathcal{V}_3 geringer als bei \mathcal{V}_0 . Deshalb ist der Vorteil der adaptivitätssensitiven Platzierung bei \mathcal{V}_3 trotz gleichmäßigerer Verteilung der Anfragen bei knapper Speicherkapazität geringer als der bei \mathcal{V}_0 .

Der Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber den G/A/*-Ansätzen ist bei Gleichverteilung der Anfragen über die verschiedenen Endgerätetypen (\mathcal{V}_4) geringer als bei \mathcal{V}_0 und steigt mit zunehmender Konzentration der Anfragen auf ausgewählte Clienttypen (vgl. Ergebnisse für \mathcal{V}_1). Das ist damit zu erklären, dass die adaptivitätssensitiven G/VA/*-Ansätze bei Konzentration der Anfragen auf wenige Clienttypen mehr von der Möglichkeit der Replikation statisch voradaptierter Repräsentation profitieren können.

Der Einfluss der verschiedenen Anfrageverteilungen \mathcal{V}_0 , \mathcal{V}_1 , \mathcal{V}_2 , \mathcal{V}_3 und \mathcal{V}_4 auf die relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen bezüglich herkömmlicher Platzierungsmechanismen ist in vielen Fällen nicht signifikant. Oft liegen die Mittelwerte der relativen Leistungsfähigkeit so eng beieinander, dass sich die 95%-Konfidenzintervalle für die verschiedenen Anfrageverteilungen überschneiden (diese Fälle sind in Tab. 6.8 durch geklammerte Differenzen gekennzeichnet).

6.6.9 Heterogenität der Clientpopulation

A-CDNs werden benötigt, um Inhalte an Clients mit heterogenen Anforderungen an die Repräsentation der Inhalte auszuliefern. In der vorliegenden Arbeit werden adaptivitätssensitive Platzierungsmechanismen vorgeschlagen, um die Adaptierbarkeit der Inhalte an die heterogenen Anforderungen der Clients bei der Platzierung der Replikate in einem A-CDN zu berücksichtigen. Folglich sollten adaptivitätssensitive Platzierungsmechanismen bei einer größeren Heterogenität der Eigenschaften der Clients deutlichere Vorteile gegenüber herkömmlichen Platzierungsmechanismen mit sich bringen als bei einer homogeneren Clientpopulation.

Um diese These zu belegen, wurden neben den Experimenten mit den 10 verschiedenen Typen von Endgeräten entsprechend Tabelle 5.3 auf Seite 104 (Modell \mathcal{V}_0) Experimente mit 4 verschiedenen Typen von Endgeräten mit Displayauflösungen von 128×160 , 240×320 und 1024×768 Bildpunkten (Modell \mathcal{V}_5 in Tab. E.5 auf Seite 190) durchgeführt. Die Ergebnisse der Analysen sind in Form der Differenz zwischen der relativen Leistungsfähigkeit bei 4 und bei 10 Typen von Endgeräten in Tabelle 6.8 (letzte Spalte) dargestellt.

Die Ergebnisse in Tabelle 6.8 bestätigen die These größerer Vorteile der adaptivitätssensitiven Platzierung gegenüber den Ansätzen der Klasse $K/V/*$ bei einer größeren Heterogenität der Clientpopulation. So beträgt zum Beispiel an der Stelle $sc(h) = 100$ KB der relative Vorteil bei nur 4 Typen von Endgeräten ca. 2% (Inhaltsadaption nur im A-CDN) bzw. 4% (auch Adaptionmöglichkeiten in den Originalservern), während bei 10 verschiedenen Typen von Endgeräten ein Vorteil von ca. 6% bzw. 8% realisiert wird. Zu erklären ist der größere Vorteil bei einer größeren Heterogenität der Clients dadurch, dass mit einer steigenden Zahl von Endgerätetypen auch die Zahl der benötigten Repräsentationen wächst. Somit wirkt sich der Nachteil, der durch die Replikation ausschließlich statisch voradaptierter Repräsentationen bei den $K/V/*$ -Ansätzen entsteht, mit zunehmender Heterogenität der Clients mehr aus.

Da bei den $G/A/*$ -Ansätzen ausschließlich generische Repräsentationen repliziert werden, spielt bei diesen Ansätzen die Heterogenität der Clientpopulation keine Rolle. Mit zunehmender Zahl von benötigten Repräsentationen können A-CDNs der Klasse $G/VA/*$ mit adaptivitätssensitiven Platzierungsmechanismen aber weniger von der Möglichkeit der Replikation statisch voradaptierter Repräsentationen profitieren. Folglich sinkt der Vorteil der adaptivitätssensitiven Ansätze gegenüber den $G/A/*$ -Ansätzen sogar leicht mit zunehmender Heterogenität. An der Stelle $sc(h) = 100$ KB zum Beispiel um 1 bis 2 Prozentpunkte. Im Vergleich zu dem großen Vorteil von ca. 44% (Inhaltsadaption nur im A-CDN) bzw. 88% (auch Adaptionmöglichkeiten in den Originalservern) bei 10 Typen von Endgeräten ist das jedoch vernachlässigbar.

6.6.10 Systemgröße

Im Simulationsmodell nach Kapitel 5 wird ein System mit 15 Zugangspunkten, jeweils 5 von jedem der 3 betrachteten Typen von Zugangspunkten, zugrunde gelegt. Dementsprechend verfügt das System über 15 Surrogate, jeweils ein Surrogat in jedem AS, in dem sich ein Zugangspunkt befindet. Die Einschränkung auf 15 Zugangspunkte wurde getroffen, um die Rechenzeit für die Simulationen zu begrenzen. Wie aus Tabelle 4.1(b) auf Seite 96 hervorgeht, hängt die Zeitkomplexität des Algorithmus zur adaptivitätssensitiven Platzierung von Replikaten in A-CDNs bei der im Simulator verwendeten Implementierung (Greedy-Algorithmus 4.6, wobei optimale Adaptionspfade mittels des Dijkstra-Algorithmus 4.1 ermittelt werden) über $O(|C| \cdot |H|^2 \log|H|)$ von der Systemgröße ab. Deshalb würde eine Verdopplung der Zahl der Zugangspunkte und Surrogate zu einer um ca. Faktor 10 höheren Laufzeit führen.

Reale CDNs können deutlich mehr Surrogate umfassen. Akamai Technologies zum Beispiel betreibt weltweit über 14 000 Surrogate in 1100 Netzwerken [oV04a]. Wenn man davon ausgeht, dass in jedem Netzwerk ein CDN-Rechenzentrum mit durchschnittlich 13 Surrogaten betrieben wird, dann entspricht das 1100 Surrogaten gemäß dem Systemmodell aus Abschnitt 3.2.1, da die Surrogate in einem CDN-Rechenzentrum im Modell als ein Surrogat betrachtet werden. Kleinere CDN-Betreiber hingegen haben teilweise deutlich weniger Surrogate im Betrieb, Mirror Image zum Beispiel lediglich 22 [VP03].

Um den Einfluss der Zahl der betrachteten Surrogate und Clientknoten auf die Leistungsfähigkeit der verschiedenen Ansätze abzuschätzen, wurde eine Serie von Experimenten mit weniger als 15 Zugangspunkten und Surrogaten durchgeführt und die Ergebnisse dieser Experimente mit denen für 15 Zugangspunkte und Surrogate verglichen. Dazu wurde der Parameter N_d , der die Zahl der Zugangspunkte pro Typ bestimmt, zwischen 1 und 4 variiert. Damit ergeben sich Systemmodelle mit 3 bis 12 Zugangspunkten und Surrogaten. Die übrigen Eigenschaften des Simulationsmodells wurden konstant gehalten. Insbesondere blieb bei der Einschränkung der betrachteten Zugangspunkte und Clients die Verteilung der Eigenschaften der Clients und deren Netzwerkanbindungen unverändert. Der Einfluss der Heterogenität der Clienteneigenschaften auf die Ergebnisse wurde in Abschnitt 6.6.9 separat untersucht.

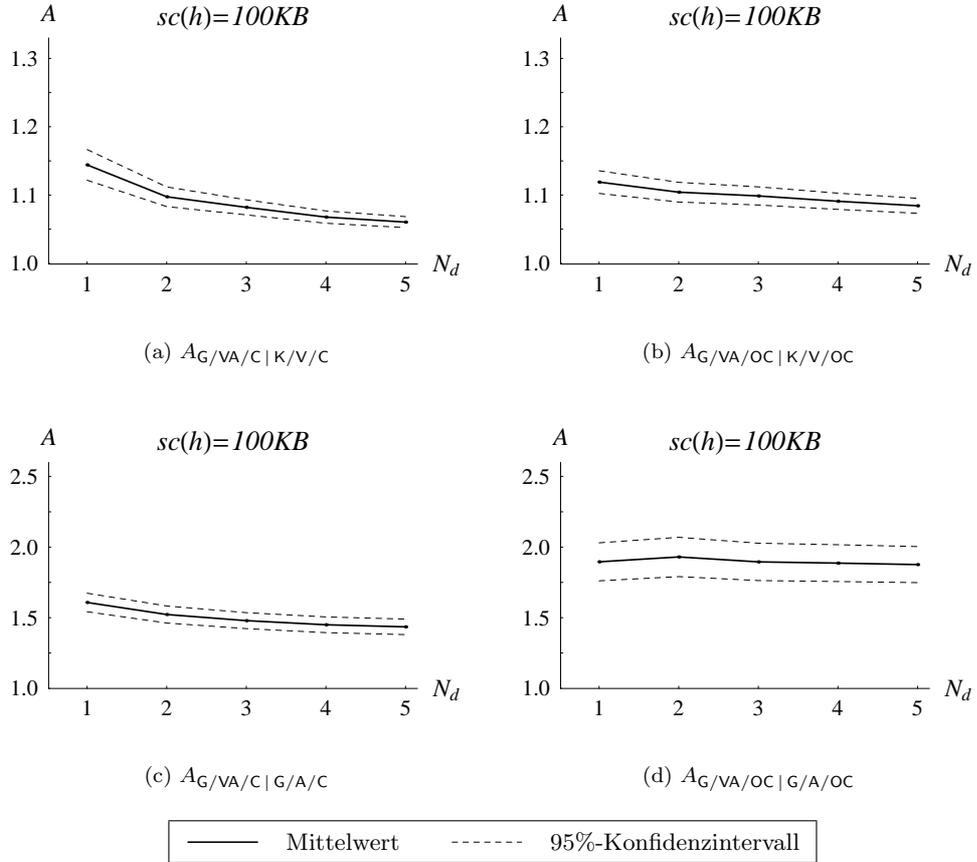


Abbildung 6.10: Relative Leistungsfähigkeit in Abhängigkeit von der Systemgröße

Die Ergebnisse dieser Analysen sind für eine Speicherkapazität von 100 KB pro Surrogat in Abbildung 6.10 dargestellt. Die Diagramme zeigen, dass der Vorteil der adaptivitätssensitiven Platzierung – mit Ausnahme des Vorteils gegenüber dem G/A/OC-Ansatz – mit zunehmender Systemgröße (Parameter N_d) abnimmt. Besonders deutlich ist dieser Effekt für $A_{G/VA/C|K/V/C}$ zu erkennen. Eine Korrelation zwischen der Systemgröße und $A_{G/VA/OC|G/A/OC}$ ist hingegen nicht nachweisbar.

Der sinkende Vorteil der adaptivitätssensitiven Platzierung mit zunehmender Systemgröße ist damit zu erklären, dass die Gesamtspeicherkapazität des A-CDN mit der zunehmenden Zahl von Surrogaten steigt. Folglich sind die K/V/*- und G/VA/*-Ansätze in der Lage, mehr verschiedene Repräsentationen der Inhaltsobjekte zu replizieren. Zwar können nicht alle Repräsentationen im nächstgelegenen Surrogat eines Client repliziert werden, aber der Zugriff auf ein adaptiertes Replikat in einem entfernten Surrogat ist oft günstiger, als der Zugriff auf das generische Originalreplikat im Originalserver. Das gilt besonders dann, wenn die Originalserver nicht über Adaptionmöglichkeiten verfügen (Abb. 6.10(a)) und demzufolge nicht nur die Kosten und Verzögerungen durch dynamische Inhaltsadaption, sondern auch durch den Transfer der generischen Repräsentation zu einem Surrogat den Gewinn des A-CDN-Betreibers schmälern.

Der Zusammenhang zwischen der Systemgröße und dem Mittelwert der relativen Leistungsfähigkeit $A_{G/VA/C|K/V/C}$ und $A_{G/VA/OC|K/V/OC}$ kann sehr gut durch die Funktion

$$f(N_d) = \alpha \cdot e^{-\beta N_d} + \gamma \quad (6.8)$$

angenähert werden (Bestimmtheit bei Speicherkapazitäten von 50 KB, 100 KB und 300 KB zwischen 0,99 und 1,00). Durch Extrapolation des näherungsweise Verlaufes ergibt sich, dass der Vorteil der adaptivitätssensitiven Platzierung gegenüber dem K/V/C-Ansatz zum Beispiel

bei einer Speicherkapazität von 100 KB pro Surrogat für beliebige Systemgrößen nie weniger als ca. 6% beträgt, also nur vernachlässigbar weniger als an der Stelle $N_d = 5$. Die untere Grenze des relativen Vorteils adaptivitätssensitiver Platzierung von Replikaten gegenüber dem K/V/OC-Ansatz liegt für $sc(h) = 100$ KB bei ca. 7%, nur ein Prozentpunkt weniger als in einem System mit 15 Surrogaten.

Die Abhängigkeit zwischen der Systemgröße und dem mittleren Vorteil adaptivitätssensitiver Platzierung gegenüber dem G/A/C-Ansatz kann bei Speicherkapazitäten von 50 KB und 100 KB pro Surrogat ebenfalls sehr gut durch Formel 6.8 angenähert werden (Bestimmtheit von 1,00). Durch Extrapolation kann damit für $sc(h) = 100$ KB eine untere Grenze für $A_{G/VA/OC|G/A/C}$ von 1,42 (im Vergleich zu 1,44 an der Stelle $N_d = 5$) abgeschätzt werden. Bei einer Speicherkapazität von 300 KB pro Surrogat ist die Abhängigkeit zwischen der Systemgröße und der relativen Leistungsfähigkeit sehr schwach ausgeprägt (Korrelationskoeffizient $r = 0,03$), so dass keine Näherung bestimmt werden kann.

Auch die Ursache für den geringeren Nachteil des G/A/C-Ansatzes gegenüber adaptivitätssensitiven Platzierungsmechanismen bei größeren Systemen liegt in der größeren Gesamtspeicherkapazität des A-CDN. Dadurch können mehr Inhaltsobjekte im A-CDN repliziert werden und es sind weniger Zugriffe auf die Originalserver erforderlich. Da die Originalserver beim G/A/C-Ansatz nicht über Adaptionmöglichkeiten verfügen, ist ein Zugriff auf ein Replikat in einem entfernten Surrogat günstiger als der Zugriff auf ein Originalreplikat in einem Originalserver.

Wenn die Originalserver ebenfalls über Adaptionmöglichkeiten verfügen (G/A/OC), so sind die Kosten und Verzögerungen für die Adaption eines Originalreplikates im Originalserver und eines Replikates in einem entfernten Surrogat vergleichbar. Folglich ist keine Korrelation zwischen der Systemgröße und $A_{G/VA/OC|G/A/OC}$ nachweisbar ($r = 0,00$).

Zusammenfassend lässt sich sagen, dass der Vorteil adaptivitätssensitiver Platzierungsmechanismen bei steigender Systemgröße sinkt, der adaptivitätssensitive Ansatz aber auch bei sehr großen Systemen mit mehreren hundert oder tausend Surrogaten einen Vorteil gegenüber herkömmlichen Platzierungsmechanismen mit sich bringt, der nur geringfügig unter dem in einem System mit 15 Surrogaten liegt (vorausgesetzt, dass die festgestellten Zusammenhänge extrapolierbar sind).

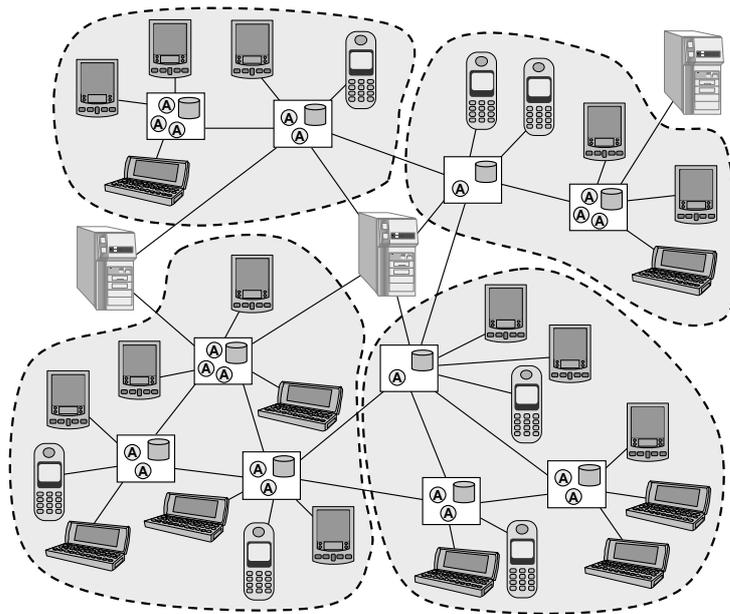


Abbildung 6.11: Untergliederung des A-CDNs in Regionen

Allerdings ist kaum zu erwarten, dass die global koordinierte Platzierung von Replikaten in einem A-CDN mit mehreren hundert weltweit verteilten Surrogaten skaliert. Deshalb müssen in einem A-CDN dieser Größenordnung heuristische Platzierungsmechanismen eingesetzt werden, deren Parameterbereich auf eine Teilmenge der Surrogate und Clientknoten eingeschränkt ist. Dies lässt sich erreichen, indem das A-CDN in Regionen unterteilt wird und für jede Region unabhängig von den anderen Regionen eine Platzierung von Replikaten bestimmt wird, wobei nur Surrogate und Anfragen von Clients innerhalb der Region berücksichtigt werden (Abb. 6.11). Die einzelnen Regionen in einem solchen A-CDN können wie unabhängige A-CDNs mit einer eingeschränkten Zahl von Surrogaten und Clients betrachtet werden. Wie in diesem Abschnitt gesehen, bringt adaptivitätssensitive Platzierung in den Regionen mit wenigen Surrogaten einen größeren Vorteil mit sich als in einem A-CDN mit Hunderten von Surrogaten.

6.6.11 Zahl der betrachteten Objekte

Die Menge K der Objekte in der Anfragelast ist in den Simulationsexperimenten auf 100 Objekte beschränkt. Aufgrund der geringen Zahl von Objekten ist auch der mittlere Speicherplatzbedarf zur vollständigen Replikation aller benötigten Repräsentationen von lediglich 489 KB sehr gering im Vergleich zur Festplattenkapazität verfügbarer Rechnersysteme in einer Größenordnung von mehreren hundert Gigabyte. In der Praxis ist die Zahl der Objekte deutlich höher. Zum Beispiel wird in [Kel01] von 36,6 Millionen Inhaltsobjekten berichtet, die innerhalb von 16 Tagen von 37 169 Clients des WebTV-Netzwerks³ angefordert wurden. Wolman et al. haben im Mai 1999 Zugriffe auf 18,4 Millionen Inhaltsobjekte von 22 984 WWW-Clients an der University of Washington protokolliert [WVS⁺99]. Laut [BJS02] liegt der Anteil der Bildobjekte in der Anfragelast eines WWW-Proxies in einer Größenordnung um 65%. Folglich sind grob geschätzt 12 Millionen der Inhaltsobjekte im WWW-Protokoll der University of Washington Bildobjekte. Die Zahl der Inhaltsobjekte einer einzelnen Web-Site liegt Größenordnungen darunter. So wurden zum Beispiel von der Web-Site www.france98.com zur Fußball-Weltmeisterschaft 1998 20 728 verschiedene Inhaltsobjekte angefordert [AJ99]. Somit werden Zahlen von mehreren Millionen Inhaltsobjekten nur bei A-CDNs erreicht, die mehrere hundert große Inhaltsanbieter zu ihren Kunden zählen.

Um den Einfluss der Zahl der Objekte auf die Simulationsergebnisse abzuschätzen, wurde eine Serie von Experimenten durchgeführt, bei der die Zahl der Objekte $|K|$ innerhalb $\{10, 20, 40, 60, 80\}$ variiert wurde. In Abbildung 6.12 sind die Ergebnisse für $|K| \in \{20, 60\}$ in Form der relativen Leistungsfähigkeit des adaptivitätssensitiven Ansatzes für eine Konfiguration ohne Adaptionmöglichkeiten in den Originalservern ($*/*/C$) denen für $|K| = 100$ gegenübergestellt. Die Ergebnisse für $|K| \in \{10, 40, 80\}$ werden zugunsten der Übersichtlichkeit in Abbildung 6.12 nicht gezeigt. Aus dem gleichem Grund wird auf die Darstellung der Konfidenzintervalle verzichtet.

Abbildung 6.12(a) zeigt, dass der Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber dem $K/V/C$ -Ansatz bei gleicher Speicherkapazität der Surrogate mit zunehmender Zahl von Objekten steigt. Das ist durch den größeren Speicherplatzbedarf einer größeren Zahl von Inhaltsobjekten zu erklären. Somit werden bestimmte Schwellwerte der relativen Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen erst bei höheren Speicherkapazitäten unterschritten. Zum Beispiel erreicht der $G/VA/C$ -Ansatz bei 20 Objekten nur bei Speicherkapazitäten unter 33 KB pro Surrogat einen Vorteil von mehr als 5% gegenüber dem $K/V/C$ -Ansatz. Bei $|K| = 100$ hingegen liegt der Vorteil des $G/VA/C$ -Ansatzes bei Speicherkapazitäten bis zu 124 KB über 5%.

Der Zusammenhang zwischen der Zahl der Objekte und der mittleren Speicherkapazität, bei der ein bestimmter Schwellwert der relativen Leistungsfähigkeit unterschritten wird, ist für verschiedene Schwellwerte der relativen Leistungsfähigkeit in Abbildung 6.13 dargestellt. Die Speicherkapazitäten für die verschiedenen Schwellwerte wurden dabei durch eine stück-

³WebTV ist ein ISP, der Internetzugang für Set-Top-Boxen anbietet, mittels denen ein konventioneller Fernseher zur Darstellung von WWW-Inhalten genutzt werden kann. WebTV-Clients waren im Jahr 2000, als die Daten erhoben wurden, über ein Modem mit einer durchschnittlichen Datenrate von 33,6 kBit/s angebunden.

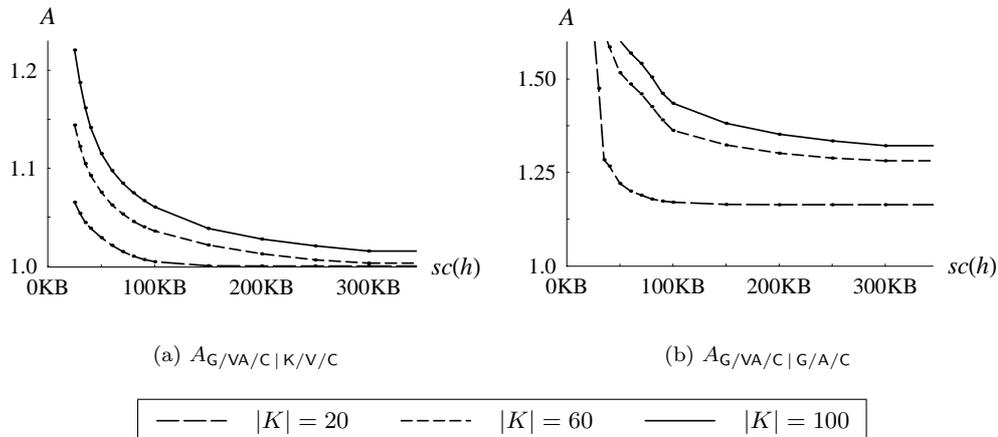
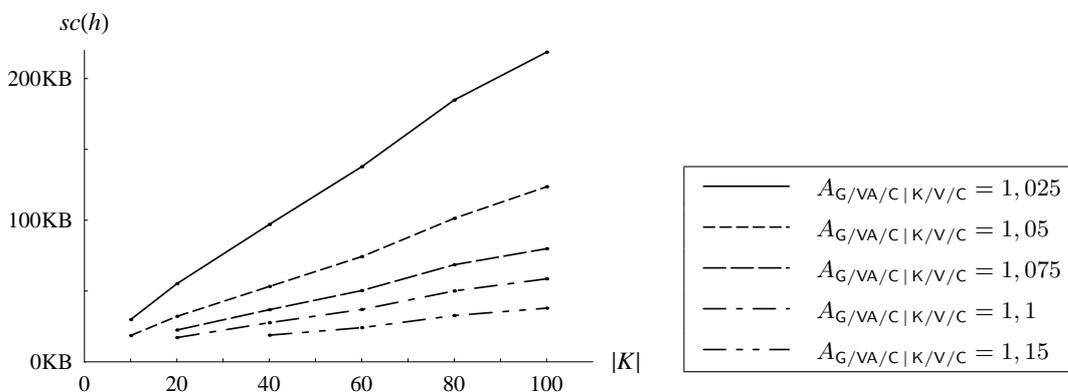


Abbildung 6.12: Relative Leistungsfähigkeit bei verschiedenen Zahlen von Inhaltsobjekten

weise lineare Näherung der Inversen der in Abbildung 6.12(a) dargestellten Abhängigkeiten ermittelt. Um den Fehler durch die stückweise lineare Näherung zu minimieren, wurden in Ergänzung zu den Ergebnissen der Experimente mit Speicherkapazitäten entsprechend Tabelle 5.1 auf Seite 103 in zusätzlichen Experimenten mit $sc(h) \in \{60 \text{ KB}, 70 \text{ KB}, 80 \text{ KB}, 90 \text{ KB}\}$ weitere Stützstellen ermittelt (dafür entfielen die Experimente mit $sc(h) = 75 \text{ KB}$).

Abbildung 6.13 zeigt, dass die Speicherkapazität, bei der ein bestimmter Schwellwert der relativen Leistungsfähigkeit unterschritten wird, mit steigendem $|K|$ linear wächst (Bestimmtheit der linearen Näherung zwischen 0,99 und 1,00). Durch Extrapolation des linearen Zusammenhangs ergibt sich, dass zum Beispiel für $|K| = 10^6$ der G/VA/C-Ansatz einen Vorteil von mindestens 5% gegenüber dem K/V/C-Ansatz mit sich bringt, wenn die Speicherkapazität pro Surrogat weniger als 1,1 GB beträgt. Bei weniger als 0,5 GB beträgt der Vorteil mehr als 10%.

Der Vorteil der adaptivitätssensitiven Platzierung gegenüber dem G/A/C-Ansatz wächst ebenfalls mit zunehmender Zahl von Objekten (Abb. 6.12(b)), da der Speicherplatzbedarf zur vollständigen Replikation der generischen Repräsentationen steigt. Allerdings ist die Korrelation der Zahl der Objekte mit der relativen Leistungsfähigkeit bezüglich des G/A/C-Ansatzes schwächer ausgeprägt als diejenige mit der relativen Leistungsfähigkeit bezüglich des K/V/C-Ansatzes (z.B. für $sc(h) = 100 \text{ KB}$ $r = 0.36$ im Vergleich zu $r = 0.61$).

Abbildung 6.13: Grenzwerte der Speicherkapazität $sc(h)$, bei denen der Mittelwert der relativen Leistungsfähigkeit $A_{G/VA/C|K/V/C}$ bestimmte Schwellwerte überschreitet

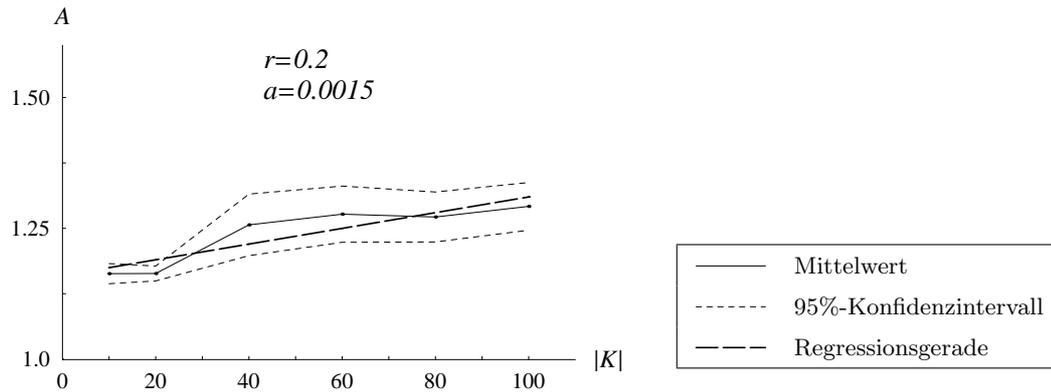


Abbildung 6.14: Relative Leistungsfähigkeit $A_{G/VA/C|G/A/C}$ für $sc(h) \rightarrow \infty$ in Abhängigkeit von der Zahl der Objekte $|K|$

Auch bei unbegrenzter Speicherkapazität ($sc(h) \rightarrow \infty$) ist eine leichte positive Korrelation zwischen der Zahl der Objekte und der relativen Leistungsfähigkeit $A_{G/VA/C|G/A/C}$ nachweisbar (Abb. 6.14). Diese ist nicht durch einen höheren Speicherplatzbedarf zur Replikation einer höheren Zahl von Inhaltsobjekten zu erklären. Vielmehr ist dieser Effekt auf die heavy-tailed Verteilung der Größen der primären Repräsentationen zurückzuführen. Die starke Asymmetrie der Verteilung hat zur Folge, dass bei einer steigenden Zahl von Objekten die Wahrscheinlichkeit steigt, dass das Lastmodell sehr große Objekte umfasst. Da der Vorteil adaptivitätssensitiver Platzierung gegenüber dem G/A/C-Ansatz bei sehr großen Objekten deutlich größer ist als bei Objekten mit kleinen primären Repräsentationen (vgl. Abs. 6.6.7), hat dieser statistische Effekt eine leichte positive Korrelation der Zahl der Objekte mit $A_{G/VA/C|G/A/C}$ zur Folge.

6.7 Quellen der Vorteile der adaptivitätssensitiven Platzierung

Wie in den vorangegangenen Abschnitten gesehen, bringt adaptivitätssensitive Platzierung von Replikaten in Bezug auf die Zielfunktion $Profit_{\Sigma, \Lambda}(P_R)$ Vorteile gegenüber dem Einsatz von herkömmlichen Platzierungsmechanismen in A-CDNs mit sich. Die Vorteile der adaptivitätssensitiven Platzierung können auf verschiedene Ursachen zurückgeführt werden. In diesem Abschnitt wird diskutiert, wie die verschiedenen Eigenschaften des adaptivitätssensitiven Ansatzes zu dessen Vorteil beitragen.

In Abschnitt 1.3 wurde dargestellt, dass herkömmliche Ansätze der Klasse K/V/* zwei Probleme in sich bergen, die durch adaptivitätssensitive Verfahren überwunden werden:

1. mangelnde Unterstützung generischer Anfragen
2. fehlende Möglichkeit zur dynamischen Inhaltsadaption

Beide Probleme tragen dazu bei, dass in A-CDNs der Klasse K/V/* die Wiederverwendbarkeit der Replikate eingeschränkt ist, was einen erhöhten Speicherplatzbedarf in den Surrogaten zur Folge hat. Um zu ermitteln, wie diese beiden Faktoren zum Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber herkömmlichen Platzierungsmechanismen der Klasse K/V/* beitragen, wurden zusätzlich Experimente mit A-CDNs der Klasse G/V/* durchgeführt. A-CDNs dieser Klasse akzeptieren generische Anfragen. Inhaltsadaption findet aber ausschließlich bei der Verteilung, nicht bei der Auslieferung der Inhalte statt. Folglich replizieren A-CDNs der Klasse G/V/* nur vollständig adaptierte Repräsentationen⁴.

⁴Da A-CDNs der Klasse G/V/* generische Anfragen unterstützen und verschiedene Repräsentationen replizieren können, werden adaptivitätssensitive Mechanismen zur Platzierung von Replikaten benötigt.

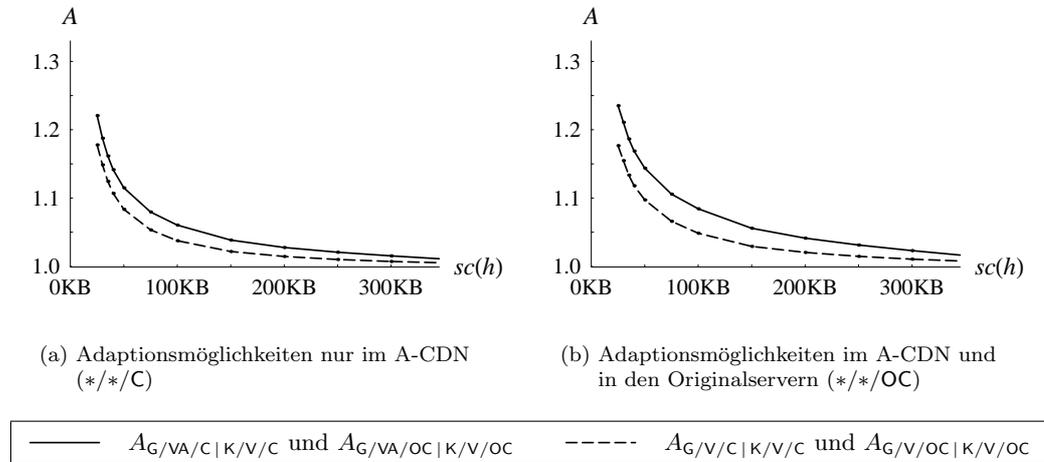


Abbildung 6.15: Quellen der Vorteile der G/VA/*-Ansätze gegenüber den K/V/*-Ansätzen

Somit stellen A-CDNs der Klasse G/V/* eine Zwischenlösung zwischen A-CDNs der Klasse K/V/* und der Klasse G/VA/* dar, bei der das Problem der mangelnden Unterstützung generischer Anfragen, nicht aber das Problem der fehlenden Möglichkeiten zur dynamischen Inhaltsadaption überwunden ist. Deshalb erlauben Erkenntnisse über die Leistungsfähigkeit der G/V/*-Ansätze Rückschlüsse auf die Quellen der Vorteile der adaptivitätssensitiven Platzierungsmechanismen in A-CDNs der Klasse G/VA/*.

In Abbildung 6.15 ist die relative Leistungsfähigkeit der G/V/*-Ansätze bezüglich der K/V/*-Ansätze der G/VA/*-Ansätze bezüglich der K/V/*-Ansätze gegenübergestellt. Die Diagramme zeigen, dass der Vorteil der G/VA/*-Ansätze bei Speicherkapazitäten um 200 KB ungefähr doppelt so groß ist, wie der der G/V/*-Ansätze. Das bedeutet, dass der Vorteil von A-CDNs der Klasse G/VA/* mit adaptivitätssensitiven Platzierungsmechanismen, der auf die Unterstützung generischer Anfragen zurückzuführen ist, und der, der aus der Möglichkeit der dynamischen Inhaltsadaption hervorgeht, vergleichbar groß sind. Bei geringeren Speicherkapazitäten steigt die Bedeutung der Unterstützung generischer Anfragen, so dass an der Stelle $sc(h) = 50$ KB mehr als $\frac{2}{3}$ des Vorteils der G/VA/*-Ansätze gegenüber den K/V/*-Ansätzen darauf zurückzuführen ist. Bei Speicherkapazitäten über 200 KB liegt die Hauptquelle des Vorteils der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs der Klasse G/VA/* in der Möglichkeit zur dynamischen Inhaltsadaption.

Wenn die Toleranz der Clients gegenüber unterschiedlichen Repräsentationen geringer ist, wie durch das Qualitätsmodell auf Basis der alternativen Qualitätsfunktion entsprechend Formel 6.7 auf Seite 136 ausgedrückt, dann spielt die Unterstützung generischer Anfragen eine geringere Rolle für die Leistungsfähigkeit der adaptivitätssensitiven Ansätze. Für das alternative Qualitätsmodell ist der Vorteil des adaptivitätssensitiven Ansatzes an der Stelle $sc(h) = 50$ KB überwiegend (zu über 52%) auf dynamische Inhaltsadaption zurückzuführen, für Speicherkapazitäten von 200 KB pro Surrogat sogar zu über 72%.

Die mangelnde Unterstützung generischer Anfragen durch A-CDNs der Klasse K/V/* bedeutet, dass solche A-CDNs oft mehr verschiedene Repräsentationen an die Clients ausliefern als ein A-CDN mit einem adaptivitätssensitiven Platzierungsmechanismus, denn sie sind nicht in der Lage, sich die Toleranz der Clients bezüglich verschiedener Repräsentationen zu Nutze zu machen. Folglich müssen auch mehr verschiedene Repräsentationen im A-CDN repliziert werden. Um dieses Problem zu umgehen, könnte ein A-CDN-Betreiber die Zahl der unterstützten Repräsentationen willkürlich einschränken. Eine Einschränkung der unterstützten Repräsentationen hätte zur Folge, dass mehr Clients die gleichen Repräsentationen anfordern müssten und damit die Wiederverwendbarkeit der Replikate erhöht würde.

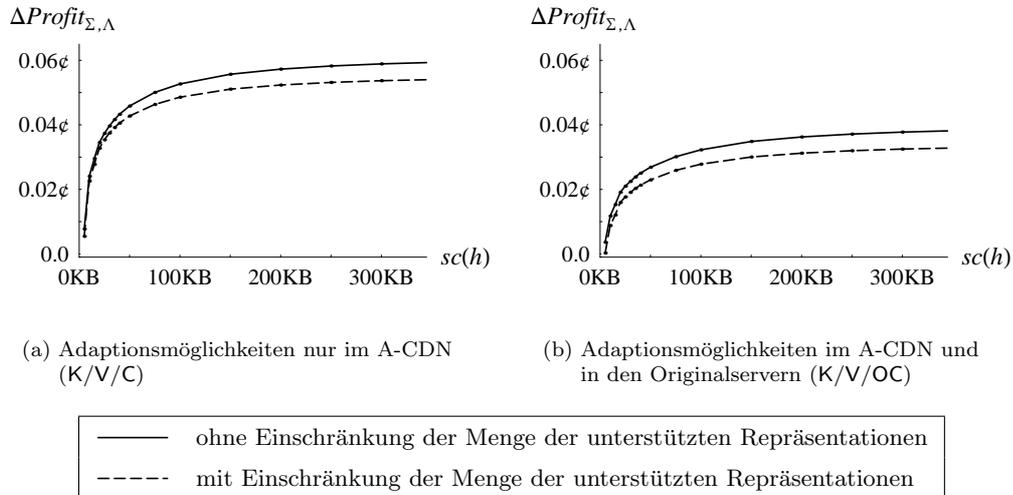


Abbildung 6.16: Absoluter Zuwachs an Gewinn pro Anfrage in einem A-CDN der Klasse K/V/* mit und ohne Einschränkung der Menge der unterstützten Repräsentationen (in Eurocent)

Um die Chancen dieses Ansatzes zu evaluieren, wurde eine Serie von Experimenten durchgeführt, bei denen die Menge der unterstützten Bildauflösungen auf $\{101 \times 80, 160 \times 160, 240 \times 320, 1024 \times 768\}$ beschränkt wurde. Die Auflösungen 160×160 , 240×320 und 1024×768 sind die Auflösungen der Endgerädetypen, die die meisten Anfragen initiieren. Die Auflösung von 101×80 Bildpunkten muss unterstützt werden, damit alle Anfragen aller Clients beantwortet werden können. Die Ergebnisse dieser Experimente sind in Abbildung 6.16 denen für A-CDNs der Klasse K/V/* gegenübergestellt, die alle Displayauflösungen aller Endgerädetypen unterstützen.

Die Diagramme zeigen, dass sich die Einschränkung der Zahl der unterstützten Repräsentationen negativ auf den Gewinn des A-CDN-Betreibers auswirkt. Das ist darauf zurückzuführen, dass die Einschränkung der Zahl der Repräsentationen für viele Anfragen eine Minderung der Qualität bedeutet. Die negativen Auswirkungen dieses Qualitätsverlustes auf den Gewinn des A-CDN-Betreibers überwiegen die positiven Effekte, die sich aus der Erhöhung der Wiederverwendbarkeit der Replikatate ergeben. Um im Endeffekt eine Steigerung des Gewinns des A-CDN-Betreibers zu erreichen, müsste die Menge der unterstützten Repräsentationen für die verschiedenen Inhaltsobjekte individuell angepasst werden. Eine solche individuelle Anpassung der Menge der ausgelieferten Repräsentationen findet in A-CDNs mit adaptivitätssensitiven Platzierungsmechanismen statt.

Als Nachteil von A-CDNs der Klasse G/A/*, die herkömmliche Platzierungsmechanismen zur Replikation der generischen Repräsentationen verwenden, wurden in Abschnitt 1.3 die Kosten und Verzögerungen infolge der dynamischen Inhaltsadaption bei jeder Anfrage genannt. Darüber hinaus kann sich aber auch der höhere Speicherplatzbedarf einer generischen Repräsentation gegenüber einer adaptierten oder teilweise adaptierten negativ auf die Leistungsfähigkeit der Ansätze der Klasse G/A/* im Vergleich zu A-CDNs der Klasse G/VA/* auswirken.

Aufschluss darüber, inwieweit sich der zweite Faktor auf den Vorteil der adaptivitätssensitiven Platzierung gegenüber den Ansätzen der Klasse G/A/* auswirkt, geben die Experimente in Abschnitt 6.6.5, bei denen die Adaptionzeiten um einen Faktor tf variiert wurden. In Abbildung 6.17 sind die Ergebnisse dieser Experimente in Form der relativen Vorteile $A_{G/VA/C|G/A/C}$ (Abb. 6.17(a)) und $A_{G/VA/OC|G/A/OC}$ (Abb. 6.17(b)) für $tf = 1$ und $tf = 0$ gegenübergestellt. Da bei $tf = 0$ Inhaltsadaption keine Kosten und Verzögerungen verursacht, ist der Vorteil des adaptivitätssensitiven Ansatzes nur mit der Einsparung an Speicherplatz bei der Replikation adaptierter Repräsentationen gegenüber der Replikation der generischen Repräsentationen

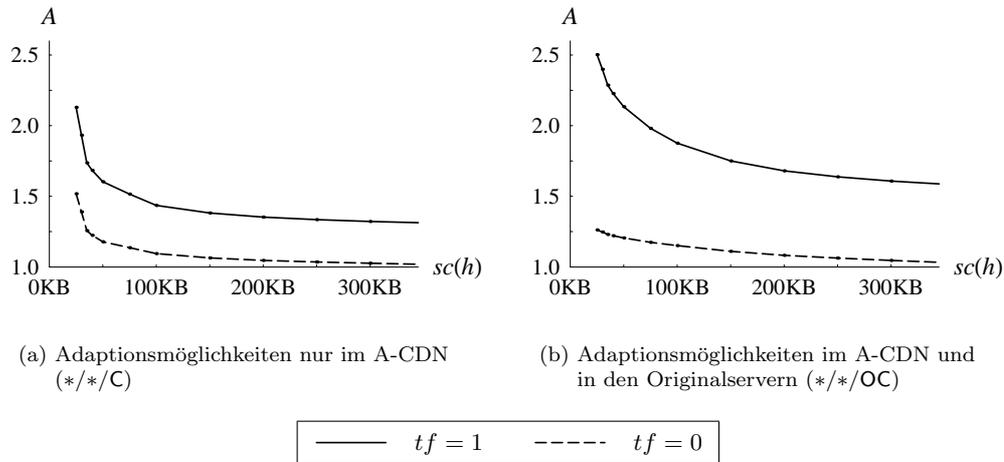


Abbildung 6.17: Quellen der Vorteile der G/VA/*-Ansätze gegenüber den G/A/*-Ansätzen

zu erklären. Bei $tf = 1$ hingegen tragen beide Faktoren zum Vorteil der adaptivitätssensitiven Platzierungsmechanismen bei. Somit drückt die Differenz zwischen den beiden Kurven den Einfluss der Kosten und Verzögerungen durch die Adaption aus.

Wie aus den Diagrammen zu ersehen ist, spielt der höhere Speicherplatzbedarf generischer Repräsentationen nur eine untergeordnete Rolle. Zum Beispiel ist der Vorteil der adaptivitätssensitiven Platzierung für $tf = 1$ an der Stelle $sc(h) = 100$ KB zu durchschnittlich 78% (Abb. 6.17(a)) bzw. 83% (Abb. 6.17(b)) auf die Nutzung von statisch voradaptierten Repräsentationen zurückzuführen. Lediglich 22% bzw. 17% können mit dem geringeren Speicherplatzbedarf adaptierter Repräsentationen erklärt werden. Bei Speicherkapazitäten von über durchschnittlich 462 KB bzw. 458 KB spielt der Speicherplatzbedarf der generischen Repräsentationen überhaupt keine Rolle. Dann reicht die Kapazität der Surrogate aus, alle Inhaltsobjekte in allen Surrogaten, bei denen diese Inhaltsobjekte angefragt werden, in der generischen Repräsentation zu replizieren.

6.8 Adaptivitätssensitive Platzierung von Replikaten in hosting A-CDNs

Abschnitt 6.5 hat gezeigt, dass bei Speicherkapazitäten unter 487 KB pro Surrogat adaptivitätssensitive Platzierung von Replikaten in A-CDNs nicht nur einen Vorteil bezüglich der Bewertungsmetrik $\Delta Profit_{\Sigma, \Lambda}$ mit sich bringt, sondern auch mehr Anfragen durch die Surrogate des A-CDN beantwortet werden können und somit weniger Zugriffe auf die Originalserver notwendig sind. Bei relaying A-CDNs, wie sie bei der Modellierung in Kapitel 3 betrachtet werden, ist die Reduzierung der Last in den Originalservern aber nicht direkt als Optimierungsziel formuliert. Im Gegensatz dazu müssen in hosting CDNs (vgl. Abs. 1.1.4) alle Anfragen durch die Surrogate des CDNs beantwortet werden. Originalserver stehen nicht zur Beantwortung von Anfragen zur Verfügung. Folglich muss für hosting CDNs die Reduzierung der Last in den Originalservern auf Null als Nebenbedingung des Optimierungsziels formuliert werden. Deshalb ist zu erwarten, dass der adaptivitätssensitive Ansatz zur Platzierung von Replikaten in hosting A-CDNs größere Vorteile mit sich bringt als in relaying A-CDNs.

Um die Vorteile adaptivitätssensitiver Platzierungsmechanismen in hosting A-CDNs zu evaluieren, wurde eine Serie von Experimenten durchgeführt, bei denen das Ziel der Beantwortung aller Anfragen durch die Surrogate des A-CDN in die Zielfunktion integriert wurde. Hierzu wird eine pauschale Strafe für Zugriffe auf Originalserver definiert. Jeder Adaptionspfad, der von ei-

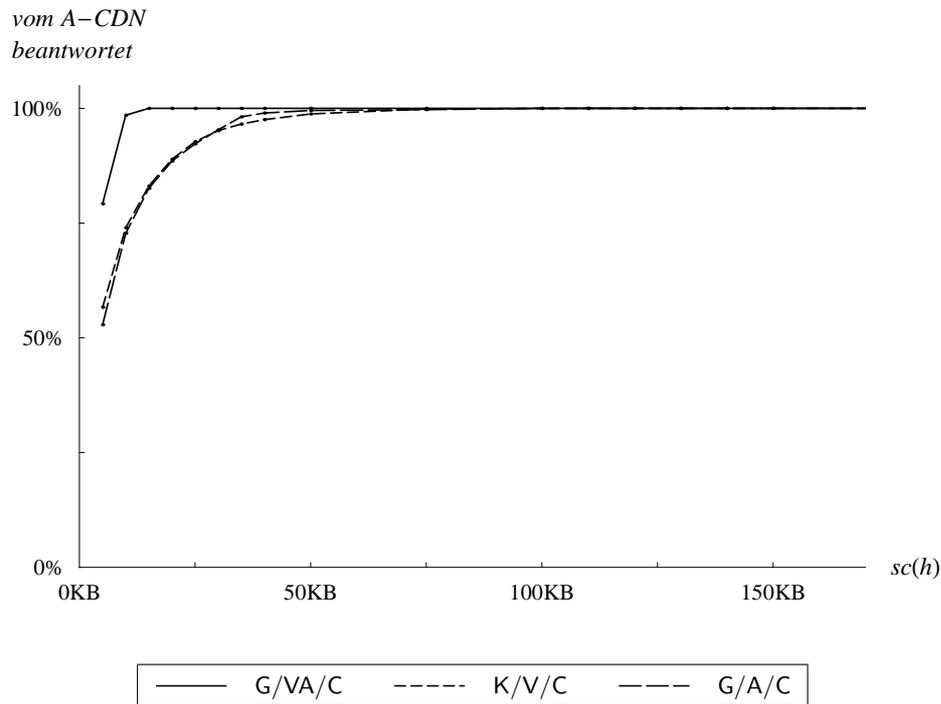


Abbildung 6.18: Anteil der Anfragen, die durch Replikate in den Surrogaten des hosting A-CDN beantwortet werden

nem Originalreplikate ausgeht, wird pauschal mit zusätzlichen Kosten von 100 EUR belastet⁵. Aufgrund dieser Kosten für die Beantwortung von Anfragen durch die Originalserver wird jeder Platzierungsmechanismus eine Platzierung ermitteln, bei der, wenn möglich, alle Anfragen durch Replikate in den Surrogaten beantwortet werden. Bei jeder Platzierung, bei der nicht alle Anfragen durch Replikate in den Surrogaten des A-CDN beantwortet werden können, wird der absolute Zuwachs $\Delta Profit_{\Sigma, \Lambda}$ an Gewinn pro Anfrage negativ sein⁶. Indem die Zielsetzung der Beantwortung aller Anfragen durch die Surrogate des A-CDNs über die Zusatzkosten in die Zielfunktion integriert und nicht als Nebenbedingung formuliert wird, können zur Ermittlung einer Platzierung von Replikaten in hosting A-CDNs die Algorithmen aus Abschnitt 4.2 verwendet werden. Formal handelt es sich dabei um eine Relaxation der Nebenbedingung. Ergänzt man das relaxierte Problem jedoch durch die Nebenbedingung $\Delta Profit_{\Sigma, \Lambda} \geq 0$, so ist diese Problemformulierung dem ursprünglichen Problem der Platzierung von Replikaten in hosting A-CDNs äquivalent.

Abbildung 6.18 zeigt, dass durch die Definition der Zusatzkosten für Zugriffe auf Originalserver bei allen Ansätzen zur Platzierung von Replikaten in A-CDNs Platzierungen ermittelt werden, bei denen mehr Anfragen durch die Surrogate des A-CDN beantwortet werden können als bei den entsprechenden Platzierungen, die für relaying A-CDNs ermittelt werden (vgl. Abb. 6.4 auf Seite 131, man beachte den unterschiedlichen Maßstab auf der Abszisse). Während bei einem adaptivitätssensitiven Platzierungsmechanismus bereits eine Speicherkapazität von 15 KB pro Surrogat ausreicht, um alle Anfragen durch das A-CDN zu beantworten,

⁵Da bei hosting A-CDNs die Originalserver nicht zur Beantwortung von Anfragen bereitstehen, werden diese Kosten für keine Anfrage tatsächlich anfallen. Somit ist die tatsächliche Höhe der Kosten auch unerheblich. Sie müssen nur hinreichend hoch sein, dass der Platzierungsalgorithmus, wenn möglich, jedes Inhaltsobjekt mindestens einmal repliziert, so dass alle Anfragen durch das A-CDN beantwortet werden können.

⁶Bei der Ermittlung des mittleren Gewinns $Profit_{\Sigma, \Lambda}(\emptyset)$ pro Anfrage ohne die Nutzung von Replikation im A-CDN wird keine Strafe für Zugriffe auf die Originalserver berücksichtigt.

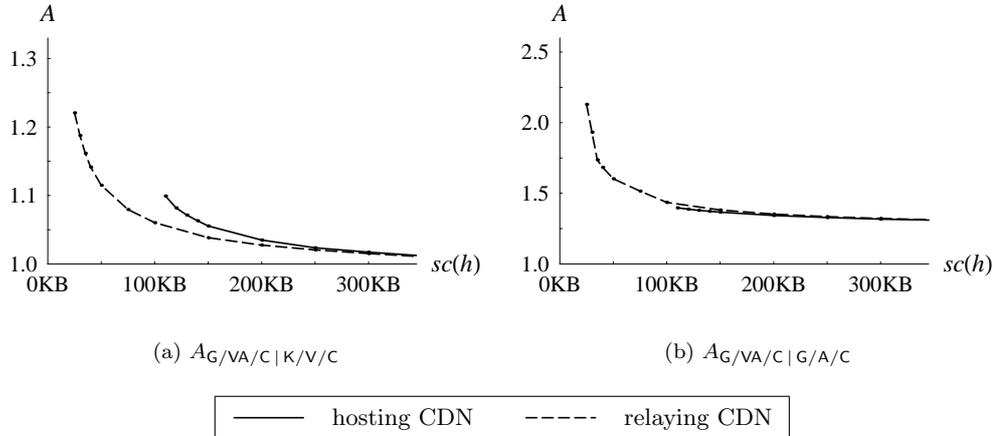


Abbildung 6.19: Vergleich der relativen Leistungsfähigkeit adaptivitätssensitiver Platzierung in hosting und relaying A-CDNs

werden bei Verwendung eines herkömmlichen Platzierungsmechanismus mehr als 100 KB pro Surrogat benötigt. Folglich können in hosting A-CDNs bei knapper Speicherkapazität der Surrogate ausschließlich adaptivitätssensitive Platzierungsmechanismen eingesetzt werden. Aber auch bei Speicherkapazitäten über 100 KB ist der Vorteil des adaptivitätssensitiven Ansatzes gegenüber dem K/V/C-Ansatz größer als in relaying A-CDNs (Abb. 6.19(a)). Bezüglich dem G/A/C-Ansatz waren für $sc(h) \geq 110$ KB keine signifikanten Unterschiede zwischen hosting und relaying A-CDNs (Abb. 6.19(b)) nachweisbar.

6.9 Zusammenfassung der Ergebnisse

In den vorangegangenen Abschnitten wurde gezeigt, dass die adaptivitätssensitive Platzierung von Replikaten in A-CDNs Vorteile gegenüber dem Einsatz von herkömmlichen Platzierungsmechanismen mit sich bringen kann, die Vorteile sich aber für die verschiedenen Ansätze zum Einsatz herkömmlicher Platzierungsmechanismen und für verschiedene System- und Lastmodelle signifikant unterscheiden.

A-CDNs der Klasse G/A/* mit herkömmlichen Platzierungsmechanismen, bei denen nur die generischen Repräsentationen repliziert werden können und Inhaltsobjekte ausschließlich dynamisch adaptiert werden, zeigen in der Praxis grundsätzlich eine schlechtere Leistungsfähigkeit als A-CDNs mit adaptivitätssensitiven Platzierungsmechanismen. In den Experimenten auf Basis des Simulationsmodells in Kapitel 5 wurde ein mittlerer Vorteil des adaptivitätssensitiven Ansatzes von mindestens 29% bzw. 54% nachgewiesen, je nachdem, ob Inhaltsadaption nur im A-CDN oder auch in den Originalservern stattfinden kann. Das ist hauptsächlich auf die Kosten und Verzögerungen infolge der dynamischen Inhaltsadaption bei jeder Anfrage zurückzuführen. Je größer die Adaptionkosten und -verzögerungen, desto dominanter ist der Vorteil des adaptivitätssensitiven Ansatzes. Nur wenn Inhaltsadaption ohne Kosten und Verzögerungen möglich wäre und hinreichend Speicherkapazität zur Verfügung stünde, dass der größere Speicherplatzbedarf der generischen Repräsentationen gegenüber teilweise voradaptierten Repräsentationen nicht relevant ist, sind die Ergebnisse für A-CDNs der Klasse G/A/* mit denen der Klasse G/VA/* mit adaptivitätssensitiven Platzierungsmechanismen vergleichbar. Die Komplexität des Platzierungsalgorithmus in A-CDNs der Klasse G/A/* liegt jedoch um einen Faktor $|T||M|$ unter der in A-CDNs der Klasse G/VA/*, so dass die Ausführungszeiten eines adaptivitätssensitiven Platzierungsalgorithmus, insbesondere wenn sehr viele verschiedene Repräsentationen berücksichtigt werden, deutlich über denen des Platzierungsalgorithmus in A-CDNs der Klasse G/A/* liegen können.

Die relative Leistungsfähigkeit von A-CDN-Ansätzen mit adaptivitätssensitiven Platzierungsmechanismen bezüglich A-CDNs der Klasse $K/V/*$, bei denen Inhaltsadaption nur während der Verteilung, jedoch nicht bei der Auslieferung der Inhalte stattfindet, ist stark vom Anfrageverhalten der Clients und der verfügbaren Speicherkapazität in den Surrogaten des A-CDN abhängig. Während unter bestimmten Bedingungen ein Vorteil von über 23% nachgewiesen werden konnte, war in anderen Szenarien kein Vorteil des adaptivitätssensitiven Ansatzes feststellbar. Jedoch ist unter der Bedingung, dass die Platzierungsalgorithmen in der Lage sind, optimale Platzierungen zu ermitteln, die Leistungsfähigkeit des adaptivitätssensitiven Ansatzes nie schlechter als die bei Verwendung eines herkömmlichen Platzierungsmechanismus. Da in den Experimenten mit heuristischen Platzierungsalgorithmen gearbeitet wurde, konnte in wenigen Fällen ein Nachteil des adaptivitätssensitiven Ansatzes gemessen werden. Dieser Nachteil war jedoch vernachlässigbar. Auch hinsichtlich der Komplexität weist ein adaptivitätssensitiver Platzierungsalgorithmus keine signifikanten Nachteile gegenüber einem herkömmlichen Platzierungsmechanismus entsprechend einem $K/V/*$ -Ansatz auf. Die obere Schranke der Komplexität ist in beiden Fällen gleich.

Die relative Leistungsfähigkeit adaptivitätssensitiver Platzierungsmechanismen in Bezug auf die Ansätze der Klasse $K/V/*$ wird in erster Linie durch die Speicherkapazität der Surrogate und die Überlappung der Anfragelast der verschiedenen Clientklassen bestimmt. Bei knapper Speicherkapazität (in den Experimenten $sc(h) \leq 50$ KB) und vollständiger Überlappung der Anfragelast der verschiedenen Clientklassen war ein Vorteil von über 19% (G/VA/C) bzw. über 23% (G/VA/OC) nachweisbar. Wenn sich die Anfragelasten der verschiedenen Klassen von Clients nicht überlappen, kann durch die Nutzung von adaptivitätssensitiven Platzierungsmechanismen jedoch nur ein marginaler Vorteil realisiert werden. A-CDNs sind also nur für solche Dienste sinnvoll, bei denen die Clients verschiedener Klassen zu einem großen Teil auf die selben Inhaltsobjekte zugreifen.

Bei steigender Speicherkapazität $sc(h)$ der Surrogate sinkt der Vorteil adaptivitätssensitiver Platzierungsmechanismen gegenüber herkömmlichen Platzierungsmechanismen der Klasse $K/V/*$. Bei hinreichend großer Speicherkapazität bringt die adaptivitätssensitive Platzierung überhaupt keine Vorteile mit sich, da in diesem Fall die vollständige Replikation aller benötigten Repräsentationen in statisch voradaptierter Form optimal ist. In den Experimenten entsprechend dem Simulationsmodell in Kapitel 5 war bei Speicherkapazitäten von durchschnittlich mehr als 489 KB kein Vorteil nachweisbar und bereits bei über 128 KB (Inhaltsadaption nur im A-CDN) bzw. 176 KB (Adaptionsmöglichkeiten auch in den Originalservern) lag der Vorteil unter 5%. Der geringe Speicherplatzbedarf liegt in der geringen Zahl von nur 100 Objekten, die in den Experimenten betrachtet wurden, und der geringen Größe der betrachteten Bildobjekte (die mittlere Größe der primären Repräsentationen lag um 6 KB) begründet. Mit steigender Zahl von Objekten wächst der Speicherplatzbedarf und damit auch die Speicherkapazitäten, bei denen bestimmte Schwellwerte der relativen Leistungsfähigkeit adaptivitätssensitiver Ansätze unterschritten werden, linear an. Aber auch bei einer Million Bildern genügt eine Speicherkapazität von ca. 1,1 GB, dass der Vorteil des adaptivitätssensitiven Ansatzes unter 5% fällt.

Sollen durch das A-CDN Bilder mit einer größeren Auflösung der primären Repräsentation verteilt werden, erhöht sich der Speicherplatzbedarf der Surrogate ebenfalls. Wenn die räumliche Auflösung der primären Repräsentationen entsprechend Abschnitt 6.6.7 mindestens 84010 Bildpunkte umfasst, erhöht sich der Speicherplatzbedarf für die vollständige Replikation aller benötigten Repräsentationen um ca. Faktor 3,2 und die Speicherkapazität, bei der der Vorteil adaptivitätssensitiver Platzierung gegenüber den $K/V/*$ -Ansätzen unter 5% sinkt, um bis zu Faktor 1,9. Genauso hat auch eine größere Heterogenität der Eigenschaften der Clients einen höheren Speicherplatzbedarf in den Surrogaten zur Folge. Bei der Erhöhung der Zahl der Endgerätetypen von 4 auf 10 wurde zum Beispiel eine Zunahme des Speicherplatzbedarfs zur vollständigen Replikation aller benötigten Repräsentationen um ca. Faktor 1,5 ermittelt. Der Speicherplatzbedarf für einen Schwellwert der relativen Leistungsfähigkeit von 5% wächst um bis zu Faktor 2,3.

Heutige Rechnersysteme können aber deutlich höhere Speicherkapazitäten zur Verfügung stellen, als zur Replikation von Bildern benötigt werden. Deshalb ist nicht zu erwarten, dass

sich in A-CDNs zur Auslieferung von Bildern an heterogene Clients durch den Einsatz von adaptivitätssensitiven Platzierungsmechanismen signifikante Vorteile gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen, die nur statisch voradaptierte Repräsentationen replizieren, erzielen lassen.

Im Gegensatz zu Bildern im WWW liegt der Speicherplatzbedarf von Videostreamen bei Video-on-Demand-Diensten um Größenordnungen höher. Chesire et al. [CWVL01] haben bei Analysen der Protokolle der Zugriffe auf Videos im Internet durch Clients an der University of Washington im April 2000 einen Median der Datenvolumina, die innerhalb der Video-Sitzungen übertragen wurden, von 0,5 MB ermittelt. Der Median der Größe der primären Repräsentation der Bilder im Simulationsmodell beträgt im Vergleich dazu lediglich 3,3 KB. Das heißt, der Speicherbedarf der primären Repräsentationen liegt bei den Videos, die in [CWVL01] betrachtet wurden, ungefähr um Faktor 155 höher als bei den Bildern, die in der vorliegenden Arbeit die Basis der Evaluation bildeten. Allerdings waren 81% der Videostreamen in der Studie von Chesire et al. mit Datenraten unter 56 kbit/s kodiert. Deshalb ist davon auszugehen, dass die Videos stark voradaptiert waren, um auch für Clients mit Modemanbindung nutzbar zu sein. Durch die Nutzung von A-CDNs können Videoinhalte für die Bedürfnisse einzelner Nutzer optimiert werden. In diesem Fall sollten die Originalserver Videoinhalte mit optimaler Qualität zur Verfügung stellen. Werden Videodaten hochauflösend und ohne wahrnehmbare Qualitätseinbußen kodiert, ist mit deutlich höheren Datenraten in einer Größenordnung von über 500 kbit/s zu rechnen [Dip02].

Darüber hinaus sind Videos, die heute über das WWW verteilt werden, sehr kurz. Der Median der Abspielzeit der Videos in der Studie in [CWVL01] betrug nur 2,2 Minuten. Das bedeutet aber, dass die Länge von Videos im WWW seit 1997, als 90% der Videos kürzer als 45 Sekunden waren [AS98], signifikant zugenommen hat. Dieser Trend wird durch aktuelle Arbeiten, wie zum Beispiel die von Li et al. [LCKN03], bestätigt. Die Autoren haben bei einer Studie im Februar und März 2003 einen Median von ca. 3 Minuten ermittelt. Wenn A-CDNs auch zur Auslieferung von Spielfilmen oder Schulungsmaterialien in Form digitaler Videos verwendet werden, dann sind laut [ASP00] mittlere Abspielzeiten von ca. 75 Minuten zu erwarten. Das durchschnittliche Datenvolumen eines Videostreams wird in [ASP00] mit 121 Mbyte angegeben (wobei der größte Teil der Videos mit Datenraten zwischen 150 kbit/s und 250 kbit/s kodiert ist).

Wenn Inhaltsobjekte dieser Größenordnung durch ein A-CDN ausgeliefert werden sollen, ist der Speicherplatzbedarf für die vollständige Replikation aller benötigten Repräsentationen deutlich höher, so dass der Einsatz von A-CDN mit adaptivitätssensitiven Platzierungsmechanismen der Klasse G/VA/* in diesem Fall gerechtfertigt ist, vorausgesetzt, es existieren Möglichkeiten zur dynamischen Inhaltsadaptation unter Echtzeitbedingungen. Wenn keine geeigneten Verfahren zur dynamischen Inhaltsadaptation von Videostreamen unter Echtzeitbedingungen zur Verfügung stehen, kann der Einsatz adaptivitätssensitiver Verfahren in A-CDNs der Klasse G/V/* sinnvoll sein, um generische Anfragen berücksichtigen zu können. Wie in Abschnitt 6.7 gezeigt, lässt sich dadurch bei knapper Speicherkapazität (in den Experimenten in Abs. 6.7 bei 50 KB oder weniger) ein Vorteil von über 10% gegenüber den K/V/*-Ansätzen realisieren. Da die Zeitkomplexität des adaptivitätssensitiven Platzierungsalgorithmus der des Platzierungsalgorithmus gemäß dem K/V/*-Ansatz entspricht, lässt sich dieser Vorteil ohne signifikant erhöhten Aufwand zur Ermittlung einer Platzierung von Replikaten erreichen.

Kapitel 7

Abschließende Bemerkungen

7.1 Zusammenfassung

In der vorliegenden Arbeit wurden Möglichkeiten zur adaptivitätssensitiven Platzierung von Replikaten in Adaptiven Content Distribution Networks (A-CDNs) diskutiert und bezüglich ihrer Vorteile gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen in A-CDNs evaluiert.

Ausgehend von aktuellen Entwicklungstendenzen in Bezug auf den Zugriff auf Inhalte im Internet, wie der zunehmenden Bedeutung des mobilen Zugriffs und dem Trend zu mehr multimedialen Inhalten, wurde zunächst ein Anwendungsszenario abgeleitet, in dem mobile und stationäre Clients zur gleichen Zeit auf die selben multimedialen Inhalte im Internet zugreifen. Um den Anforderungen einer steigenden Zahl mobiler und stationärer Internetnutzer mit stark heterogenen Endgeräten und Netzwerkanbindungen beim Zugriff auf multimediale Inhalte gerecht zu werden, wurden in dieser Arbeit Adaptive Content Distribution Networks (A-CDN) vorgeschlagen. A-CDNs sind anwendungsübergreifende, verteilte Infrastrukturen, die auf Grundlage der verteilten Replikation von Inhalten und Inhaltsadaption eine skalierbare Auslieferung von adaptierbaren Inhalten an heterogene Clients ermöglichen. Durch Inhaltsadaption wird die Repräsentation der Inhalte, d.h. deren Darstellungsform, an die Eigenschaften und Fähigkeiten der Endgeräte und der Netzwerkanbindungen der Clients angepasst. Durch verteilte Replikation der Inhalte in den Surrogaten eines A-CDN kann die Last in den Originalservern der Inhaltsanbieter reduziert und die räumliche Lokalität der Zugriffe erhöht werden. Dadurch sinkt die Netzwerklast. Darüber hinaus werden die Antwortzeiten reduziert, was schließlich zu einer Verbesserung der Dienstqualität führt.

Die Platzierung der Replikat in einem A-CDN wird durch den Platzierungsmechanismus des A-CDN gesteuert. Anders als in herkömmlichen CDNs, die keine Inhaltsadaption berücksichtigen, muss ein Platzierungsmechanismus in einem A-CDN nicht nur entscheiden, welches Inhaltsobjekt in welchem Surrogat repliziert werden soll, sondern darüber hinaus, in welcher Repräsentation bzw. in welchen Repräsentationen das Inhaltsobjekt zu replizieren ist. Herkömmliche, nicht-adaptivitätssensitive Platzierungsmechanismen sind nicht in der Lage, verschiedene Repräsentationen eines Inhaltsobjektes zu berücksichtigen. Sollen solche Platzierungsmechanismen in A-CDNs eingesetzt werden, so kann aus Sicht des Platzierungsmechanismus entweder

1. jede Repräsentation eines Inhaltsobjektes als ein separates Objekt betrachtet (Ansätze der Klasse $K/V/*$ entsprechend der Klassifikation in Abschnitt 1.3) oder
2. jedes Inhaltsobjekt mit seiner generischen Repräsentation gleichgesetzt werden (Ansätze der Klasse $G/A/*$).

Bei der ersten Variante repliziert das A-CDN nur statisch voradaptierte Repräsentationen. Somit werden die Replikat ausschließlich bei der Verteilung, jedoch nicht bei der Auslieferung adaptiert. Nur wenn eine Anfrage durch ein generisches Originalreplikat vom Originalserver beantwortet wird, findet gegebenenfalls dynamische Inhaltsadaption statt. Außerdem muss eine Anfrage bei dieser Variante eindeutig einer Repräsentation zuzuordnen sein. Das bedeutet,

dass in der Anfrage eine konkrete Repräsentation spezifiziert werden muss (konkrete Anfrage), obwohl ein Client durchaus in der Lage ist, verschiedene Repräsentationen des gleichen Inhaltsobjektes, wenn auch mit unterschiedlicher Qualität, zu akzeptieren. Die Beschränkung auf statische Inhaltsadaption und konkrete Anfragen schränkt die Wiederverwendbarkeit der Replikate ein.

Wenn der Platzierungsmechanismus des A-CDN nur generische Repräsentationen der Inhaltsobjekte repliziert (Variante 2), findet Inhaltsadaption nur dynamisch, bei der Auslieferung der Inhalte an die Clients statt. Ausschließlich dynamische Inhaltsadaption verursacht Kosten und Antwortzeitverzögerungen bei jeder Anfrage. Andererseits kann in diesem Fall jedes Replikate zur Beantwortung aller Anfragen nach dem entsprechenden Inhaltsobjekt verwendet werden. Ebenso besteht keine Notwendigkeit, dass in einer Anfrage explizit eine konkrete Repräsentation spezifiziert wird. Wenn in einer Anfrage nur das geforderte Inhaltsobjekt und die Eigenschaften und Fähigkeiten des Client angegeben werden (generische Anfrage), kann das A-CDN die unter den gegebenen Bedingungen (Eigenschaften und Fähigkeiten des Client, Platzierung der Replikate, Systemzustand) optimale Repräsentation ermitteln und an den Client ausliefern.

Aufgrund der genannten Probleme beim Einsatz herkömmlicher Platzierungsmechanismen in A-CDNs wurden in der vorliegenden Arbeit adaptivitätssensitive Mechanismen zur Platzierung von Replikaten in A-CDNs vorgeschlagen. Adaptivitätssensitive Platzierungsmechanismen berücksichtigen die Adaptierbarkeit der Inhalte bei der Bestimmung einer Platzierung von Replikaten. Sie können verschiedene Repräsentationen (generische, teilweise und vollständig adaptierte) eines Inhaltsobjektes replizieren, Anfragen mit verschiedenen Repräsentationen beantworten und Repräsentationen zur Beantwortung einer Anfrage durch dynamische Inhaltsadaption in andere Repräsentationen überführen. Das heißt, adaptivitätssensitive Platzierungsmechanismen unterstützen generische Anfragen und können statische und dynamische Inhaltsadaption flexibel miteinander kombinieren.

Das Ziel der Arbeit war, die Überlegenheit des adaptivitätssensitiven Ansatzes gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen in A-CDNs in der Praxis zu evaluieren. Dazu sind

- auf Grundlage eines Modells eines Systems zum adaptiven Content Networking das Optimierungsproblem der Platzierung von Replikaten in A-CDNs und das inhärente Teilproblem der Komposition von Adaptionspfaden formalisiert,
- Verfahren zur Lösung der Optimierungsprobleme entwickelt,
- in einem Simulator implementiert und
- evaluiert worden.

Die Optimierung erfolgt hinsichtlich der Kosten der Dienstleistung, der Qualität der ausgelieferten Repräsentationen der Inhaltsobjekte und der Dienstqualität der Auslieferung. Da das Problem der Platzierung von Replikaten in A-CDNs NP-schwer ist, wie in der Arbeit bewiesen wurde, sind heuristische Greedy-Algorithmen, die von einer globalen Sicht auf die Anfragemenge und den Systemzustand ausgehen, zur Bestimmung einer näherungsweise optimalen Platzierung von Replikaten entwickelt worden. Obwohl die praktische Einsetzbarkeit statisch optimierender Platzierungsalgorithmen, die eine globale Sicht auf das System voraussetzen, fraglich ist, sind in der vorliegenden Arbeit statisch optimierende Algorithmen gewählt worden, da diese erlauben, auf Grundlage nahezu optimaler Lösungen die Möglichkeiten und Grenzen des Konzeptes der adaptivitätssensitiven Platzierung abzuschätzen.

Da die Komposition von Adaptionspfaden zur dynamischen Inhaltsadaption ein inhärentes Teilproblem des Problems der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs darstellt, wurden in der Arbeit auch Algorithmen zur Komposition optimaler Adaptionspfade diskutiert. Es wurde eine Abbildung des Problems auf das Optimierungsproblem der Suche nach einem kürzesten Pfad in einem gewichteten gerichteten Graphen vorgestellt, die den Einsatz eines beliebigen Kürzester-Pfad-Algorithmus zur Komposition optimaler Adaptionspfade erlaubt.

Zur Evaluation der Vorteile des adaptivitätssensitiven Ansatzes zur Platzierung von Replikaten in A-CDNs wurde ein global im Internet verteiltes A-CDN, welches zur Auslieferung von JPEG-Bildern an mobile und stationäre Internetnutzer verwendet wird, als praktisches Einsatzszenario gewählt und in einem Simulator implementiert. Dabei wurden zwei Formen der Adaption von JPEG-Bildern berücksichtigt: (1) Skalierung (genauer gesagt, Verringerung) der räumlichen Auflösung unter Beibehaltung des Seitenverhältnisses und (2) Erhöhung der Komprimierung (verlustbehaftet) durch Requantisierung mit vermindertem JPEG-Qualitätsfaktor. Die Qualität bei der Auslieferung der JPEG-Bilder wurde durch ein mehrdimensionales Qualitätsmodell beschrieben, welches die räumliche Auflösung, den JPEG-Qualitätsfaktor und die Antwortzeit als Qualitätsdimensionen berücksichtigt. Im Simulationsmodell wurden mobile und stationäre Clients mit zehn verschiedenen Typen von Endgeräten und vier verschiedenen Typen von Netzwerkanbindungen berücksichtigt. Die Abhängigkeit der Simulationsergebnisse von den im Simulationsmodell getroffenen Annahmen wurde in Sensitivitätsanalysen untersucht.

Die Ergebnisse der Evaluation der adaptivitätssensitiven Platzierung von Replikaten bezüglich ihrer Vorteile gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen in A-CDNs lassen sich folgendermaßen zusammenfassen:

1. Adaptivitätssensitive Platzierung von Replikaten in A-CDNs ist möglich.
2. Platzierungen von Replikaten in A-CDNs, die durch adaptivitätssensitive Platzierungsmechanismen ermittelt werden, sind hinsichtlich der Zielfunktion, welche die Kosten der Dienstleistung, die Qualität der ausgelieferten Repräsentationen der Inhaltsobjekte und die Dienstqualität der Auslieferung berücksichtigt, in vielen Fällen Platzierungen, die mittels herkömmlicher Platzierungsmechanismen bestimmt werden, überlegen.
3. Der Vorteil gegenüber herkömmlichen Platzierungsmechanismen, bei denen ausschließlich statisch voradaptierte Repräsentationen repliziert werden (Klasse K/V/*), hängt stark vom Anfrageverhalten der Clients und der Speicherkapazität der Surrogate des A-CDN ab. Bei hinreichend großer Speicherkapazität bringt der adaptivitätssensitive Ansatz keine Vorteile mit sich. Ebenso ist der Vorteil vernachlässigbar, wenn Clients unterschiedlicher Clientklassen nicht auf die selben Inhaltsobjekte zugreifen.

Für das untersuchte Einsatzszenario der Verteilung von Bildern ist davon auszugehen, dass moderne Rechnersysteme hinreichend Speicherkapazität zur Verfügung stellen können, so dass sich durch die adaptivitätssensitive Platzierung von Replikaten keine signifikanten Vorteile gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen, bei denen nur statisch voradaptierte Repräsentationen repliziert werden, erzielen lassen. Bei der Auslieferung anderer Inhaltstypen, wie zum Beispiel bei Video-on-Demand-Diensten, kann der Speicherplatzbedarf in den Surrogaten des A-CDN aber deutlich größer sein, so dass in diesen Fällen der Einsatz adaptivitätssensitiver Platzierungsmechanismen gerechtfertigt ist. Ob bei steigenden Kapazitäten und damit verbundenen sinkenden Preisen von Festplattenspeichern der adaptivitätssensitive Ansatz in der Zukunft langfristig relevante Vorteile mit sich bringen wird, hängt davon ab, inwieweit neue Anwendungsbereiche (z.B. Online-Videotheken, Telelearning, Online-Spiele) und steigende Nutzeranforderungen zukünftig einen Bedarf nach immer höheren Speicherkapazitäten begründen.

4. Aufgrund der Adaptionkosten und der adaptionsbedingten Verzögerungen bei ausschließlich dynamischer Inhaltsadaption sind A-CDNs mit adaptivitätssensitiven Platzierungsmechanismen solchen mit herkömmlichen Platzierungsmechanismen, die nur generische Repräsentationen replizieren (Klasse G/A/*), hinsichtlich der Zielfunktion grundsätzlich überlegen. Der Vorteil ist umso ausgeprägter, je höher die Adaptionkosten, je größer die Verzögerungen infolge der dynamischen Inhaltsadaption und je deutlicher der Größenunterschied zwischen generischen und adaptierten Repräsentationen ist.
5. Adaptivitätssensitive Platzierungsmechanismen ermitteln theoretisch nie schlechtere Platzierungen als herkömmliche Platzierungsmechanismen. Werden, wie in der vorliegenden Arbeit, heuristische Platzierungsalgorithmen verwendet, kann der adaptivitätssensitive Ansatz aber in Ausnahmefällen den Vergleichsansätzen unterlegen sein. Der Nachteil ist jedoch vernachlässigbar.

Selbst wenn keine geeigneten Verfahren zur dynamischen Inhaltsadaption zur Verfügung stehen oder dynamische Inhaltsadaption aufgrund der hohen Rechenzeitanforderungen nicht praktikabel ist, lassen sich durch die Unterstützung generischer Anfragen bei Verwendung adaptivitätssensitiver Platzierungsmechanismen (in A-CDNs der Klasse G/V/*) Vorteile gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen realisieren.

Hinsichtlich der Komplexität des Platzierungsalgorithmus weisen adaptivitätssensitive Platzierungsmechanismen keine signifikanten Nachteile gegenüber herkömmlichen Platzierungsmechanismen, bei denen ausschließlich statisch voradaptierte Repräsentationen repliziert werden (K/V/*), auf. Die obere Schranke der Komplexität ist in beiden Fällen gleich. Wenn durch einen herkömmlichen Platzierungsmechanismus nur generische Repräsentationen in den Surrogaten des A-CDN repliziert werden (G/A/*), so ist die Komplexität des Platzierungsalgorithmus jedoch um einen Faktor $|T||M|$ niedriger. Folglich kann der Aufwand zur Berechnung einer Platzierung von Replikaten beim adaptivitätssensitiven Ansatz insbesondere, wenn sehr viele verschiedene Repräsentationen berücksichtigt werden, deutlich über dem beim Einsatz eines herkömmlichen Platzierungsmechanismus, der nur generische Repräsentationen repliziert, liegen.

Die grundlegenden Konzepte der Problemmodellierung, die entwickelten Algorithmen zur Platzierung von Replikaten und zur Komposition von Adaptionspfaden sowie das Simulationsmodell und die zentralen Ergebnisse der Evaluation wurden in verschiedenen internationalen Publikationen der Öffentlichkeit vorgestellt [BB03, BB04, Buc04].

7.2 Ausblick

Aus der vorliegenden Arbeit ergeben sich zahlreiche Anknüpfungspunkte für weiterführende Arbeiten, sowohl im Hinblick auf eine verbesserte Modellierung und weitergehende experimentelle Untersuchungen als auch hinsichtlich der Weiterentwicklung der Ansätze in Richtung praktischer Einsetzbarkeit.

7.2.1 Modellierung und experimentelle Untersuchungen

Bei der Modellierung des Problems der Platzierung von Replikaten in A-CDNs und der Komposition von Adaptionspfaden in dieser Arbeit wurden zahlreiche Einschränkungen und Vereinfachungen vorgenommen. Durch die Aufhebung solcher Einschränkungen und Vereinfachungen ergeben sich interessante Ansatzpunkte für weiterführende Arbeiten.

Einbeziehung von Strukturadaptionen. Die Modellierung der Inhaltsadaption in der vorliegenden Arbeit beschränkt sich auf Konvertierungsoperationen. Strukturadaptionen, die ein Inhaltsobjekt in mehrere unabhängige Objekte dekomponieren oder ein Objekt aus mehreren Objekten komponieren, wurden nicht betrachtet. Solche Strukturadaptionen können aber im Zusammenspiel mit der Replikation in einem A-CDN in verschiedenen Anwendungsszenarien sehr interessant sein. Zum Beispiel kann durch Strukturadaption aus einem Replikat eines hierarchisch kodierten Videostroms mit geringer Qualität und einer Erweiterungsschicht, die von einem anderen Surrogat oder dem Originalserver bezogen wird, ein Videostrom hoher Qualität komponiert und ausgeliefert werden.

Szenarien, in denen neben Konvertierungsoperationen auch Strukturadaptionen berücksichtigt werden, können jedoch nicht auf Basis des Paradigmas pfadbasierter Adaption, welches in dieser Arbeit bei der Modellierung der Inhaltsadaption zugrunde gelegt wurde, realisiert werden. Sie erfordern komplexe Adaptiongraphen, wie sie in [Mar03] und [Spr04] diskutiert werden. Während die Algorithmen zur Komposition von Adaptionspfaden dafür erweitert werden müssen, sind die Greedy-Algorithmen zur Bestimmung einer Platzierung von Replikaten im A-CDN auch anwendbar, wenn Strukturadaptionen mitberücksichtigt werden.

Berücksichtigung der Kosten zur Verteilung der Inhalte. Darüber hinaus wurden bei der Problemformulierung in der vorliegenden Arbeit die Kosten zur Verteilung der Inhalte nicht berücksichtigt. Wenn die Verteilung im Vergleich zur Auslieferung sehr selten ist, wie im beschriebenen Einsatzszenario aufgrund der seltenen Aktualisierungen von Multimediaobjekten anzunehmen ist, sind die Verteilungskosten gegenüber den Auslieferungskosten vernachlässigbar. Wenn Inhaltsobjekte jedoch regelmäßig aktualisiert werden, müssen die Kosten für die regelmäßige Verteilung der aktualisierten Inhaltsobjekte bei der Optimierung der Platzierung der Replikate berücksichtigt werden.

Bei der Verteilung werden die verschiedenen Repräsentationen der Replikate eines Inhaltsobjektes durch Inhaltsadaption aus der generischen Repräsentation des Originalreplikates gewonnen. Allerdings ist es nicht sinnvoll, jedes Replikat durch einen Adaptionspfad zu erzeugen, der vom Originalreplikat ausgeht. Wird die Repräsentation eines Replikates aus einem bereits vorher erzeugten Replikat generiert, so können dadurch die Verteilungskosten und die Last in den Originalservern reduziert werden. Die Reihenfolge, in welcher die verschiedenen Replikate auseinander und aus dem Originalreplikat erzeugt werden müssen, kann durch einen Algorithmus zur Konstruktion eines Minimalgerüsts in dem durch die Menge der Replikate erzeugten Untergraph der transitiven Hülle eines Problemgraphen, ähnlich dem in Abschnitt 4.1.1 eingeführten (jedoch ohne Berücksichtigung der Qualität der verschiedenen Repräsentationen), ermittelt werden. Die Wurzel eines solchen Minimalgerüsts ist das Originalreplikat. Die Replikate in den Surrogaten des A-CDN bilden die inneren Knoten und Blattknoten. Mit den Kanten des Minimalgerüsts werden gleichzeitig die benötigten Adaptionspfade bestimmt.

Werden bei der Beschreibung des Problems der Platzierung von Replikaten in A-CDNs die Verteilungskosten berücksichtigt, kann die Monotonieeigenschaft der Nutzwertfunktion der Greedy-Heuristik (vgl. Definition 4.1 auf Seite 92) verletzt sein, so dass die Greedy-Heuristik beliebig schlechte Lösungen liefern kann. Ist das der Fall, müssen andere Näherungsverfahren, wie zum Beispiel *Simulated Annealing* oder evolutionäre Optimierung [MF00], zum Einsatz kommen.

Kapazitätsbeschränkung für Adaptions- und Transferoperationen. Eine Vereinfachung bei der Modellierung in der vorliegenden Arbeit stellt die Annahme dar, dass Adaptions- und Transferoperationen nicht kapazitätsbeschränkt sind. In der Praxis konkurrieren parallel ausgeführte Adaptions- und Transferoperationen um endliche Ressourcen, wie die CPU-Zeit der Surrogate bzw. Adaptionsserver oder die Übertragungskapazität der Netzwerkverbindungen.

Da die zukünftigen Zeitpunkte der Anfragen bei der Ermittlung einer Platzierung von Replikaten noch nicht bekannt sind, erfordert die Berücksichtigung der Rechenkapazität und der Übertragungskapazität als begrenzte Ressourcen eine stochastische Modellierung. Ein Ansatz für eine solche stochastische Modellierung wird in [KHRR01] verwendet. In dieser Arbeit wird mittels eines stochastischen Rucksack-Modells die Bandbreite der Netzwerkanbindung eines Proxy-Caches für Multimediaströme modelliert. Diese Modellierung erlaubt, die Wahrscheinlichkeit einer Überlastung der Netzwerkanbindung vorherzusagen. Allerdings basiert die Vorhersage auf einer Modellierung des Anfragestroms als Überlagerung von Poisson-Prozessen. Diese Modellierung ignoriert jegliche zeitliche Lokalität im Anfragestrom und ist deshalb in der Praxis nur eingeschränkt anwendbar.

Betriebswirtschaftliche Hintergründe. In der vorliegenden Arbeit wird die Minimierung der Gesamtkosten für die Ausführung von Adaptionsoperationen und Netzwerktransfers als Optimierungsziel berücksichtigt. Dabei wird nicht unterschieden, wem die Kosten entstehen. In der Praxis müssen die Kosten für Adaptionsoperationen und Netzwerktransfers von verschiedenen Partnern übernommen werden. Typischerweise tragen die Nutzer die Kosten für die Netzwerkanbindung der Clients, der A-CDN-Betreiber die Kosten für die Bereitstellung von Adaptions- und Speicherkapazitäten im A-CDN sowie für die Netzwerkanbindung der Surrogate und die Inhaltsanbieter die Betriebskosten für die Originalserver und deren Netzwerkanbindung.

Für eine differenziertere Betrachtung der Kosten bei der Modellierung des Problems der Platzierung von Replikaten in A-CDNs muss ein konkretes Geschäfts- und Abrechnungsmodell

zugrunde gelegt werden. Darüber hinaus müssen die Wechselwirkungen zwischen den Kosten für die Nutzer und deren Nutzungsverhalten sowie die Auswirkungen des Nutzungsverhaltens auf die Zahlungsbereitschaft der Inhaltsanbieter als Kunden des A-CDN-Betreibers modelliert werden. Solche Betrachtungen, die die betriebswirtschaftlichen Aspekte des Betriebs von A-CDNs vertiefen, verbleiben für weiterführende Arbeiten.

Quantitative Evaluation von A-CDNs zur Echtzeitauslieferung von Videostreamen.

In den Simulationsexperimenten in dieser Arbeit wurde ein Anwendungsszenario eines A-CDN zur Auslieferung von JPEG-Bildern untersucht. Dabei hat sich gezeigt, dass aufgrund der geringen Größe von JPEG-Bildern der Speicherplatzbedarf der Surrogate so gering ist, dass davon ausgegangen werden kann, dass moderne Rechnersysteme hinreichend Speicherkapazität zur Verfügung stellen können, so dass sich durch die adaptivitätssensitive Platzierung keine Vorteile gegenüber dem Einsatz herkömmlicher Platzierungsmechanismen erzielen lassen. Der Speicherplatzbedarf zur vollständigen Replikation aller benötigten Repräsentationen von Videostreamen ist jedoch deutlich höher. Wenn aufgrund des Speicherplatzbedarf der verschiedenen Repräsentationen der Inhaltsobjekte nicht davon ausgegangen werden kann, dass hinreichend Speicherkapazität verfügbar ist, so ist der Einsatz adaptivitätssensitiver Platzierungsmechanismen gerechtfertigt. Während die Ergebnisse der Sensitivitätsanalysen diese qualitative Aussage zulassen, erlauben sie jedoch keine quantitativen Aussagen darüber, bei welchen Speicherkapazitäten bei der Auslieferung von Videostreamen welche Vorteile des adaptivitätssensitiven Ansatzes erreicht werden. Aus diesem Grund stellt die Modellierung und quantitative Evaluation von A-CDNs zur Echtzeitauslieferung von Videostreamen einen wichtigen Ansatzpunkt für weiterführende Arbeiten dar.

Die Ursache für die mangelnde Übertragbarkeit der quantitativen Ergebnisse auf A-CDNs zur Echtzeitauslieferung von Videostreamen ist insbesondere in einem unterschiedlichen Modell zur Bewertung der Qualität von Videos und der Dienstqualität der Auslieferung von Videos zu suchen. Aufgrund der langen Abspielzeiten von Videostreamen spielen die Netzwerklatenzen, die sich als Verzögerung des Startzeitpunktes der Videodarstellung bemerkbar machen, für die Dienstqualität eine geringere Rolle als bei Bildern. Im Simulationsmodell, welches den Untersuchungen in dieser Arbeit zugrunde lag, haben die Latenzen im Internet jedoch einen signifikanten Einfluss auf die Vorteile des adaptivitätssensitiven Ansatzes. Spielen Netzwerklatenzen für die Dienstqualität eine untergeordnete Rolle, so fallen auch die Vorteile des adaptivitätssensitiven Ansatzes geringer aus. Andererseits stehen die Netzwerklatenzen im Simulationsmodell nur stellvertretend für Einflussfaktoren, die durch die Netzwerkdistancen bestimmt werden. Im Fall der Echtzeitübertragung von Videostreamen wirken sich Netzwerkdistancen aber nicht nur über die Latenz auf die Dienstqualität aus. Verzögerungsschwankungen, Paketverluste und Engpässe in der Datenrate nehmen mit steigender Netzwerkdistanz ebenfalls zu und haben signifikanten Einfluss auf die Qualität bei der Echtzeitübertragung von Videos. Deshalb ist davon auszugehen, dass Netzwerkdistancen trotz der geringeren Relevanz der Verzögerungen einen großen Einfluss auf die Qualität der ausgelieferten Videos und die Dienstqualität der Echtzeitauslieferung haben. Verzögerungsschwankungen, Paketverluste und Engpässe in der Datenrate in Abhängigkeit von der Netzwerkdistanz werden durch die Modellierung von Netzwerktopologien in dieser Arbeit aber nicht erfasst. Das bedeutet, dass eine quantitative Evaluation von A-CDNs zur Auslieferung von Videostreamen eine detailliertere Modellierung des Internets erfordert.

7.2.2 A-CDNs in der Praxis

In der vorliegenden Arbeit wird das Konzept der adaptivitätssensitiven Platzierung von Replikaten in A-CDNs im Vergleich zum Einsatz herkömmlicher Platzierungsmechanismen in A-CDNs evaluiert. Die praktische Realisierung von A-CDNs mit adaptivitätssensitiven Platzierungsmechanismen ist Gegenstand weiterführender Arbeiten. Für eine praktische Realisierung sind andere Anforderungen an die Platzierungs- und Anfrageweiterleitungsmechanismen zu stellen als für die Evaluation des Konzeptes. Deshalb müssen die in dieser Arbeit entwickelten Ansätze für die praktische Realisierung von A-CDNs überarbeitet und weiterentwickelt werden.

Skalierbare Platzierungsmechanismen. In dieser Arbeit wurden auf Grundlage eines formalen Problemmodells statisch optimierende Platzierungsalgorithmen, die eine globale Sicht auf den aktuellen Systemzustand und die Anfragelast voraussetzen, entwickelt und in Simulationsexperimenten analysiert. Statisch optimierende Platzierungsalgorithmen auf Grundlage globaler System- und Lastinformationen sind in der Lage, eine näherungsweise optimale Platzierung von Replikaten in A-CDNs zu ermitteln. Deshalb erlauben sie, die Möglichkeiten und Grenzen des Konzeptes der adaptivitätssensitiven Platzierung von Replikaten in einem A-CDN abzuschätzen. Die Platzierungsalgorithmen wurden jedoch nicht für den praktischen Einsatz in A-CDN-Implementierungen entwickelt. Sie skalieren nicht mit Einsatzszenarien, in denen viele tausend heterogene Clients auf Millionen von Inhaltsobjekten zugreifen, die in mehreren hundert oder gar mehreren tausend Surrogaten eines A-CDNs repliziert werden können. Da sich der Systemzustand und die Anfragelast in der Praxis dynamisch ändern, kann darüber hinaus nicht davon ausgegangen werden, dass das Monitoring- und das Accountingsystem eines A-CDN in der Lage sind, dem Platzierungsmechanismus eine aktuelle, globale Sicht auf das System zur Verfügung zu stellen. Aufgrund dessen werden in der Praxis skalierbare heuristische Platzierungsmechanismen benötigt, die auf Grundlage von lokalen System- und Lastinformationen, gegebenenfalls ergänzt durch grobe Abschätzungen stark aggregierter globaler Informationen, eine Platzierung von Replikaten ermitteln. Die Entwicklung von solchen skalierbaren heuristischen Platzierungsmechanismen ist Gegenstand weiterführender Arbeiten.

Eine Möglichkeit, die Skalierbarkeit des Platzierungsalgorithmus durch die Beschränkung auf lokale Informationen zu verbessern, stellt die in Abschnitt 6.6.10 vorgestellte Untergliederung des A-CDN in Regionen dar. Innerhalb jeder Region wird unabhängig von den anderen Regionen eine Platzierung von Replikaten bestimmt, wobei nur Surrogate und Anfragen innerhalb der Region berücksichtigt werden.

Um die Platzierung der Replikate in einem A-CDN unmittelbar den Änderungen im Systemzustand oder der Anfragelast anzupassen, bieten sich dynamische Verfahren an. Dynamische Verfahren ermitteln nicht periodisch eine neue Platzierung, sondern überwachen kontinuierlich, inwieweit die Platzierung der Replikate den aktuellen Anforderungen entspricht und passen die Platzierung gegebenenfalls reaktiv einem sich ändernden Systemzustand oder einer sich ändernden Anfragelast an. Aus Sicht des Autors sind dafür insbesondere schwellwert-basierte Verfahren (vgl. Tab. 2.1 auf Seite 25) geeignet, wie sie zum Beispiel in [RA99] vorgeschlagen werden. Diese Verfahren sind einfach zu implementieren und gut skalierbar. Außerdem können über den Nutzwert der Replikate in schwellwert-basierten Verfahren neben den Anfrageraten die Kosten und die Qualität bei der Beantwortung der Anfragen berücksichtigt werden.

Skalierbare Anfrageweiterleitungsmechanismen. Genauso wie skalierbare heuristische Platzierungsmechanismen werden für die praktische Umsetzung von A-CDNs auch skalierbare Realisierungen der Anfrageweiterleitungsmechanismen benötigt.

Prinzipiell sind die in dieser Arbeit vorgeschlagenen Algorithmen zur Komposition von Adaptionspfaden auch in der Praxis einsetzbar. Allerdings kann in der Praxis nicht von einer globalen Kenntnis des Systemzustandes ausgegangen werden. Dieses Problem kann ebenfalls durch die Beschränkung auf lokale Systeminformationen, wie zum Beispiel durch die Untergliederung des A-CDN in Regionen, die zur Verbesserung der Skalierbarkeit der Platzierungsalgorithmen vorgeschlagen wurde, adressiert werden.

Architekturdetails. Neben der Entwicklung von skalierbaren Platzierungs- und Anfrageweiterleitungsmechanismen sind für eine praktische Realisierung weitere Details in Bezug auf die Architektur von A-CDNs zu klären. Diese betreffen insbesondere die Gestaltung des Monitoring-, des Accounting- und des Metadaten-Verteilungssystems.

So werden Verfahren benötigt, wie die Informationen über die Eigenschaften und Fähigkeiten der Endgeräte und der Netzwerkanbindungen der Clients gesammelt, aggregiert und den Platzierungs- und Anfrageweiterleitungsmechanismen in skalierbarer Form zur Verfügung gestellt werden können. Da Informationen über die Eigenschaften und Fähigkeiten der Endgeräte und der Netzwerkanbindungen aus den Anfragen der Clients gewonnen und in das Mo-

dell des Systemzustandes integriert werden müssen, ist in A-CDNs eine enge Integration von Monitoring- und Accountingsystem angebracht. Die Frage der Gestaltung der Kooperation dieser Teilsysteme verbleibt für zukünftige Arbeiten.

Auch die Realisierung des Metadaten-Verteilsystems geht über den Fokus der vorliegenden Arbeit hinaus. Um der Forderung nach minimalen Antwortzeitverzögerungen durch den Zugriff auf Metadaten gerecht zu werden, müssen die Metadaten im A-CDN verteilt gecacht oder repliziert werden. Geeignete Mechanismen dazu können sich an Verfahren orientieren, die sich in herkömmlichen CDNs zur verteilten Replikation nicht adaptierbarer Inhalte bewährt haben. Es ist jedoch zu klären, welche Wechselwirkungen sich zwischen der Replikation der Metadaten und der Replikation der Inhalte im A-CDN ergeben. Schließlich müssen Formate und Schnittstellen zum Austausch von Metadaten definiert werden, die eine Interaktion des Metadaten-Verteilsystems mit den Originalservern der Inhaltsanbieter erlauben.

Anhang A

Korrektheit des Dijkstra-Algorithmus zur Ermittlung kürzester Pfade in einem Problemgraphen $\Gamma_{c,k}$

In Abschnitt 4.1.2 wird ein Dijkstra-Algorithmus (Algorithmus 4.1 auf Seite 80) zur Ermittlung kürzester Pfade in einem Problemgraphen $\Gamma_{c,k}$ zur Komposition optimaler Adaptionspfade vorgeschlagen. Beweise der Korrektheit des Dijkstra-Algorithmus zur Ermittlung kürzester Pfade in Graphen in der Literatur (z.B. in [CLRS01]) setzen voraus, dass die Kanten im Graphen nichtnegativ gewichtet sind. In einem Problemgraphen $\Gamma_{c,k}$ gemäß Abschnitt 4.1.1 sind jedoch alle Kanten, die vom virtuellen Startknoten $source_k$ ausgehen, negativ gewichtet. Dass der Dijkstra-Algorithmus auch in diesem Fall anwendbar ist – vorausgesetzt, dass alle übrigen Kanten nichtnegativ gewichtet sind – wird im Folgenden bewiesen.

Im Beweis wird die folgende Eigenschaft kürzester Pfade in Graphen ausgenutzt, die hier als Satz formuliert werden soll:

Satz A.1 *Wenn ein Pfad $\theta = x \xrightarrow{\xi_1} \dots \xrightarrow{\xi_i} y \xrightarrow{\xi_j} \dots \xrightarrow{\xi_k} z$ der kürzeste Pfad zwischen x und z ist ($l(\theta) = \delta(x, z)$), dann ist der Teilpfad $\theta' = x \xrightarrow{\xi_1} \dots \xrightarrow{\xi_i} y$ der kürzeste Pfad zwischen x und y ($l(\theta') = \delta(x, y)$).*

Der Beweis von Satz A.1 ist trivial: Gäbe es einen kürzeren Pfad von x nach y , wäre die Verkettung dieses Pfades mit dem Restpfad $y \xrightarrow{\xi_j} \dots \xrightarrow{\xi_k} z$ kürzer als $l(\theta)$.

Der Beweis der Korrektheit des Dijkstra-Algorithmus erfolgt durch vollständige Induktion über der Zahl der Iterationen i der **while**-Schleife in Zeile 5 des Algorithmus 4.1 auf Seite 80. Die Beweisidee basiert auf [CLRS01].

Beweis A.1 (Korrektheit von Algorithmus 4.1 auf Seite 80)

Voraussetzung:

- Alle Kanten $x \rightarrow y \in E(\Gamma_{c,k})$ mit $x \neq source_k$ sind nichtnegativ gewichtet:

$$\forall x \rightarrow y \in E(\Gamma_{c,k}). (x \neq source_k) \Rightarrow (w(x \rightarrow y) \geq 0)$$

Behauptung:

- Für alle Knoten x in der Menge $Perm$ bezeichnet d_x die Länge des kürzesten Pfades zwischen dem Startknoten $source_k$ und x :

$$\forall x \in Perm. d_x = \delta(source_k, x)$$

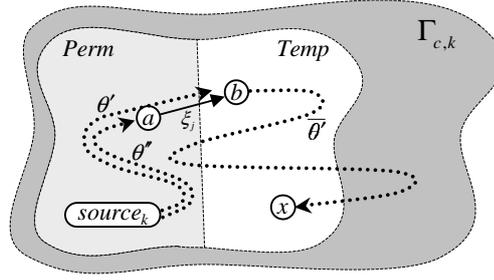


Abbildung A.1: Illustration zum Beweis der Korrektheit des Dijkstra-Algorithmus

Induktionsanfang:

- Nach 0 Iterationen ist $Perm = \emptyset$. Damit ist die Behauptung erfüllt.
- Nach einer Iteration ist $Perm = \{source_k\}$. d_{source_k} ist mit 0 initialisiert. Da $\delta(source_k, source_k) = 0$, ist die Behauptung erfüllt.

Induktionsvoraussetzung:

- Die Behauptung gilt nach der $(i - 1)$ -ten Iteration ($i > 1$).

Induktionsbehauptung:

- Die Behauptung gilt auch nach der i -ten Iteration ($i > 1$).

Induktionsschluss:

- (i) In der i -ten Iteration werde x als Arbeitsknoten gewählt.
- (ii) Es sei

$$\theta = source_k \xrightarrow{\varepsilon} \dots \xrightarrow{\xi_i} a \xrightarrow{\xi_j} b \xrightarrow{\xi_k} \dots \xrightarrow{\xi_l} x$$

ein Pfad im Problemgraphen $\Gamma_{c,k}$ mit $l(\theta) = \delta(source_k, x)$ (d.h., θ ist der kürzeste Pfad zwischen $source_k$ und x). Ferner seien

$$\begin{aligned} \theta' &= source_k \xrightarrow{\varepsilon} \dots \xrightarrow{\xi_i} a \xrightarrow{\xi_j} b \\ \theta'' &= source_k \xrightarrow{\varepsilon} \dots \xrightarrow{\xi_i} a \\ \bar{\theta}' &= b \xrightarrow{\xi_k} \dots \xrightarrow{\xi_l} x \end{aligned}$$

Teilpfade von θ mit $V(\theta'') \subseteq Perm$ und $b \notin Perm$ (d.h., $a \xrightarrow{\xi_j} b$ ist die erste Kante in θ , die aus $Perm$ herausführt; da $source_k \in Perm$ und $x \notin Perm$, muss eine solche Kante existieren). Dabei gilt $l(\theta') = l(\theta'') + w(a \xrightarrow{\xi_j} b)$ und $l(\theta) = l(\theta') + l(\bar{\theta}')$.

- (iii) Da in der i -ten Iteration x als Arbeitsknoten gewählt wird, muss $x \in Temp$ sein. Daraus folgt, dass in einer früheren Iteration ein Pfad der Länge d_x von $source_k$ nach x bestimmt worden ist. Alle Pfade zwischen $source_k$ und x haben mindestens die Länge $\delta(source_k, x)$. Somit gilt $d_x \geq \delta(source_k, x)$.
- (iv) Aus (ii) und Satz A.1 folgt, $l(\theta') = \delta(source_k, b)$ und $l(\theta'') = \delta(source_k, a)$. Demzufolge gilt $\delta(source_k, b) = \delta(source_k, a) + w(a \xrightarrow{\xi_j} b)$ und $\delta(source_k, x) = \delta(source_k, b) + l(\bar{\theta}')$.
- (v) Die Knoten des Teilpfades $\bar{\theta}'$ sind allesamt verschieden vom Knoten $source_k$. Das folgt daraus, dass jeder Knoten in $\bar{\theta}'$ mindestens eine eingehende Kante in $\Gamma_{c,k}$ hat (die

Kante, die ihn mit seinem Vorgänger in θ verbindet), $source_k$ aber gemäß der Definition in Formel 4.4 auf Seite 78 keine eingehende Kante aufweist. Da vorausgesetzt wird, dass alle Kanten, die nicht von $source_k$ ausgehen, nicht negativ gewichtet sind, gilt $\forall e \in E(\overline{\theta'}). w(e) \geq 0$. Folglich ist $l(\overline{\theta'}) \geq 0$. Im Zusammenhang mit (iv) ergibt sich $\delta(source_k, x) \geq \delta(source_k, b)$.

- (vi) In dem vorangegangenen Iterationsschritt i_a ($i_a < i$), in dem a Arbeitsknoten war und in $Perm$ aufgenommen worden ist, wurde $d_b := d_a + w(a \xrightarrow{\xi_j} b)$ gesetzt und b in $Temp$ aufgenommen. Da d_a im Iterationsschritt i_a nicht mehr verändert wird und laut Induktionsvoraussetzung nach dem Iterationsschritt i_a $d_a = \delta(source_k, a)$ gilt, ist nach dem Iterationsschritt i_a $d_b = \delta(source_k, a) + w(a \xrightarrow{\xi_j} b)$. Mit (iv) bedeutet das, dass $d_b = \delta(source_k, b)$. Da es keinen kürzeren Pfad von $source_k$ nach b gibt, bleibt d_b in den nachfolgenden Iterationen unverändert. Es gilt also auch im Iterationsschritt i : $d_b = \delta(source_k, b)$.
- (vii) Aus der Tatsache, dass im i -ten Iterationsschritt x als Arbeitsknoten gewählt wird, folgt, dass es kein $z \in Temp$ gibt, für das $d_z < d_x$ ist. Da laut (vi) $b \in Temp$ ist, folgt $d_b \geq d_x$.
- (viii) Aus (iii), (v), (vi) und (vii) folgt: $d_x \geq \delta(source_k, x) \geq \delta(source_k, b) = d_b \geq d_x$. Somit gilt: $d_x = \delta(source_k, x)$.
- (ix) Im i -ten Iterationsschritt wurde x der Menge $Perm$ hinzugefügt. Gemäß (viii) gilt $d_x = \delta(source_k, x)$. Alle anderen Knoten $z \in Perm$ waren bereits nach dem $(i-1)$ -ten Iterationsschritt Element in $Perm$. Für sie gilt gemäß der Induktionsvoraussetzung $d_z = \delta(source_k, z)$. Daraus folgt, dass nach dem i -ten Iterationsschritt die Behauptung erfüllt ist. \square

Anhang B

Monotonie der Nutzwertfunktion

In Abschnitt 4.2.3 wird die Nutzwertfunktion $benefit_{\Sigma,\Lambda}(\Delta P_R, P_{R(i-1)})$ für die Greedy-Heuristik zur adaptivitätssensitiven Platzierung von Replikaten in A-CDNs (Formel 4.43 auf Seite 93) eingeführt. Wie in [GM99] gezeigt, muss eine solche Nutzwertfunktion die Monotonieeigenschaft gemäß Definition 4.1 auf Seite 92 aufweisen, damit die Greedy-Heuristik nicht beliebig schlechte Ergebnisse liefern kann. Die Nutzwertfunktion $benefit_{\Sigma,\Lambda}(\Delta P_R, P_{R(i-1)})$ ist monoton gemäß Definition 4.1, wie im Folgenden bewiesen wird.

Beweis B.1 (Monotonie der Nutzwertfunktion $benefit_{\Sigma,\Lambda}$)

Voraussetzung:

$$benefit_{\Sigma,\Lambda}(\Delta P_R, P_R) = \frac{f(\Delta P_R \cup P_R) - f(P_{R(i-1)})}{\sum_{(k,h,(t,m)) \in \Delta P_R \cup P_R} sf(k, (t, m)) - \sum_{(k,h,(t,m)) \in P_R} sf(k, (t, m))}$$

Behauptung:

$$\forall \Delta P_R \in PR_{P_0}. \forall \Delta P'_R \in PR_{P_0}. \forall P_R \in PR_{P_0}.$$

$$benefit_{\Sigma,\Lambda}(\Delta P_R \cup \Delta P'_R, P_R) \leq \max \{benefit_{\Sigma,\Lambda}(\Delta P_R, P_R), benefit_{\Sigma,\Lambda}(\Delta P'_R, P_R)\}$$

Beweis:

(i) Aus der Voraussetzung folgt:

$$\begin{aligned} & benefit_{\Sigma,\Lambda}(\Delta P_R, P_R) \\ &= \frac{Profit_{\Sigma,\Lambda}(\Delta P_R \cup P_R) - Profit_{\Sigma,\Lambda}(P_R)}{\sum_{(k,h,(t,m)) \in \Delta P_R \cup P_R} sf(k, (t, m)) - \sum_{(k,h,(t,m)) \in P_R} sf(k, (t, m))} \\ &= \frac{\sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot (Profit_{\Sigma,(c,k)}(\Delta P_R \cup P_R) - Profit_{\Sigma,(c,k)}(P_R))}{\sum_{(k,h,(t,m)) \in \Delta P_R - P_R} sf(k, (t, m))} \end{aligned}$$

(ii) Es sei $\beta = \sum_{(k,h,(t,m)) \in \Delta P_R - P_R} sf(k, (t, m))$. Somit gilt:

$$\begin{aligned} & benefit_{\Sigma,\Lambda}(\Delta P_R, P_R) \\ &= \frac{1}{\beta} \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot (Profit_{\Sigma,(c,k)}(\Delta P_R \cup P_R) - Profit_{\Sigma,(c,k)}(P_R)) \end{aligned}$$

(iii) Mit $\beta' = \sum_{(k,h,(t,m)) \in \Delta P'_R - P_R} sf(k, (t, m))$ gilt analog:

$$\begin{aligned} & benefit_{\Sigma,\Lambda}(\Delta P'_R, P_R) \\ &= \frac{1}{\beta'} \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot (Profit_{\Sigma,(c,k)}(\Delta P'_R \cup P_R) - Profit_{\Sigma,(c,k)}(P_R)) \end{aligned}$$

(iv) Ebenfalls folgt aus der Voraussetzung:

$$\begin{aligned}
& \text{benefit}_{\Sigma, \Lambda}(\Delta P_R \cup \Delta P'_R, P_R) \\
&= \frac{\sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R))}{\sum_{(k,h,(\mathbf{t},\mathbf{m})) \in (\Delta P_R - P_R) \cup (\Delta P'_R - P_R)} sf(k, (\mathbf{t}, \mathbf{m}))} \\
&= \frac{1}{\beta + \beta'} \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R))
\end{aligned}$$

(v) Es seien Req_0 , $Req_{\Delta P_R}$, $Req_{\Delta P'_R}$, $Req_{\Delta P_R \Delta P'_R}$ und $Req_{\Delta P'_R \Delta P_R}$ disjunkte Teilmengen von $(C \times K)$ mit $Req_0 \cup Req_{\Delta P_R} \cup Req_{\Delta P'_R} \cup Req_{\Delta P_R \Delta P'_R} \cup Req_{\Delta P'_R \Delta P_R} = (C \times K)$. Dabei bezeichnet

- Req_0 alle Anfragen, die weder von der Materialisierung der Replika in ΔP_R noch der in $\Delta P'_R$ profitieren,
- $Req_{\Delta P_R}$ alle Anfragen, die von der Auswahl von ΔP_R , aber nicht von $\Delta P'_R$ profitieren,
- $Req_{\Delta P'_R}$ alle Anfragen, die von der Auswahl von $\Delta P'_R$, aber nicht von ΔP_R profitieren,
- $Req_{\Delta P_R \Delta P'_R}$ alle Anfragen, die von der Auswahl von ΔP_R und $\Delta P'_R$ profitieren, aber mehr von ΔP_R , und
- $Req_{\Delta P'_R \Delta P_R}$ alle Anfragen, die von der Auswahl von ΔP_R und $\Delta P'_R$ profitieren, aber nicht mehr von ΔP_R .

Formal sind die Mengen Req_0 , $Req_{\Delta P_R}$, $Req_{\Delta P'_R}$, $Req_{\Delta P_R \Delta P'_R}$ und $Req_{\Delta P'_R \Delta P_R}$ folgendermaßen definiert:

$$\begin{aligned}
Req_0 &= \{(c, k) \mid ((c, k) \in C \times K) \wedge \\
&\quad (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) = \text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) = \text{Profit}_{\Sigma, (c,k)}(P_R))\} \\
Req_{\Delta P_R} &= \{(c, k) \mid ((c, k) \in C \times K) \wedge \\
&\quad (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) > \text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) = \text{Profit}_{\Sigma, (c,k)}(P_R))\} \\
Req_{\Delta P'_R} &= \{(c, k) \mid ((c, k) \in C \times K) \wedge \\
&\quad (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) > \text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) = \text{Profit}_{\Sigma, (c,k)}(P_R))\} \\
Req_{\Delta P_R \Delta P'_R} &= \{(c, k) \mid ((c, k) \in C \times K) \wedge \\
&\quad (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) > \text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) > \text{Profit}_{\Sigma, (c,k)}(P_R))\} \\
Req_{\Delta P'_R \Delta P_R} &= \{(c, k) \mid ((c, k) \in C \times K) \wedge \\
&\quad (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) \geq \text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) > \text{Profit}_{\Sigma, (c,k)}(P_R))\}
\end{aligned}$$

(vi) Aus (ii) und (v) folgt:

$$\begin{aligned}
& \text{benefit}_{\Sigma, \Lambda}(\Delta P_R, P_R) \\
&= \frac{1}{\beta} \left(\sum_{(c,k) \in \text{Req}_0} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right. \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R \Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad \left. + \sum_{(c,k) \in \text{Req}_{\Delta P'_R \Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right) \\
&= \frac{1}{\beta} \left(\sum_{(c,k) \in \text{Req}_{\Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right. \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R \Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad \left. + \sum_{(c,k) \in \text{Req}_{\Delta P'_R \Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right)
\end{aligned}$$

(vii) Es sei

$$\begin{aligned}
\alpha &= \sum_{(c,k) \in \text{Req}_{\Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R \Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P'_R \Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)).
\end{aligned}$$

$$\text{Somit gilt: } \text{benefit}_{\Sigma, \Lambda}(\Delta P_R, P_R) = \frac{\alpha}{\beta}$$

(viii) Mit

$$\begin{aligned}
\alpha' &= \sum_{(c,k) \in \text{Req}_{\Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R \Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P'_R \Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)).
\end{aligned}$$

$$\text{folgt analog aus (iii) und (v): } \text{benefit}_{\Sigma, \Lambda}(\Delta P'_R, P_R) = \frac{\alpha'}{\beta'}$$

(ix) Aus (iv) und (v) folgt:

$$\begin{aligned}
& \text{benefit}_{\Sigma, \Lambda}(\Delta P_R \cup \Delta P'_R, P_R) \\
&= \frac{1}{\beta + \beta'} \sum_{(c,k) \in C \times K} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&= \frac{1}{\beta + \beta'} \left(\sum_{(c,k) \in \text{Req}_0} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right. \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R \Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad \left. + \sum_{(c,k) \in \text{Req}_{\Delta P'_R \Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup \Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right) \\
&= \frac{1}{\beta + \beta'} \left(\sum_{(c,k) \in \text{Req}_{\Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right. \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad + \sum_{(c,k) \in \text{Req}_{\Delta P_R \Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \\
&\quad \left. + \sum_{(c,k) \in \text{Req}_{\Delta P'_R \Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right) \\
&= \frac{1}{\beta + \beta'} \left(\alpha + \alpha' - \sum_{(c,k) \in \text{Req}_{\Delta P_R \Delta P'_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P'_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right. \\
&\quad \left. - \sum_{(c,k) \in \text{Req}_{\Delta P'_R \Delta P_R}} \lambda_{c,k} \cdot (\text{Profit}_{\Sigma, (c,k)}(\Delta P_R \cup P_R) - \text{Profit}_{\Sigma, (c,k)}(P_R)) \right) \\
&\leq \frac{\alpha + \alpha'}{\beta + \beta'}
\end{aligned}$$

(x) Ohne Beschränkung der Allgemeinheit sei $\text{benefit}_{\Sigma, \Lambda}(\Delta P_R, P_R) \geq \text{benefit}_{\Sigma, \Lambda}(\Delta P'_R, P_R)$.
Somit gilt:

$$\begin{aligned}
\frac{\alpha}{\beta} &\geq \frac{\alpha'}{\beta'} \\
\frac{\alpha}{\beta} \cdot \beta' &\geq \alpha' \\
\alpha + \frac{\alpha}{\beta} \cdot \beta' &\geq \alpha + \alpha' \\
\frac{\alpha}{\beta} \cdot \beta + \frac{\alpha}{\beta} \cdot \beta' &\geq \alpha + \alpha' \\
\frac{\alpha}{\beta} \cdot (\beta + \beta') &\geq \alpha + \alpha' \\
\frac{\alpha}{\beta} &\geq \frac{\alpha + \alpha'}{\beta + \beta'}
\end{aligned}$$

(xi) Aus (vii), (ix) und (x) folgt:

$$\begin{aligned} \mathit{benefit}_{\Sigma,\Lambda}(\Delta P_{\mathbb{R}} \cup \Delta P'_{\mathbb{R}}, P_{\mathbb{R}}) &\leq \frac{\alpha}{\beta} \\ \mathit{benefit}_{\Sigma,\Lambda}(\Delta P_{\mathbb{R}} \cup \Delta P'_{\mathbb{R}}, P_{\mathbb{R}}) &\leq \mathit{benefit}_{\Sigma,\Lambda}(\Delta P_{\mathbb{R}}, P_{\mathbb{R}}) \\ \mathit{benefit}_{\Sigma,\Lambda}(\Delta P_{\mathbb{R}} \cup \Delta P'_{\mathbb{R}}, P_{\mathbb{R}}) &\leq \max\{\mathit{benefit}_{\Sigma,\Lambda}(\Delta P_{\mathbb{R}}, P_{\mathbb{R}}), \mathit{benefit}_{\Sigma,\Lambda}(\Delta P'_{\mathbb{R}}, P_{\mathbb{R}})\} \end{aligned}$$

□

Anhang C

Empirische Untersuchung und Modellierung der Eigenschaften von JPEG-Bildern im WWW

Die Modellierung der Eigenschaften von JPEG-Bildern und der Adaptionsoptionen im Simulationsmodell (vgl. Kap. 5) beruhen auf empirischen Untersuchungen der Medieneigenschaften und Transcoding-Eigenschaften von JPEG-Bildern, die in einem Anfrageprotokoll des WWW-Proxy `web-cache.inf.tu-dresden.de` an der Fakultät Informatik der TU Dresden referenziert werden. Dieser Proxy wird standardmäßig von allen PCs in den Studentenkabinetten des Fakultätsrechenzentrums, verschiedenen Computern von Fakultätsmitarbeitern und einigen privaten Computern in Studentenwohnheimen verwendet.

Auf `web-cache.inf.tu-dresden.de` lief im Juni 2003, als das Anfrageprotokoll aufgezeichnet wurde, Squid/2.3 STABLE4 [Wes03]. Die Protokolldatei `access.log` des Proxy enthält für jede Anfrage einen Eintrag, der die in Tabelle C.1 aufgeführten Informationen umfasst. Für die Untersuchungen wird das Protokoll aller Anfragen vom 11.06.2003 herangezogen. Um den

Tabelle C.1: Informationen in `access.log`

Feld	Beschreibung
<code>time</code>	Zeitstempel des Eintrags
<code>duration</code>	Zeit zur Beantwortung der Anfrage
<code>client address</code>	IP-Adresse des Client
<code>result codes</code>	Cache-Erfolg (HIT, MISS ...) und HTTP-Ergebniscode
<code>bytes</code>	Umfang der Antwort (Daten + Header)
<code>request method</code>	Anfragemethode (GET, HEAD, POST ...)
URL	URL des angefragten Objektes
<code>rfc931</code>	nicht benutzt
<code>hierarchy code</code>	gibt an, wie die Anfrage beantwortet wurde (z.B. durch Weiterleitung an einen anderen Proxy oder an den Originalserver)
<code>type</code>	Inhaltstyp des Objektes gemäß HTTP-Header

Eine detaillierte Beschreibung wird in [Wes03] gegeben.

Anforderungen des Datenschutzes zu genügen, werden die IP-Adressen der Clients vor den Analysen anonymisiert und die Query-Anteile der URLs (vgl. [BLFIM98]) entfernt.

Für die Analyse werden ausschließlich Einträge herangezogen, die entweder einen Inhaltstyp `image/jpeg` im Feld `type` haben oder deren URL einen Dateinamen mit der Endung `jpg`, `JPG`, `jpeg` oder `JPEG` enthält. Darüber hinaus werden nur Einträge berücksichtigt, die Objekte außerhalb der lokalen Domänen `tu-dresden.de`, `htw-dresden.de` und `sax.de` referenzieren. Da Objekte aus den lokalen Domänen nicht gecacht werden, sind die meisten Clients ohnehin so konfiguriert, dass sie keine Anfragen nach diesen Objekten an den Proxy schicken. Außerdem ist davon auszugehen, dass das Anfrageverhalten von Mitarbeitern und Studenten für Objekte aus den lokalen Domänen von dem für Objekte im Internet abweicht. Mehrfachvorkommen von URLs in der Protokolldatei werden ebenfalls eliminiert.

C.1 Analyse der Medieneigenschaften

Da die Protokolldatei des Squid-Proxy keine Informationen über die Medieneigenschaften der referenzierten Bilder enthält, wurden alle referenzierten Bilder nochmals aus dem Internet heruntergeladen und analysiert. Diese Analyse wurde am 31.07.2003 durchgeführt. Aufgrund der Dynamik im WWW, dem Fehlen eines gültigen Cookie oder des Query-Anteils des URL beim wiederholten Herunterladen sowie durch Fehler der WWW-Server konnte nicht für jeden URL eine gültige JFIF-Datei (Austauschformat für JPEG-Bilder [Ham92]) heruntergeladen werden. Für die 27 066 verfügbaren Bilder werden bei der Analyse die folgenden Eigenschaften ermittelt:

- räumliche Auflösung (`pix-x`, `pix-y`)
- JPEG-Qualitätsfaktor
- Farbmodell

Während die räumliche Auflösung und das Farbmodell direkt aus dem Header einer JFIF-Datei ausgelesen werden können, ist der JPEG-Qualitätsfaktor nicht unmittelbar verfügbar. Dieser wird deshalb mittels des Programms `initQual` von Chandra und Ellis von der Duke University [CSE99] aus den Quantisierungsmatrizen im Header abgeschätzt.

Natürlich erlaubt dieses Verfahren nur die Bestimmung der Eigenschaften der Objektpräsentationen, die an einen Client ausgeliefert werden. Da jedoch beim Internetzugriff von Clients im Festnetz heutzutage im Allgemeinen keine dynamische Adaption von Bildern zum Einsatz kommt, ist davon auszugehen, dass diese Eigenschaften auch der Repräsentation auf dem Originalserver entsprechen.

Für die Modellierung werden die ermittelten Informationen nochmals gefiltert. So werden alle Bilder

- mit weniger als 3 Punkten in einer Dimension,
- mit mehr als 300 Punkten in der einen Dimension und 10 oder weniger Punkten in der anderen Dimension oder
- mit 1000 oder mehr Punkten in der einen Dimension und weniger als 25 Punkten in der anderen Dimension

aussortiert. Bilder mit solchen Abmessungen sind im Allgemeinen keine echten Abbildungen auf einer WWW-Seite, sondern werden als Abstandshalter, Trennlinien oder zur Hintergrundgestaltung verwendet. Für solchartige Objekte ist eine Inhaltsadaption nicht sinnvoll. Deshalb werden sie bereits bei der Modellierung der Anfragelasten ausgeschlossen. Im Ergebnis der Filterung verbleiben 21 763 JPEG-Bilder für die Modellbildung.

Die räumliche Auflösung der Bilder wird durch die Auflösung in x- (pix-x) und y-Dimension (pix-y) bestimmt. Da pix-x und pix-y miteinander korreliert sind (Abb. C.1(a)), wird die Auflösung durch das Seitenverhältnis $a = \text{pix-x}/\text{pix-y}$ und die Zahl der Bildpunkte $\text{pix} = \text{pix-x} \cdot \text{pix-y}$ der Bilder modelliert. Diese Größen sind nicht miteinander korreliert (Abb. C.1(b)) und die Auflösung lässt sich aus ihnen leicht bestimmen:

$$\text{pix-x} = \sqrt{\text{pix} \cdot a} \quad (\text{C.1})$$

$$\text{pix-y} = \sqrt{\text{pix} \cdot \frac{1}{a}} \quad (\text{C.2})$$

Die Verteilung des Seitenverhältnisses wird durch die Überlagerung von drei logarithmischen Normalverteilungen mit zwei zusätzlichen Sprüngen bei den Seitenverhältnissen 1:1 und 4:3 modelliert (siehe auch Abb. C.2(a)):

$$F(a) = 0,3 \cdot \Phi\left(\frac{\ln a - 0,77}{1,08}\right) + 0,13 \cdot \Phi\left(\frac{\ln a + 0,31}{0,11}\right) + 0,22 \cdot \Phi\left(\frac{\ln a - 0,31}{0,16}\right) + 0,23 \cdot S(a-1) + 0,12 \cdot S\left(a - \frac{4}{3}\right) \quad (\text{C.3})$$

In der Formel der Verteilungsfunktion $F(a)$ steht

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{1}{2}z^2} dz \quad (\text{C.4})$$

für die Verteilungsfunktion der normierten Normalverteilung und

$$S(x) = \begin{cases} 0 & \text{falls } x \leq 0 \\ 1 & \text{falls } x > 0 \end{cases} \quad (\text{C.5})$$

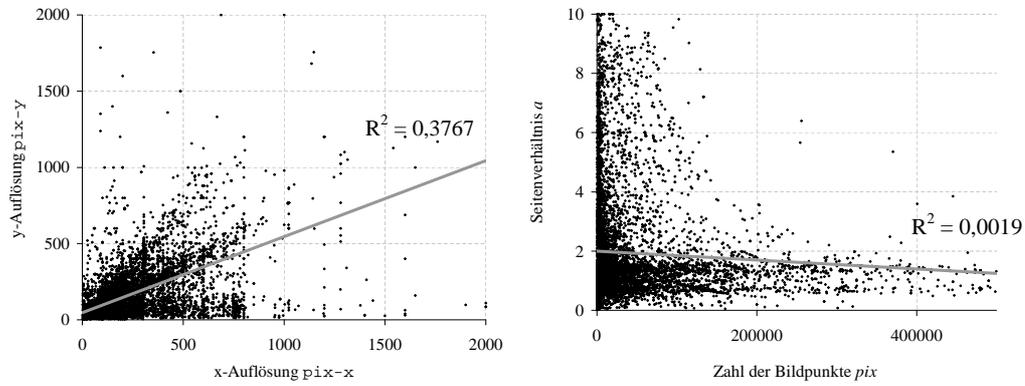
für eine Sprungfunktion mit einem 0-1-Sprung an der Stelle $x = 0$.

Die Zahl der Bildpunkte folgt einer hybriden, heavy-tailed Verteilung, deren Rumpf einer logarithmischen Normalverteilung und deren Schwanz einer Pareto-Verteilung entspricht. Hybride Lognormal/Pareto-Verteilungen werden in mehreren Arbeiten zur Modellierung von WWW-Verkehr zur Beschreibung der Verteilung der Größe von WWW-Objekten [BBBC99, CT99, BC98] und WWW-Bildern [BJS02] vorgeschlagen. Da keine Bilder mit weniger als 9 Punkten betrachtet werden, ist die Verteilung der Zahl der Bildpunkte jedoch um 9 Punkte verschoben. Der Übergangspunkt zwischen Rumpf und Schwanz der Verteilung liegt bei $\text{pix} = 84\,009$. Damit ergibt sich die folgende Verteilung der Zahl der Bildpunkte (siehe auch Abb. C.2(b)):

$$F(a) = \begin{cases} 0,88 \cdot \frac{\Phi\left(\frac{\ln(\text{pix}-9)-9,1}{1,2}\right)}{\Phi\left(\frac{\ln 84\,000-9,1}{1,2}\right)} & \text{falls } \text{pix} \leq 84\,009 \\ 1 - 0,12 \cdot \left(\frac{84\,000}{\text{pix}-9}\right)^{1,37} & \text{falls } \text{pix} > 84\,009 \end{cases} \quad (\text{C.6})$$

Die Verteilung des JPEG-Qualitätsfaktors und des verwendeten Farbmodells (Graustufen Y und Echtfarben YCbCr) sind ebenfalls bestimmt worden (Tab. C.2 auf Seite 179). Sie werden im Simulationsmodell (Kap. 5) jedoch nicht verwendet.

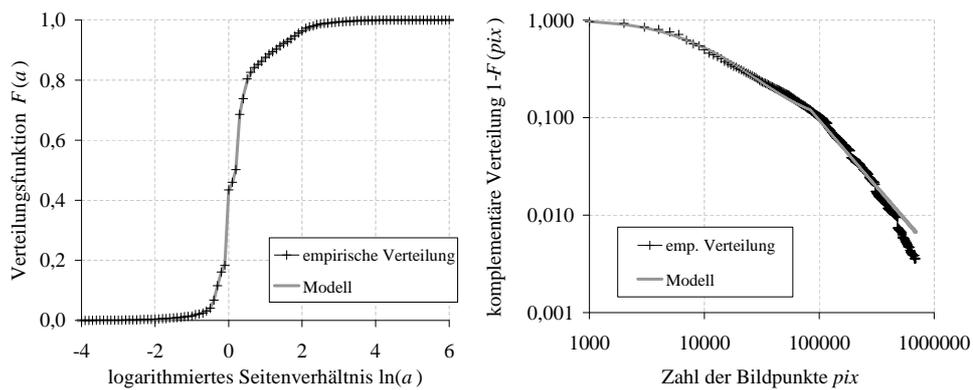
Die Popularität der JPEG-Bilder in der Protokolldatei ist Zipf-ähnlich [BCF⁺99] verteilt. Der Zipf-Parameter beträgt $z = 0,6$. Eine Korrelationsanalyse hat gezeigt, dass die Popularität nicht mit den untersuchten Medieneigenschaften Zahl der Bildpunkte, Seitenverhältnis, JPEG-Qualitätsfaktor und Farbmodell korreliert ist.



(a) Korrelation zwischen Auflösung in x-
(pix-x) und y-Dimension (pix-y)

(b) Korrelation zwischen Zahl der Bildpunkte
 pix und Seitenverhältnis a

Abbildung C.1: Korrelationsanalyse zur Modellierung der räumlichen Auflösung



(a) Seitenverhältnis a

(b) Zahl der Bildpunkte pix

Abbildung C.2: Empirische Verteilungsfunktion und Modellbildung

Tabelle C.2: Empirische Verteilung des JPEG-Qualitätsfaktors und des Farbmodells

JPEG-Qualitätsfaktor	relative Häufigkeit
0-5	0,005%
6-10	0,04%
11-15	0,27%
16-20	0,67%
21-25	0,43%
26-30	0,11%
31-35	0,06%
36-40	1,82%
41-45	0,25%
46-50	4,73%
51-55	1,69%
56-60	0,31%
61-65	6,20%
66-70	3,73%
71-75	6,92%
76-80	12,54%
81-85	24,25%
86-90	20,57%
91-95	8,48%
96-100	6,91%

(a) JPEG-Qualitätsfaktoren

Farbmodell	relative Häufigkeit
YCbCr	98,9%
Y	1,1%

(b) Farbmodelle

C.2 Transcoding-Analyse

Zur Modellierung der Adaptionoperationen im Simulationsmodell wird für eine Menge von Testbildern ermittelt, wie sich deren Medieneigenschaften, genauer gesagt deren Dateigröße, bei der Ausführung von Adaptionoperationen verändern und welche Ausführungszeiten für die Adaptionoperationen einzuplanen sind. Die Testbilder werden dazu mit verschiedenen Skalierungsfaktoren in ihrer räumlichen Auflösung skaliert und mit verschiedenen JPEG-Qualitätsfaktoren kodiert (Tab. C.3). Dabei werden jeweils nur die JPEG-Qualitätsfaktoren berücksichtigt, die kleiner als der JPEG-Qualitätsfaktor des Ausgangsbildes sind.

Tabelle C.3: Zielparameter der Adaptionoperationen in der Transcoding-Analyse

Skalierungsfaktoren:	0,95	0,85	0,75	0,65	0,55	0,45	0,35	0,25	0,15	0,05										
JPEG-Qualitätsfaktoren:	95	90	85	80	75	70	65	60	55	50	45	40	35	30	25	20	15	10	5	0

Die Testumgebung für die Transcoding-Analyse ist in Java mittels des Java Advanced Imaging Toolkit JAI 1.1.2 für Windows [oV99b] entwickelt worden. Als Skalierungsmethode wird bikubische Interpolation verwendet. Die Quantisierungsmatrizen für die JPEG-Kodierung werden entsprechend dem Vorschlag der Independent JPEG Group (IJG) [LGB⁺98] ermittelt. Hierzu wird zunächst der JPEG-Qualitätsfaktor jqf in einen Skalierungsfaktor scl umgerechnet:

$$scl = \begin{cases} \left\lceil \frac{5000}{jqf} \right\rceil & \text{falls } jqf \geq 50 \\ 200 - 2 \cdot jqf & \text{sonst} \end{cases} \quad (\text{C.7})$$

Mittels dieses Skalierungsfaktors werden anschließend aus den Standard-Quantisierungsmatrizen ($sqy_{u,v}$) und ($sqc_{u,v}$) die Quantisierungsmatrizen ($qy_{u,v}$) und ($qc_{u,v}$) zur Quantisierung der DCT-Koeffizienten für Helligkeits- bzw. Farbwerte ermittelt:

$$qy_{u,v} = \frac{scl \cdot sqy_{u,v} + 50}{100} \quad (\text{C.8})$$

$$qc_{u,v} = \frac{scl \cdot sqc_{u,v} + 50}{100} \quad (\text{C.9})$$

Die Eigenschaften der Testplattform sind in Tabelle C.4 zusammengefasst. Bei der Ausführung der Adaption werden jeweils die Skalierungs- und Kodierungszeit gemessen und die Größe des kodierten JPEG-Bildes bestimmt. Zur Ermittlung der Zeiten zum Dekodieren bei verschiedenen Auflösungen und JPEG-Qualitätsfaktoren werden die Zielbilder im Anschluss erneut

Tabelle C.4: Eigenschaften der Testplattform

CPU	AthlonXP 1600+
Hauptspeicher	1 GB
Betriebssystem	Windows 2000 Professional
Java VM	Java 2 Runtime Environment, Standard Edition (build 1.4.2_01-b06) Java HotSpot(TM) Client VM (build 1.4.2_01-b06, mixed mode) 64 MB Heap-Speicher

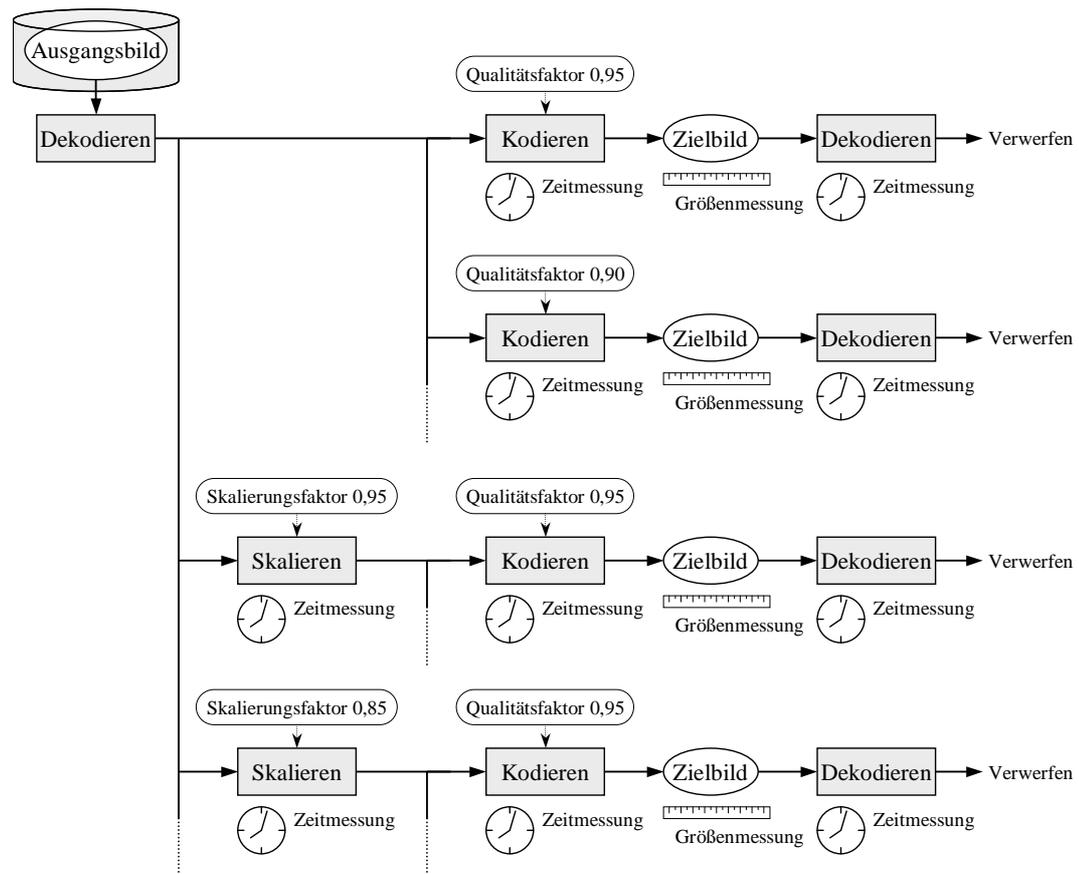


Abbildung C.3: Schema der Transcoding-Analyse

dekodiert. Dieses Vorgehen ist in Abbildung C.3 schematisch dargestellt. Um Verzögerungen durch Festplattenzugriffe zu vermeiden, wird das Ausgangsbild vor Ausführung der ersten Operation geladen, dekodiert und im Hauptspeicher gepuffert. Ebenso werden das Zielbild und alle Zwischenergebnisse nicht auf die Festplatte geschrieben, sondern im Hauptspeicher gepuffert. Um zufällige Fehler und Schwankungen bei der Zeitmessung auszuschließen, werden Zeitmessungen grundsätzlich fünfmal durchgeführt, anschließend der größte Messwert verworfen und der Mittelwert \bar{X} und die Standardabweichung S über den verbliebenen Messwerten ermittelt. Ist $S \geq 0,1 \cdot \bar{X}$, so werden die Ergebnisse verworfen und die Messungen wiederholt, andernfalls wird der Mittelwert \bar{X} als Messwert der Zeitmessung für die anschließende Modellbildung übernommen.

Für die Analyse werden aus den JPEG-Bildern, die im Rahmen der Analyse der Medien-eigenschaften (Anhang C.1) heruntergeladen worden sind, zufällig gleichverteilt 83 Testbilder ausgewählt, die eine Auflösung von

- mindestens 100 Punkten in der x-Dimension oder
- mindestens 80 Punkten in der y-Dimension

aufweisen. Bilder geringerer Auflösung werden nicht berücksichtigt. Bei diesen ist eine Skalierung um die in Tabelle C.3 genannten Skalierungsfaktoren nicht sinnvoll. Die generierten Repräsentationen wären teilweise sehr klein und die verschiedenen Repräsentationen würden

sich nur unwesentlich voneinander unterscheiden. Zudem müssen Bilder mit geringerer Auflösung für die im Simulationsmodell berücksichtigten Endgeräte (vgl. Tab. 5.3 auf Seite 104) ohnehin nicht skaliert werden.

Aus den Messergebnissen werden mittels der Methode der kleinsten Quadrate die in Tabelle C.5(a) dargestellten Modelle für die Dateigröße und die Ausführungszeiten für die einzelnen Teiloperationen der Adaption ermittelt. Die Güte der Modelle ist in Tabelle C.5(b) illustriert.

Tabelle C.5: Ergebnisse der Transcoding-Analyse

	Bildgröße [Bytes]
Zielbild	$\mathbf{size} := \lfloor p_{1k} + p_{2k} \cdot blk^{p_{3k}} + (e^{2.53 - 0.0127 \cdot scl} - 1) \cdot (p_{4k} - p_{5k} \cdot e^{-p_{6k} \cdot blk}) \rfloor$
Teiloperation	Ausführungszeit [ms]
Dekodieren	$t_{dec} = 1,54 + 0,015 \cdot blk^{1.07}$
Skalieren	$t_{scl} = 0,80 + 0,036 \cdot blk^{0.92}$
Kodieren	$t_{enc} = 0,68 + 0,015 \cdot blk^{1.07}$

blk = Zahl der Blöcke von 8×8 Bildpunkten (im Zielbild)

scl = Skalierungsfaktor für JPEG-Quantisierung (vgl. Formel C.7 auf Seite 180)

p_{1k}, \dots, p_{6k} = bildspezifische Kodierungsparameter (vgl. Tab. C.6)

(a) Modelle

	Bestimmtheit	absoluter Fehler		relativer Fehler	
		99%-Quantil	95%-Quantil	99%-Quantil	95%-Quantil
Bildgröße	0,99352	381,6 Bytes	189,8 Bytes	3,5%	7,5%
Dekodierzeit	0,98897	2,4268 ms	0,8445 ms	21,2%	16,5%
Skalierungszeit	0,96705	3,2765 ms	1,3865 ms	105,6%	86,5%
Kodierzeit	0,99423	2,3229 ms	0,6332 ms	17,6%	13,0%

(b) Güte der Modelle

Tabelle C.6: JPEG-Kodierungsparameter $p_{1k}, p_{2k}, p_{3k}, p_{4k}, p_{5k}, p_{6k}$ der analysierten Bilder

Nr.	p_{1k}	p_{2k}	p_{3k}	p_{4k}	p_{5k}	p_{6k}
1	252,032	125,886411	0,566515	540,0908387	532,1165533	0,003043134
2	294,104	66,69766868	0,69069	2790,349007	2737,451208	0,000582935
3	327,896	98,7419617	0,581028	532,080137	538,0613303	0,004869151
4	378,905	69,4335375	0,665486	688,3101498	679,3745195	0,003187435
5	383,157	84,05572737	0,592125	348,6671307	344,0140675	0,005351559
6	397,054	33,73345247	0,833261	19310,10124	19274,00451	0,000127417
7	397,399	36,25836089	0,676833	2366,930604	2335,386195	0,000245703
8	405,311	75,13526331	0,585315	17602,73486	17578,16087	7,40952E-05
9	409,963	47,02267819	0,682384	489,2112783	508,3766198	0,003997957
10	418,544	37,33346092	0,75301	4333,32448	4279,162323	0,000392794
11	436,47	73,56136825	0,579499	285,8987596	281,7281363	0,007717774
12	448,046	39,3470105	0,711522	1748,847552	1704,441126	0,000601104
13	451,122	35,03438473	0,680394	2095,906918	2047,863945	0,000371064
14	455,266	52,84483616	0,674871	406,5772049	406,1296935	0,005595119
15	462,894	85,86531809	0,560445	179,2365417	185,041949	0,013884064
16	474,172	39,42499476	0,738748	2396,533269	2379,207077	0,000767875
17	477,396	22,58689792	0,805075	5525,000646	5415,456998	0,000219728
18	500,318	39,44865686	0,698727	656,4360023	665,6690004	0,002196029
19	508,88	37,54649381	0,792389	974,2658514	970,4622948	0,003078496
20	510,417	27,11182555	0,764179	2932,397349	2877,688224	0,000500109
21	514,227	12,51590205	0,891129	5143,082846	5137,96595	0,000242087
22	516,371	41,82816849	0,700358	953,2528668	930,4260045	0,001622998
23	518,333	22,14518298	0,808679	3697,161473	3649,562406	0,000393279
24	521,473	37,12534908	0,603966	78,28750593	102,5809815	0,01088841
25	529,285	47,64558638	0,69555	584,3280849	573,5772063	0,003287982
26	530,653	33,20930142	0,75573	1539,992031	1523,297315	0,00104409
27	533,016	22,0633972	0,779448	1743,735608	1715,418839	0,000633378
28	534,093	51,47364865	0,714847	376,0233432	395,5244282	0,008954078
29	535,616	53,9933019	0,685265	610,4284568	596,997276	0,002933132
30	537,724	44,44489023	0,746227	959,8961318	979,2554624	0,003113311
31	539,094	34,05578788	0,751458	632,5792345	621,6284399	0,003575029
32	539,707	48,35985368	0,702256	28685814,81	28685793,28	5,65441E-08
33	547,238	36,27250441	0,658043	58015056,73	58015022,9	1,44576E-08
34	547,99	33,16682071	0,812056	2415,681749	2397,813479	0,001052288
35	548,835	37,1554328	0,691826	1097,83496	1081,098087	0,001311949
36	556,732	33,74424891	0,722795	294,7392101	302,7152367	0,009165547
37	557,786	44,47245461	0,715887	298,1025275	297,931189	0,011087114
38	559,224	32,99843051	0,776701	710,0287868	693,133451	0,002637018
39	561,104	21,72144454	0,820582	4830,729485	4818,955829	0,000388154
40	561,65	24,94042547	0,852764	4213,327198	4225,583136	0,000622992
41	563,294	29,79230087	0,812157	463,8063497	466,46013	0,006489265

Fortsetzung auf der folgenden Seite

Fortsetzung von Tabelle C.6

Nr.	p_{1k}	p_{2k}	p_{3k}	p_{4k}	p_{5k}	p_{6k}
42	564,121	29,76817888	0,730757	149800152,1	149800138,1	8,06418E-09
43	564,26	34,2566258	0,7201	261,4779005	268,6669613	0,010581207
44	569,106	35,66857325	0,734299	232,4193292	230,7725401	0,007947942
45	570,529	31,83448066	0,764902	343,8864874	345,838724	0,006708822
46	571,179	31,62000532	0,786706	121620088,2	121620068,5	1,37114E-08
47	573,099	25,96527322	0,79566	1167,960302	1154,707738	0,001610337
48	573,964	35,16425206	0,784669	307,7575544	309,3683229	0,009726974
49	577,651	30,15558058	0,741798	276,2513709	272,8801125	0,008686019
50	577,657	26,21600232	0,75746	280,434204	279,1285841	0,007271932
51	582,521	34,41663421	0,789409	797,8700151	790,5054497	0,003035723
52	585,028	29,43634102	0,816409	966,4246957	959,5769888	0,002866319
53	585,296	34,89766774	0,666359	646,1432753	629,5797609	0,002036901
54	586,569	24,13589134	0,77043	451,8505855	442,9525043	0,004129524
55	587,176	27,36897301	0,676263	33642909,35	33642898,47	3,94174E-08
56	587,385	26,74720548	0,703304	463,082637	453,1859608	0,003282822
57	592,207	23,90768607	0,77297	205,2055186	206,783541	0,011515511
58	592,397	21,84954287	0,752499	748,557266	731,920975	0,001656686
59	594,555	21,12379225	0,812062	518,5768264	516,1320019	0,004185705
60	596,166	25,46971247	0,822531	1088,365	1075,630791	0,002330678
61	596,942	29,13703368	0,844832	1894,344638	1884,75393	0,001496197
62	599,637	6,713899972	0,76333	81,08109565	88,67983734	0,00389685
63	602,608	21,09107573	0,851008	778,0335532	761,4180505	0,002493813
64	603,084	13,94400965	0,8703	461,1911072	455,489448	0,004107538
65	603,911	14,80611099	0,835007	851,0033923	858,7111055	0,002144103
66	606,273	26,9728283	0,792497	83688556,07	83688535,16	1,70932E-08
67	606,384	26,13172256	0,788628	745,8542423	741,8021405	0,002825199
68	607,994	21,68303158	0,803456	86151050,9	86151036,36	1,52651E-08
69	608,939	24,01671389	0,857615	630,570307	626,7843554	0,004499253
70	612,844	19,08838433	0,870195	1120,544509	1108,765541	0,001972903
71	614,439	19,05024572	0,891392	2806,262345	2821,891124	0,001113828
72	614,901	17,12946363	0,690184	35,64710613	37,74913459	0,049670634
73	617,026	21,86571751	0,792692	463,5550331	452,7511109	0,003722972
74	618,886	16,51286873	0,800153	1582,119265	1571,463376	0,00082672
75	622,637	17,21239907	0,913943	3140,641871	3143,092764	0,000910283
76	625,221	21,09656013	0,79988	82690292,06	82690274,28	1,72775E-08
77	626,243	19,57865613	0,88185	66307444,72	66307432,33	2,81633E-08
78	627,726	13,37650754	0,793331	71,79518667	79,92770595	0,024502035
79	629,901	28,83038887	0,859961	2455,668056	2446,913101	0,001279918
80	630,692	12,93141989	0,873887	516,3583	510,0213312	0,003763365
81	631,586	11,43169773	0,939285	4511,509739	4514,783668	0,00051039
82	632,32	16,00018045	0,873615	424,7970731	424,8484642	0,005890375
83	635,158	9,890948045	0,959373	14617249,83	14617244,95	1,20997E-07

Anhang D

CPU-Kosten verschiedener Rechenzentren

Die Wahl geeigneter CPU-Kosten für die Abbildung von Adaptionszeiten auf Adaptionskosten in Abschnitt 5.4 beruht auf einer WWW-Recherche von CPU-Kosten, die bei der Nutzung von Rechenkapazitäten in verschiedenen Rechenzentren berechnet werden. Die Ergebnisse dieser Recherche sind in Tabelle D.1 zusammengefasst. Für das Simulationsmodell wurden in Anlehnung an Nr. 1, 2 und 6 CPU-Kosten in Höhe von 5 EUR/h gewählt, weil dort Plattformen zugrunde liegen, deren Prozessorleistung pro Prozessor in der Größenordnung der Prozessorleistung des Rechners liegen, mit dessen Hilfe die Transcoding-Analyse (Anhang C.2) durchgeführt wurde, die die Grundlage für die Modellierung der Adaptionszeiten bildet.

Tabelle D.1: CPU-Kosten verschiedener Rechenzentren

Nr.	Organisation	Plattform	Preis pro Stunde	Quelle
1	Ohio Supercomputer Center	Intel 32-Bit Cluster	5 USD	http://oscinfo.osc.edu/software/general.shtml
2	Ohio Supercomputer Center	Intel 64-Bit Cluster	5 USD	http://oscinfo.osc.edu/software/general.shtml
3	Ohio Supercomputer Center	SunFire 6800	10 USD	http://oscinfo.osc.edu/software/general.shtml
4	Ohio Supercomputer Center	Cray SV1	50 USD	http://oscinfo.osc.edu/software/general.shtml
5	Univ. of Washington, Department of Atmospheric Sciences	Sun SPARC	20 USD	http://www.atmos.washington.edu/~neallog/Accounting.html
6	Univ. of Washington, IT Services	UNIX Workstation (nicht näher spezifiziert)	6 USD	http://www.washington.edu/computing/unix/rates.html
7	Univ. of Edinburgh, HPCx Ltd.	IBM POWER4+ Regatta Knoten	0,785 GBP	http://www.hpcx.ac.uk/services/policies/charging_model.html
8	University of Texas, IT Services	Cray J916	18 USD (Sep. 1995)	http://www.utexas.edu/cc/newsletter/sep95/craycharges.html

Anhang E

Verteilung der Anfragen auf die Clients

Die Lastmodelle, die den Simulationsexperimenten zugrunde liegen (vgl. Abschnitt 5.5.2), basieren auf Annahmen über den Anteil der Anfragen, die von den verschiedenen Clients mit unterschiedlichen Eigenschaften der Endgeräte und Netzwerkanbindungen initiiert werden. In Tabelle 5.8 auf Seite 116 ist eine grobe Abschätzung eines möglichen realen Anfrageverhaltens angegeben. Da empirische Daten zum Anfrageverhalten zukünftiger mobiler und stationärer Internetnutzer nicht verfügbar sind, kann die Anfrageverteilung in der Praxis von dieser Abschätzung abweichen. Um den Einfluss einer abweichenden Anfrageverteilung auf die Simulationsergebnisse einschätzen zu können, werden auch Experimente mit verschiedenen alternativen Verteilungen der Anfragen auf die Clients durchgeführt. Es werden die folgenden Alternativen betrachtet:

- \mathcal{V}_1 : Der Anteil der Anfragen von Clients im Festnetz beträgt 60%. Lediglich 40% der Anfragen werden von mobilen Clients (in zellularen Mobilfunknetzen und an WLAN-Hotspots) initiiert (Tabelle E.1 auf Seite 188).
- \mathcal{V}_2 : Anfragen werden zu gleichen Teilen von mobilen Clients in zellularen Mobilfunknetzen, von mobilen Clients an WLAN-Hotspots und von stationären Clients im Festnetz initiiert. Somit steigt der Anteil der Anfragen von Clients an WLAN-Hotspots gegenüber der Verteilung in Tabelle 5.8 auf Seite 116 von 20% auf 33,3%. Das Verhältnis der Anfragen von mobilen Clients in zellularen Mobilfunknetzen gegenüber denen von stationären Clients im Festnetz bleibt 1:1 (Tabelle E.2 auf Seite 188).
- \mathcal{V}_3 : Der Anteil der Anfragen von Clients in zellularen Mobilfunknetzen, die eine sehr schmalbandige Anbindung mit hohen Kommunikationskosten haben, beträgt 60%. Das Verhältnis der Anfragen von mobilen Clients an WLAN-Hotspots und stationären Clients im Festnetz bleibt unverändert bei 1:2 (Tabelle E.3 auf Seite 189).
- \mathcal{V}_4 : Die Anfragen verteilen sich gleichmäßig auf alle Endgerätetypen. Somit initiieren die Clients jedes Endgerätetyps 10% aller Anfragen (Tabelle E.4 auf Seite 189).

Zusätzlich werden Experimente durchgeführt, bei denen die Zahl der Endgerätetypen und damit die Heterogenität der Clientpopulation eingeschränkt ist. Dabei werden nur 4 verschiedene Typen von Endgeräten betrachtet, wobei pro Clientklasse nur ein Endgerätetyp berücksichtigt wird. Die Anfragen verteilen sich in diesem Fall entsprechend dem Modell \mathcal{V}_5 in Tabelle E.5 auf Seite 190.

Tabelle E.1: Alternative Anfrageverteilung \mathcal{V}_1 (mehr Anfragen aus dem Festnetz)

1 Endgerät (Displayauflösung)	2 Zugangspunkt	3 Netzwerk- verbindung	4 Client- klasse	5 Anteil am Anfragevolumen		
			i	σ_i	τ_d	
Smart Phone, niedrige Auflösung (101x80)	Mobilfunk- Basisstation	GPRS	1	26,7%	13,3%	4,44%
Smart Phone, mittlere Auflösung (128x160)						4,44%
Smart Phone, hohe Auflösung (176x208)						4,44%
Palm PDA, niedrige Auflösung (160x160)			2		2,22%	
Palm PDA, hohe Auflösung (320x320)					1,11%	
Pocket PC PDA (240x320)					3,78%	
Handheld PC PDA (640x240)			3		0,53%	
Sub-Notebook (800x600)					0,80%	
Laptop (1024x768)					1,33%	
Palm PDA, niedrige Auflösung (160x160)	WLAN- Hotspot	IEEE 802.11b	4	13,3%	6,67%	2,07%
Palm PDA, hohe Auflösung (320x320)					1,07%	
Pocket PC PDA (240x320)			5		3,53%	
Handheld PC PDA (640x240)					1,33%	
Sub-Notebook (800x600)					2,00%	
Laptop (1024x768)					3,33%	
Arbeitsplatzcomputer (1024x768)	Festnetz- Einwahlpunkt	ISDN	6	60,0%	60,0%	30,0%
Arbeitsplatzcomputer (1024x768)		DSL				30,0%

Tabelle E.2: Alternative Anfrageverteilung \mathcal{V}_2 (mehr Anfragen von WLAN-Clients)

1 Endgerät (Displayauflösung)	2 Zugangspunkt	3 Netzwerk- verbindung	4 Client- klasse	5 Anteil am Anfragevolumen		
			i	σ_i	τ_d	
Smart Phone, niedrige Auflösung (101x80)	Mobilfunk- Basisstation	GPRS	1	33,3%	16,7%	5,56%
Smart Phone, mittlere Auflösung (128x160)						5,56%
Smart Phone, hohe Auflösung (176x208)						5,56%
Palm PDA, niedrige Auflösung (160x160)			2		4,17%	
Palm PDA, hohe Auflösung (320x320)					13,3%	2,08%
Pocket PC PDA (240x320)					7,08%	
Handheld PC PDA (640x240)			3		0,67%	
Sub-Notebook (800x600)					3,33%	1,00%
Laptop (1024x768)					1,67%	
Palm PDA, niedrige Auflösung (160x160)	WLAN- Hotspot	IEEE 802.11b	4	33,3%	16,7%	5,17%
Palm PDA, hohe Auflösung (320x320)					2,67%	
Pocket PC PDA (240x320)			5		8,83%	
Handheld PC PDA (640x240)					3,33%	
Sub-Notebook (800x600)					16,7%	5,00%
Laptop (1024x768)					8,33%	
Arbeitsplatzcomputer (1024x768)	Festnetz- Einwahlpunkt	ISDN	6	33,3%	33,3%	16,7%
Arbeitsplatzcomputer (1024x768)		DSL				16,7%

Tabelle E.3: Alternative Anfrageverteilung \mathcal{V}_3 (mehr Anfragen von Clients in zellularen Mobilfunknetzen)

1 Endgerät (Displayauflösung)	2 Zugangspunkt	3 Netzwerk- verbindung	4 Client- klasse	5 Anteil am Anfragevolumen				
			i	σ_i	τ_d			
Smart Phone, niedrige Auflösung (101x80)	Mobilfunk- Basisstation	GPRS	1	60,0%	30,0%	10,0%		
Smart Phone, mittlere Auflösung (128x160)						10,0%		
Smart Phone, hohe Auflösung (176x208)						10,0%		
Palm PDA, niedrige Auflösung (160x160)			2		13,3%	24,0%	7,50%	
Palm PDA, hohe Auflösung (320x320)							3,75%	
Pocket PC PDA (240x320)							12,7%	
Handheld PC PDA (640x240)			3			26,7%	6,00%	1,20%
Sub-Notebook (800x600)								1,80%
Laptop (1024x768)								3,00%
Palm PDA, niedrige Auflösung (160x160)	WLAN- Hotspot	IEEE 802.11b	4	13,3%			6,67%	2,07%
Palm PDA, hohe Auflösung (320x320)								1,07%
Pocket PC PDA (240x320)								3,53%
Handheld PC PDA (640x240)			5		6,67%		1,33%	
Sub-Notebook (800x600)							2,00%	
Laptop (1024x768)							3,33%	
Arbeitsplatzcomputer (1024x768)	Festnetz- Einwahlpunkt	ISDN	6	26,7%	26,7%	13,3%		
Arbeitsplatzcomputer (1024x768)		DSL				13,3%		

Tabelle E.4: Alternative Anfrageverteilung \mathcal{V}_4 (gleichmäßige Verteilung über die verschiedenen Endgerätetypen)

1 Endgerät (Displayauflösung)	2 Zugangspunkt	3 Netzwerk- verbindung	4 Client- klasse	5 Anteil am Anfragevolumen				
			i	σ_i	τ_d			
Smart Phone, niedrige Auflösung (101x80)	Mobilfunk- Basisstation	GPRS	1	60%	30%	10%		
Smart Phone, mittlere Auflösung (128x160)						10%		
Smart Phone, hohe Auflösung (176x208)						10%		
Palm PDA, niedrige Auflösung (160x160)			2		30%	15%	5%	
Palm PDA, hohe Auflösung (320x320)							5%	
Pocket PC PDA (240x320)							5%	
Handheld PC PDA (640x240)			3			10%	15%	5%
Sub-Notebook (800x600)								5%
Laptop (1024x768)								5%
Palm PDA, niedrige Auflösung (160x160)	WLAN- Hotspot	IEEE 802.11b	4	30%			15%	5%
Palm PDA, hohe Auflösung (320x320)								5%
Pocket PC PDA (240x320)								5%
Handheld PC PDA (640x240)			5		15%		5%	
Sub-Notebook (800x600)							5%	
Laptop (1024x768)							5%	
Arbeitsplatzcomputer (1024x768)	Festnetz- Einwahlpunkt	ISDN	6	10%	10%	5%		
Arbeitsplatzcomputer (1024x768)		DSL				5%		

Tabelle E.5: Alternative Anfrageverteilung \mathcal{V}_5 (eingeschränkte Zahl von Endgerätetypen)

1	2	3	4	5	6	7
Endgerät (Displayauflösung)	Zugangspunkt	Netzwerk- verbindung	Client- klasse	Anteil am Anfragevolumen		
			i		σ_i	τ_d
Smart Phone, mittlere Auflösung (128x160)	Mobilfunk- Basisstation	GPRS	1	40%	20%	20%
Pocket PC PDA (240x320)			2		16%	16%
Laptop (1024x768)			3		4%	4%
Pocket PC PDA (240x320)	WLAN- Hotspot	IEEE 802.11b	4	20%	10%	10%
Laptop (1024x768)			5		10%	10%
Arbeitsplatzcomputer (1024x768)	Festnetz- Einwahlpunkt	DSL	6	40%	40%	40%

Abkürzungen

Abb.	Abbildung
Abs.	Abschnitt
A-CDN	Adaptives Content Distribution Network
ADN	Application Delivery Network
Alg.	Algorithmus
AS	Autonomes System (eine Menge von Teilnetzen im Internet, welche als Einheit administriert werden)
BGP	Border Gateway Protocol (Verbindungsschichtprotokoll zum Routing zwischen AS)
bzw.	beziehungsweise
ca.	circa
CDI	Content Distribution Internetworking (Arbeitsgruppe der IETF)
CDN	Content Distribution Network
d.h.	das heißt
DNS	Domain Name System (Namensdienst im Internet)
entspr.	entsprechend
EUR	Euro
GB	Gigabyte (2^{30} Bytes)
ggf.	gegebenenfalls
GPRS	General Packet Switched Radio (GSM-Erweiterung für paketvermittelte Datenkommunikation)

GSM	Global System for Mobile Communications (Europäischer Standard für digitale zellulare Mobilkommunikation; GSM umfasst Sprach- und Datenkommunikation)
HSCSD	High Speed Circuit Switched Data (GSM-Erweiterung für leitungsvermittelte Datenkommunikation mit erhöhten Datenraten bis zu 57,6 kbit/s durch Kanalbündelung)
HTTP	Hypertext Transfer Protokoll (Anwendungsschichtprotokoll für den Zugriff auf WWW-Seiten)
IEEE 802.11	Familie von Standards für drahtlose lokale Netze
IETF	Internet Engineering Task Force
insbes.	insbesondere
IP	Internet Protocol (Verbindungsschichtprotokoll im Internet)
ISP	Internet Service Provider
Kap.	Kapitel
KB	Kilobyte (2^{10} Bytes)
kbit/s	Kilobits (10^3 Bits) pro Sekunde
LAN	Local Area Network (lokales Netz)
LRU	Least Recently Used (ein Cache-Ersetzungs-Verfahren, bei dem das Objekt verdrängt wird, dessen Referenzierung am längsten zurück liegt)
MB	Megabyte (2^{20} Bytes)
Mbit/s	Kilobits (10^6 Bits) pro Sekunde
Mio.	Million
Mrd.	Milliarde
ms	Millisekunden (10^{-3} Sekunden)
MSN	Microsoft Network (ISP-Service der Microsoft Corp.)
PDA	Personal Digital Assistant
RTP	Realtime Protocol (Transportprotokoll zur Echtzeitübertragung von Medienströmen)

RTSP	Realtime Streaming Protocol (Anwendungsschichtprotokoll zur Steuerung der Echtzeitübertragung von Medienströmen)
s.	siehe
sog.	so genannt
T1	Punkt-zu-Punkt-Netzwerkanbindung mit einer Datenrate von 1544 Mbit/s
Tab.	Tabelle
TCP	Transmission Control Protocol (Transportprotokoll im Internet)
tlw.	teilweise
UMTS	Universal Mobile Telecommunication System (Mobilfunknetz der dritten Generation)
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
usw.	und so weiter
Var.	Variante
vgl.	vergleiche
vs.	versus
WAP	Wireless Application Protocol (Protokollsuite für Anwendungen auf mobilen Endgeräten)
WLAN	Wireless Local Area Network (drahtloses lokales Netz)
WWW	World Wide Web
z.B.	zum Beispiel

Symbole

a	Seitenverhältnis eines Bildes
$A_{\text{Ansatz}_1 \text{Ansatz}_2}$	relative Leistungsfähigkeit des A-CDN-Ansatzes Ansatz_1 bezüglich des A-CDN-Ansatzes Ansatz_2 (Bewertungsmetrik zur Evaluation)
$\forall x \in X. \mathcal{P}(x)$	für alle x in der Menge X gilt das Prädikat $\mathcal{P}(x)$
B	Menge der Originalserver in einem System Σ
b	ein Originalserver aus der Menge B
blk	Zahl der Blöcke von 8×8 Bildpunkten in einem Bild
C	Menge der Städte, die an eine Einrichtung anzuschließen sind (k -Median-Problem)
C	Menge der Clientknoten in einem System Σ
c	ein Clientknoten aus der Menge C
c_{cpu}	Kosten pro Zeiteinheit CPU-Nutzung
c_{ij}	Kosten zum Anschluss der Stadt j an die Einrichtung am Ort i (k -Median-Problem)
cf_o	Kostenfunktion einer Operation o
$cost_\pi$	Kosten eines Adaptionspfades π
$\mathbf{D}(f)$	Definitionsbereich der Abbildung f
$\delta(x, y)$	Distanz zwischen den Knoten x und y (Länge des kürzesten Pfades zwischen x und y)
del	Qualitätsdimension der Antwortzeit
del_{tol}	Toleranzschwelle bzgl. der Antwortzeit
$E(\Gamma)$	Menge der Kanten eines Graphen Γ
$\exists x \in X. \mathcal{P}(x)$	es existiert ein x in der Menge X , für das das Prädikat $\mathcal{P}(x)$ gilt
F	Menge der Orte, an denen eine Einrichtung eröffnet werden kann (k -Median-Problem)

Φ	Verteilungsfunktion der normierten Normalverteilung
$\Gamma_{c,k}$	Problemgraph zur Suche nach dem optimalen Adaptionpfad zur Beantwortung der Anfrage (c, k)
$\Gamma_{Req_{c,k}}$	Problemgraph zur Suche nach dem jeweils optimalen Adaptionpfad zur Beantwortung aller Anfrage in der Menge $Req_{c,k}$
H	Menge der Surrogate in einem System Σ
H_{loc}	Menge der Surrogate, die bei der verteilten Realisierung der Greedy-Algorithmen durch den lokalen Platzierungsmanager loc vertreten werden
h	ein Surrogat aus der Menge H
I	Menge der Orte, an denen eine Einrichtung eröffnet wird ($I \subseteq F$) (k -Median-Problem)
i	ein Ort aus der Menge F bzw. I (k -Median-Problem)
j	eine Stadt aus der Menge C (k -Median-Problem)
jqf_{exh}	Schwellwert des JPEG-Qualitätsfaktors für erschöpfende Qualität
jqf_{ins}	Schwellwert des JPEG-Qualitätsfaktors für unzureichende Qualität
$jqf_{t,o}$	Ziel-JPEG-Qualitätsfaktor der Operation o
K	Menge der Inhaltsobjekte in einem System Σ
k	ein Inhaltsobjekt aus der Menge K
K_s	Teilmenge der Inhaltsobjekte, die von Clients aller Klassen angefordert werden
k_s	ein Inhaltsobjekt aus der Teilmenge K_s
\bar{K}_s	Teilmenge der Inhaltsobjekte, die von Clients nur einer Klasse angefordert werden
\bar{k}_s	ein Inhaltsobjekt aus der Teilmenge \bar{K}_s
K_{ci}	Teilmenge der Inhaltsobjekte, die von Clients der Klasse i angefordert werden
$l(\theta)$	Länge des Pfades θ in einem Graphen
Λ	Anfragelast (Matrix $(\lambda_{c,k})$ der Anfrageraten)
$\lambda_{c,k}$	Anfragerate einer Anfrage (c, k)
λ_k	Popularität des Objektes k (aggregierte Anfragerate aller Clients nach k)
$leaf(\pi)$	Blattknoten eines Adaptionpfades π

$\mathcal{L}oc$	Menge der lokalen Platzierungsmanager für eine verteilte Realisierung der Greedy-Algorithmen
loc	ein lokaler Platzierungsmanager aus der Menge $\mathcal{L}oc$
$M \subseteq M_1 \times \dots \times M_\mu$	Menge der Medienqualitäten
$\mathbf{m} = (m_1, \dots, m_\mu)$	Tupel von Medienqualitätsmerkmalen aus M
$\max X$	größtes Element der Menge X
$X = \max!$	der Ausdruck X ist zu maximieren
mhc	Funktion der Clientfähigkeiten (bildet einen Clientknoten c auf die Menge der Tupel von Typ-, Medienqualitäts- und Servicequalitätsmerkmalen ab, die Objektausprägungen beschreiben, die das Endgerät von c darstellen kann)
\mathbb{N}	Menge der natürlichen Zahlen (einschließlich 0)
N	Menge der Knoten in einem System Σ
n	ein Knoten der Menge N
N_d	Zahl der Zugangspunkte für Clients des Typs d (im Simulationsmodell für alle Clienttypen gleich)
NP	Komplexitätsklasse, die alle Probleme umfasst, für die in polynomialer Zeit für eine potentielle Lösung (Zertifikat) geprüft werden kann, ob sie tatsächlich eine Lösung ist; es ist jedoch nicht Voraussetzung, dass ein Algorithmus existiert, der eine Lösung in polynomialer Zeit findet
O	Menge der Operationen (Adaptions- und Transferoperationen) in einem System Σ
o	eine Operation (Adaptions- oder Transferoperation) der Menge O
O_{adapt}	Menge der Adaptionsoperationen
o_{adapt}	eine Adaptionsoperation der Menge O_{adapt}
O_{trans}	Menge der Transferoperationen
o_{trans}	eine Transferoperation der Menge O_{trans}
O	obere Schranke der Komplexität
P	Komplexitätsklasse, die alle Probleme umfasst, die in polynomialer Zeit gelöst werden können
P	Platzierung von Replikaten (Originalreplikate in Originalservern und Replikate in Surrogaten)
p	Preisniveau (konstanter Faktor, um den die nominellen Preise $price_{c,k}$ in der Sensitivitätsanalyse variiert werden)
P_R	Platzierung von Replikaten in den Surrogaten
P_0	Platzierung von Originalreplikaten in den Originalservern

π	ein Adaptionpfad
Π_P	Menge aller Adaptionpfade, die sich bei einer bestimmten Platzierung P konstruieren lassen
$p_{1k} \cdots p_{6k}$	bildspezifische JPEG-Kodierungsparameter eines Bildes k
pix	Zahl der Bildpunkte eines Bildes
pix_{ins}	Schwellwert der Zahl der Bildpunkte für unzureichende Qualität
pix_{exh}	Schwellwert der Zahl der Bildpunkte für erschöpfende Qualität
PR_{P_0}	Menge der potentiellen Replikat bei der Platzierung P_0 von Originalreplikaten
$PR_{P_0}^H$	Menge der potentiellen Replikat in einem Surrogat aus der Menge H bei der Platzierung P_0 von Originalreplikaten
$price_{c,k}$	nomineller Preis eines Objektes k bei Auslieferung an einen Client, der durch c repräsentiert wird
$profit_{\pi,c}$	Profit eines Adaptionpfades π , mittels dessen ein Objekt an einen Client ausgeliefert wird, der durch c repräsentiert wird
$Profit_{\Sigma,(c,k)}(P_R)$	Profit einer Platzierung P_R bezüglich einer konkreten Anfrage (c,k)
$Profit_{\Sigma,\Lambda}(P_R)$	Profit einer Platzierung P_R bezüglich der Anfragelast Λ
$\Delta Profit_{\Sigma,\Lambda}$	mittlerer Zuwachs an Gewinn pro Anfrage durch die Replikation im A-CDN (Bewertungsmetrik zur Evaluation)
qf	Qualitätsfunktion (ordnet einer Ausprägung von Objektmerkmalen eines Objektes einen Qualitätswert zu)
qf_d	Qualitätsfunktion in der Qualitätsdimension d
qf_{del}	Qualitätsfunktion in der Qualitätsdimension del (Antwortzeit)
qf_{res}	Qualitätsfunktion in der Qualitätsdimension res (räumliche Bildauflösung)
qf_{sqd}	Qualitätsfunktion in der Qualitätsdimension sqd (subjektiv wahrgenommene Degeneration des Bildes)
$qual_{\pi,c}$	Qualität eines Adaptionpfades π aus Sicht eines Client, der durch c repräsentiert wird
$\mathbf{R}(f)$	Wertebereich der Abbildung f
\mathbb{R}	Menge der reellen Zahlen
\mathbb{R}_+	Menge der positiven reellen Zahlen
\mathbb{R}_{0+}	Menge der nicht-negativen reellen Zahlen
$\mathbb{R}(a,b)$	Menge der reellen Zahlen x im offenen Intervall $a < x < b$
$\mathbb{R}[a,b]$	Menge der reellen Zahlen x im abgeschlossenen Intervall $a \leq x \leq b$
r_k	Rang des Objektes k in einer Liste, in der alle Objekte entsprechend ihrer Popularität sortiert sind
$Req_{c,k}$	Menge der Anfragen (c',k) aller Clients $c' \in C$, die sowohl bzgl. der Qualitätsfunktion als auch bzgl. des Preissystems äquivalent zu c sind ($c' \equiv_{q,p} c$), nach dem Objekt k

res	Qualitätsdimension der räumlichen Auflösung
$rev_{\pi,c}$	Erlös eines Adaptionspfades π zur Auslieferung eines Objektes an einen Client, der durch c repräsentiert wird
$root(\pi)$	Wurzelknoten eines Adaptionspfades π
$S \subseteq S_1 \times \dots \times S_\sigma$	Menge der Servicequalitäten
s	Anteil der Anfragen nach Objekten, die von Clients aller Klassen angefordert werden
$\mathbf{s} = (s_1, \dots, s_\sigma)$	Tupel von Servicequalitätsmerkmalen aus S
\mathbf{s}_{init}	Tupel der Initialwerte der Servicequalitätsmerkmale \mathbf{s}
S	Standardabweichung einer Stichprobe von Simulations- oder Messergebnissen
Σ	Modell eines Systems zum adaptiven Content Networking
σ_i	relative Anfraghäufigkeit der Clients der Klasse i
$satisfies$	Relation, die ausdrückt, welche Adaptionspfade welche Anfragen beantworten können
sc	Funktion der Speicherkapazität der Surrogate
sf	Größenfunktion, die jeder Repräsentation jedes Objektes deren Größe in Bytes zuordnet
$sink_{c,k}$	virtueller Zielknoten in einem Problemgraphen $\Gamma_{c,k}$
$Sink_{Req_{c,k}}$	Menge der virtuellen Zielknoten in einem Problemgraphen $\Gamma_{Req_{c,k}}$
Sol_Σ	Lösungsraum für die erschöpfende Suche nach einer optimalen Platzierung von Replikaten im A-CDN
$source_k$	virtueller Startknoten in einem Problemgraphen $\Gamma_{c,k}$
sqd	Qualitätsdimension der subjektiv wahrgenommenen Degeneration des Bildes
$T \subseteq T_1 \times \dots \times T_\tau$	Menge der Typen
$\mathbf{t} = (t_1, \dots, t_\tau)$	Tupel von Typmerkmalen aus T
t_{dec}	Ausführungszeit der JPEG-Dekodieroperation
t_{enc}	Ausführungszeit der JPEG-Kodieroperation
t_{scl}	Ausführungszeit der Skalierungsoperation für Bilder
θ	ein Pfad in einem Problemgraphen $\Gamma_{c,k}$
tf	Faktor zur Variation der Ausführungszeiten der Adaptionsoptionen in der Sensitivitätsanalyse
\mathcal{V}_i	Modell der Verteilung der Anfragen auf die Clients
$V(\Gamma)$	Menge der Knoten eines Graphen Γ

w	Gewichtsfunktion in einem Problemgraphen $\Gamma_{c,k}$
$x_{0,k}$	Auflösung der primären Repräsentation eines Bildes k in x-Dimension
x_c	Displayauflösung des Endgerätes eines Clientknoten c in x-Dimension
$x_{\min,k}$	kleinste Auflösung in x-Dimension, bei der ein Bild k nutzbar ist
$x_{t,o}$	Zielauflösung einer Skalierungsoperation o in x-Dimension
\bar{X}	Mittelwert einer Stichprobe von Simulations- oder Messergebnissen
$y_{0,k}$	Auflösung der primären Repräsentation eines Bildes k in y-Dimension
y_c	Displayauflösung des Endgerätes eines Clientknoten c in y-Dimension
$y_{\min,k}$	kleinste Auflösung in y-Dimension, bei der ein Bild k nutzbar ist
$y_{t,o}$	Zielauflösung einer Skalierungsoperation o in y-Dimension
\mathbb{Z}	Menge der ganzen Zahlen
\mathbb{Z}_+	Menge der positiven ganzen Zahlen
$\mathbb{Z}(a, b)$	Menge der ganzen Zahlen x im offenen Intervall $a < x < b$
$\mathbb{Z}[a, b]$	Menge der ganzen Zahlen x im abgeschlossenen Intervall $a \leq x \leq b$
z	Parameter der Zipf-ähnlichen Verteilung der Objektpopularität
2^X	Potenzmenge einer Menge X
\wp	Zuordnung von Kostenfunktionen cf_o zu den Operationen $o \in O$
$ X $	Kardinalität einer Menge X
$\lceil x \rceil$	obere Gauß-Klammer (kleinste ganze Zahl, die größer als x ist)
$\lfloor x \rfloor$	untere Gauß-Klammer (größte ganze Zahl, die kleiner als x ist)
$\{x\}$	gerundeter Wert von x
$[x]_{\equiv}$	Äquivalenzklasse (Menge aller Elemente einer Grundgesamtheit, die bzgl. der Äquivalenzrelation \equiv äquivalent zu x sind)
$f : A \rightarrow B$	partielle Abbildung aus A in B
$f : a \mapsto b$	partielle Abbildung, die a auf b abbildet
$\{x \mid \mathcal{P}(x)\}$	Menge aller x , für die das Prädikat $\mathcal{P}(x)$ erfüllt ist
\equiv_q	Äquivalenzrelation, die Äquivalenz zweier Clientknoten bzgl. der Qualitätsfunktion ausdrückt
\equiv_p	Äquivalenzrelation, die Äquivalenz zweier Clientknoten bzgl. des Preissystems ausdrückt
$\equiv_{q,p}$	Äquivalenzrelation, die Äquivalenz zweier Clientknoten bzgl. der Qualitätsfunktion und des Preissystems ausdrückt

Literatur

- [AJ99] ARLITT, MARTIN und JIN, TAI: *Workload Characterization of the 1998 World Cup Web Site*. Technischer Bericht HPL-1999-35R1, HP Labs, Palo Alto, CA, USA, 1999.
- [AS98] ACHARYA, SOAM und SMITH, BRIAN: *An Experiment to Characterize Videos Stored on the Web*. In: *SPIE/ACM Conference on Multimedia Computing and Networking (MMCN'98)*, Seiten 166–178, San Jose, CA, USA, 1998.
- [ASP00] ACHARYA, SOAM; SMITH, BRIAN und PARNES, PETER: *Characterizing User Access To Videos On The World Wide Web*. In: *SPIE/ACM Conference on Multimedia Computing and Networking (MMCN 2000)*, Seiten 130–141, San Jose, CA, USA, 2000.
- [BB03] BUCHHOLZ, SVEN und BUCHHOLZ, THOMAS: *Adaptive Content Networking*. In: *International Symposium on Information and Communication Technologies (ISICT 03)*, Dublin, Irland, 2003.
- [BB04] BUCHHOLZ, SVEN und BUCHHOLZ, THOMAS: *Replica Placement in Adaptive Content Distribution Networks*. In: *ACM Symposium on Applied Computing (SAC'04)*, Nicosia, Zypern, 2004.
- [BBBC99] BARFORD, PAUL; BESTAVROS, AZER; BRADLEY, ADAM D. und CROVELLA, MARK E.: *Changes in Web Client Access Patterns – Characteristics and Caching Implications*. *World Wide Web Journal, Special Issue on Characterization and Performance Evaluation*, 2(1-2):15–28, 1999.
- [BC98] BARFORD, PAUL und CROVELLA, MARK E.: *Generating Representative Web Workloads for Network and Server Performance Evaluation*. In: *Performance '98/ACM SIGMETRICS '98*, Seiten 151–160, Madison, WI, USA, 1998.
- [BCC⁺95] BESTAVROS, AZER; CARTER, ROBERT L.; CROVELLA, MARK E.; CUNHA, CARLOS R.; HEDDAYA, ABDELSALAM und MIRDAD, SULAIMAN A.: *Application-Level Document Caching in the Internet*. In: *International Workshop on Services in Distributed and Networked Environments (SDNE '95)*, Whistler, Kanada, 1995.
- [BCF⁺99] BRESLAU, LEE; CAO, PEI; FAN, LI; PHILLIPS, GRAHAM und SHENKER, SCOTT: *Web Caching and Zipf-like Distributions: Evidence and Implications*. In: *IEEE Infocom 1999*, New York, NY, USA, 1999.
- [BCH⁺02] BARBIR, A.; CHEN, R.; HOFMANN, M.; ORMAN, H. und PENNO, R.: *An Architecture for Open Pluggable Edge Services (OPES), Internet-Draft (11.12.2002)*. 2002.
- [BCNS03] BARBIR, B.; CAIN, B.; NAIR, R. und SPATSCHECK, O.: *Known Content Network (CN) Request-Routing Mechanisms*. RFC 3568, 2003.

- [BEH⁺01] BRUSSEE, R.; EERTINK, H.; HUIJSEN, W.; HULSEBOSCH, B.; ROUGOOR, M.; TEEUW, W.; WIBBELS, M. und ZANDBELT, H.: *Content Distribution Networks: State of the art*. Technischer Bericht TI/RS/2001/027, Telematica Instituut, Enschede, Niederlande, 2001.
- [Bel57] BELLMAN, RICHARD E.: *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1957.
- [BFK⁺00] BADRINATH, B.; FOX, ARMANDO; KLEINROCK, LEONARD; POPEK, GERALD; REIHER, PETER und SATYANARAYANAN, M.: *A Conceptual Framework for Network and Client Adaptation*. IEEE Mobile Networks and Applications, 5(4), 2000.
- [BGMP01] BUYUKKOKTEN, ORKUT; GARCIA-MOLINA, HECTOR und PAEPCKE, ANDREAS: *Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices*. In: *International World Wide Web Conference (WWW 2001)*, Seiten 652–662, Hong Kong, China, 2001.
- [BGS99] BICKMORE, TIMOTHY; GIRGENSOHN, ANDREAS und SULLIVAN, JOSEPH W.: *Web Page Filtering and Re-Authoring for Mobile Users*. The Computer Journal - Special Issue on Mobile Computing, 42(6):534–546, 1999.
- [BHH04] BUCHHOLZ, SVEN; HAMANN, THOMAS und HÜBSCH, GERALD: *Comprehensive Structured Context Profile (CSCP): Design and Experiences*. In: *IEEE Workshop on Context Modeling and Reasoning (CoMoRea'04)*, Seiten 43–47, Orlando, FL, USA, 2004.
- [BJS02] BUCHHOLZ, SVEN; JAENSCH, STEFFEN und SCHILL, ALEXANDER: *Flexible Web Traffic Modeling for New Application Domains*. In: *IASTED International Conference on Applied Modelling and Simulation (AMS 2002)*, Cambridge, MA, USA, 2002.
- [BKV98] BRUSILOVSKY, PETER; KOBZA, ALFRED und VASSILEVA, JULITA: *Adaptive Hypertext and Hypermedia*. Kluwer Academic Publishers, Dordrecht, 1998.
- [BL03] BAUER, ANDREAS und LEHNER, WOLFGANG: *On Solving the View Selection Problem in Distributed Data Warehouse Architectures*. In: *15th International Conference on Scientific and Statistical Database Management (SSDBM2003)*, Cambridge, MA, USA, 2003.
- [BLFIM98] BERNERS-LEE, T.; FIELDING, R.; IRVINE, U.C. und MASINTER, L.: *Uniform Resource Identifiers (URI): Generic Syntax*. RFC 2396, 1998.
- [BM02] BRUSILOVSKY, PETER und MAYBURY, MARK T.: *From Adaptive Hypermedia to the Adaptive Web*. Communications of the ACM: Special issue: The adaptive web, 45(5):30–33, 2002.
- [Bru01] BRUSILOVSKY, PETER: *Adaptive Hypermedia*. User Modeling and User-Adapted Interaction, 11:87–110, 2001.
- [Buc01] BUCHHOLZ, SVEN: *Heterogeneous Information Access and Distribution for Value-Added Portal Services; Part 1: Extraction, Distribution and Update of Context Information*. Projektbericht Hades-Projekt, Deliverable, Part 1 (intern), Technische Universität Dresden, Fakultät Informatik, Dresden, 2001.
- [Buc04] BUCHHOLZ, SVEN: *Experimental Evaluation of the Performance Advantages of Adaptation-aware Replica Placement in Adaptive CDNs*. In: *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04)*, San Jose, CA, USA, 2004.

- [CBC95] CUNHA, CARLOS R.; BESTAVROS, AZER und CROVELLA, MARK E.: *Characteristics of WWW Clients-based Traces*. Technischer Bericht BU-CS-95-010, Boston University, Computer Science Department, Boston, MA, USA, 1995.
- [CDMF00] CHEVERST, KEITH; DAVIES, NIGEL; MITCHELL, KEITH und FRIDAY, ADRIAN: *Experiences of developing and deploying a context-aware tourist guide: the GUIDE project*. In: *International Conference on Mobile Computing and Networking (MOBICOM)*, Seiten 20–31, Boston, MA, USA, 2000.
- [CDN+96] CHANKHUNTHOD, ANAWAT; DANZIG, PETER B.; NEERDAELS, CHUCK; SCHWARTZ, MICHAEL F.; WORRELL und J., KURT: *A Hierarchical Internet Object Cache*. In: *USENIX Technical Conference*, San Diego, CA, USA, 1996.
- [CFL77] CORNUELOS, GERARD; FISHER, MARSHALL L. und L., NEMHAUSER. GEORGE: *Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms*. Management Sciences, 23(8):789–810, 1977.
- [CGP02] CUCCHIARA, RITA; GRANA, CONSTANTINO und PRATI, ANDREA: *A Framework for Semantic Video Transcoding*. In: *Workshop Percezione e Visione delle Macchine*, Siena, Italien, 2002.
- [CGSEV01] CHANDRA, SURENDAR; GEHANI, ASHISH; SCHLATTER ELLIS, CARLA und VAHDAT, AMIN: *Transcoding Characteristics of Web Images*. In: *SPIE/ACM Conference on Multimedia Computing and Networking (MMCN'01)*, Seiten 135–149, San Jose, CA, USA, 2001.
- [Cha02] CHANG, SHIH-FU: *Optimal Video Adaptation and Skimming Using a Utility-Based Framework*. In: *International Workshop on Digital Communications (IWDC 2002)*, New York, NY, USA, 2002.
- [CHLS02] CHANG, Y.; HAN, R.; LI, C. und SMITH, J.: *Secure Transcoding of Internet Content*. In: *International Workshop on Intelligent Multimedia Computing and Networking (IMMCN)*, Durham, NC, USA, 2002.
- [Chu03] CHUA, BEVERLY: *Impact of Spatio-Temporal Resolution on Video Quality*. Technischer Bericht TR 03/34, CSIRO ICT Centre, Australia, 2003.
- [CKS01] CIDON, ISRAEL; KUTTEN, SHAY und SOFFER, RAN: *Optimal Allocation of Electronic Content*. In: *IEEE Infocom 2001*, Anchorage, AL, USA, 2001.
- [CLRS01] CORMEN, THOMAS H.; LEISERSON, CHARLES E.; RIVEST, RONALD L. und STEIN, CLIFFORD: *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2001.
- [CMI00] CHANDRASEKARAN, SIRISH; MADDEN, SAMUEL und IONESCU, MIHUT: *Ninja Paths: An Architecture for Composing Services over Wide Area Networks*. 2000.
- [CMT01] COOPER, I.; MELVE, I. und TOMLINSON, G.: *Internet Web Replication and Caching Taxonomy*. RFC 3040, 2001.
- [CR99] CHANDRA, KAVITHA und REIBMAN, AMY R.: *Modeling One- and Two-layer Variable Bit Rate Video*. IEEE/ACM Transactions on Networking (TON), 7(3):398–413, 1999.
- [CSE99] CHANDRA, SURENDAR und SCHLATTER ELLIS, CARLA: *JPEG Compression Metrics as a Quality Aware Image Transcoding*. In: *2nd USENIX Symposium on Internet Technologies & Systems (USITS'99)*, Boulder, CO, USA, 1999.
- [CSR98] CANDAN, KASIM S.; SUBRAHMANIAN, V. S. und RANGAN, P. VENKAT: *Collaborative Multimedia Systems: Synthesis of Media Objects*. IEEE Transactions on Knowledge and Data Engineering, 10(3):433–457, 1998.

- [CT99] CROVELLA, MARK E. und TAQQU, MURAD S.: *Estimating the Heavy Tail Index from Scaling Properties*. Methodology and Computing in Applied Probability, 1(1):55–79, 1999.
- [CTW01] CHOI, SUMI; TURNER, JONATHAN und WOLF, TILMAN: *Configuring Sessions in Programmable Networks*. In: *IEEE Infocom 2001*, Anchorage, AL, USA, 2001.
- [CWVL01] CHESIRE, MAUREEN; WOLMAN, ALEC; VOELKER, GEOFFREY M. und LEVY, HENRY M.: *Measurement and Analysis of a Streaming-Media Workload*. In: *3rd USENIX Symposium on Internet Technologies and Systems*, Seiten 1–12, San Francisco, CA, USA, 2001.
- [CYH00] CARDELLINI, VALERIA; YU, PHILIP S. und HUANG, YUN-WU: *Collaborative Proxy System for Distributed Web Content Transcoding*. In: *International Conference on Information and Knowledge Management*, Seiten 520–527, McLean, VA, USA, 2000.
- [DCTR03] DAY, M.; CAIN, B.; TOMLINSON, G. und RZEWSKI, P.: *A Model for Content Internetworking (CDI)*. RFC 3466, 2003.
- [DFKM97] DOUGLIS, FRED; FELDMANN, ANJA; KRISHNAMURTHY, BALACHANDER und MOGUL, JEFFREY: *Rate of Change and other Metrics: a Live Study of the World Wide Web*. In: *USENIX Symposium on Internet Technologies and Systems*, Monterey, CA, USA, 1997.
- [Dij59] DIJKSTRA, EDSGER W.: *A note on two problems in connection with graphs*. Numerical Mathematics, 1:395–412, 1959.
- [Dip02] DIPERT, BRIAN: *Video characterization creates hands-on headaches, part two*. EDN, (8):81–88, 2002.
- [dK01] KOCH, NORA PARCUS DE: *Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process*. München, 2001.
- [DMF97] DUSKA, B.; MARWOOD, D. und FEELEY, M.: *The Measured Access Characteristics of World-Wide-Web Client Proxy Caches*. In: *USENIX Symposium on Internet Technologies and Systems*, Monterey, CA, USA, 1997.
- [DMP⁺02] DILLEY, JOHN; MAGGS, BRUCE; PARIKH, JAY; PROKOP, HARALD; SITARAMAN, RAMESH und WEIHL, BILL: *Globally Distributed Content Delivery*. IEEE Internet Computing, 7(5):50–58, 2002.
- [EB98] ECKERT, MICHAEL P. und BRADLEY, ANDREW P.: *Perceptual Models Applied to Still Image Compression*. Signal Processing, 70(3):177–200, 1998.
- [EC03] ELSON, J. und CERPA, A.: *Internet Content Adaptation Protocol (ICAP)*. RFC 3507, 2003.
- [ES98] EFFELSBERG, WOLFGANG und STEINMETZ, RALF: *Video Compression Techniques*. dpunkt-Verlag, Heidelberg, 1998.
- [FB96a] FOX, ARMANDO und BREWER, ERIC A.: *Reducing WWW Latency and Bandwidth Requirements by Real-time Distillation*. In: *5th International WWW Conference*, Paris, Frankreich, 1996.
- [FB96b] FREED, N. und BORENSTEIN, N.: *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. RFC 2046, 1996.

- [FF62] FORD, L. R. und FULKERSON, D. R.: *Flows in Networks*. Princeton University Press, Princeton, NJ, USA, 1962.
- [FGCB98] FOX, ARMANDO; GRIBBLE, STEVE; CHAWATHE, YATIN und BREWER, ERIC A.: *Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives*. Special Issue of IEEE Personal Communications on Adaptation, 1998.
- [FGM⁺99] FIELDING, R.; GETTYS, J.; MOGU, J.; FRYSTYK, H.; MASINTER, L.; LEACH, P. und BERNERS-LEE, T.: *Hypertext Transfer Protocol – HTTP/1.1*. RFC 2616, 1999.
- [FHMW03] FIALA, ZOLTAN; HINZ, MICHAEL; MEISSNER, KLAUS und WEHNER, FRANK: *A Component-based Approach for Adaptive, Dynamic Web Documents*. Journal of Web Engineering, 2(1&2):58–73, 2003.
- [FJJ⁺01] FRANCIS, P.; JAMIN, S.; JIN, C.; JIN, Y.; RAZ, D.; SHAVITT, Y. und ZHANG, L.: *IDMaps: A Global Internet Host Distance Estimation Service*. IEEE/ACM Transactions on Networking, 9(5):525–540, 2001.
- [FK03] FU, XIAODONG und KARAMCHETI, VIJAY: *Why Path-Based Adaptation? Performance Implications of Different Adaptation Mechanisms for Network Content Delivery*. Technischer Bericht TR2003-843, New York University, Department of Computer Science, New York, NY, USA, 2003.
- [FRE⁺02] FERRERIA, VINCENT; RUDENKO, ALEXEY; EUSTICE, KEVIN; GUY, RICHARD; RAMAKRISHNA, V. und REIHER, PETER: *Panda: Middleware to Provide the Benefits of Active Networks to Legacy Applications*. In: *2002 DARPA Active Networks Conference and Exposition (DANCE'02)*, San Francisco, CA, USA, 2002.
- [FSAK01] FU, XIAODONG; SHI, WEISONG; AKKERMAN, ANATOLY und KARAMCHETI, VIJAY: *CANS: Composable, Adaptive Network Services Infrastructure*. In: *USENIX Symposium on Internet Technologies and Systems (USITS'01)*, San Francisco, CA, USA, 2001.
- [FT87] FREDMAN, MICHAEL und TARJAN, ROBERT: *Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms*. Journal of the Association for Computing Machinery, 34(3):596–615, 1987.
- [FXZM03] FAN, XIN; XIE, XING; ZHOU, HE-QIN und MA, WEI-YING: *Looking into Video Frames on Small Displays*. In: *ACM International Conference on Multimedia*, Berkeley, CA, USA, 2003.
- [GBZS01] GÖBEL, STEFFEN; BUCHHOLZ, SVEN; ZIEGERT, THOMAS und SCHILL, ALEXANDER: *Device Independent Representation of Web-based Dialogs and Contents*. In: *IEEE Youth Forum in Computer Science and Engineering (YUFORIC'01)*, Valencia, Spanien, 2001.
- [Geo74] GEOFFRION, A. M.: *Lagrangean Relaxation for Integer Programming*. Mathematical Programming, 2:82–114, 1974.
- [Gla94] GLASSMANN, STEVEN: *A Caching Relay for the World Wide Web*. In: *International World Wide Web Conference*, Seiten 69–76, Genf, Schweiz, 1994.
- [GM99] GUPTA, HIMANSHU und MUMICK, SINGH: *Selection of Views to Materialize Under a Maintenance Cost Constraint*. In: *7th International Conference on Database Theory (ICDT'99)*, Seiten 453–470, Jerusalem, Israel, 1999.
- [GSZ⁺03] GRASSEL, GUIDO; SPRIESTERSBACH, AXEL; ZIEGERT, THOMAS; WASMUND, MICHAEL; DERMLER, GABRIEL und MERRICK, ROLAND: *An XHTML Language Profile for Single Source Authoring to Enhance the Access from Mobile Devices to Web Enterprise Applications*. In: *WWW 2003 Developers Day*, Budapest, Ungarn, 2003.

- [Gup97] GUPTA, HIMANSHU: *Selection of Views to Materialize in a Data Warehouse*. In: *Int'l Conference on Database Theory*, Delphi, Griechenland, 1997.
- [GWB⁺01] GRIBBLE, STEVEN D.; WELSH, MATT; BEHREN, ROB VON; BREWER, ERIC A.; CULLER, DAVID; BORISOV, N.; CZERWINSKI, S.; GUMMADI, R.; HILL, J.; JOSEPH, ANTONY D.; KATZ, RANDY H.; MAO, Z. M.; ROSS, S. und ZHAO, B.: *The Ninja Architecture for Robust Internet-Scale Systems and Services*. *Journal of Computer Networks*, 35(4), 2001.
- [Ham92] HAMILTON, ERIC: *JPEG File Interchange Format, Version 1.02*. <http://www.w3.org/Graphics/JPEG/jfif3.pdf> [Zugriff am 26.08.2004], 1992.
- [HBR⁺98] HAN, R.; BHAGWAT, P.; R., LAMAIRE; MUMMERT, T.; PERRET, V. und RUBAS, J.: *Dynamic Adaptation in an Image Transcoding Proxy for Mobile Web Browsing*. *IEEE Personal Communications Magazine*, 5(6):8–17, 1998.
- [Hei04] HEIMRICH, THOMAS: *Output Constraints in Multimedia Database Systems*. In: *International Workshop on Multimedia Data and Document Engineering (MDDE-04)*, Seiten 143 – 150, Washington DC, USA, 2004.
- [HRU96] HARINARAYAN, VENKY; RAJARAMAN, ANAND und ULLMAN, JEFFREY D.: *Implementing data cubes efficiently*. In: *ACM International Conference on Management of Data (SIGMOD'96)*, Seiten 205–216, Montreal, Kanada, 1996.
- [HWC74] HELD, MICHAEL; WOLFE, PHILIP und CROWDER, HARLAN P.: *Validation of Subgradient Optimization*. *Mathematical Programming*, 6:62–88, 1974.
- [JB00] JIN, SHUDONG und BESTAVROS, AZER: *Sources and Characteristics of Web Temporal Locality*. In: *The IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (Mascots2000)*, San Francisco, CA, USA, 2000.
- [JM01] JACOBS, APRIL und MEARS, JENNIFER: *Content peer groups fall flat*. <http://www.nwfusion.com/news/2001/1105infra.html> [Zugriff am 25.08.2004], 2001.
- [Jör99] JÖRDING, TANJA: *Temporary User Modeling for Adaptive Product Presentations in the Web*. In: *International Conference on User Modeling*, Banff, Kanada, 1999.
- [JV99] JAIN, KAMAL und VAZIRANI, VIJAY V.: *Primal-Dual Approximation Algorithms for Metric Facility Location and k-Median Problems*. In: *IEEE Symposium on Foundations of Computer Science (FOCS)*, Seiten 2–13, New York, NY, USA, 1999.
- [JWJ02] JARED WINICK, JARED und JAMIN, SUGIH: *Inet-3.0: Internet Topology Generator*. Technischer Bericht CSE-TR-456-02, University of Michigan, 2002.
- [Kar04] KARLSSON, MAGNUS: *persönliche Kommunikation (Email vom 10.08.2004)*. 2004.
- [KB94] KRISHNAKUMAR, NARAYANAN und BERNSTEIN, ARTHUR J.: *Bounded Ignorance: A Technique for Increasing Concurrency in a Replicated System*. *ACM Transactions on Database Systems (TODS)*, 19(4):586–625, 1994.
- [Kel01] KELLY, TERENCE: *Thin-Client Web Access Patterns: Measurements from a Cache-Busting Proxy*. In: *Sixth International Web Caching and Content Delivery Workshop*, Boston, MA, USA, 2001.
- [KHRR01] KANGASHARJU, JUSSI; HARTANTO, FELIX; REISSLEIN, MARTIN und ROSS, KEITH W.: *Distributing Layered Encoded Video through Caches*. In: *IEEE Infocom 2001*, Seiten 1791–1800, Anchorage, AL, USA, 2001.

- [Kin03] KING, RAWLSON: *Netli Unveils World's First Application Delivery Network*. http://www.netli.com/Art_5.html (Zugriff am 27.10.2004), 2003.
- [KK03] KARLSSON, MAGNUS und KARAMANOLIS, CHRISTOS: *Bounds on the Replication Cost for QoS*. Technischer Bericht HPL-2003-156, HP Labs, Palo Alto, CA, USA, 2003.
- [KKM02] KARLSSON, MAGNUS; KARAMANOLIS, CHRISTOS und MAHALINGAM, MALLIK: *A Framework for Evaluating Replica Placement Algorithms*. Technischer Bericht HPL-2002-219, HP Labs, Palo Alto, CA, USA, 2002.
- [KKO98] KANGASHARJU, JUSSI; KWON, YOUNGGAP und ORTEGA, ANTONIO: *Design and Implementation of a Soft Caching Proxy*. In: *3rd International WWW Caching Workshop (WCW'98)*, Manchester, UK, 1998.
- [Kly99] KLYNE, G.: *A Syntax for Describing Media Feature Sets*. RFC 2533, 1999.
- [KM02] KARLSSON, MAGNUS und MAHALINGAM, MALLIK: *Do We Need Replica Placement Algorithms in Content Delivery Networks*. In: *7th International Workshop on Web Content Caching and Distribution (WCW'02)*, Boulder, CO, USA, 2002.
- [KPR99] KORUPOLU, MADHUKAR R.; PLAXTON, C. GREG und RAJARAMAN, RAJMOHAN: *Placement Algorithms for Hierarchical Cooperative Caching*. In: *ACM-SIAM Symposium on Discrete algorithms*, Seiten 586–595, Baltimore, MD, USA, 1999.
- [KRR01] KANGASHARJU, JUSSI; ROBERTS, JAMES und ROSS, KEITH W.: *Object Replication Strategies in Content Distribution Networks*. In: *6th International Workshop on Web Content Caching and Distribution (WCW'01)*, Boston, MA, USA, 2001.
- [KRW01] KELLER, RALPH; RAMAMIRTHAM, JEYASHANKHER und WOLF, TILMAN: *Active pipes: Program composition for programmable networks*. In: *IEEE Conference on Military Communications (MILCOM)*, McLean, VA, USA, 2001.
- [KRW⁺04] KLYNE, GRAHAM; REYNOLDS, FRANKLIN; WOODROW, CHRIS; OHTO, HIDETAKA; HJELM, JOHAN; BUTLER, MARK H. und TRAN, LUU: *Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0*. In: *W3C Recommendation*, 2004.
- [KRWP00] KELLER, RALPH; RAMAMIRTHAM, JEYASHANKHER; WOLF, TILMAN und PLATTNER, BERNHARD: *Programming Active Networks Using Active Pipes*. Technischer Bericht WUCS-00-27, Washington University, St. Louis, WA, USA, 2000.
- [KW02] KRASIC, CHARLES und WALPOLE, JONATHAN: *Quality-Adaptive Media Streaming by Priority Drop*. Technischer Bericht CSE-02-015, Department of Computer Science and Engineering, Oregon Graduate Institute, 2002.
- [KWC03] KIM, JAE-GON; WANG, YONG und CHANG, SHIH-FU: *Content-Adaptive Utility-Based Video Adaptation*. In: *IEEE International Conference on Multimedia & Expo (ICME)*, Baltimore, MD, USA, 2003.
- [KWZ01] KRISHNAMURTHY, BALACHANDER; WILLS, CRAIG und ZHANG, YIN: *On the Use and Performance of Content Distribution Networks*. In: *ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, USA, 2001.
- [LA94] LUOTONEN, ARI und ALTIS, KEVIN: *World-Wide Web Proxies*. *Computer Networks and ISDN Systems*, 27(2):147–154, 1994.
- [LCKN03] LI, MINGZHE; CLAYPOOL, MARK; KINICKI, ROBERT und NICHOLS, JAMES: *Characteristics of Streaming Media Stored on the Web*. Technischer Bericht WPI-CS-TR-03-18, Worcester Polytechnic Institute, Worcester, MA, USA, 2003.

- [Len97] LENZ, RICHARD: *Adaptive Datenreplikation in verteilten Systemen*. B.G. Teubner Verlagsgesellschaft, Stuttgart, Leipzig, 1997.
- [LGB⁺98] LANE, TOM; GLADSTONE, PHILIP; BOUCHER, JIM; CROCKER, LEE; MINGUILON, JULIAN; ORTIZ, LUIS; PHILLIPS, GEORGE; ROSSI, DAVIDE; VOLLBEDING, GUIDO und WEIJERS, GE: *The Independent JPEG Group's JPEG software Release 6b*. Independent JPEG Group (IJG), <http://www.iijg.org/files/jpegsrc.v6b.tar.gz> [Zugriff am 20.06.2003], 1998.
- [LGI⁺99] LI, BO; GOLIN, MORDECAI J.; ITALIANO, GIUSEPPE F.; DENG, XIN und SOHRABY, KAZEM: *On the Optimal Placement of Web Proxies in the Internet*. In: *IEEE Infocom 1999*, New York, NY, USA, 1999.
- [LL00] LEIGHTON, F. THOMAS und LEWIN, DANIEL M.: *Global Hosting System*. US Patent No. 6,108,703, United States Patent and Trademark Office, USA, 2000.
- [LL02] LUM, WAI YIP und LAU, FRANCIS C.M.: *On Balancing Between Transcoding Overhead and Spatial Consumption in Content Adaptation*. In: *ACM International Conference on Mobile Computing and Networking (MobiCom 2002)*, Atlanta, GA, USA, 2002.
- [LMP01] LUPARELLO, DARIO; MUKHERJEE, SARIT und PAUL, SANJOY: *Streaming Media Traffic: An Empirical Study*. In: *International Workshop on Web Caching and Content Distribution*, Boston, MA, USA, 2001.
- [LTEZG03] LU, XIAOXIANG; TAO, SHU; EL ZARKI, MAGDA und GUERIN, ROCH: *Quality-based Adaptive Video Over the Internet*. In: *Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS'03)*, Orlando, FL, USA, 2003.
- [LW99] LAZZARO, JOHN und WAWRZYNEK, JOHN: *JPEG Quality Transcoding using Neural Networks Trained with a Perceptual Error Measure*. *Neural Computation*, 11(1):267–296, 1999.
- [Mar03] MARDER, ULRICH: *Multimedia-Metacomputing in Web-basierten multimedialen Informationssystemen*. Logos Verlag, Berlin, 2003.
- [MBC⁺00] MA, WEI-YING; BEDNER, ILJA; CHANG, GRACE; KUCHINSKY, ALLAN und ZHANG, HONGJIANG: *A Framework for Adaptive Content Delivery in Heterogeneous Network Environments*. In: *SPIE/ACM Conference on Multimedia Computing and Networking (MMCN 2000)*, San Jose, CA, USA, 2000.
- [MF00] MICHALEWICZ, ZBIGNIEW und FOGEL, DAVID B.: *How to Solve It: Modern Heuristics*. Springer-Verlag, Heidelberg, Berlin, 2000.
- [MSL99] MOHAN, RAKESH; SMITH, JOHN R. und LI, CHUNG-SHENG: *Adapting Multimedia Internet Content for Universal Access*. *IEEE Transactions on Multimedia*, 1(1):104–114, 1999.
- [MSS02] MANJUNATH, B. S.; SALEMBIER, PHILIPPE und SIKORA, THOMAS: *Introduction to MPEG-7: Multimedia Content Description Interface*. John Wiley & Sons, Inc., New York, NY, USA, 2002.
- [NPB01] NAKANO, AKIHIRO; PETERSON, LARRY und BAVIER, ANDY: *Constructing End-to-End Paths for Playing Media Objects*. In: *IEEE Open Architectures and Network Programming*, Seiten 117–128, New York, NY, USA, 2001.
- [NPS95] NOBLE, BRIAN D.; PRICE, MORGAN und SATYANARAYANAN, M.: *A Programming Interface for Application-Aware Adaptation in Mobile Computing*. *Computing Systems*, 8(4):345–363, 1995.

- [NS03] NEPAL, SURYA und SRINIVASAN, UMA: *DAVE: A System for Quality Driven Adaptive Video Delivery*. In: *ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'03)*, Seiten 223–230, Berkeley, CA, USA, 2003.
- [NZ02] NG, T. S. EUGENE und ZHANG, HUI: *Predicting Internet Network Distance with Coordinates-Based Approaches*. In: *IEEE Infocom 2002*, New York, NY, USA, 2002.
- [OCAV97] ORTEGA, ANTONIO; CARIGNANO, FABIO; AYER, SERGE und VETTERLI, MARTIN: *Soft Caching: Web Cache Management for Images*. In: *IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, Princeton, NJ, USA, 1997.
- [oV98] OHNE VERFASER: *GVU's 9th WWW User Survey, Internet Shopping (Part 2) Summary*. http://www.gvu.gatech.edu/user_surveys/survey-1998-04/reports/1998-04-Netshop2.html [Zugriff am 01.09.2004], 1998.
- [ÖV99a] ÖZSU, M. TAMER und VALDURIEZ, PATRICK: *Principles of Distributed Database Systems*. Prentice Hall, Upper Saddle River, NJ, USA, 1999.
- [oV99b] OHNE VERFASER: *Programming in Java Advanced Imaging*. Sun Microsystems, Inc., http://java.sun.com/products/java-media/jai/forDevelopers/jai1_0_1guide-unc/ [Zugriff am 19.08.2004], 1999.
- [oV00a] OHNE VERFASER: *Cisco, Cable & Wireless, Digital Island, Genuity, GlobalCenter, Mirror Image Internet, NaviSite, Network Appliance, PSINet, and ServInt form Content Alliance*. Cisco Systems Inc., http://www.cisco.com/warp/public/146/pressroom/2000/aug00/sp_082800.htm [Zugriff am 25.08.2004], 2000.
- [oV00b] OHNE VERFASER: *Konsortium für Webbeschleuniger gegründet*. <http://www.heise.de/newsticker/meldung/11464> [Zugriff am 25.08.2004], 2000.
- [oV01a] OHNE VERFASER: *Universal Mobile Telecommunications System (UMTS); Service aspects: Services and Service Capabilities*. In: *ETSI TS 122 105 v3.10.0*. European Telecommunications Standards Institute (<http://www.etsi.org/>), 2001.
- [oV01b] OHNE VERFASER: *Untersuchung: Bringt i-mode den Durchbruch für WAP?* <http://www.golem.de/0111/17102.html> [Zugriff am 02.06.2004], 2001.
- [oV02a] OHNE VERFASER: *IBM WebSphere Everyplace Access - An Architectural Overview*. IBM Corp.; <http://www7.software.ibm.com/vadd-bin/ftpd1?1/vadc/wsdd/pdf/WEAArchitecture0909.pdf> [Zugriff am 20.08.2004], 2002.
- [oV02b] OHNE VERFASER: *Content Distribution Market Size and Share Study*. NetsEdge Research Group; http://www.netsedgeonline.com/pdf_files/market_share_final.pdf [Zugriff am 25.08.2004], 2002.
- [oV02c] OHNE VERFASER: *Market for CDN Products and Services Will Continue to Grow Says NetsEdge Research Group*. NetsEdge Research Group; http://www.netsedgeonline.com/press_releases/2002_content_dist_study.html [Zugriff am 25.08.2004], 2002.
- [oV02d] OHNE VERFASER: *WAP Forum Releases*. <http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html> [Zugriff am 20.08.2004], 2002.
- [oV03] OHNE VERFASER: *The Content Distribution Internetworking (cdi) Working Group in the Applications Area of the IETF has concluded*. <ftp://ops.ietf.org/pub/lists/2003/cdn.0306> [Zugriff am 25.08.2004], 2003.

- [oV04a] OHNE VERFASSEN: *Company Information*. Akamai Technologies, Inc., http://www.akamai.com/en/html/about/company_information.html [Zugriff am 25.08.2004], 2004.
- [oV04b] OHNE VERFASSEN: *Press Release: Akamai Reports Record Revenue and Profits*. Akamai Technologies, Inc., <http://www.akamai.com/en/html/about/press/press469.html> [Zugriff am 25.08.2004], 2004.
- [oV04c] OHNE VERFASSEN: *Annual Report on Form 10-K for the Fiscal Year Ended December 31, 2003*. Akamai Technologies, Inc., http://www.akamai.com/en/html/investors/10k_2003.htm [Zugriff am 25.08.2004], 2004.
- [oV04d] OHNE VERFASSEN: *1&1 Internet AG: 1 & 1 Professional*. <http://hosting.1und1.de/xml/order/HostingDev> [Zugriff am 18.08.04], 2004.
- [oV04e] OHNE VERFASSEN: *Langenscheidt Fremdwörterbuch Online*. <http://www.langenscheidt.de/fremdwb/> [Zugriff am 19.08.2004], 2004.
- [oV04f] OHNE VERFASSEN: *Vodafone live!* <http://www.vodafone.de/live/index.html> [Zugriff am 19.08.2004], 2004.
- [oV04g] OHNE VERFASSEN: *Mathematica Version 5 Documentation* Wolfram Research, Inc., <http://documents.wolfram.com/v5/> [Zugriff am 30.12.2004], 2004.
- [oV05] OHNE VERFASSEN: *Wrist Net Square FX3005*. <http://www.fossil.com/shopping/product/detailmain.jsp?itemType=PRODUCT&RS=1&itemID=21845&keyword=FX+3005> [Zugriff am 05.01.2005], 2005.
- [PB02] PODLIPNIG, STEFAN und BÖSZÖRMENYI, LASZLO: *Replacement Strategies for Quality Based Video Caching*. In: *IEEE International Conference on Multimedia and Expo (ICME)*, Lausanne, Schweiz, 2002.
- [PB03] PODLIPNIG, STEFAN und BÖSZÖRMENYI, LASZLO: *A Survey of Web Cache Replacement Strategies*. *ACM Computing Surveys*, 35(4):374–398, 2003.
- [Pen03] PENG, GANG: *CDN: Content Distribution Network*. Technischer Bericht TR-125, Experimental Computer Systems Lab, Department of Computer Science, State University of New York, Stony Brook, NY, USA, 2003.
- [PM93] PENNEBAKER, WILLIAM B. und MITCHELL, JOAN L.: *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, 1993.
- [QPV01] QIU, LILI; PADMANABHAN, VENKATA N. und VOELKER, GEOFFREY M.: *On the Placement of Web Server Replicas*. In: *IEEE Infocom 2001*, Seiten 1587–1596, Anchorage, AL, USA, 2001.
- [RA99] RABINOVICH, MICHAEL und AGGARWAL, AMIT: *RaDaR: A scalable architecture for a global Web hosting service*. In: *International World Wide Web Conference (WWW'99)*, Toronto, Kanada, 1999.
- [RDG03] RZEWSKI, P.; DAY, M. und GILLETTI, D.: *Content Internetworking (CDI) Scenarios*. RFC 3570, 2003.
- [RGE01] RADOSLAVOV, PAVLIN; GOVINDAN, RAMESH und ESTRIN, DEBORAH: *Topology-Informed Internet Replica Placement*. In: *International Workshop on Web Caching and Content Distribution*, Boston, MA, USA, 2001.
- [RK01] REJAIE, REZA und KANGASHARJU, JUSSI: *Mocha: A Quality Adaptive Multimedia Proxy Cache for Internet Streaming*. In: *International Workshop on Network and Operating Systems Support for Digital Audio and Video*, Port Jefferson, NY, USA, 2001.

- [RMV99] ROADKNIGHT, CHRIS; MARSHALL, IAN und VEARER, DEBORAH: *File Popularity Characterization*. In: *Second Workshop on Internet Server Performance*, Atlanta, GA, USA, 1999.
- [Rob02] ROBBEN, MATTHIAS: *Internetnutzung in Europa – ein Puzzle mit 1000 Teilen?* <http://www.ecin.de/marktbarometer/europa2/> [Zugriff am 24.08.2004], 2002.
- [Rou04] ROUSSKOV, A.: *OPES Callout Protocol Core, Internet-Draft (5. Mai 2004)*. 2004.
- [RS02] RABINOVICH, MICHAEL und SPATSHECK, OLIVER: *Web Caching and Replication*. Addison-Wesley, 2002.
- [RXA03] RABINOVICH, MICHAEL; XIAO, ZHEN und AGGARWAL, AMIT: *Computing on the Edge: A Platform for Replicating Internet Applications*. In: *8th International Workshop on Web Content Caching and Distribution (WCW'03)*, Hawthorne, NY, USA, 2003.
- [RYHE00] REJAIE, REZA; YU, HAOBO; HANDLEY, MARK und ESTRIN, DEBORAH: *Multimedia Proxy Caching Mechanism for Quality Adaptive Streaming Applications in the Internet*. In: *IEEE Infocom 2000*, Tel Aviv, Israel, 2000.
- [SC02] SEHGAL, ANSHUL und CHOU, PHILIP A.: *Cost-Distortion Optimized Caching Of Streaming Media*. In: *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Orlando, FL, USA, 2002.
- [SLB04] SHEN, BO; LEE, SUNG-JU und BASU, SUJOY: *Caching Strategies in Transcoding-enabled Proxy Systems for Streaming Media Distribution Networks*. *IEEE Transactions on Multimedia, Special Issue on Streaming Media*, 6(2), 2004.
- [Spr04] SPRINGER, THOMAS: *Ein komponentenbasiertes Meta-Modell kontextabhängiger Adaptionsgraphen für mobile und ubiquitäre Anwendungen*. Dissertation, Technische Universität Dresden, 2004.
- [SRL98] SCHULZRINNE, H.; RAO, A. und LANPHIER, R.: *Real Time Streaming Protocol (RTSP)*. RFC 2326, 1998.
- [SSPS04] SIVASUBRAMANIAN, SWAMINATHAN; SZYMANIAK, MICHAL; PIERRE, GUILLAUME und STEEN, MAARTEN VAN: *Web Replica Hosting Systems Design*. Technischer Bericht IR-CS-001, Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, Niederlande, 2004.
- [STHK01] SCHILIT, BILL N.; TREVOR, JONATHAN; HILBERT, DAVID und KOH, T. K.: *m-Links: An Infrastructure for Very Small Internet Devices*. In: *ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Rom, Italien, 2001.
- [Tin76] TINHOFER, GOTTFRIED: *Methoden der angewandten Graphentheorie*. Springer-Verlag, Wien, New York, 1976.
- [TRAR99] TORRES-ROJAS, FRANCISCO J.; AHAMAD, MUSTAQUE und RAYNAL, MICHEL: *Timed Consistency for Shared Distributed Objects*. In: *ACM Symposium on Principles of Distributed Computing*, Seiten 163–172, Atlanta, GA, USA, 1999.
- [TS03] TANENBAUM, ANDREW S. und STEEN, MAARTEN VAN: *Verteilte Systeme*. Pearson Studium, München, 2003.
- [TZC02] TANG, XUEYAN; ZHANG, FAN und CHANSON, SAMUEL T.: *Streaming Media Caching Algorithms for Transcoding Proxies*. In: *International Conference on Parallel Processing (ICPP'02)*, Seiten 287–295, Vancouver, Kanada, 2002.

- [VC94] VISHWANATH, MOHAN und CHOU, PHILIP A.: *An Efficient Algorithm for Hierarchical Compression of Video*. In: *IEEE International Conference on Image Processing (ICIP-94)*, Austin, TX, USA, 1994.
- [VP03] VAKALI, ATHENA und PALLIS, GEORGE: *Content Delivery Networks: Status and Trends*. *IEEE Internet Computing*, 7(6):68–74, 2003.
- [WAF99] WEE, SUSIE J.; APOSTOLOPOULOS, JOHN G. und FEAMSTER, NICK: *Field-to-frame Transcoding with Spatial and Temporal Downsampling*. In: *IEEE International Conference on Image Processing*, Kobe, Japan, 1999.
- [Wan99] WANG, JING: *A Survey of Web Caching Schemes for the Internet*. *ACM SIGCOMM Computer Communication Review*, 29(5):36–46, 1999.
- [Wes03] WESSELS, DUANE: *Squid Web Proxy Cache*. <http://www.squid-cache.org/> [Zugriff am 10.08.2004], 2003.
- [WKL⁺99] WALPOLE, JONATHAN; KRASIC, CHARLES; LIU, LING; MAIER, DAVID; PU, CALTON; MCNAMEE, DYLAN und STEERE, DAVID: *Quality of Service Semantics for Multimedia Database Systems*. In: *IFIP TC-2/WG 2.6 Working Conference on Database Semantics - Semantics Issues in Multimedia (DS-8)*, Rotorua, Neuseeland, 1999.
- [WSA02] WEE, SUSIE; SHEN, BO und APOSTOLOPOULOS, JOHN: *Compressed-Domain Video Processing*. Technischer Bericht HPL-2002-282, HP Labs, Palo Alto, CA, USA, 2002.
- [WSJ99] WOLSKI, RICH; SPRING, NEIL T. und JIM, HAYES.: *The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing*. *Journal of Future Generation Computing Systems*, 15(5-6):757–768, 1999.
- [WVS⁺99] WOLMAN, ALEC; VOELKER, GEOFFREY M.; SHARMA, NITIN; CARDWELL, NEAL; KARLIN, ANNA und LEVY, HENRY M.: *On the Scale and Performance of Cooperative Web Proxy Caching*. *Operating Systems Review*, 34(5):16–31, 1999.
- [YCE⁺97] YOSHIKAWA, CHAD; CHUN, BRENT; EASTHAM, PAUL; VAHDAT, AMIN; ANDERSON, THOMAS und CULLER, DAVID: *Using Smart Clients to Build Scalable Services*. In: *USENIX 1997 Annual Technical Conference*, Anaheim, CA, USA, 1997.
- [YREP01] YARVIS, MARK D.; REIHER, PETER; EUSTICE, KEVIN und POPEK, GERALD J.: *Conductor: Enabling Distributed Adaptation*. Technischer Bericht CSD-TR-010025, UCLA, Los Angeles, CA, USA, 2001.
- [YRP99] YARVIS, MARK D.; REIHER, PETER und POPEK, GERALD J.: *Conductor: A Framework for Distributed Adaptation*. In: *Workshop on Hot Topics in Operating Systems (HotOS VII)*, Seiten 44–49, Rio Rico, AZ, USA, 1999.