

**LEHRSTUHL FÜR MULTIMEDIA UND SIGNALVERARBEITUNG**  
UNIVERSITÄT ERLANGEN-NÜRNBERG  
Prof. Dr.-Ing. Andre Kaup

# **Multimediaübertragungsverfahren für Internet und Mobilfunk**

**Diplomarbeit**

von

**cand. Ing. Stefan Petrausch**

Betreuende Hochschullehrer : Prof. Dr.-Ing. Andre Kaup,  
Dipl.-Ing. Wolfgang Sörgel

Forchheim, 6. August 2002

Hier kommt die Aufgabenstellung hin !

## **Erklärung**

Ich versichere, daß ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und daß die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Forchheim, den 6. August 2002

(Stefan Petrausch)

Bayreuther Str. 94  
D-91301 Forchheim

# Inhaltsverzeichnis

<b>Aufgabenstellung</b>	<b>ii</b>
<b>Erklärung</b>	<b>iii</b>
<b>Zusammenfassung</b>	<b>vi</b>
<b>Formelzeichen und Abkürzungen</b>	<b>vii</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Multimedia- <i>Streaming</i> auf mobile Empfänger . . . . .	1
1.1.1 Geschichtlicher Hintergrund . . . . .	1
1.1.2 <i>Transcoding</i> . . . . .	2
1.2 Bisherige Arbeiten . . . . .	2
1.3 Problematik der Untersuchung . . . . .	3
<b>2 Informationstheoretische Betrachtungen</b>	<b>6</b>
2.1 Gekoppelte Kanäle . . . . .	6
2.1.1 Der Übertragungskanal . . . . .	6
2.1.2 Informationsübertragung . . . . .	7
2.1.3 Einschub einer Verarbeitungseinrichtung . . . . .	7
2.2 Kanalmodelle . . . . .	8
2.2.1 Allgemein . . . . .	8
2.2.2 Der <i>Binary Symmetric Channel</i> BSC . . . . .	9
2.2.3 Unsymmetrische binäre Kanäle . . . . .	15
2.2.4 <i>Erasure</i> -Kanäle . . . . .	18
2.3 Kanäle mit Gedächtnis . . . . .	21
2.3.1 Markovmodelle . . . . .	21
2.3.2 BSC-Modell . . . . .	23
2.3.3 <i>Erasure</i> -Modell . . . . .	27
2.4 Verknüpfungen verschiedenartiger Kanäle . . . . .	28
2.4.1 Das verknüpfte Kanalmodell . . . . .	29
2.4.2 Die Kapazitätsberechnung . . . . .	30
2.4.3 Ergebnisse und Vergleich . . . . .	31

<b>3</b>	<b>Verzerrung bei angewandten Codierverfahren</b>	<b>34</b>
3.1	Allgemein . . . . .	34
3.1.1	Die Datenübertragung . . . . .	34
3.1.2	Verzerrung . . . . .	35
3.2	Paketorientierte Übertragung . . . . .	36
3.2.1	Verzerrung für einen Kanal . . . . .	36
3.2.2	Zwei Kanäle . . . . .	37
3.3	Ungleicher Fehlerschutz . . . . .	40
3.3.1	Begriffsklärungen . . . . .	40
3.3.2	Pakete mit Daten unterschiedlicher Wichtigkeit . . . . .	42
3.3.3	Minimierung der Verzerrung $D$ . . . . .	43
3.3.4	Verzerrung für einen Kanal . . . . .	45
3.3.5	Zwei Kanäle . . . . .	46
<b>4</b>	<b>Verifikation</b>	<b>50</b>
4.1	Ausgangsmaterial . . . . .	50
4.1.1	Das Video . . . . .	50
4.1.2	Der Kanal und die Codierung . . . . .	51
4.1.3	Die PSNR-Messung . . . . .	51
4.2	Das Programm und seine Funktionen . . . . .	52
4.2.1	Der allgemeine Ablauf . . . . .	52
4.2.2	Bearbeitung eines Videos . . . . .	52
4.2.3	Die Optimierung der Redundanzverteilung . . . . .	53
4.3	Messung . . . . .	55
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>57</b>
<b>A</b>	<b>Kapazität des Gilbert-Elliott Kanals nach Mushkin</b>	<b>60</b>
A.1	Modell und allgemeine Definitionen . . . . .	60
A.2	Abgeleitete Größen . . . . .	61
A.3	Rekursive Berechnung von $q_n$ . . . . .	62
A.4	Die Kapazität des Gilbert-Elliott Kanals und ihre Eigenschaften . . . . .	63
<b>B</b>	<b>Kapazität zweier gekoppelter Gilbert-Elliott Kanäle</b>	<b>65</b>
B.1	Modell und Vorgehensweise . . . . .	65
B.2	Rekursive Berechnung der bedingten Zustandsverteilung . . . . .	65
B.3	Ermittlung der Kapazität und Unterschiede zu Mushkin . . . . .	67

# Zusammenfassung

Diese Diplomarbeit widmet sich der Problematik der Übertragung von Multimedia-Inhalten über diverse Datenetzwerke, wie beispielsweise das Internet, zu einer Basisstation und von dort aus über einen Funkkanal auf mobile Empfänger, vorzugsweise Handys. Dabei wird das Hauptaugenmerk auf die Kanalcodierung, spezieller die Behandlung und Verarbeitung dieser in der Basisstation, gerichtet. Es werden zwei prinzipielle Vorgehensweisen unterschieden, einerseits das Durchschleifen der Kanalcodierung und nur minimale Anpassung an die verschiedenen Übertragungswege, wie die Beachtung der Transportprotokolle, und andererseits die komplette Neucodierung des Kanalcodes, *Transcoding* genannt ( nicht zu verwechseln mit dem *Transcoding* bei der Quellcodierung), um bereits aufgetretene Fehler durch den Kanalcode zu korrigieren, womit die Basisstation als Signalverstärker fungiert. Dabei wird kein Wert auf spezielle Fehlerschutzcodes gelegt, da die Vielfalt dieser enorm groß ist und die Wahl einiger weniger Codes das Ergebnis verfälscht, sondern es wird nach prinzipiellen Eigenschaften derartiger Datenübertragungen gesucht.

Zu diesem Zweck wird in der Arbeit nach einem einleitendem Kapitel, das nochmals die Problematik erläutert und auf verwandte Arbeiten hinweist, im zweiten Kapitel die Situation informationstheoretisch behandelt und es werden somit Grenzwerte und Richtlinien für den zu erwartenden Unterschied herausgearbeitet. Dabei werden die verschiedensten Effekte und die unterschiedlichsten Übertragungsarten modelliert und untersucht, um dem faktischen Problem immer näher zu kommen. Im folgenden Kapitel wird eine exemplarische, sehr allgemein gehaltene, algebraische Blockcodierung zur Fehlerkorrektur angenommen um an ihr die Auswirkungen von begrenzter Codewortlänge und anderen praktischen Problemen bei der Kanalcodierung zu untersuchen. Besonders wird dabei auch auf unterschiedlichen Fehlerschutz bei der Codierung eingegangen, wie er gerade für Multimediainhalte angewandt wird und deutliche Gewinne bringt. Im letzten Kapitel wird ein Übertragungsszenario mit Hilfe eines C-Programmes und eines am Lehrstuhl bereits vorhandenen *Video-Streaming* Systems simuliert und versucht die Ergebnisse der vorherigen Kapitel zu verifizieren.

Im Anhang befinden sich die Beschreibungen zu komplizierteren Algorithmen die in der Arbeit benötigt wurden und der Ausblick gibt Hinweise darauf, wie die in dieser Arbeit gewonnenen Ergebnisse eingesetzt und genutzt werden können.

# Formelzeichen und Abkürzungen

## Operatoren

Hier einige in dieser Arbeit verwendeten Abkürzungen und Schreibweisen:

$\lfloor \cdot \rfloor$	nächste ganze Zahl $z \in \mathbb{Z}$ kleiner als das Argument
$\mathcal{E}\{\cdot\}$	Erwartungswert des Ausdrucks
$I(\cdot; \cdot)$	Transinformation zwischen den beiden Zufallsvariablen
$\Pr(\cdot)$	Wahrscheinlichkeit für ein Ereignis
$B(N; p; i) :=$	$p^i(1-p)^{N-i} \cdot \binom{N}{i}$ Binomialverteilung
$\mathcal{C} :=$	$\{c_1, c_2, \dots, c_K\}$ Zustandsraum für den Markovprozess
$\mathcal{X} :=$	$\{x_1, x_2, \dots, x_N\}$ Menge aller Eingangsvariablen
$\mathcal{Y} :=$	$\{y_1, y_2, \dots, y_N\}$ Menge aller Ausgangsvariablen
$\mathbf{z}_n :=$	$(z_1, z_2, z_3, \dots, z_n)$ Folge von Zufallsvariablen

## Formelzeichen

Nachfolgend die in dieser Diplomarbeit verwendeten Formelzeichen:

$?$	Ausgangssymbol Auslöschung
$\mu_i$	Gedächtnis des $i$ -ten Kanals
$\rho_i$	<i>Good to Bad Ratio</i> des $i$ -ten Kanals
$\rho_{ij}$	Verzerrungsmaß zur Berechnung der mittleren Verzerrung
<b>B,G</b>	Zustände des Gilbert-Elliott Kanalmodells
$b, g$	Übergangswahrscheinlichkeiten des Gilbert-Elliott Kanalmodells
$C$	maximal mögliche Kapazität ( also meist mit <i>Transcoding</i> )
$C_{BSC}$	Kapazität eines <i>Binary Symmetric Channels</i>
$C_{GilEll}$	Kapazität eines Gilbert-Elliott Kanals
$C_i$	Kapazität des $i$ -ten Teilkanals
$C_N$	Kapazität ohne <i>Transcoding</i>
$C_{NM}$	Kapazität ohne Berücksichtigung des Kanalgedächtnisses
$C_{SI}$	Kapazität bei perfekter Kenntnis des Kanalzustandes
$C_T$	Index $T$ bedeutet Einheit in bit/s
$c_k$	Kanalzustände
$D$	mittlere Verzerrung bei der Datenübertragung

$D^{(h)}$	Zufallsgröße: dargestellter Layer in der $H$ -ten Layerhierarchie
$D_l^{(h)}$	Ereignis: in der $h$ -ten Layerhierarchie wird der $l$ -te Layer dargestellt
$D_i$	mittlere Verzerrung des $i$ -ten Teilkanals
$D_{max}$	maximal tolerierte Verzerrung bei der Übertragung
<b>E</b>	Zustand Auslöschung ( <i>Erasure</i> ) des Verbundmodells
$E$	Übergangswahrscheinlichkeit und Aufenthaltswahrscheinlichkeit für den Zustand <b>E</b>
$F$	Anzahl der aufgetretenen Fehler
$f_l$	Anzahl der Redundanz-Byte des $l$ -ten Layer
$H$	Anzahl der Layerhierarchien
$i, j$	Zählvariablen
$K$	Anzahl der Nachrichtensymbole
$K_e$	vergleichbare Nachrichtenlänge bei gleichem Fehlerschutz für alle Layer
$k_l$	Anzahl der Byte des $l$ -ten Layer
$L^{(h)}$	Tiefe der $h$ -ten Layerhierarchie
$M$	Anzahl der Eingangssymbole
$M_B$	Wortlänge oder Blockgröße
$M_X, M_Y$	Mächtigkeit von $X$ bzw. $Y$
$m$	Anzahl der bit zur Darstellung von $M$
$N$	Codewortlänge
$n_l$	Größe des Codeworts des $l$ -ten Layer in Byte
$P$	Zustandsübergangsmatrix
$p$	Fehlerwahrscheinlichkeit
$\bar{p}$	durchschnittliche Fehlerrate eines MEC
$p^{(0)}, p^{(1)}$	Fehlerrate eines BSC's für das Eingangssymbol $X = 0$ bzw. $X = 1$
$p_B, p_G$	Fehlerwahrscheinlichkeiten des Gilbert-Elliott Modells
$p_G$	Fehlerwahrscheinlichkeit des Gesamtkanals
$p_M$	mittlere Fehlerwahrscheinlichkeit, für die $C$ maximiert wird
$R, R_0$	Datenübertragungsrate
$S, S'$	Ein-, Ausgangsvariable des Servers
$S_n$	Zustand eines Markovmodells zum Zeitpunkt $n$
$T$	Dauer einer Kanalbenutzung
$X$	Eingangs-Zufallsvariable
$x_i$	Realisierung der Zufallsvariablen $X$
$Y$	Ausgangs-Zufallsvariable
$y_i$	Realisierung der Zufallsvariablen $Y$



# Kapitel 1

## Einführung

### 1.1 Multimedia-Streaming auf mobile Empfänger

#### 1.1.1 Geschichtlicher Hintergrund

Das vergangene Jahrzehnt war geprägt von der rasanten Entwicklung des Internets. Die Notwendigkeit des Datenaustauschs zwischen den verschiedenen Wissenschaftlern und Forschungseinrichtungen war schon lange Zeit vorher bekannt und verwirklicht worden, aber durch die enorme Verbreitung des *Personal Computers* in alle Bereiche des modernen Lebens, sowohl bei der Arbeit als auch in der Freizeit, wuchs auch bei Privatpersonen mehr und mehr der Bedarf nach derartigen Datenaustausch-Diensten. Und wiederum durch die technische Entwicklung angetrieben und von privaten Konsumenten nachgefragt, wurden multimediale Inhalte, wie Texte, Bilder, Klänge, Musik und Video, immer wichtiger für die Computer und somit auch für das Internet.

Es wurden diverse Quellcodierverfahren für Musik und Video entwickelt und standardisiert, wie 1994 MP3 ( siehe [16]) und Jahr 2000 MPeG 4 ( siehe [17]) um der begrenzten Bandbreite ( eigentlich Datenübertragungsrate) des Internets Rechnung zu tragen, womit nun einerseits kompakte Datenströme zur Übertragung bereit stehen und andererseits, speziell durch die Video-Codierverfahren, erweiterte Ansprüche an die Kanalcodierung gestellt werden, wie ungleicher Fehlerschutz.

Und andererseits erlebte auch der Sektor der mobilen Kommunikation, spätestens seit der Einführung des GSM-Standards 1991 in Europa ( siehe [19]) eine unvergleichlichen Aufschwung. Die nunmehr weite Verbreitung von mobilen Telefonen und die kommende Einführung des UMTS-Standards, und damit der Verfügbarkeit von ausreichend Bandbreite, führen zu einer Verschmelzung von Mobilfunksystemen und Internet. Der künftige Anwender möchte mit seinem Handy Inhalte aus dem Internet, insbesondere multimediale Inhalte, von einem beliebigen Ort auf der Welt in ansprechender Qualität empfangen können.

Es wird also immer mehr nötig sein große Datenströme über den Verbund von Internet und Mobilfunkkanal auf das Handy zu *streamen*.

### 1.1.2 Transcoding

Eine derartige Anordnung der Kanäle, zum *Streamen* von Daten, benötigt wie jede moderne digitale Übertragungstechnik Codes zur Fehlerkorrektur, sogenannte Fehlerschutzcodes, um die während der Übertragung anfallenden Fehler erkennen und beseitigen zu können. Die Natur dieser Kommunikationsstrecke erlaubt ganz prinzipiell zwei Methoden der Anwendung von Fehlerschutzcodes:

- Vollkommen getrennte Kanalcodierung für Internet und Mobilfunk:  
Die Basisstation empfängt das Codewort, das über das Internet gesendet wurde, entfernt die vom Sender als Kanalcodierung hinzugefügte Redundanz und erhält somit die gesendete Information. Dabei muß kein Unterschied zwischen FEC-Codes (*Forward Error Correction-Codes*) und ARQ-Protokollen (*Automatic Repeat reQuest-Protokollen*) gemacht werden, da beide mit der eben erwähnten Redundanz arbeiten.  
Liegen genug Daten im Puffer der Basisstation vor, so werden diese neu encodiert und über den Mobilfunkkanal geschickt, an dessen Ende das Handy diesen Kanalcode wieder entfernt um an die eigentlich gesendete Information zu kommen.  
Diese Umcodierung der Kanalcodierung wird in der Literatur wie beispielsweise in [13] *Transcoding* genannt, ist aber nicht zu verwechseln mit dem *Video-Transcoding*, also einer neuen Quellcodierung des Videosignals, so wie dieser Begriff in [15] verwendet wird.
- Gemeinsame Kanalcodierung für Internet und Mobilfunk:  
Der oben beschriebene Vorgang wird nur einmal durchgeführt. Am Sender werden die Daten encodiert, ihnen wird Redundanz hinzugefügt und decodiert werden sie erst am Empfänger wieder. Die Basisstation leitet die Daten nur weiter, bzw. verrichtet nur das Nötigste, wie die Anpassung des Datenstroms an die verschiedenen Protokolle und Blockgrößen. Sie schleift die Daten einfach durch, es wird kein *Transcoding* betrieben.

Und eben dieser Unterschied zwischen den beiden Methoden ist Gegenstand dieser Arbeit, wir wollen im Weiteren untersuchen, welchen Unterschied es macht und vor allem wie groß dieser ist, wenn *Transcoding* betrieben wird oder nicht.

## 1.2 Bisherige Arbeiten

Zum Thema Video-Streaming über das Internet sind einige aktuelle Veröffentlichungen vorhanden, wie beispielsweise übersichtshalber in [14]. Viele Autoren verlegen sich aber vor allem auf die Aspekte der effizienten Quellcodierung unter Berücksichtigung einer passenden Paketierung zur Übertragung im Internet. Ungleicher Fehlerschutz bzw. *unequal error-protection* auf Englisch spielt dabei eine wichtige Rolle. Denn einerseits muß die Anpassung des Datenstroms an die unterschiedlichen Kapazitäten des Internets gegeben sein, man spricht vom skalierbarem *Video-Streaming* ( zum Beispiel in [10]), und andererseits kann damit die beobachtete Störung im Video minimiert werden.

Dabei gehen die Autoren verschiedene Wege der Kanalcodierung:

- In der Arbeit, die vor kurzem am Lehrstuhl für Nachrichtentechnik I in Erlangen entstand ( [10]), wird die klassische Methode der Datenübertragung im Internet mittels

ARQ-Protokollen, also automatischen Wiederholungsanforderungen, behandelt. Es werden Methoden vorgestellt, wie die Übertragung zu optimieren ist, und Ergebnisse präsentiert.

- Viele Arbeiten befassen sich mit Reed-Solomon Codes, da dies besonders gute FEC-Codes ( *Forward Error Correction*-Codes) für paketorientierte Datenübertragung sind. Die prinzipielle Aufteilung der Datenblöcke wird beispielsweise in einer ebenfalls am Lehrstuhl entstandenen Arbeit [9] dargestellt. In [11] steht auch ein einfacher Algorithmus zur approximativen Suche der besten Verteilung der Redundanz auf die jeweiligen Schichten, *Layer* genannt. Und exakt diese Variante der Kanalcodierung werde ich in Kapitel 3 und 4 zur Untersuchung heranziehen.

Zum eigentlichen Problem, dem *Streamen* von Videos, bzw. Multimediadaten auf mobile Empfänger, gibt es nicht so viele Arbeiten. Viele Autoren weisen auf die Möglichkeit hin ( wie [10]), daß es sinnvoll wäre einen Proxy-Server einzusetzen, schließen dies aber eher aus Beobachtungen und Vermutungen. In [15] wird eine Übersicht geliefert, *Transcoding* wird aber nur oberflächlich behandelt.

Dem Thema dieser Diplomarbeit am nächsten kommt das Paper [13] der Chinesen T.-W. Angus Lee, S.-H. Gary Chan, Qian Zhang, Wen-Wu Zhu und Ya-Qin Zhang, in dem es speziell um die optimale Verteilung von Redundanz bei der Videoübertragung über Festnetze und drahtlose Netze geht. Es wird in dieser Arbeit auch konkret zwischen *Transcoding* und nicht *Transcoding* unterschieden und zwischen den beiden Methoden verglichen.

Allerdings wurden keine prinzipiellen Aussagen gemacht, sondern nur Meßwerte für spezielle Kanalcodierungen ermittelt und ausgewertet. Und diese Lücke soll diese Arbeit füllen, die allgemeinen und prinzipiellen Ergebnisse können dann gut mit denen von [13] verglichen werden.

### 1.3 Problematik der Untersuchung

Aber gerade das Finden von allgemeinen und prinzipiellen Ergebnissen ist auf Grund der Vielschichtigkeit und der Mannigfaltigkeit der möglichen Kanalcodierungen sehr schwer. Die Wahl eines bestimmten Codes und das Festlegen von Eigenschaften des Übertragungssystems beeinflussen das Ergebnis enorm, weshalb die Messungen von [13] nur unter ganz bestimmten Bedingungen als Referenz herangezogen werden können.

Um die beiden Optionen *Transcoding* und nicht *Transcoding* fair vergleichen zu können, muß man versuchen alle Nebeneffekte der Kanalcodierung auszublenden und nur die festen Eigenschaften des Internets auf der einen Seite und des Mobilfunkkanals auf der anderen Seite zu betrachten. Tabelle 1.1 listet die wichtigsten Beeinflussungen des Kanals durch die Codierung auf.

Oft interessieren die Effekte aus Tabelle 1.1 zwar, sind allerdings von den technischen Gegebenheiten abhängig und somit speziell und nicht im allgemeinen gültig. Um nun derlei Auswirkungen zu vermeiden, wird für die weiteren Untersuchungen die Kanalcodierung für beide Fälle immer bestmöglich optimiert, um damit die Größen fair vergleichen zu können.

Zu diesem Zweck bedienen wir uns im nächsten Kapitel, dem Kapitel 2, der Informations-

Kurzbeschreibung	Auswirkungen
Typ des Fehlerschutzcodes	Nicht jeder Kanal eignet sich für jeden Fehlerschutzcode, einige sind besonders effizient und andere hingegen nicht. Wenn man unterschiedliche Codes wählt, so muss man deren Effizienz und Wirkungsweise berücksichtigen.
Codewortlänge $N$	Je länger der Code, desto besser der Fehlerschutz, andererseits kann aber die Codewortlänge nicht beliebig groß werden. Verwenden beide Kanäle unterschiedlich lange Codes, so kann das eigentliche Ergebnis vom Effekt der Codewortlänge verdeckt werden.
Maximale Verzögerungen	Folgendes kann passieren: <ul style="list-style-type: none"> <li>• meist muss die Codewortlänge <math>N</math> gesenkt werden, was zu obigen Folgen führt</li> <li>• wenn nicht genügend Zeit für ausreichend viele Internet-Pakete zur Verfügung steht, ist die Codierung sehr schlecht angepasst, während der Mobilfunkkanal auf Grund der anderen Struktur nur wenig darunter leidet</li> <li>• sind die maximal tolerierbaren Verzögerungen kurz, so erhöht sich die Fehlerrate des Internets, aber auch nur diese</li> </ul>
Puffergrößen	Die Auswirkungen sind ähnlich denen von begrenzten Verzögerungen, allerdings ist meist die Puffergröße des mobilen Empfangsgerätes begrenzt, was die Codewortlänge des Mobilfunkkanals einschränkt.
Rechenaufwand	Je nach Code und je nach Codewortlänge kann der Aufwand zur Decodierung sehr groß sein. Wiederum im mobilen Gerät ist die Rechenkomplexität aber begrenzt, weshalb hier oft ein einfacher und kürzerer Code verwendet werden muss.
Super-Coder Effekt	Verschachtelt man mehrere Codes ineinander, so ist der resultierende Code meist deutlich besser als ein Einzeler. Dies wird bei den sogenannten Turbo-Codes ausgenutzt und kann beträchtliche Gewinne bringen.

Tabelle 1.1: Beeinflussungen durch die gewählte Kanalcodierung

theorie, die das Problem auf eine allgemeine Basis stellt, und im darauf folgenden Kapitel 3, bedienen wir uns eines prototypischen Verfahren der Blockfehlerkorrektur.

# Kapitel 2

## Informationstheoretische Betrachtungen

### 2.1 Gekoppelte Kanäle

#### 2.1.1 Der Übertragungskanal

Diese Arbeit behandelt Übertragungskanäle und deren Eigenschaften und wie sie zu optimieren sind. Daher müssen wir uns zuerst klar machen wovon wir reden. Nach Kelly (siehe in [7]) gilt:

Der Teil eines Informationsübertragungssystems, den der Entwerfer des Systems so hinzunehmen hat wie er ist, den er nicht ändern (d.h. optimieren) kann oder will, wird als Kanal bezeichnet.

Und nach Shannon (vgl. [8]) kann einem Kanal eine maximale Informationsübertragungsrate, die Kapazität  $C$  zugeordnet werden, die fest ist und auf keinen Fall überschritten werden kann. Ein echter Kanal kann also nicht mit einer größeren Rate als  $C$  übertragen, schon aus informationstheoretischer Sicht. Was aber nun, wenn der interessante Übertragungskanal rein physikalisch eine Reihenschaltung von zwei anderen Kanälen ist?

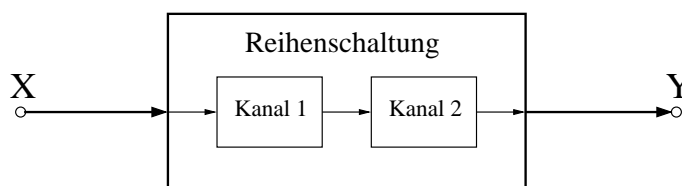


Abbildung 2.1: Reihenschaltung zweier Kanäle

Nach Kelly bildet diese Reihenschaltung von zwei Kanälen keinen Kanal im eigentlichen Sinn, oft interessiert aber eben nur die Kapazität dieser, die wir deshalb im weiteren Verlauf den *Gesamtkanal* nennen wollen.

In diesem Kapitel wird nun die Frage behandelt, wie die Kapazität des Gesamtkanals bei gegebenen, voneinander unabhängigen Teilkanälen optimiert werden kann.

### 2.1.2 Informationsübertragung

Es gilt also den Informationsfluß und somit die Transinformation  $I(X; Y)$  zwischen den beiden Zufallsvariablen  $X$  und  $Y$  ( vgl. Abbildung 2.1 ) zu maximieren. Das Maximum dieser Transinformation wird die Kapazität  $C$  des Kanals genannt, von der wir aus der Informationstheorie wissen, daß sie angibt, wieviel Information maximal pro Kanalbenutzung sicher und ohne Fehler übertragen werden kann.

$$C = \max I(X; Y)$$

Bei einem Kanal im eigentlichen Sinn nach Kelly und, wenn wir keinerlei Änderungen an obiger Reihenschaltung vornehmen, kann nur durch Variation der Eingangsvariablen  $X$  maximiert werden. Diese *normale* Kapazität  $C_N$  ermittelt sich demnach wie folgt:

$$C_N = \max_X I(X; Y)$$

### 2.1.3 Einschub einer Verarbeitungseinrichtung

Die denkbare Änderung, die wir an unserem Gesamtkanal vornehmen können, ist der Einschub einer Informationsverarbeitungseinrichtung. Sie soll die ankommenden Daten aus dem ersten Kanal entgegennehmen, Fehler korrigieren und die so gewonnene Information für den zweiten Kanal bestmöglich aufbereiten.

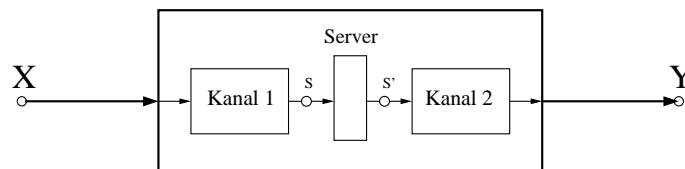


Abbildung 2.2: Server verarbeitet Zufallsvariable  $S$

Um die Kapazität für diesen modifizierten Kanal vorerst wenigstens abschätzen zu können bedienen wir uns des *Data Processing Theorem*, das besagt, daß niemals Information gewonnen werden kann und jeder Teil einer Übertragungskette nur Information verlieren kann ( vgl. [7] ).

$$I(X; Y) \leq I(X; S') \leq I(X; S) \quad (2.1)$$

Nur bei idealer Informationsverarbeitung im Server tritt das Gleichheitszeichen der zweiten Ungleichung in 2.1 in Kraft.

Da die Transinformation symmetrisch bezüglich ihrer beider Zufallsvariablen ist ( also  $I(X; Y) = I(Y; X)$  ), und unter der Annahme eines idealen Servers, läßt sich somit folgendes herleiten:

$$\begin{aligned} I(X; Y) &\leq I(X; S) \\ I(X; Y) &\leq I(S'; Y) \end{aligned}$$

⇓

$$\begin{aligned}
 I(X; Y) &\leq \min \{I(X; S); I(S'; Y)\} \\
 C &\leq \min \{C_1; C_2\}
 \end{aligned}
 \tag{2.2}$$

Dabei kann die Gleichheit unabhängig von  $C_1$  und  $C_2$  keineswegs ausgeschlossen werden! Diese wichtige Tatsache wird in Abbildung 2.3 an einem praxisnäherem Modell noch einmal verdeutlicht.

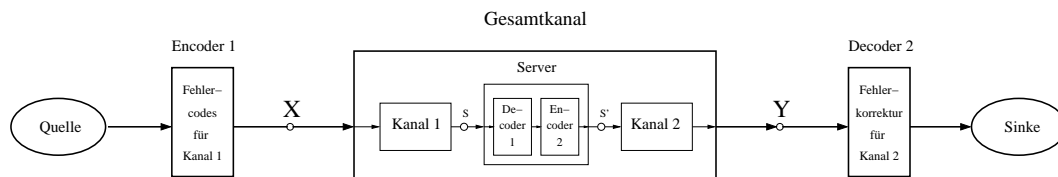


Abbildung 2.3: optimale Übertragung mit Server

Einer Datenquelle werden Informationen mit einer bestimmten Rate  $R_0$  entzogen. Diese können solange  $R_0 \leq C_1$  durch den Encoder 1 und den Decoder 1 fehlerfrei im Server ankommen. Dieser fungiert nun selber als Quelle und kann nun seinerseits Informationen mit der Rate  $R_0$  zur Verfügung stellen. Er codiert diese Daten im Encoder 2 und schickt sie durch Kanal 2. Durch den Decoder 2 kann wiederum unter der Bedingung  $R_0 \leq C_2$  genau diese Daten-Rate  $R_0$  rekonstruiert werden und an die Informationssinke abgegeben werden. Eine fehlerfreie Übertragung ist also nach dem Kanalcodierungstheorem (vgl. [7]) möglich, wenn gilt

$$R_0 \leq C_1 \wedge R_0 \leq C_2$$

⇓

$$R_0 \leq \min \{C_1; C_2\}$$

und da nur dann fehlerfrei übertragen werden kann, wenn die Code-Rate kleiner als die Kanal-kapazität ist, also  $R_0 \leq C$  stoßen wir wieder auf Gleichung 2.2.

## 2.2 Kanalmodelle

### 2.2.1 Allgemein

Die Frage, die nun behandelt werden muß, ist, wie stark oder ob überhaupt, die Kapazität mit Server  $C$  besser als die ohne jede Verarbeitungseinrichtung  $C_N$  ist.

$$C \stackrel{?}{\geq} C_N$$



Zu diesem Zweck werden im weiteren einige einfache Kanalmodelle untersucht, die sich realen Kanälen möglichst gut nähern, aber trotzdem analytisch erfassbar bleiben.

Dabei beschränken wir uns auf zeitdiskrete Kanäle, da der zeitliche Ablauf im einzelnen nicht interessiert und einer physikalischen Realisierung überlassen bleibt. Raten und Kapazitäten werden hierbei in Information pro Kanalbenutzung gerechnet. Weiterhin beschränken wir uns auf diskrete Ein- und Ausgangsvariablen, da wertekontinuierliche Variablen für die Verarbeitung ungeeignet sind, und sowieso durch Analog-Digital-Wandler in wertediskrete Variablen umgesetzt werden.

Die Kapazität eines solchen zeit- und wertdiskreten Kanals berechnet sich nach [7] wie folgt

$$\begin{aligned}
 C &:= \max_X \{I(X;Y)\} \\
 &= \max_X \left\{ \sum_{i=1}^{M_X} \sum_{j=1}^{M_Y} \Pr(X = x_i, Y = y_j) \cdot \log_2 \left[ \frac{\Pr(X = x_i | Y = y_j)}{\Pr(X = x_i)} \right] \right\}
 \end{aligned} \tag{2.3}$$

mit den Variablen

- $X$  : Zufallsvariable am Eingang
- $Y$  : Zufallsvariable am Ausgang
- $M_X$  : Mächtigkeit von  $X$
- $M_Y$  : Mächtigkeit von  $Y$

### 2.2.2 Der Binary Symmetric Channel BSC

#### Ein einzelner BSC

Das erste Kanalmodell, das untersucht werden soll, ist der *Binary Symmetric Channel* kurz BSC genannt, wie er in Abbildung 2.4 zu sehen ist.

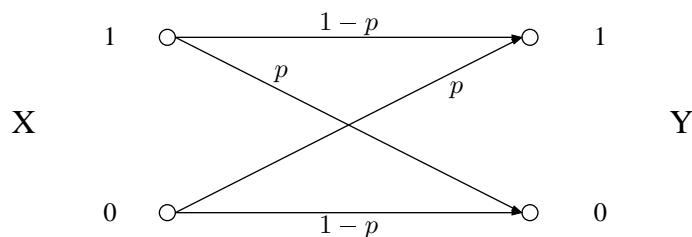


Abbildung 2.4: Der BSC

Seine Zufallsvariablen nehmen nur zwei Zustände an  $M_X = M_Y = 2$  sind also binär und er hat eine symmetrische Fehlerwahrscheinlichkeit  $p$  ( oft auch *Bit Error Rate BER* genannt), die für beide Eingangssymbole ( 1 und 0 ) gleich groß ist. Außerdem hat der Kanal keinerlei Gedächtnis. Sein Verhalten ist vollkommen unabhängig vom vorherigen oder zukünftigen Geschehen, was sich in der statistischen Unabhängigkeit seiner Übergangswahrscheinlichkeiten ausdrückt.

$$P(Y_i | X_i, Y_j | X_j) = P(Y_i | X_i) \cdot P(Y_j | X_j) \quad \forall i \neq j$$

Da  $X$  nur zwei Werte annimmt, können wir diese Zufallsvariable durch das Ereignis '1' mit der Wahrscheinlichkeit  $\Pr(X = 1)$  vollkommen beschreiben. Damit vereinfacht sich die Suche nach dem Maximum der Transinformation auf eine Extremwertbestimmung in einer Variablen.

$$C = \max_{\Pr(X=1)} \left\{ \sum_{i=1}^2 \sum_{j=1}^2 \Pr(X = x_i, Y = y_j) \cdot \log_2 \left[ \frac{\Pr(X = x_i | Y = y_j)}{\Pr(X = x_i)} \right] \right\} \quad (2.4)$$

Die Herleitung kann einfach nachvollzogen werden, ist aber auch in der einschlägigen Literatur ( zum Beispiel [7] ) zu finden und lautet:

$$C_{BSC} = 1 + p \log_2 p + (1 - p) \log_2(1 - p) \quad (2.5)$$

### Zwei gekoppelte BSC's

Für uns von Interesse sind nun zwei BSC's in Reihe geschaltet, so wie sie in Abbildung 2.5 zu sehen sind. Um die Notation zu vereinfachen werden die Bitfehlerwahrscheinlichkeiten mit  $p_1$  für den ersten und  $p_2$  für den zweiten Kanal abgekürzt.

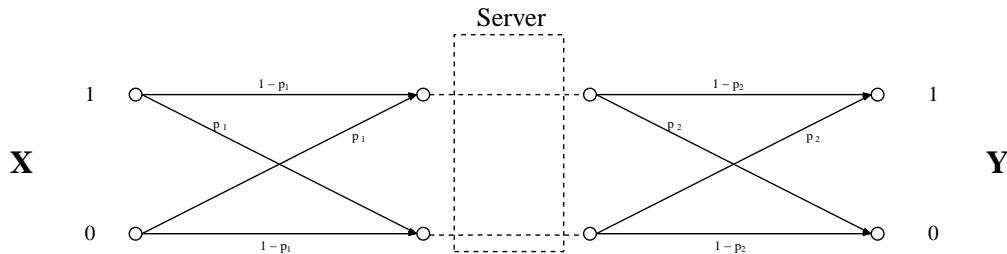


Abbildung 2.5: Zwei BSC's mit Server in der Mitte

Wenn der Server nichts macht bzw. nicht existiert, können wir einen äquivalenten BSC nach Abbildung 2.4 konstruieren, der als Gesamtbitfehlerwahrscheinlichkeit

$$\begin{aligned} p_G &= (1 - p_1)p_2 + p_1(1 - p_2) \\ &= p_1 + p_2 - 2p_1p_2 \end{aligned} \quad (2.6)$$

hat, was eine Kapazität gemäß 2.5 ergibt.

$$C_N = 1 + p_G \log_2 p_G + (1 - p_G) \log_2(1 - p_G)$$

Wird aber ein Server eingebaut und nutzt dieser die Kapazitäten der Teilkanäle ideal, so wird die Kapazität des Gesamtkanals nach Gleichung 2.2 berechnet.

$$C = \min \left\{ 1 + p_1 \log_2 p_1 + (1 - p_1) \log_2(1 - p_1); \right. \\ \left. 1 + p_2 \log_2 p_2 + (1 - p_2) \log_2(1 - p_2) \right\}$$

Nimmt man einen gegebenen Kanal mit gegebener Fehlerwahrscheinlichkeit  $p_G$  an und löst man Gleichung 2.6 nach  $p_2$  auf, so erhält man eine Formel für die Kapazität  $C$  in Abhängigkeit von  $p_G$  und  $p_1$ .

$$C = \min \left\{ 1 + p_1 \log_2 p_1 + (1 - p_1) \log_2(1 - p_1); 1 + \frac{p_G - p_1}{1 - 2p_1} \cdot \log_2 \left( \frac{p_G - p_1}{1 - 2p_1} \right) + \frac{1 - p_1 - p_G}{1 - 2p_1} \log_2 \left( \frac{1 - p_1 - p_G}{1 - 2p_1} \right) \right\} \quad (2.7)$$

Abbildungen 2.6 und 2.7 sind zwei Beispiele von Funktion 2.7, in denen für die beiden Gesamtbitfehlerwahrscheinlichkeiten  $p_G = \frac{1}{10}$  und  $p_G = \frac{3}{10}$  vorerst die Kapazitäten der Teilkanäle  $C_1$  und  $C_2$  geplottet wurden. Die Kapazität  $C$  des Gesamtkanals ergibt sich dann als Minimum von  $C_1$  und  $C_2$ , und ist zum Vergleich gegenüber der Kapazität ohne Server  $C_N$  in Abhängigkeit der Bitfehlerwahrscheinlichkeit  $p_1$  des ersten Kanals angetragen. Zusätzlich ist der Schnittpunkt von  $C_1$  und  $C_2$ , der gleichzeitig immer das Maximum von  $C$  darstellt, mit  $p_1 = p_M$  gekennzeichnet..

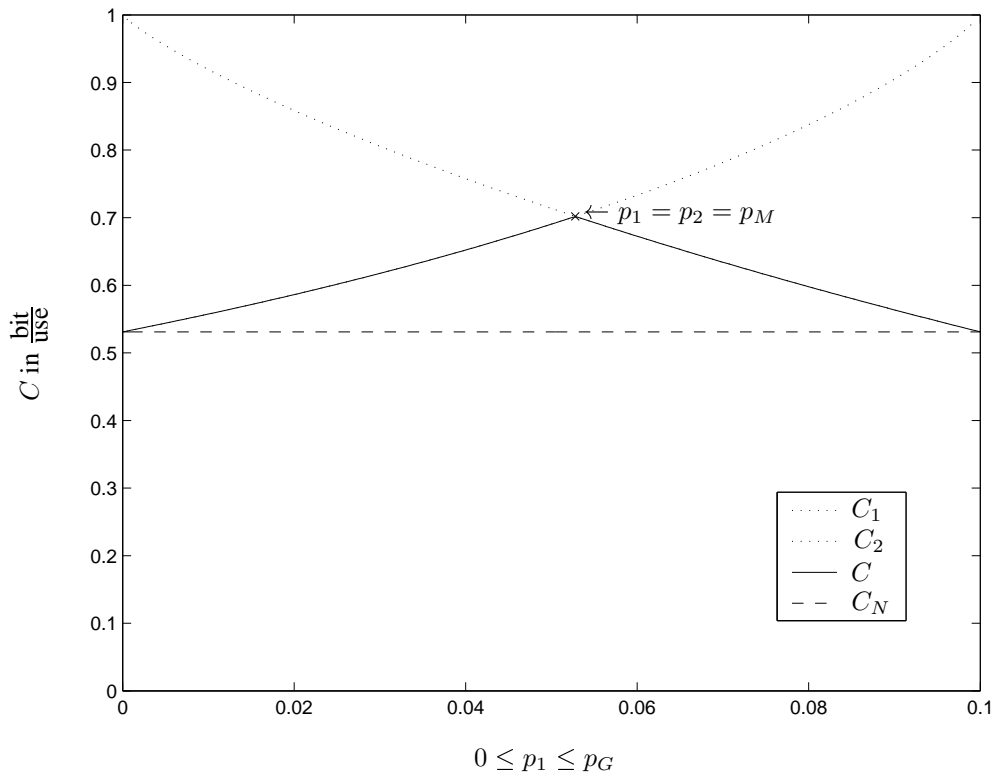


Abbildung 2.6: Erreichbare Kapazität  $C$  im Vergleich zu  $C_N$  bei  $p_G = \frac{1}{10}$ .

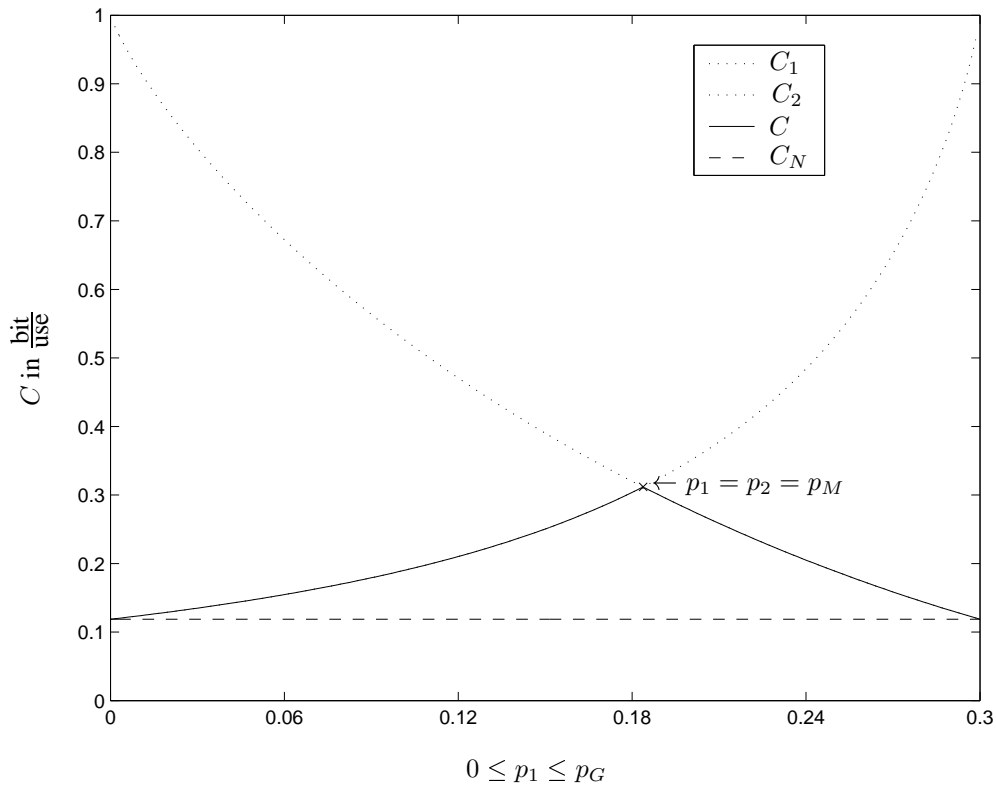


Abbildung 2.7: Erreichbare Kapazität  $C$  im Vergleich zu  $C_N$  bei  $p_G = \frac{3}{10}$ .

Wichtige Eigenschaften, die man den Bildern entnehmen kann, sind:

- die Fehlerate eines einzelnen Kanals kann nicht größer sein als die des Gesamtkanals:

$$p_1 \leq p_G$$

Der zweite Kanal kann keine Fehler korrigieren, die der erste Kanal gemacht hat, nach Gleichung 2.6 ergäben sich sonst negative Werte für  $p_2$ .

- die Kapazitäten der Teilkanäle schneiden sich nicht bei der Mitte der Fehlerrate, sondern für

$$p_M := p_1 = p_2 = \frac{1}{2} - \frac{1}{2} \sqrt{1 - 2 \cdot p_G} \geq \frac{p_G}{2} \tag{2.8}$$

- für größere Bitfehlerwahrscheinlichkeiten sind die Gewinne prinzipiell größer
- die Gesamtkapazität  $C$  ist niemals kleiner als die Kapazität ohne Server  $C_N$

⇓

Man kann den Kanal durch einen (idealen) Server prinzipiell nicht verschlechtern!

**Maximaler Gewinn**

Wie den Abbildungen 2.6 und 2.7 anzusehen ist, sind die Kapazitäten der Teilkanäle  $C_1$  und  $C_2$  im Definitionsbereich offenbar immer streng monoton steigend bzw. fallend in Bezug auf die Fehlerrate  $p_1$  des ersten Kanals, leitet man also  $C_1$  und  $C_2$  nach  $p_1$  ab, so erhält man:

$$\begin{aligned} \frac{\partial}{\partial p_1} C_1 &= \log_2 \left( \frac{p_1}{1-p_1} \right) < 0 \\ \frac{\partial}{\partial p_1} C_2 &= \frac{1}{(2p_1-1)^2} \cdot \left[ (1-2p_G) \cdot \log_2 \left( \frac{p_1+p_G-1}{p_1-p_G} \right) \right] > 0 \\ &\forall p_1 \in [0..0,5[ \end{aligned}$$

Sucht man nun gemäß Gleichung 2.2 das Maximum der Gesamtkapazität, so ist das auf Grund der streng monotonen Funktionen  $C_1(p_1)$  und  $C_2(p_1)$  offensichtlich der Schnittpunkt dieser.

$$\begin{aligned} C_{max} &= \max_{p_1} C = \max_{p_1} \min \{ C_1, C_2 \} \\ &= C_1(p_M) = C_2(p_M) \\ &= 1 + p_M \log_2 p_M + (1 - p_M) \log_2 (1 - p_M) \end{aligned}$$

mit  $p_M$  gemäß Gleichung 2.8.

In Abbildung 2.8 ist nun diese maximal mögliche Kanalkapazität  $C_{max}$  und die Kanalkapazität ohne Server  $C_N$  als Funktion der Gesamt-Bitfehlerwahrscheinlichkeit geplotet.

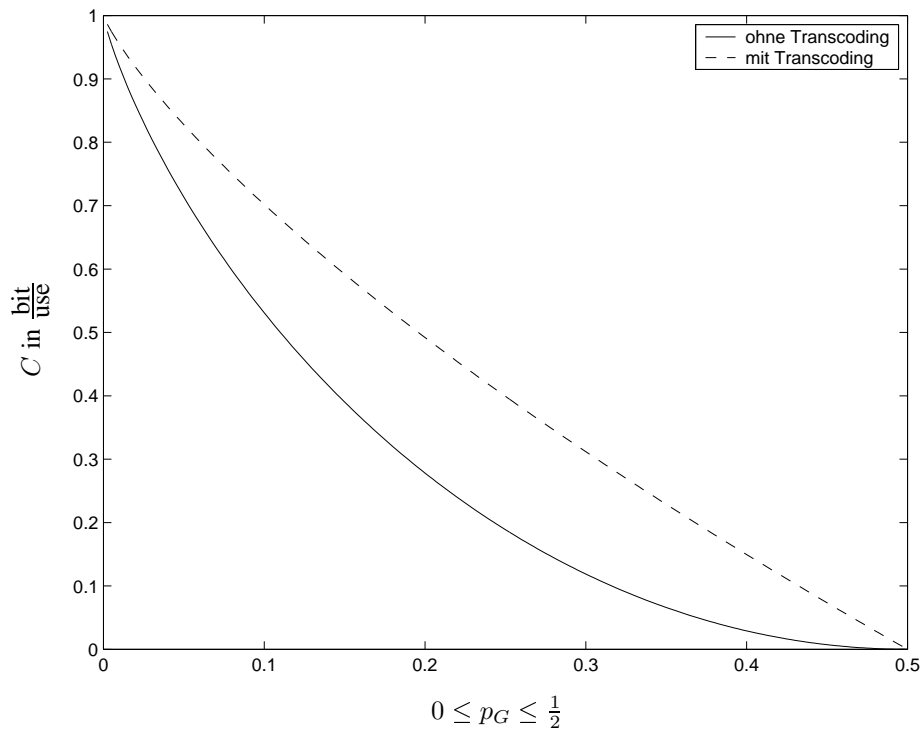


Abbildung 2.8: Maximal mögliche Kapazität mit *Transcoding*

**Maximaler Gewinn bei  $n$  Kanälen**

Erweitert man die Reihenschaltung von zwei BSC-Kanälen auf  $n$  BSC-Kanäle, so erhält man einen Gesamtkanal nach Abbildung 2.9.

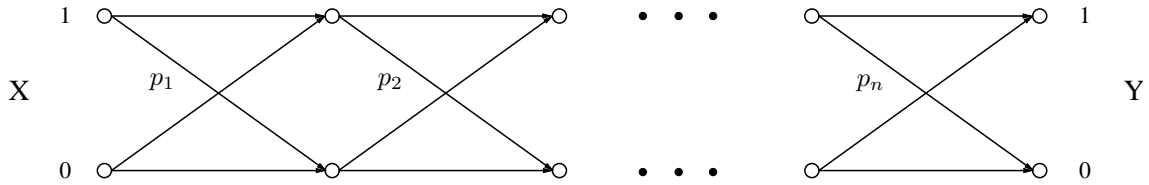


Abbildung 2.9: Reihenschaltung von  $n$  BSC-Kanälen

Die Gesamtkapazität mit *Transcoding* ist offensichtlich

$$C = \min \{C_1; C_2; C_3; \dots ; C_n\} \tag{2.9}$$

mit den einzelnen Kapazitäten

$$C_i = 1 + p_i \log_2 p_i + (1 - p_i) \log_2(1 - p_i) \tag{2.10}$$

und da 2.10 nach  $p_i$  streng monoton fallend ist wegen

$$\frac{\partial}{\partial p_i} C_i = \log_2 p_i - \log_2(1 - p_i) > 0 \quad \text{für} \quad 0 \leq p_i < \frac{1}{2}$$

wird Gleichung 2.9 dann maximiert, wenn alle Fehlerwahrscheinlichkeiten gleich sind, also  $p_1 = p_2 = \dots = p_n$  ist.

Dies ist begründet in der streng monotonen Eigenschaft der einzelnen Kapazitäten  $C_i$ , die keine lokalen Maxima zuläßt. Ebenso schließen sich Maxima aus, bei denen nicht alle Teilkanäle gleich groß sind, da durch einen abweichenden Teilkanal  $C_i$  die anderen Teilkanäle aus der Ruhelage  $p_j = p_M \quad j \neq i$  in die entgegengesetzte Richtung ausgelenkt werden.  $C$  ist also:

$$C = 1 + p_M \log_2 p_M + (1 - p_M) \log_2(1 - p_M)$$

Die normalen Kapazität ist einfach  $C_N = 1 + p_G \log_2 p_G + (1 - p_G) \log_2(1 - p_G)$  und wird auf die Berechnung der Gesamtfehlerrate  $p_G$  zurückgeführt mit

$$p_G = \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} p_M^{2i-1} (1 - p_M)^{n-2i-1} \binom{n}{2i+1} \tag{2.11}$$

wobei der Operator  $\lfloor \cdot \rfloor$  die Abrundung darstellt.

Mit Hilfe eines Algebra-Softwarepakets wie Maple kann Gleichung 2.11 nach  $p_M$  aufgelöst werden und man erhält als Ergebnisse:

$$\begin{aligned}
 n = 2 & : p_M = \frac{1}{2} - \frac{1}{2}\sqrt{1 - 2p_G} \\
 n = 3 & : p_M = \frac{1}{2} - \frac{1}{2}\sqrt[3]{1 - 2p_G} \\
 n = 4 & : p_M = \frac{1}{2} - \frac{1}{2}\sqrt[4]{1 - 2p_G} \\
 n = 5 & : p_M = \frac{1}{2} - \frac{1}{2}\sqrt[5]{1 - 2p_G} \\
 & \vdots
 \end{aligned}$$

Es ergibt sich somit folgende Abbildung für die maximale Kapazität, die durch das *Transcoding* erreicht werden kann, als Funktion der festen Gesamtfehlerrate  $p_G$ .

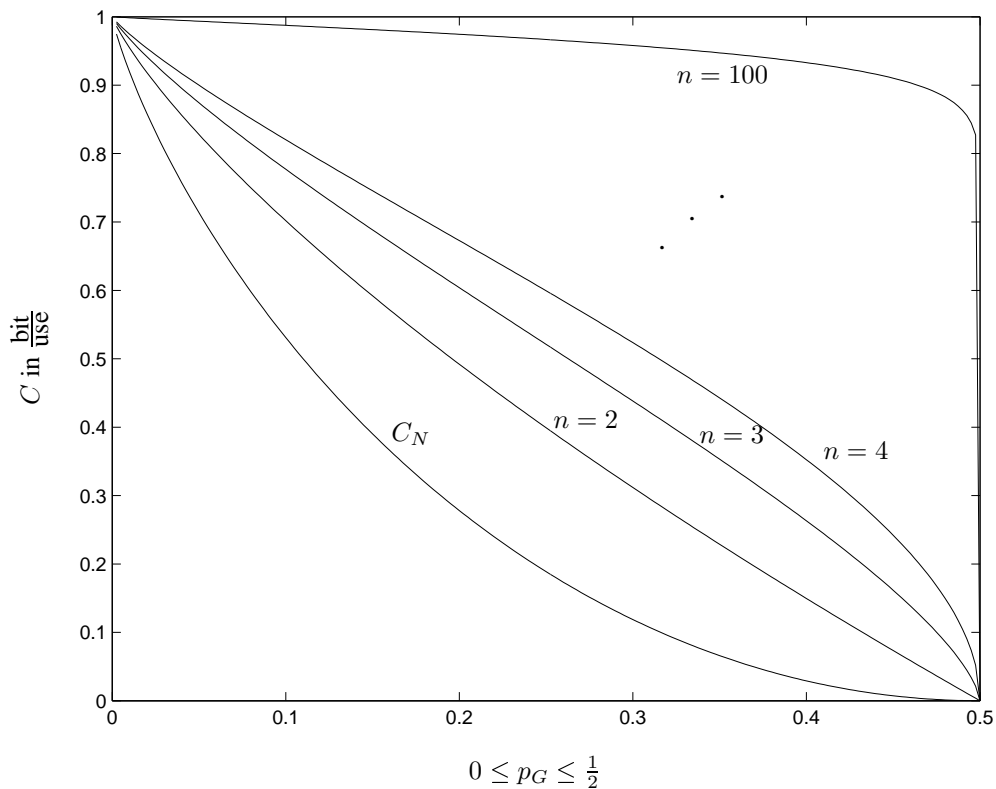


Abbildung 2.10: Kapazitäten von Reihenschaltungen von BSC-Kanälen

Hätten wir also unendlich viele Zwischenstationen, so könnten wir nach Abbildung 2.10 für jeden Wert von  $p_G$  absolut fehlerfrei übertragen. Dies widerspricht zwar unseren Erfahrungen aus dem Kinderspiel “Stille Post”, aber wir gehen ja bei unserem Modell von perfekten Zwischenstationen und nicht von Menschen aus.

### 2.2.3 Unsymmetrische binäre Kanäle

Der nächste Effekt, der uns nun interessiert, ist neben der reinen Fehlerrate die Symmetrie von Kanälen.

**Ein binärer unsymmetrischer Kanal, BnSC**

Dazu erweitern wir einfach das Modell des BSC-Kanals auf binäre unsymmetrische Kanäle. Wir haben somit, wie in Abbildung 2.11 zu sehen ist, unterschiedliche Fehlerraten für das Eingangssymbol  $X = 0$  und  $X = 1$  jeweils gekennzeichnet durch den Index.

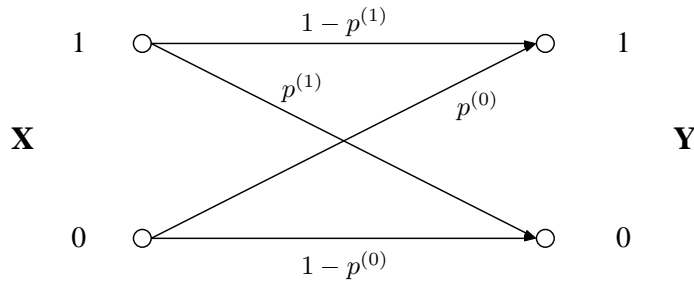


Abbildung 2.11: binärer nicht symmetrischer Kanal

Aber schon dieses einfache Modell macht die analytische Bestimmung seiner Kapazität nahezu unmöglich. Die Vorgehensweise entspricht der beim BSC-Kanal, allgemein nach der Formel 2.3 oder speziell nach der des BSC-Kanals 2.4. Zuerst die Berechnung der Transformation zwischen  $X$  und  $Y$  und dann die Maximierung dieser durch Variation von  $\Pr(X = 1)$ . Allerdings ist bereits diese Formel, die durchaus noch mit Algebra-Programmen wie Maple gewonnen werden kann, derart groß, daß sie nicht auf eine Seite passen würde. Eine Kurvenschar von Kapazitäten des binären unsymmetrischen Kanals zeigt Abbildung 2.12, in der  $p^{(1)}$  als Parameter dient und  $p^{(0)}$  die Veränderliche ist. Zum Vergleich ist die bekannte Kapazität des BSC angetragen.

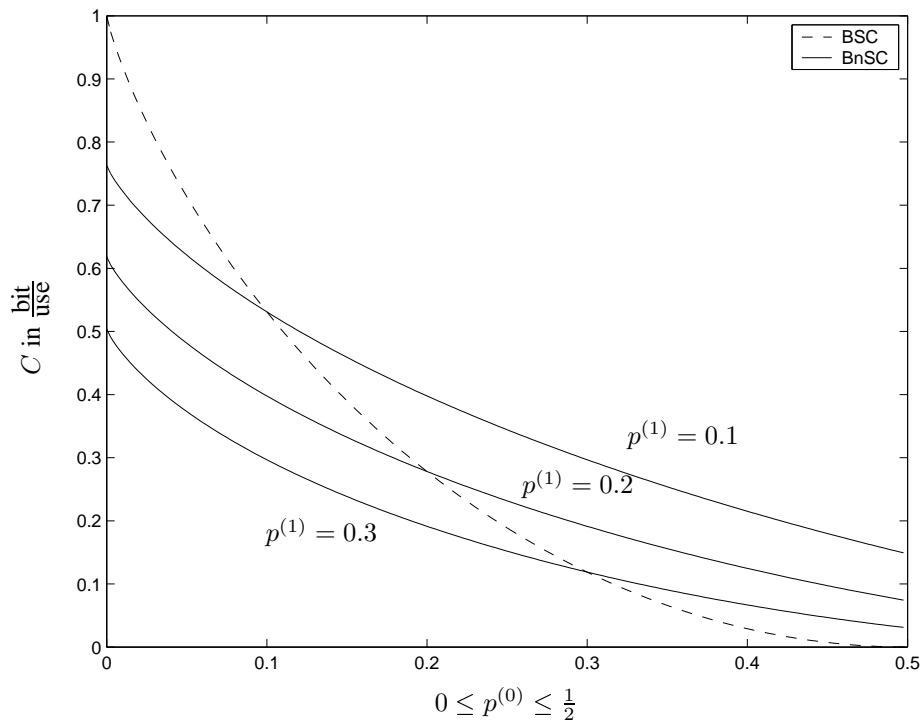


Abbildung 2.12: Kapazität des binären nicht symmetrischen Kanals



**Zwei binäre unsummetrische Kanäle, BnSC's**

Die Gesamtfehlerrate von zwei in Reihe geschalteten BnSC's kann nach den Gleichungen 2.12 und 2.13 berechnet werden.

$$p_G^{(0)} = p_1^{(1)}(1 - p_2^{(0)}) + (1 - p_1^{(1)})p_2^{(1)} \tag{2.12}$$

$$p_G^{(1)} = (1 - p_1^{(0)})p_2^{(0)} + p_1^{(0)}(1 - p_2^{(1)}) \tag{2.13}$$

Womit man dann die Gesamtkapazität ohne *Transcoding*  $C_N$  einfach durch das Minimum der beiden Teilkapazitäten bestimmen kann.

$$C = \min \left\{ C_1(p_1^{(0)}, p_1^{(1)}); C_2(p_2^{(0)}, p_2^{(1)}) \right\}$$

Um die beiden nun auf die bisherige Art und Weise zu vergleichen, ist zu beachten, daß man zwei feste Größen als Konstanten vorgeben muß, und zwar  $p_G^{(0)}$  und  $p_G^{(1)}$  und, daß man eine Funktion von zwei voneinander unabhängige Variablen  $p_1^{(0)}$  und  $p_1^{(1)}$  bekommt.

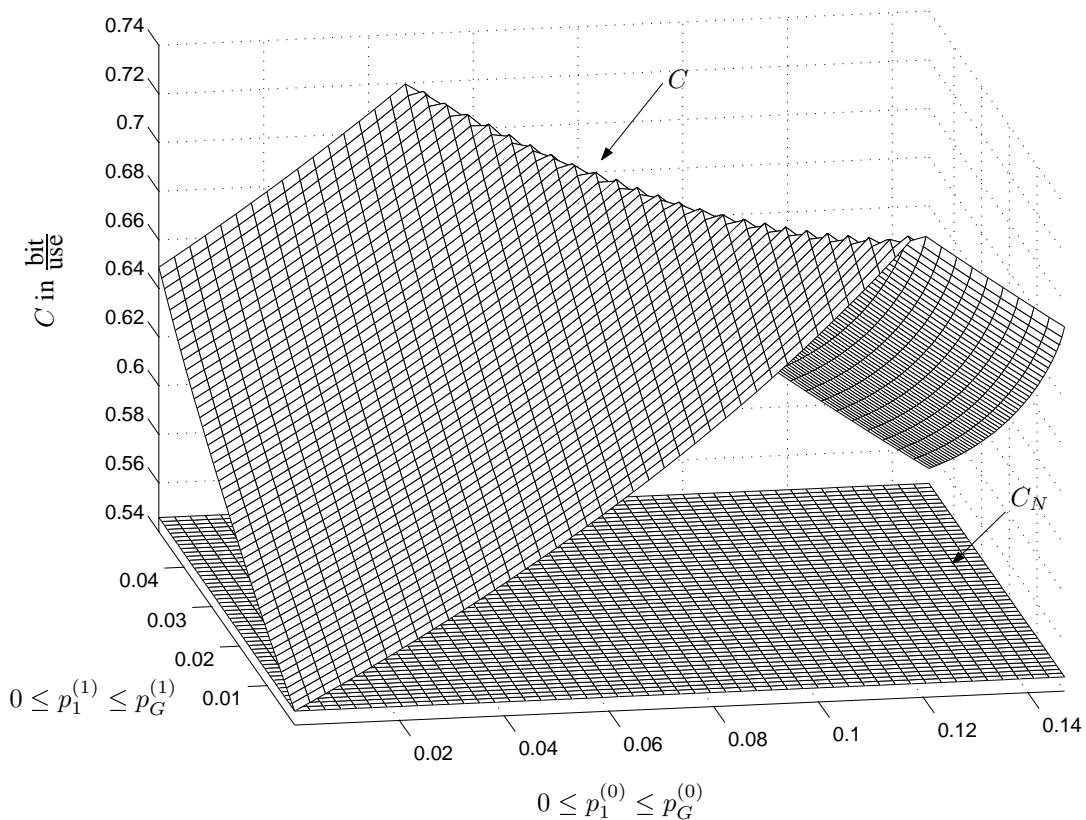


Abbildung 2.13: Kapazität des BnSC mit ( $C$ ) und ohne ( $C_N$ ) *Transcoding*

Es sind in Abbildung 2.13 wie schon in den vorangegangenen Kapiteln die Kapazitäten  $C$  und  $C_N$  des Kanals geplottet worden, unter der Annahme, daß die Gesamtfehlerraten ( hier sind es ja zwei) konstant sind. Variiert wurden die Fehlerraten des ersten Kanals.

Die Ergebnisse entsprechen im Großen und Ganzen auch denen aus Kapitel 2.2.2, mit der Ausnahme, daß das Maximum der Kapazität  $C$  für besonders nicht symmetrische Teilkanäle erreicht wird. Wie in Abbildung 2.13 zu sehen, bildet die Kurve einen Sattel und erreicht ihr Maximum für  $p_1^{(1)} = 0$ . Der zweite Kanal muß in diesem Fall gemäß Gleichungen 2.12 und 2.13 natürlich ebenfalls unsymmetrisch sein, allerdings in der jeweils anderen Richtung, das heißt  $p_2^{(1)}$  ist nicht minimal sondern maximal und für  $p_2^{(0)}$  gilt das Analoge.

Bei Anti-Symmetrischen Teilkanälen ist der Gewinn durch *Transcoding* also noch größer als bei reinen fehlerbehafteten Kanälen.

### Maximaler Gewinn

Wie schon in Kapitel 2.2.2 für den BSC, kann auch für den BnSC der maximale Gewinn durch *Transcoding*  $C_{max}$  als Funktion der Bitfehlerraten  $p_G^{(0)}$  und  $p_G^{(1)}$  ermittelt werden, und im Vergleich zu  $C_N$  geplottet werden. Abbildung 2.14 zeigt die numerisch gefundenen Werte.

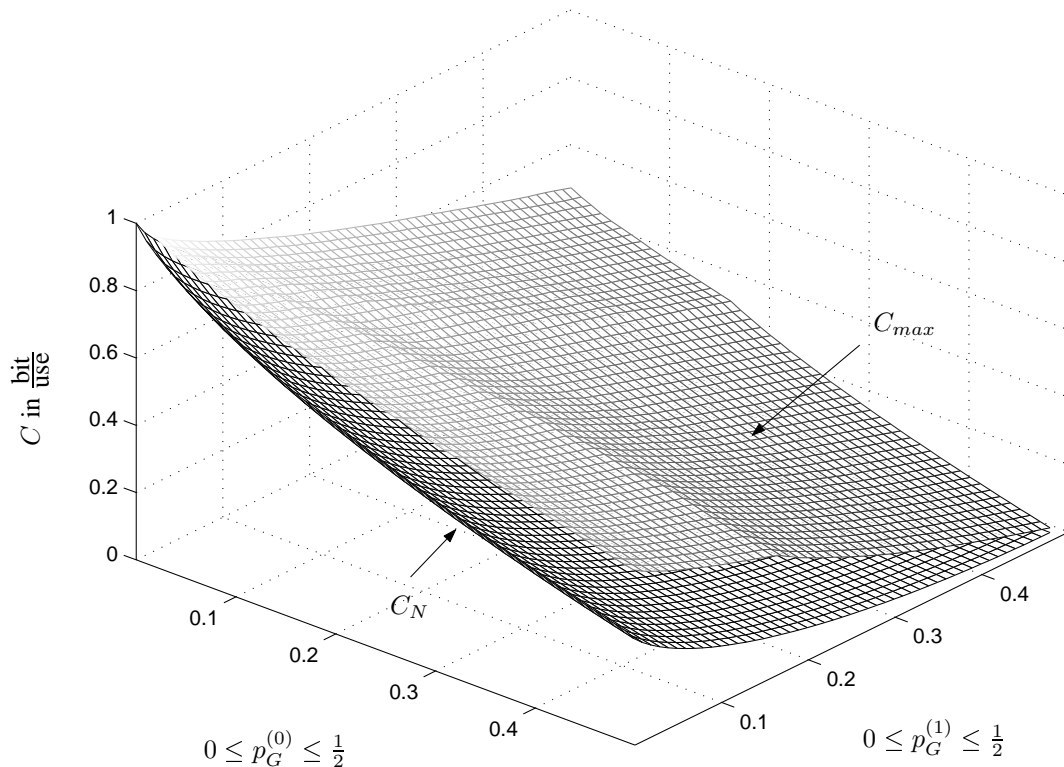


Abbildung 2.14: maximale Kapazität des BnSC mit *Transcoding*

Beachtenswert ist, daß insbesondere auch für symmetrische Gesamtkanäle die Kapazität bei der Übertragung mit *Transcoding* durch antisymmetrische Teilkanäle und nicht etwa durch symmetrische Teilkanäle optimiert wird.

### 2.2.4 Erasure-Kanäle

Eine gerade für das Internet sehr gutes Modell ist das sogenannte *Erasure-Modell*, das keine Bitfehler kennt, sondern nur Auslöschungen ( Englisch: *erasures*). Sein großer Vorteil ist, daß

es auch für nicht binäre Eingangsvariablen  $X$  praktikabel bleibt.

**Der  $M$ -ary Erasure Channel  $M$ -ary EC**

Die Eingangsvariable  $X$  muß jetzt nicht mehr binär sein, sondern kann allgemein  $M = \|X\|$  Werte annehmen. Fehler, die bei der Übertragung auftreten, werden immer sicher detektiert und angezeigt, die Fehlerwahrscheinlichkeit  $p$  ist unabhängig vom Eingangssymbol  $x_i$ . Bild 2.15 verdeutlicht das Modell nochmals.

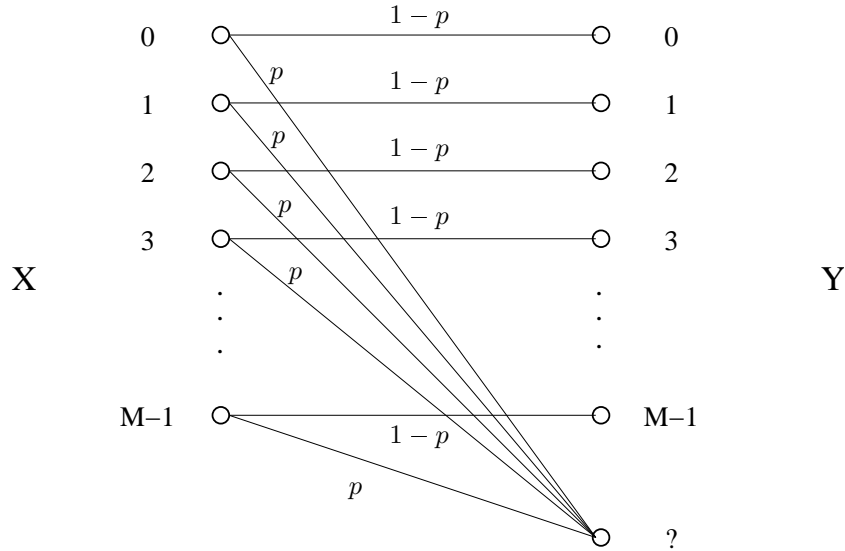


Abbildung 2.15: Der  $M$ -ary Erasure Channel

Die Kapazität kann sehr einfach berechnet werden, da es sich um einen symmetrischen (vgl. [7]) Kanal handelt, seine Übergangsmatrix  $\mathbf{K}$  läßt sich in eine gewichtete Summe von streng symmetrischen Teilkanälen, die gleichmäßig streuend und sammelnd sind, zerlegen.

$$\mathbf{K} = \begin{pmatrix} 1-p & 0 & 0 & \dots & 0 & p \\ 0 & 1-p & 0 & \dots & 0 & p \\ 0 & 0 & 1-p & \dots & 0 & p \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1-p & p \end{pmatrix} = (1-p) \cdot \underbrace{\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}}_{=S_1} + p \cdot \underbrace{\begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}}_{=S_2}$$

Die Kapazität des Gesamtkanals berechnet sich nach [7] durch die Addition der gewichteten Kapazitäten der Teilkanäle.

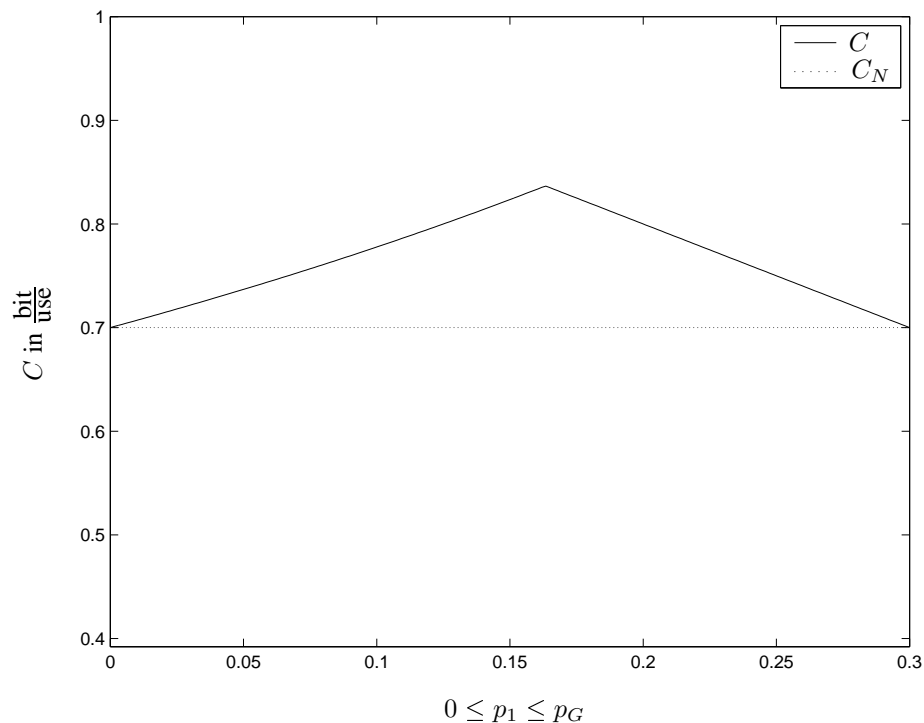
$$C = (1-p) \cdot C_1 + p \cdot C_2 = (1-p) \cdot \log_2 M + p \cdot 0 = (1-p) \log_2 M \tag{2.14}$$

**Zwei  $M$ -ary Erasure Channels  $M$ -ary EC's**

Die Kapazitäten mit und ohne *Transcoding* lassen sich besonders einfach berechnen, zu beachten ist nur, daß beim Gesamtkanal genau dann eine Auslöschung vorliegt, wenn der erste oder der zweite Kanal eine Auslöschung hatten, es handelt sich dabei um ein echtes Oder und nicht um ein exklusives Oder wie beim BSC.

$$\begin{aligned}
 C &= \min \{C_1; C_2\} \\
 &= \min \{(1 - p_1) \log_2 M; (1 - p_2) \log_2 M\} \\
 &= (1 - \max \{p_1; p_2\}) \log_2 M \\
 C_N &= (1 - p_G) \log_2 M \\
 &= (1 - p_1 - p_2 + p_1 p_2) \log_2 M \\
 &= (1 - p_1)(1 - p_2) \log_2 M
 \end{aligned}$$

Abbildung 2.16 zeigt die gewohnte Darstellung der Kapazitäten bei einer Gesamtfehlerrate von  $p_G = 0.3$ . Die Darstellung des maximalen Gewinns ergibt denen aus Kapitel 2.2.2 sehr ähnliche Kurven, dieser wird aber bei  $p_M = 1 - \sqrt{1 - p_G}$  erreicht.

Abbildung 2.16: Kapazität des  $M$ -ary EC**Unterschiedliche Blocklängen  $M$** 

Das  $M$ -ary EC-Modell eignet sich sehr gut für paketorientierte Übertragungskanäle mit Paketen der Größe  $\log_2 M$  bit und Paketfehlern. Dabei kann aber der Fall von Paketen unterschiedlicher

Größe nicht außer acht gelassen werden, da dies eher der Normalfall ist.

Zunächst muß man versuchen die Kapazität in einer anderen Einheit darzustellen, da die Einheit  $\frac{\text{bit}}{\text{use}}$  nicht mehr eindeutig ist. Es bietet sich an Kapazitäten in bit pro Zeit zu messen, also  $\frac{\text{bit}}{\text{s}}$ . Gleichung 2.14 muß folgendermassen modifiziert werden:

$$C_T = (1 - p) \cdot \frac{\log_2 M}{T} \quad (2.15)$$

wobei  $T$  die Zeit pro Kanalbenutzung ist und somit die Einheit  $\frac{\text{s}}{\text{use}}$  hat, womit sich die passende Einheit für  $C_T$  ergibt.

Die Berechnung von  $C_T$  für zwei Kanäle ist unproblematisch, bei der Berechnung der “normalen” Kapazität  $C_{T,N}$  muß man allerdings beachten, daß einer von beiden Kanälen wohl das Nadelöhr ist und der Bessere sich mit seiner Datenübertragungsrate an den Schlechteren anpassen muß.

$$C_T = \min \left\{ (1 - p_1) \frac{\log_2 M_1}{T_1}; (1 - p_2) \frac{\log_2 M_2}{T_2} \right\}$$

$$C_{T,N} = (1 - p_1)(1 - p_2) \min \left\{ \frac{\log_2 M_1}{T_1}; \frac{\log_2 M_2}{T_2} \right\}$$

Je nach Beschaffenheit der Kanäle können sich durch diese geänderten Formeln massive Unterschiede ergeben. Ist zum Beispiel der eine Kanal stark fehlerbehaftet, dafür aber mit einer hohen Datenrate, und der andere Kanal extrem zuverlässig, dafür aber auch sehr langsam, so kombinieren sich ohne *Transcoding* die Nachteile, sie multiplizieren sich im wahrsten Sinne des Wortes, währenddessen sie sich mit *Transcoding* ausgleichen würden. Ist ein Kanal hingegen bei beiden Faktoren, Fehlerrate und maximaler Datendurchsatz, der Schlechtere, so ändert sich gegenüber den bisherigen Formeln kaum etwas.

## 2.3 Kanäle mit Gedächtnis

Gerade für den Mobilfunkkanal ist das Kanalgedächtnis ein entscheidender Faktor, und da der Leitspruch “**Memory increases Capacity**” (vgl. [7]) gilt, ist zu hoffen, daß durch die Berücksichtigung des Kanalgedächtnisses die Vorteile des *Transcoding* für das *streamen* auf das Handy noch verstärkt werden.

### 2.3.1 Markovmodelle

Die beste Methode um ein Kanalgedächtnis zu modellieren ist ein zugrundeliegendes Markovmodell. Bereits 1960 und 1963 von E. N. Gilbert und E. O. Elliott in einer einfachen Form eingeführt (vgl. [1]) wurden die auf Markov basierenden Kanalmodelle immer mehr verfeinert und als Modelle für Kanäle mit Gedächtnis verifiziert, wie zum Beispiel 1978 in [3]. Auch in jüngster Vergangenheit wurden sie immer wieder herangezogen, wie beispielsweise 1996 von Hong Shen Wang (in [4]), der ihre Gültigkeit für das weit verbreitete Rayleigh Fading Kanalmodell untersuchte und im Jahre 2001, in dem in [5] die Tauglichkeit derartiger Modelle für

kabellose Netzwerke eingesetzt wurde.

Die Basis ist eine irreduzible, aperiodische und stationäre Markovkette mit endlich vielen Zuständen  $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$ , weshalb derartige Kanalmodelle auch den Namen *Finite-State Markov Channels* FSMC's tragen (vgl. [2]). Die Markov-Kette selbst ist außerdem durch ihre Übergangsmatrix  $P$  bestimmt, deren Elemente die Wahrscheinlichkeit für den Zustand  $S_{n+1}$  unter der Bedingung des vorherigen Zustands  $S_n$  darstellen.

$$P_{km} = \Pr(S_{n+1} = c_m | S_n = c_k)$$

Die Markovkette hat somit folgende Form:

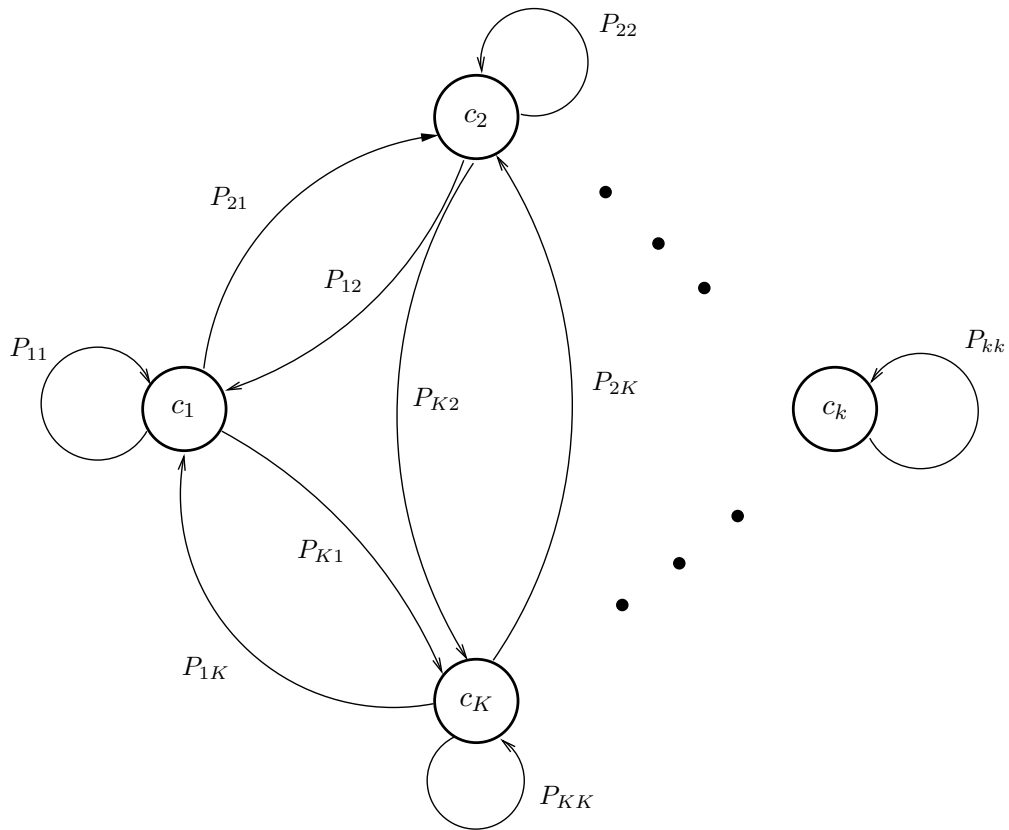


Abbildung 2.17: Markovkette

Zur Bildung des Kanalmodells, wird jedem Zustand  $c_k \in \mathcal{C}$  ein diskreter gedächtnisloser Kanal mit dem gleichen endlichem Ein- und Ausgangsalphabet  $\mathcal{X}$  und  $\mathcal{Y}$  zugeordnet. Dieser kann sehr allgemein sein und wird durch seine bedingten Übergangswahrscheinlichkeiten bestimmt.

$$\Pr(y_n | x_n, S_n)$$

$$x_n \in \mathcal{X}, y_n \in \mathcal{Y}$$

### 2.3.2 BSC-Modell

Zur Modellierung eines gedächtnisbehafteten BSC's reicht es aus von zwei Zuständen auszugehen ( vgl. [4]). Wir bedienen uns eines Gilbert-Elliott Kanals.

#### Ein Gilbert-Elliott Kanal

Es handelt sich um einen Kanal mit zwei Zuständen,  $c_1 = B$  für *Bad* und  $c_2 = G$  für *Good*, mit den Übergangswahrscheinlichkeiten  $b$  und  $g$ . Beiden Zuständen wird ein BSC mit der Fehlerwahrscheinlichkeit  $p_B$  bzw.  $p_G$  zugeordnet. Das Modell ist nochmals in Abbildung 2.18 zu sehen, genauere Definitionen und die Herleitung der Kapazität sind Anhang A zu entnehmen.

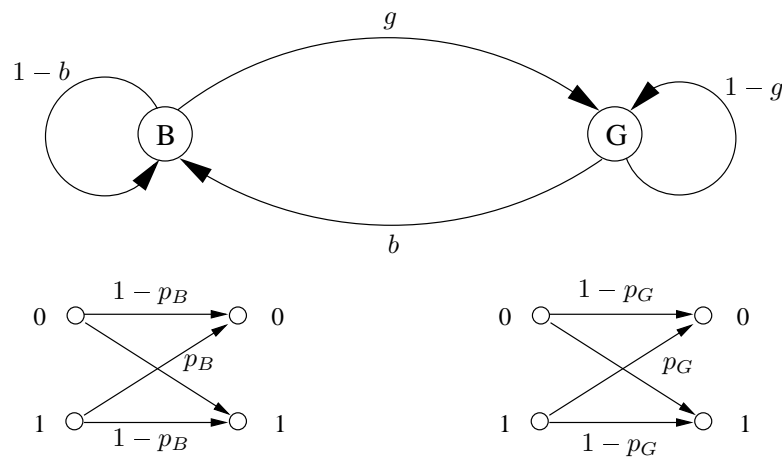


Abbildung 2.18: Gilbert-Elliott Kanalmodell

Das Gilbert-Elliott Kanalmodell ist schon lange bekannt und gut untersucht, wie beispielsweise in [1], nun gilt es dieses Modell in eine Reihenschaltung einzubauen.

#### Zwei gekoppelte Gilbert-Elliott Kanäle

Um die Auswirkungen von Gedächtnis auf das *Transcoding* allgemein zu untersuchen, werden zunächst zwei Gilbert-Elliott Kanäle in Reihe geschaltet. Auf Grund der Unabhängigkeit der beiden Teilkanäle entsteht offensichtlich ein FSMC mit vier verschiedenen Zuständen und vier verschiedenen Fehlerwahrscheinlichkeiten.

Zustand	Fehlerwahrscheinlichkeit
$BB = B_1 \wedge B_2$	$p_{BB} = (1 - p_{B1})p_{B2} + p_{B1}(1 - p_{B2})$
$BG = B_1 \wedge G_2$	$p_{BG} = (1 - p_{B1})p_{G2} + p_{B1}(1 - p_{G2})$
$GB = G_1 \wedge B_2$	$p_{GB} = (1 - p_{G1})p_{B2} + p_{G1}(1 - p_{B2})$
$GG = G_1 \wedge G_2$	$p_{GG} = (1 - p_{G1})p_{G2} + p_{G1}(1 - p_{G2})$

Für eine Abbildung dieses zusammengefassten Kanalmodells sei auf die Abbildung B.1 in Anhang B verwiesen. Die genaue Herleitung der rekursiven Berechnung der Kapazität befindet sich ebenfalls in Anhang B, hier werden im weiteren nur die Ergebnisse dieser Berechnung diskutiert.

### Mobilfunkkanal mit Gedächtnis

Als erstes wollen wir das gesamte Gedächtnis dem zweiten Kanal zuweisen, um zu untersuchen, ob sich die Vermutungen am Anfang dieses Kapitels bestätigen.

Der erste Kanal sei also gedächtnislos, mit einer geringen Fehlerrate von  $p_1 = 0.05$ . Der zweite Kanal hat im Verhältnis zum ersten eine deutlich höhere Fehlerrate von  $p_2 = 0.4$ , was sich an das Verhältnis von Internet zu Mobilfunk anlehnen soll, und er hat Gedächtnis. Dieses wird von  $\mu_2 = 0$  (kein Gedächtnis) bis  $\mu_2 = 1$  (deterministische Abhängigkeit zwischen den Zuständen) variiert. Mit den festgesetzten Kanalübergangswahrscheinlichkeiten  $p_B = 0.5$  und  $p_G = 0.01$  (die starken Unterschiede zwischen den Größen  $p_B$  und  $p_G$  erzeugen große Auswirkungen von  $\mu_2$ ), die für beide Kanäle gleich sind, können daraus alle erforderlichen Größen für das Gilbert-Elliott-Modell gebildet werden.

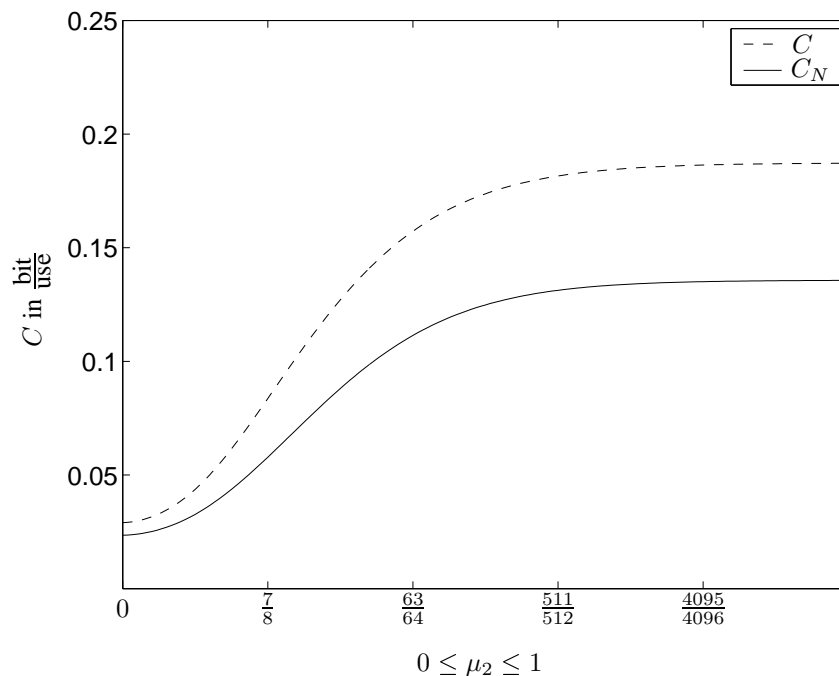


Abbildung 2.19: Kapazität mit und ohne *Transcoding* mit  $\mu_1 = 0$

Wie in Abbildung 2.19 zu sehen, stimmt die Vermutung, daß das Kanalgedächtnis des zweiten Kanals die Vorteile von *Transcoding* noch verstärkt. Offenbar profitieren beide Übertragungstechniken vom Gedächtnis des zweiten Kanals, beim Durchschleifen erhöht sich Gesamtkapazität  $C_N$  ein wenig, beim *Transcoding* allerdings beschreibt die Kurve von  $C$ , auf Grund der hohen Fehlerrate des zweiten Kanals  $p_2$ , genau den schlechteren Kanal und somit die geringere Kapazität  $C_2$ . Die Kapazität des gesamten Kanals  $C$  kann also im vollen Umfang vom Gedächtnis des zweiten Kanals profitieren und nicht nur zu einem gewissen Teil, wie beim Durchschleifen.

Um diese Anhängigkeit nochmals zu verdeutlichen ist in Abbildung 2.20 der Gewinn durch *Transcoding* für die obigen Einstellungen dargestellt. Zu beachten ist, daß der Gewinn nur deshalb so groß ausfällt, weil sehr hohe Fehlerraten ( $p_1 = 0.05$  und  $p_2 = 0.4$ ) gewählt wurden. Realistischere Werte für die Gewinne durch *Transcoding* beim Streaming über Internet und Mobilfunkkanal sind Kapitel 2.4 zu entnehmen.



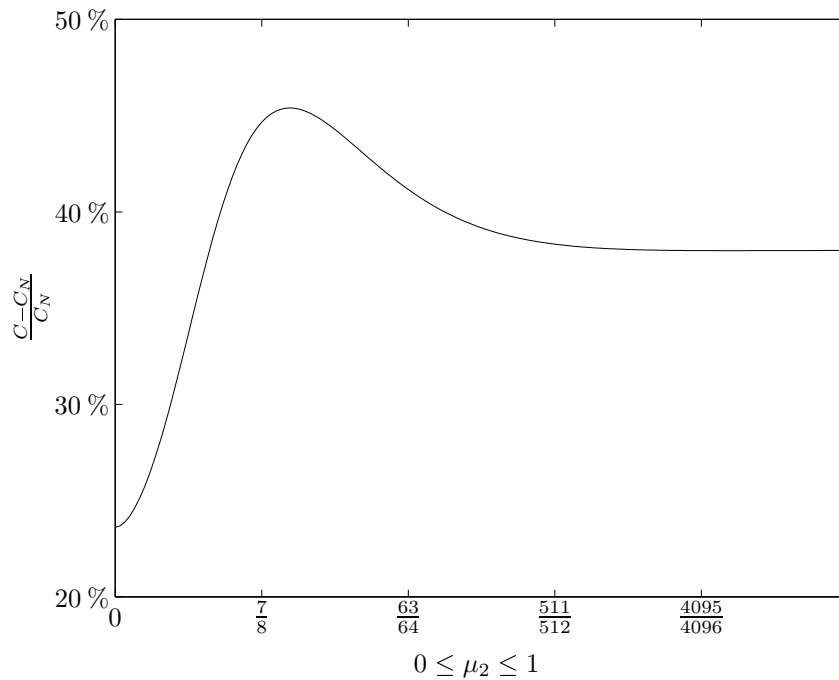


Abbildung 2.20: relativer Gewinn durch *Transcoding* bei obigen Werten

### Optimale Verteilung von Gedächtnis

Nun wollen wir die optimale Verteilung von Kanalgedächtnis auf die beiden Unterkanäle ermitteln. Wir bedienen uns hierzu einer Darstellungsweise, wie sie schon in Kapitel 2.2.2, 2.2.3 und 2.2.4 eingesetzt wurde: Zwei Teil-Kanäle seien weitestgehend identisch mit Ausnahme einer zu untersuchenden Größe, hier das Kanalgedächtnisses  $\mu$ . Diese nimmt im ersten Kanal alle möglichen Werte an, hier also  $0 \leq \mu_1 \leq \mu_G$ , während diesselbe Größe im zweiten Kanal (also  $\mu_2$ ) so angepasst wird, daß sich ein Gesamtkanal mit konstanten Eigenschaften, also insbesondere eines konstanten Kanalgedächtnisses  $\mu_G$  ergibt.

Da es aber im Gegensatz zu den Kapiteln 2.2.2, 2.2.3 und 2.2.4 keine triviale Möglichkeit gibt  $\mu_2$  aus  $\mu_1$  und  $\mu_G$  zu ermitteln, wurde in Abbildung 2.21  $\mu_2$  durch die Suche nach der konstanten Kapazität ohne *Transcoding*  $C_N$  ermittelt.  $\rho_1$  und  $\rho_2$  sind gleich und wurden so eingestellt, daß sich mit  $p_B = 0.5$  und  $p_G = 0.01$  Fehlerraten von  $p_1 = p_2 = 0.2$  einstellen, was eine Gesamtfehlerrate von  $p_G = 0.32$  ergibt.

Abbildung 2.21 sind nun einige Effekte zu entnehmen:

- die Gesamtkapazität  $C$  wird offenbar für gleiche Teilkapazitäten  $C_1$  und  $C_2$  und somit für gleiche Gedächtnisse  $\mu_1$  und  $\mu_2$  maximiert, dies entspricht der gleichmäßigen Verteilung der Fehlerrate
- die Kapazität mit *Transcoding*  $C$  ist immer größer als ohne *Transcoding*, sie ist in dieser Abbildung sogar echt größer, was durch die Fehlerraten der Teilkanäle bedingt ist
- $C_{NM}$ , das gemessen wird, wenn das Kanalgedächtnis außer acht gelassen wird, ist wie in Anhang A eine untere Grenze für  $C$  ( beide Kapazitäten wurden natürlich mit *Transcoding* ermittelt)

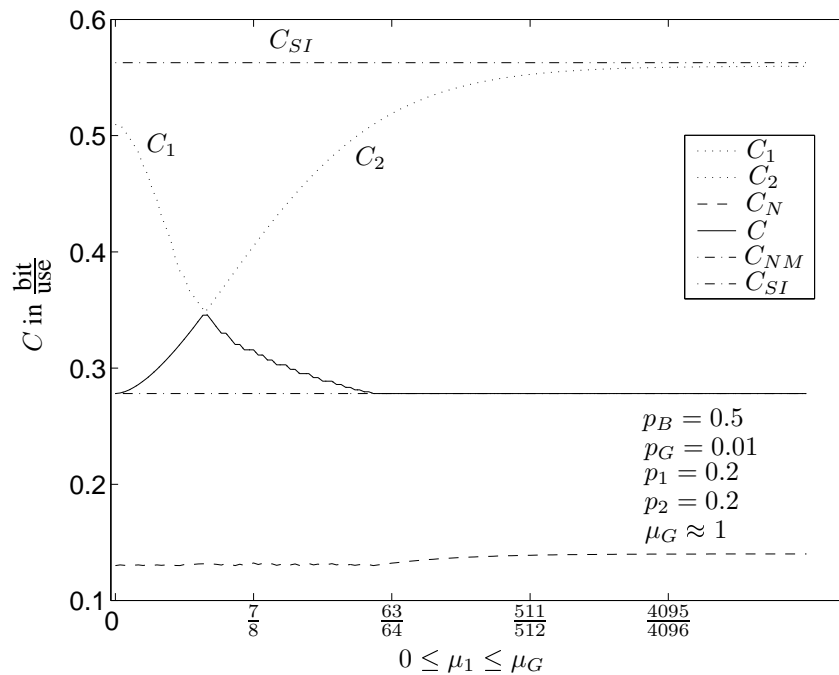


Abbildung 2.21: Abhängigkeit der Kapazitäten von der Verteilung des Gedächtnisses

- sind die Kanalzustände bekannt, so ergibt sich die Kapazität  $C_{SI}$  ( auch wieder analog zu Anhang A), die eine obere Grenze für  $C$  darstellt ( wiederum wurden beide Kapazitäten mit *Transcoding* ermittelt)
- $C_{SI}$  ist interessanterweise nicht abhängig von der Verteilung des Kanalgedächtnisses auf die beiden Teilkanäle, sind also die aktuellen Zustände beider Kanäle bekannt, so gibt es keinen zusätzlichen Vorteil durch *Transcoding* bei gedächtnisbehafteten Kanälen

### Maximierung durch gleiche Kapazitäten

Die Frage die sich auf Grund der obigen Ergebnisse stellt, ist, ob es vorteilhafter ist, beide Eigenschaften der Kanäle, Fehlerrate und Gedächtnis, gleich zu verteilen, oder ob es nur darauf ankommt, daß beide Teilkanäle die gleiche Kapazität haben.

Zur Untersuchung dieser Frage verwenden wir folgende Anordnung: von den beiden Gilbert-Elliott Kanälen eines Gesamtkanals ist der erste, sowohl mit seiner Fehlerrate  $p_1$ , als auch mit seinem Kanalgedächtnis  $\mu_1$  konstant. Das Kanalgedächtnis  $\mu_2$  des zweiten Kanals wird von 0 bis 1 variiert, dabei wird dessen Fehlerrate  $p_2$  so eingestellt, daß sich die Kapazität ohne *Transcoding*  $C_N$  nicht ändert. Erhalten wir nun in Abbildung 2.22 irgendwo ein Maximum für  $C$ , so gibt es offenbar Bedingungen anderer Art zur Maximierung der Kapazität mit *Transcoding*, andernfalls nicht.

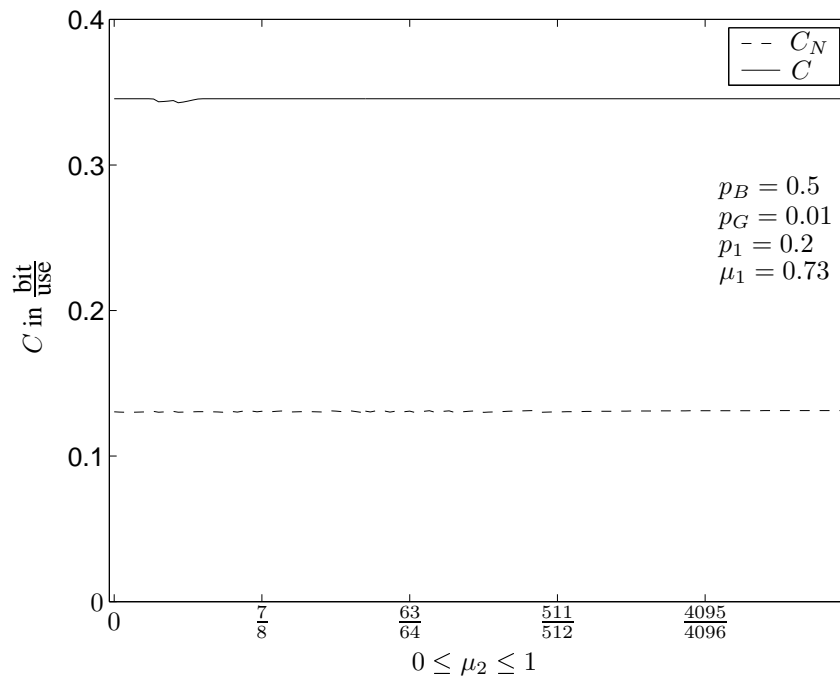


Abbildung 2.22: Kapazitäten, wenn der erste Kanal konstant ist

Die Kapazität mit *Transcoding* bleibt offenbar konstant.

⇓

Der Gewinn durch den Einbau eines Servers und durch die Umcodierung der Kanalcodes (*Transcoding*) ist umso größer, je kleiner die Gesamtkapazität ist, und je gleichmässiger diese auf die beiden Teilkanäle verteilt ist. Dabei ist es unerheblich, ob sich die Kanäle in all Ihren Eigenschaften gleichen.

### 2.3.3 Erasure-Modell

Wegen der großen praktischen Bedeutung von Blockfehlermodellen, wollen wir in diesem Kapitel das M-ary EC Modell aus Kapitel 2.2.4 mit Hilfe einer Markovkette zu einem sogenannten MEC-Modell (*Markov Erasure Channel*, vgl. [6]) erweitern.

#### Der Markov Erasure Channel MEC mit zwei Zuständen

Der einfachste Vertreter dieser Gattung ist ein MEC mit zwei Zuständen: die Markovkette ist identisch mit der aus Abbildung A.1 in Anhang A.1. Die den beiden Zuständen *Good* und *Bad* zugeordneten diskreten gedächtnislosen Kanäle sind allerdings M-ary EC's. Dabei wird dem Zustand *Bad* die Auslöschung und dem Zustand *Good* die sichere Übertragung des Pakets zugeordnet.

Eine derartige Fehlercharakteristik wird unifiliare Markovfehlerquelle (vgl. [7]) genannt, da vom empfangenen Codewort immer auf den Zustand des Kanals geschlossen werden kann. Es liegt somit eine perfekt Kenntnis des Kanalzustandes vor, womit die Kapazität des Kanals durch

die Erwartungswertbildung der Kapazitäten der einzelnen Zustände  $C(c_k)$  berechnet werden kann.

$$C = C_{SI} = \sum_{k=1}^K \Pr(S = c_k) \cdot C(c_k)$$

Speziell für zwei Zustände und das M-ary EC-Modell als Grundlage gilt (vergleiche auch Anhang A):

$$\begin{aligned} C &= \Pr(S = B) \cdot 0 + \Pr(S = G) \cdot \log_2 M \\ &= \frac{g}{b+g} \cdot \log_2 M \\ &= (1 - \bar{p}) \log_2 M \end{aligned} \tag{2.16}$$

Die Formel entspricht also weitestgehend der für einen M-ary EC in Gleichung 2.14, wobei  $\bar{p}$  die durchschnittliche Fehlerrate des Kanals ist.

### Zwei Markov Erasure Channels

Wenn man zwei solche Kanäle in Reihe schaltet, müßte man eigentlich wiederum eine Markovkette mit vier Zuständen einführen und diese rekursiv berechnen. Allerdings ist dies durch die unifiliare Natur des Kanals nicht nötig. Kommt ein Block beim Empfänger an, so ist bekannt, ob er richtig ist oder nicht. Für die Ermittlung der Kapazität ist es unerheblich zu wissen, mit welcher Wahrscheinlichkeit der nächste Block richtig ist oder nicht. Es zählt, wie oben schon, nur die Tatsache das der aktuelle Block richtig ist und mit welcher Wahrscheinlichkeit er richtig ist.

⇓

$$C_N = \mathcal{E} \{C(c_k)\} = (1 - \bar{p}_1)(1 - \bar{p}_2) \log_2 M$$

Mit der aus Gleichung 2.16 bereits bekannten Formel für die Kapazität mit *Transcoding*  $C$  als Minimum der beiden Teilkapazitäten hat man praktisch identische Gleichungen wie bei den M-ary EC Modellen.

## 2.4 Verknüpfungen verschiedenartiger Kanäle

Mit den so eingeführten Kanalmodellen können wir uns wieder dem eigentlichen Problem widmen und ein den tatsächlichen Gegebenheiten des *Streamings* von Multimediainhalten auf Handys gut angepasstes Kanalmodell bilden.

### 2.4.1 Das verknüpfte Kanalmodell

Dazu wird als Modell für das Internet das MEC-Modell und für den Mobilfunkkanal ein Gilbert-Elliott Modell herangezogen. Dies hat den Vorteil, daß beide Modelle auf Markovketten basieren und somit analog Kapitel 2.3.1 zu einer Markovkette mit vier Zuständen und vier diskreten gedächtnislosen Kanalmodellen vereint werden können. Das Modell ist in Abbildung 2.23 zu sehen, B und G stehen wie beim Gilbert-Elliott Modell für *Bad* und *Good*, das E bzw.  $\bar{E}$  steht für Auslöschung ( Englisch *Erasure*) bzw. keine Auslöschung.

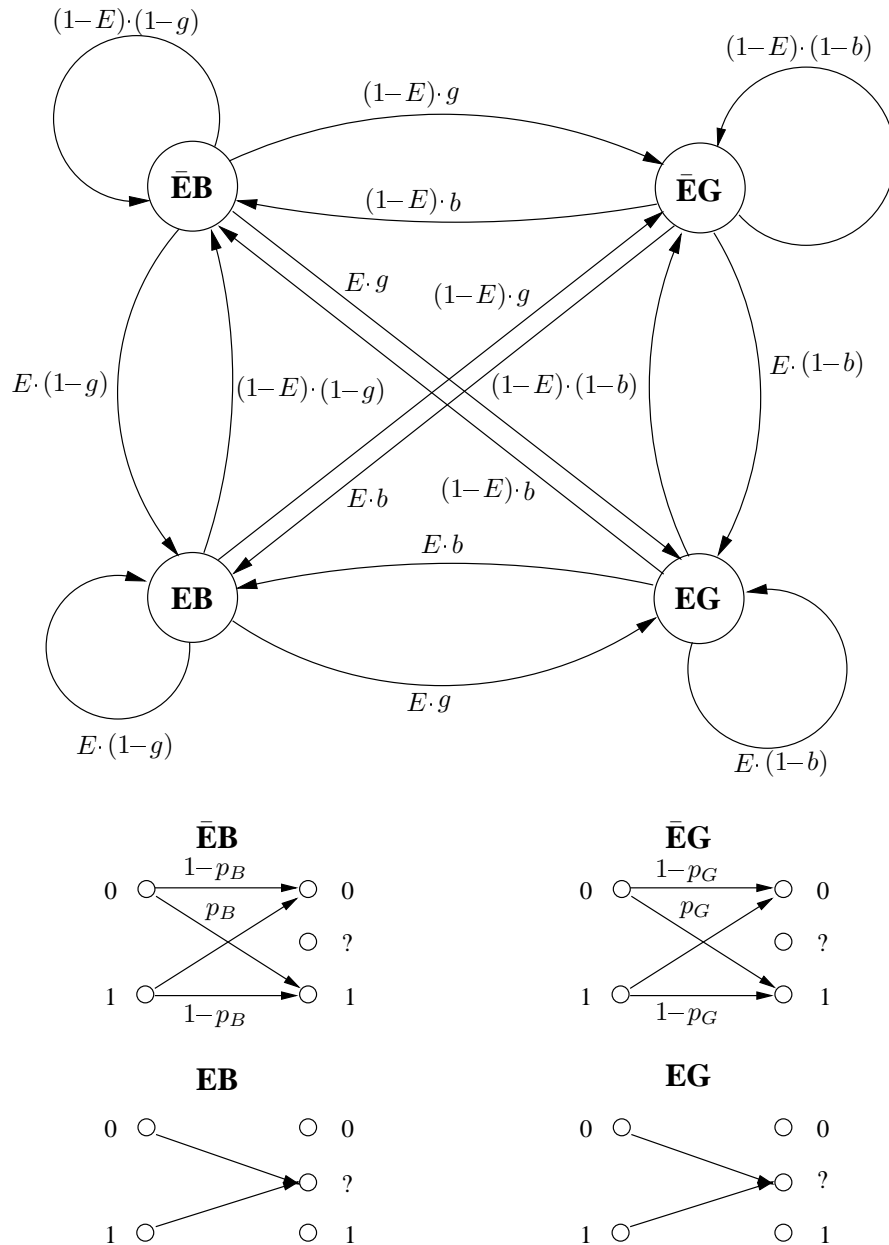


Abbildung 2.23: Zustände des verknüpften Kanalmodells

Setzt man das Kanalgedächtnis des MEC auf Null, was nach dem vorherigem Abschnitt keinerlei Auswirkungen auf die Kapazitätsberechnung hat, so erhält man die Variable  $E$  sowohl als

Zustandsaufenthaltswahrscheinlichkeit für den Zustand E, als auch als Übergangswahrscheinlichkeit, wie in Abbildung 2.23 zu sehen ist.

Bisher allerdings gilt dieses Modell nur, wenn die Blockgröße des Internets  $M = 2$  ist und nur die beiden Eingangssymbole 0 und 1 annimmt. Dieses Problem kann man umgehen, wenn man die Stationarität des Markovmodells aufgibt. Man schränkt die Blockgröße auf Potenzen der Zahl 2 ein ( $M = 2^m$ ) und überträgt durch das Internet immer  $m$  Bit. Befindet man sich also am Anfang eines Blocks, so gilt das Markovmodell nach Abbildung 2.23, befindet man sich aber in der Mitte eines Blocks, so kann es keine Übergänge zwischen den Zuständen E und  $\bar{E}$  geben, es entfallen also einige Zweige aus Abbildung 2.23, ein anderes Markovmodell wird angewandt.

## 2.4.2 Die Kapazitätsberechnung

Die rekursive Berechnung der Kapazität nach [2] verlangt eigentlich eine stationäre Markovkette, wäre demnach ungeeignet für obiges Modell. Zwei Tatsachen erlauben uns aber trotzdem nach diesem Algorithmus zu rechnen:

- die Berechnung der Kapazität eines FSMC nach [2] erlaubt instationäre Eingangssignale (vgl. Kapitel B.3), indem sie eine Mittelung über alle Rekursionstiefen vornimmt
- betrachtet man in unserem Modell nur jeden  $m$ -ten Schritt, so ist es durchaus stationär, eine Mittelung über mehrere Blöcke wird somit konvergieren

Somit kann man nach [2] die Kanalkapazität der Reihenschaltung von MEC und Gilbert-Elliott Kanals berechnen, was in der Mex-Datei `C_n.c` geschehen ist. Man muß allerdings strikt gemäß der allgemeinen Vorgehensweise in Anhang B verfahren, da sich kaum Vereinfachungen der allgemeinen Formeln ergeben.

Bei der Zustandsübergangsmatrix muß zwischen der im Block  $P_B$  und der am Blockende  $P$  unterschieden werden:

$$P = \begin{pmatrix} (1-g)(1-E) & g(1-E) & (1-g)E & gE \\ b(1-E) & (1-b)(1-E) & bE & (1-b)E \\ (1-g)(1-E) & g(1-E) & (1-g)E & gE \\ b(1-E) & (1-b)(1-E) & bE & (1-b)E \end{pmatrix}$$

$$P_B = \begin{pmatrix} 1-g & g & 0 & 0 \\ b & 1-b & 0 & 0 \\ 0 & 0 & 1-g & g \\ 0 & 0 & b & 1-b \end{pmatrix}$$

Außerdem muß wegen des Fragezeichensymbols, also der Blockauslöschung bei der  $B$ -Matrix (Diagonalmatrix mit den Elementen  $\Pr(y|S = c_k)$ ) zwischen den beiden Fällen  $Y = 0 \wedge 1$  und  $Y = ?$  unterschieden werden, was zu unterschiedlichen Werten für den ersten Erwartungswert in Gleichung B.1 führt, die von der Mex-Datei `modell_rho` berechnet werden. Die  $B$ -Matrizen sind:

$$B_{0,1} = \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$B_{?} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Die  $D$ -Matrix ( Diagonalmatrix mit den Elementen  $\Pr(y|x, S = c_k)$ ), die zur rekursiven Ermittlung der  $\pi_n$  benötigt wird, kann nun drei verschiedenen Formen annehmen:

$$D_{x=y} = \begin{pmatrix} 1 - p_B & 0 & 0 & 0 \\ 0 & 1 - p_G & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad D_{x \neq y} = \begin{pmatrix} p_B & 0 & 0 & 0 \\ 0 & p_G & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$D_{y=?} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Gewappnet mit diesen Größen ist es nun möglich mit [2] bzw. mit den Anweisungen aus Anhang B die Kapazität  $C_N$  des Gesamtkanals rekursiv zu ermitteln.

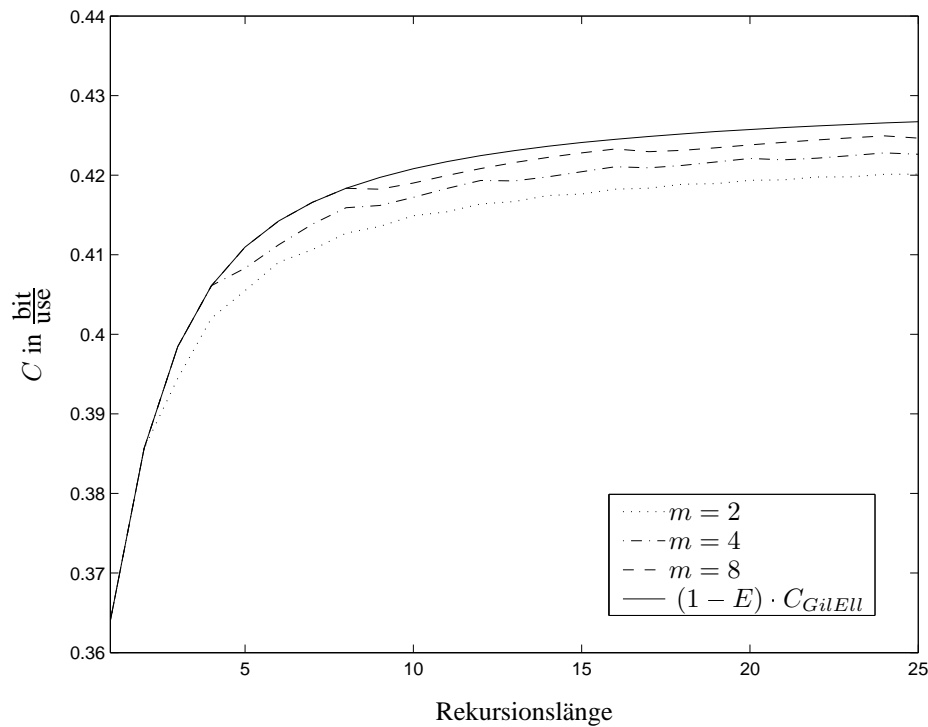
### 2.4.3 Ergebnisse und Vergleich

Essenziell sind zuallererst Betrachtungen der Konvergenz des Algorithmus in  $C_n$ . c. Man erkennt nämlich, daß man die Blocklänge  $m$  nicht allzu groß machen darf, da sich die Mittelung über die verschiedenen Rekursionstiefen, wie schon im vorherigen Kapitel erwähnt, über einige Blocklängen erstrecken muß. Dies hat aber auf Grund der exponentiellen Zunahme der Rechenkomplexität Grenzen.

In Abbildung 2.24 sind deshalb die gewählten Blocklängen  $m$  mit 2,4 und 8 bit sehr gering gehalten. Es wurde als Fehlerrate des Internets  $E = 0.2$  gewählt. Der Gilbert-Elliott Kanal wurde nur zum beispielshalber auf die Werte  $\rho = 3, \mu = \frac{3}{4}, p_B = 0.499$  und  $p_G = 0.01$  eingestellt, das resultierende Ergebniss ist allerdings für alle Werte von  $E, \rho, \mu, p_B$  und  $p_G$  gleich.

Wie in Abbildung 2.24 zu sehen gilt offenbar:

- der Algorithmus konvergiert immer gegen einen bestimmten Wert
- zu Beginn macht das Ergebnis alle  $m$  Werte einen leichten Knick nach unten, was aber für große Rekursionstiefen nachläßt

Abbildung 2.24: Berechnete Kapazität  $C_N$  für verschiedene  $m$ 

- je länger der Block, desto größer die Kapazität
- für große Blöcke strebt die Kapazität des Verbundmodells offenbar gegen die mit  $(1 - E)$  multiplizierte Kapazität  $C_{GilEU}$  des Gilbert-Elliott Kanals

Wenn man also von realen Blocklängen ausgeht, die mehrere Hundert Byte und nicht wenige Bit wie in Abbildung 2.24 betragen, so kann man guten Gewissens die folgende Näherungsformel für die Kapazität des Verbundmodells ohne *Transcoding* aufstellen:

$$C_N \approx (1 - E) \cdot C_{GilEU} \quad (2.17)$$

Geht man weiterhin von realen Zuständen aus, so ist der Mobilfunkkanal immer schlechter als der Internetkanal, die Kapazität mit *Transcoding* ist in realen Szenarios demnach immer die Kapazität des Gilbert-Elliott Kanals:

$$C = C_{GilEU} \quad (2.18)$$

Somit kann eine sehr einfache Näherungsformel für den zu erwartenden Gewinn durch *Transcoding* beim *streamen* von Daten über die Reihenschaltung von Internet und Mobilfunk aufgestellt werden, Gleichung 2.19

$$\frac{C - C_N}{C_N} \approx \frac{E}{1 - E} \quad (2.19)$$



Oder andersherum ausgedrückt: Der Verlust, den man hinnehmen muß, wenn man keinen Server nutzt, ist genau

$$\frac{C - C_N}{C} \approx E \quad (2.20)$$

Vergleicht man nun die experimentel ermittelten Werten von [13] mit denen durch die Formeln 2.19 und 2.20 ermittelten, so stimmen diese sehr gut überein. In [13] wurde beispielsweise bei einer Paketfehlerrate des Internets von  $E = 2\%$  und einer maximalen Paketverlustrate von 0.03 nach dem Decodieren für *Transcoding* eine Datenrate von 980 bit/s und ohne *Transcoding* eine Datenrate von 960 bit/s gemessen. Es errechnet sich daraus ein Gewinn von

$$\frac{980 - 960}{960} = 2.08\% ,$$

während die Näherungsformel 2.19 einen Wert von

$$\frac{2\%}{1 - 2\%} = 2.04\%$$

ergibt.

# Kapitel 3

## Verzerrung bei angewandten Codierverfahren

### 3.1 Allgemein

Nach den allgemeinen und eher theoretischen Berechnungen von Kapitel 2 wollen wir uns in diesem Kapitel der praktischen Implementierung, genauer gesagt, den Einschränkungen dieser widmen, um die Unterschiede zwischen den beiden diskutierten Übertragungsverfahren herauszustellen. Und um diese Unterschiede vergleichen zu können führen wir das Maß der Verzerrung (englisch: *Distortion*)  $D$  ein.

#### 3.1.1 Die Datenübertragung

Bei der angewandten Datenübertragung gehen wir von einigen Voraussetzungen und Einschränkungen aus, wie sie bei der tatsächlichen Verwirklichung eben anfallen. Irgendwann muss man sich auf Grund der gegebenen Eigenschaften des Systems, physikalischer oder auch technischer Natur, auf bestimmte Übertragungsstandards einigen. Daraus resultieren Einschränkungen, wie eine begrenzte Codewortlänge  $N$ , ein gewählter Code mit seinem begrenzten Fehlerschutz  $N - K$ , eine feste Block- bzw. Wortlänge  $M_B$  und bestimmte Fehlereigenschaften der Übertragung (beispielsweise die Fehlerrate  $p$ ). In Abbildung 3.1 ist die Übertragung der Daten nochmals veranschaulicht und Tabelle 3.1 listet obige Einschränkungen die getroffen werden müssen nochmals auf und beschreibt mögliche Gründe und Entscheidungskriterien dafür.

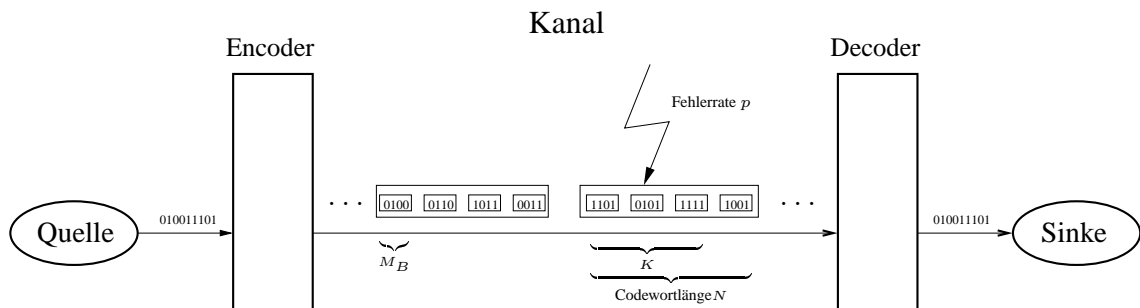


Abbildung 3.1: Angewandte Datenübertragung

Name	Beschreibung	Ursachen
$M_B$	Anzahl der Bit pro Wort bzw. Datenblock.	<ul style="list-style-type: none"> <li>• Architektur des Systems ( z.B. Internet, TDMA-Verfahren)</li> <li>• Art des Fehlerschutzcodes</li> </ul>
$p$	Fehlerwahrscheinlichkeit	<ul style="list-style-type: none"> <li>• Sendeleistung</li> <li>• physikalische Gegebenheiten des Kanals</li> <li>• maximale Verzögerungen</li> <li>• mögliche wortinterne Fehlerschutzcodes und tiefere Protokollebenen</li> </ul>
$N$	Codewortlänge in Wörtern	<ul style="list-style-type: none"> <li>• Effizienz des Fehlerschutzcodes</li> <li>• maximale Puffergrößen</li> <li>• maximale Verzögerungen</li> </ul>
$K$	Nachrichtenwörter und somit $N - K$ Redundanzwörter	<ul style="list-style-type: none"> <li>• Art des Fehlerschutzcodes</li> <li>• maximale Wahrscheinlichkeit des Codierversagens ( <math>\Rightarrow</math> Wichtigkeit der Daten)</li> </ul>

Tabelle 3.1: Größen aus Abbildung 3.1 und Entscheidungskriterien für diese

### 3.1.2 Verzerrung

Gerade die Wahl der Anzahl der Redundanzwörter  $N - K$ , wie auch in Tabelle 3.1 zu sehen, ist entscheidend für die Qualität der Datenübertragung und muß für den Übertragungskanal und die gewählte Codierung optimiert werden.

Als Maß für diese Qualität steht die mittlere Verzerrung  $D$  ( englisch *Distortion*), deren allgemeine Definition ( vgl. [7]) in Gleichung 3.1 zu sehen ist.

$$D = \sum_{i=1}^{M_x} \sum_{j=1}^{M_y} \Pr(X = x_i, Y = y_j) \cdot \rho_{ij} \quad (3.1)$$

Sie ist einfach die Summe über die Wahrscheinlichkeiten aller möglichen Paare von Ein- und Ausgangssymbolen gewichtet mit dem Verzerrungsmaß  $\rho_{ij}$ . Dabei muß es sich bei den Ein- und Ausgangssymbolen um so große Hypersymbole handeln, daß Abhängigkeiten zwischen diesen ausgeschlossen werden können.

Dem Verzerrungsmaß  $\rho_{ij} \in \mathbb{R}_0^+$  kommt dabei die wichtige Aufgabe zu, Fehler zu gewichten. Es kann je nach Bedarf auch nach vollkommen subjektiven Kriterien gewählt werden, kann also auch auf Datenströme mit verschiedenen wichtigen Informationen angewandt werden. Für eine vollkommen fehlerfreie Übertragung  $x_i = y_j$  gilt wohl  $\rho_{ij} = 0$  aber sonst kann auch die Einheit frei gewählt werden.

## 3.2 Paketorientierte Übertragung

Wenn nun  $M_B$  sehr groß wird, so reden wir nicht mehr von Wörtern sondern von Datenblöcken. Und wenn diese entweder richtig oder falsch sind, auftretende Fehler also den Block als Ganzes zerstören, und wenn außerdem die Inhalte der Blöcke und die Wahrscheinlichkeiten für Fehler durch die Voraussetzung  $M_B \gg 1$  nahezu unabhängig voneinander sind, so liegt ein Szenario gemäß Kapitel 2.2.4 vor.

### 3.2.1 Verzerrung für einen Kanal

Unter diesen Voraussetzungen bietet sich als Verzerrungsmaß  $\rho_{ij}$  die Hammingdistanz an

$$\rho_{ij} = \begin{cases} 0 & \text{für } i = j \\ 1 & \text{für } i \neq j \end{cases}$$

womit sich die Formel für die mittlere Verzerrung für diesen Fall stark vereinfacht.

$$D = \Pr(X \neq Y) = p$$

Werden solche Datenströme mit einem Reed-Solomon-Code der Länge  $N$  mit  $K$  Nachrichtenstellen und somit  $N - K$  Redundanzstellen kodiert, so können auf jeden Fall  $F = N - K$  Auslöschungen, also Blockfehler korrigiert werden (vgl. [11]), womit für  $N \rightarrow \infty$  eine fehlerfreie ( $D = 0$ ) Datenübertragung gewährleistet wäre. Die mittlere Verzerrung berechnet sich unter diesen Bedingungen wie folgt:

$$\begin{aligned} D &= \Pr(F > N - K) \\ &= \sum_{i=F+1}^N \binom{N}{i} \cdot p^i (1-p)^{N-i} \\ &= \sum_{i=F+1}^N B(N; p; i) \\ &= 1 - \sum_{i=0}^F B(N; p; i) \end{aligned} \tag{3.2}$$

wobei  $B(N; p; i)$  die wohlbekannte Binomialverteilung (siehe z.B. [21]) und  $F$  die Anzahl der aufgetretenen Fehler ist.

Abbildung 3.2 stellt die Verzerrung mit Codierung der ohne Codierung, die in beiden Fällen der Empfangswahrscheinlichkeit der Pakete entspricht, für eine Codewortlänge  $N$  von 255 Blöcken und 200 Nachrichtenstellen  $K$  gegenüber. Sehr deutlich wird dabei die polarisierende Eigenschaft derartiger Blockcodes. Bis zu einer bestimmten Fehlerwahrscheinlichkeit, überträgt der Kanal fast fehlerfrei, aber danach eigentlich gar nicht mehr. Diese Eigenschaft wird auch bei zwei Kanälen deutlich zu sehen sein.

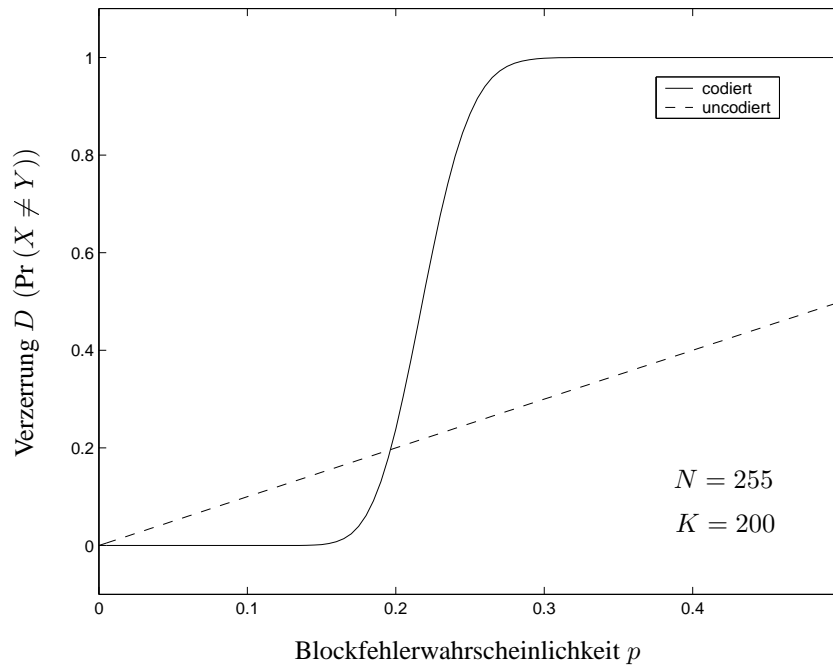


Abbildung 3.2: Verzerrung für Paketorientierte Kanäle

### 3.2.2 Zwei Kanäle

Hier gilt es zu unterscheiden, ob die Kodierung für jeden Kanal extra geschieht oder auf beide Kanäle angewandt wird.

#### mit Transcoding

Kodierversagen des gesamten Kanals tritt immer dann auf, wenn mindestens bei einem der beiden Teilkanäle Kodierversagen auftritt. Da beide Ereignisse voneinander unabhängig sind, können sie sehr einfach nach Gleichung 3.3, mit  $D_1$  und  $D_2$  nach Gleichung 3.2, berechnet werden (Nomenklatur gemäß Kapitel 2).

$$D = D_1 + D_2 - D_1 \cdot D_2 \quad (3.3)$$

#### ohne Transcoding

Werden die Datenblöcke im Server einfach weitergeleitet, so kann das gesamte System als ein einziger Kanal mit der Fehlerrate  $p_G = p_1 + p_2 - p_1 \cdot p_2$  und einer einzigen Kodierung angesehen werden. Die Verzerrung berechnet sich gemäß Gleichung 3.2 mit der Gesamtfehlerrate  $p_G$ .

$$D_N = D_G$$

## Ergebnisse

Da es beim *Transcoding* prinzipiell zwei verschiedene Kanäle und zwei verschiedene Codes gibt, kann die Codierung hinsichtlich ihrer Codewortlänge und hinsichtlich der Anzahl der Quellwörter, wenn beispielsweise der eine Kanal mehr Daten pro Zeit übertragen kann als der andere, viel besser an den Kanal angepasst werden. Verwendet man kein *Transcoding*, muß man sich immer auf den kleinsten Nenner einigen.

Aber abgesehen von diesem Effekt erzeugt die Übertragung mit *Transcoding* weniger Verzerrung als dieselbige ohne. In Abbildung 3.3 wird analog zum Kapitel 2.2.4 die Verzerrung mit *Transcoding*  $D$  gegenüber der ohne  $D_N$  dargestellt. Dabei wird Letztere konstant gehalten, was auf Grund der streng monotonen Eigenschaften der Binomialverteilung eine konstante Gesamtlehrrate  $p_G$  zur Folge hat. Als Parameter wird die Fehlerletrate des ersten Kanals  $p_1$  herangezogen.

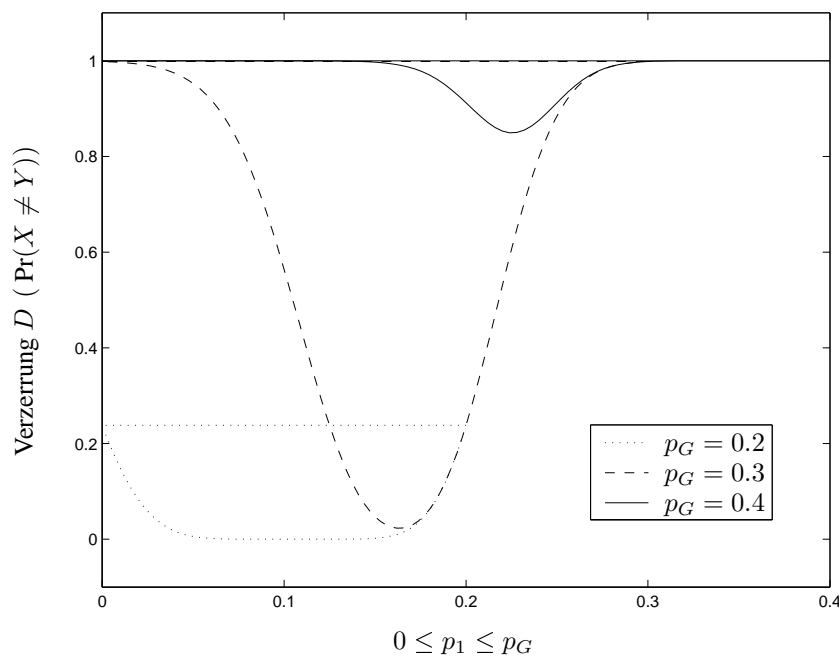


Abbildung 3.3: Vergleich der Verzerrungen

Wie in Abbildung 3.3 gut zu sehen, ist die Verzerrung mit *Transcoding* zwar immer kleiner als ohne, und sie wird wie schon in Kapitel 2.2.4 durch gleiche Fehlerwahrscheinlichkeiten  $p_1 = p_2 = 1 - \sqrt{1 - p_G}$  maximiert, ihr Unterschied zur Verzerrung ohne *Transcoding* hängt aber sehr stark von der Gesamtlehrrate  $p_G$  ab. Nur bei Gesamtlehrraten nahe der Sprungstelle der Verzerrung (vgl. Abbildung 3.2) kommt es zu starken Gewinnen.

## Rate-Distortion-Funktion

Wir gehen deshalb den anderen Weg und führen die sogenannte *Rate-Distortion-Funktion* ein, wir berechnen also mit welcher Rate  $K/N$  wir maximal übertragen können, wenn wir die Verzerrung  $D_{max}$  tolerieren, berechnen also nicht wie bisher die Verzerrung  $D$  für eine feste Rate

$K/N$ . Abbildung 3.4 zeigt nun die Übertragungsrate  $R$  bei maximaler tolerierter Verzerrung  $D_{max}$  und fester Gesamtfehlerwahrscheinlichkeit  $p_G$ , wenn die Blockfehlerwahrscheinlichkeit des ersten Kanals  $p_1$  variiert wird.

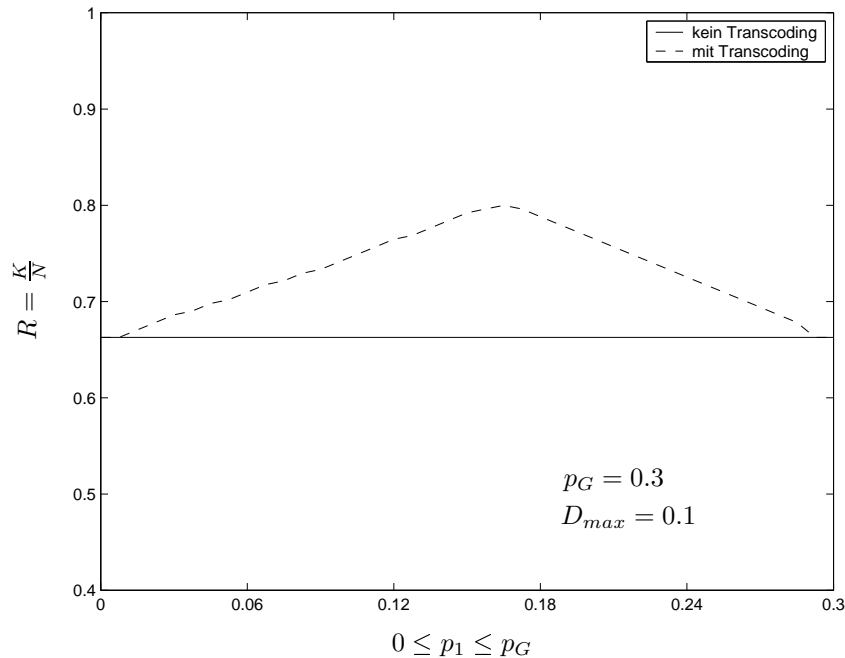
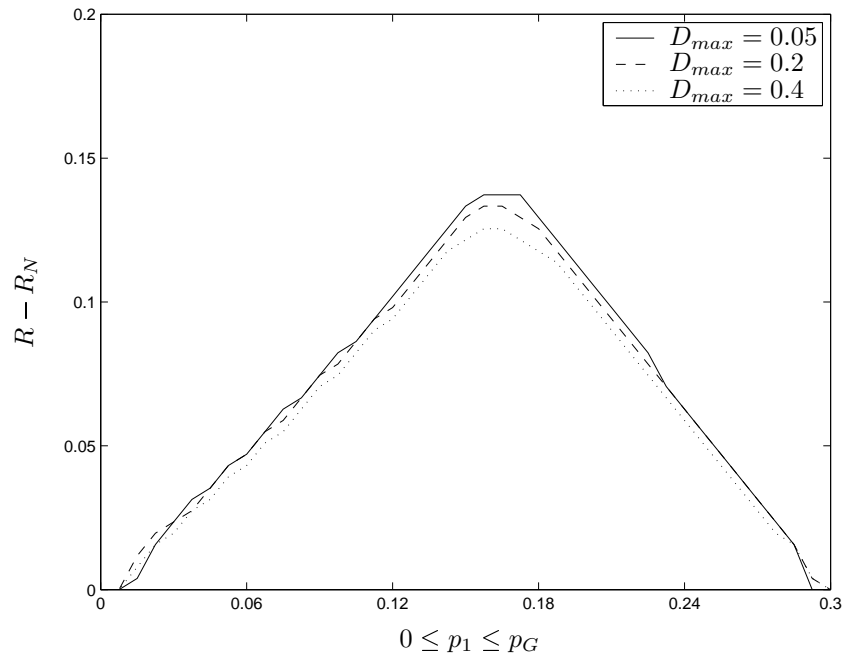


Abbildung 3.4: maximale Rate  $R$ , wenn  $D \leq 0.1$

Die Ergebnisse entsprechen sowohl in Form als auch in den Größenordnungen sehr gut denen aus Kapitel 2.2.4, es kann fast die Kapazität  $C$  als Übertragungsrate  $R$  genutzt werden, die Kapazität ist demnach eine aussagekräftige Größe. Aus informationstheoretischer Sicht sind dabei zwei Faktoren zu berücksichtigen:

- Je länger das Codewort  $N$ , desto höher die Übertragungsrate  $R$ , sie kann aber nicht größer als die Kapazität  $C$  werden, solange man keine Verzerrung  $D$  akzeptiert. Prinzipiell könnte für  $N \rightarrow \infty$  mit  $R = C$  übertragen werden.
- Je mehr Verzerrung  $D_{max}$  toleriert wird, desto höher kann die Übertragungsrate  $R$  sein, sie kann dann sogar größer als die Kapazität werden.

Abbildung 3.5 verdeutlicht deswegen die Abhängigkeit des Gewinns an Übertragungsrate von der tolerierten Verzerrung  $D_{max}$ , dieser ist nämlich ein wenig geringer, wenn mehr Verzerrung toleriert wird. Dies ist begründet in der allgemeinen Natur von Rate-Distortion Funktionen  $R(D_{max})$ . Wird keine Verzerrung toleriert, so kann maximal mit der Kapazität als Übertragungsrate gearbeitet werden ( $R(0) = C$ ), wird allerdings totale Verzerrung toleriert, so können alle Codeblöcke  $N$  Nachrichtenblöcke  $K$  sein, womit  $R(1) = N/K = 1$  gilt. Beim *Transcoding* ist die Kapazität  $C$  prinzipiell höher (vgl. Kapitel 2), deshalb kann die Datenrate mit größeren tolerierten Verzerrungen  $D_{max}$  nicht mehr so stark steigen wie beim Durchschleifen der Daten.

Abbildung 3.5: Gewinn an Rate durch *Transcoding*

### 3.3 Ungleicher Fehlerschutz

Ein neuer Effekt, der bereits in Kapitel 3.1.2 angedeutet wurde und bei den informationstheoretischen Betrachtungen aus Kapitel 2 aber nicht zum Tragen kam, ist der von Daten unterschiedlicher Wichtigkeit.

#### 3.3.1 Begriffsklärungen

Gerade bei der Übertragung von multimedialen Inhalten sind Daten in Blöcke unterschiedlicher Wichtigkeit, sogenannte *Layer*, aufgeteilt. Audio- und Videodaten sind unterschiedlich wichtig, und auch Videodaten selber sind in ihrer Qualität und somit in ihrer Grösse skalierbar und enthalten demnach unterschiedlich wichtige *Layer*.

Abbildung 3.6 stellt beispielsweise einen Videocoder mit stark unterschiedlich wichtigen *Layer* nach [10] dar.

Der *Base Layer* enthält einen *intra-predicted frame* und drei *inter-predicted frames* in einer niedrigen zeitlichen und örtlichen Auflösung. Der *Enhancement Layer* hingegen enthält Daten zum zeitlichen und örtlichen Verfeinern der Videosequenz, die den *Base Layer* als Grundlage zur Vorhersage (englisch: *Prediction*) benötigen. Offensichtlich verursacht der Verlust von Daten des *Base Layer* wesentlich mehr Störung als der Verlust von Daten aus dem *Enhancement Layer*.

Außerdem wären noch zwei weitere Eigenschaften der Datenübertragung von multimedialen Inhalten zu berücksichtigen, die sich ebenfalls aus obigen Beispiel in Abbildung 3.6 gut ableiten lassen:

- hierarchische Daten:



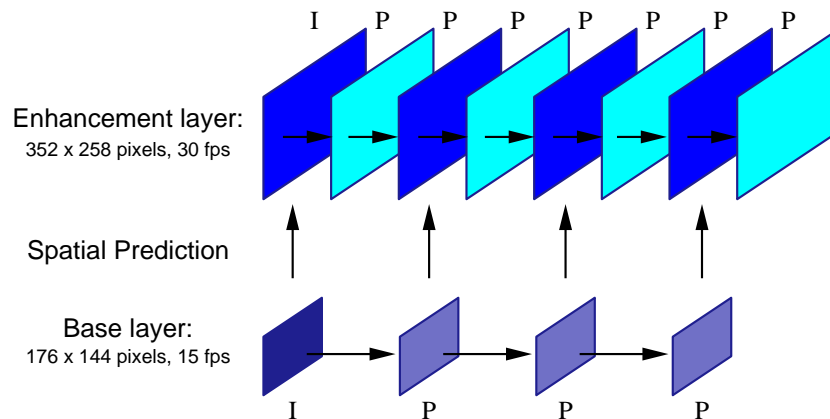


Abbildung 3.6: Beispiel eines Videocoders

Multimediale Daten sind oft hierarchisch in ihren Inhalten, wenn einer der wichtigeren Layer verloren geht, sind die Layer niedriger Priorität uninteressant, da sie ohne den wichtigeren Layer nicht mehr zur Verbesserung des Endresultats beitragen können. Fällt in obigen Beispiel der *Base Layer* aus, so nützen die Daten des *Enhancement Layer* auch nichts mehr. Ein Gegenbeispiel wären Audio- und Videodaten in unterschiedlichen Layern, die zwar auch verschieden wichtig, aber nicht abhängig voneinander sind.

- **Blockfehler:**  
Wenn Codierversagen bei einem bestimmten Layer auftritt, so ist dieser meist total verloren. Viele Fehlerschutzcodes ( wie beispielsweise Reed-Solomon-Codes) haben schon die Eigenschaft, daß bei einem Codierversagen der gesamte Datenblock verloren ist, aber auch die Quellcodierung der Videodaten ist wegen der angewandten Prediktion anfällig für Fehlerfortpflanzungen bei Bitfehlern. Je nach Layer, in dem der Fehler auftritt, ist es ratsamer einen defekten Layer ganz zu verwerfen, als ihn zur Decodierung heranzuziehen.

Es gilt nun, das Verzerrungsmaß  $\rho_{ij}$  so anzupassen, daß es all diesen Anforderungen gerecht wird.

Zu diesem Zweck sei also  $H$  die Anzahl der voneinander unabhängigen Layer-Hierarchien ( zum Beispiel Audio und Video) und  $L^{(h)}$  mit  $h \in \{1, \dots, H\}$  sei die Tiefe der  $h$ -ten Layer-Hierarchie. Es bilde die Zufallsgröße  $D^{(h)}$  das Ereignis: 'in der  $h$ -ten Layer-Hierarchie wird gerade der  $l$ -te Layer dargestellt' auf die Zahlen  $D_l^{(h)}$  ab ( analog zu [10]).  $D_l^{(h)}$  ist somit die mittlere Verzerrung die hinzukommt, wenn nur der  $l$ -te Layer der  $h$ -ten Layerhierarchie dargestellt wird und somit die höheren Layer ( Ordnung  $l+1$  bis  $L^{(h)}$ ) nicht. Dafür müssen natürlich, wegen der hierarchischen Struktur, die Layer 1 bis  $l$  korrekt empfangen worden sein, Layer  $l+1$  aber nicht. Ebenso ist  $D_0^{(h)}$  die Verzerrung, wenn für diese Layerhierarchie gar keine Daten empfangen worden sind.

Mit diesen eingeführten Variablen erhalten wir als neue Formel für die mittlere Verzerrung Gleichung 3.4.

$$\begin{aligned}
 D &= \mathcal{E} \left\{ \sum_{h=1}^H D^{(h)} \right\} \\
 &= \sum_{h=1}^H \mathcal{E} \{ D^{(h)} \} \\
 &= \sum_{h=1}^H \sum_{l=1}^{L^{(h)}} \Pr \left( D^{(h)} = D_l^{(h)} \right) \cdot D_l^{(h)}
 \end{aligned} \tag{3.4}$$

Bemerkenswert ist hierbei der Übergang von Zeile 1 auf Zeile 2 in Gleichung 3.4. Da diese simple stochastische Rechenregel auch für abhängige Zufallsvariablen gilt, ist es für die mittlere Verzerrung also unerheblich, ob beispielsweise durch die Codierung die Ereignisse von Fehlern in den verschiedenen Layerhierarchien voneinander abhängig sind oder nicht.

### 3.3.2 Pakete mit Daten unterschiedlicher Wichtigkeit

Bei der Übertragung solcher Daten in Blöcken bzw. Paketen ist wiederum einiges zu beachten. Die prinzipielle Wirkungsweise von Fehlerschutzcodes ist die einer Solidargemeinschaft. Viele Wörter ( hier Datenblöcke) schließen sich zu einem Codewort zusammen um den Fehler-eigenschaften des Kanals besser trotzen zu können, und dies funktioniert umso besser je mehr Wörter es sind. Auf Grund begrenzter Ressourcen, wie Puffergröße, maximaler Verzögerung und maximaler Decodierkomplexität, kann aber das Codewort nicht beliebig groß werden. Um also maximalen Fehlerschutz für einen Layer zu erhalten, muß dieser auf alle Wörter des Codesymbols möglichst gleichmäßig verteilt sein.

Bei einer paketorientierten Übertragung ist dies glücklicherweise für jeden Layer möglich, ohne Einschränkungen für die anderen Layer hinzunehmen. Zur optimalen Übertragung werden die  $k_l$  Daten der unterschiedlich großen Layer mit ihrem  $n_l - k_l$  Fehlerschutzstellen also gleichmäßig, wie in Abbildung 3.7 zu sehen ist, auf die Blöcke verteilt.

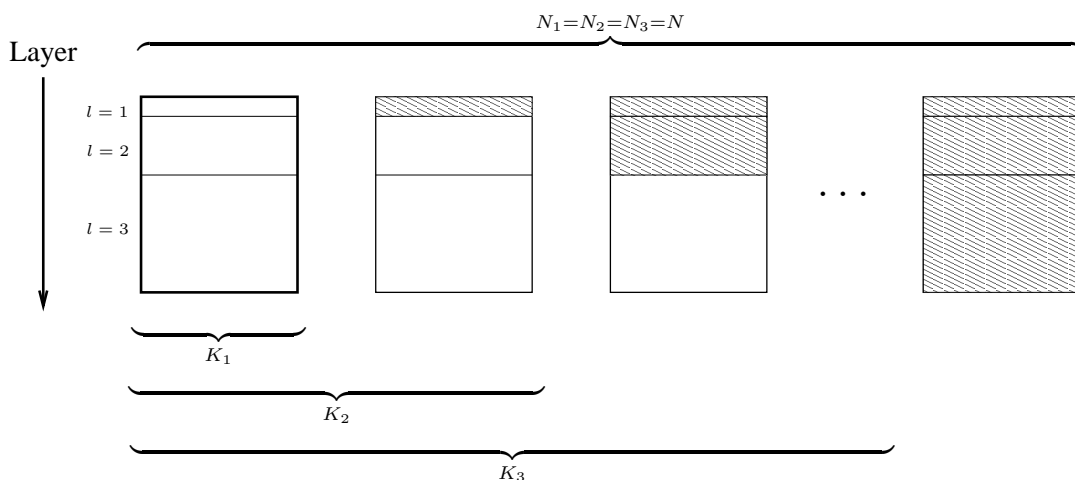


Abbildung 3.7: Ungleicher Fehlerschutz

Der Schlüssel zur Optimierung ist die Zuordnung von unterschiedlichem Fehlerschutz zu den verschiedenen Layern. Die hellen Flächen in Abbildung 3.7 entsprechen den Datenmenge  $k_l$  der durchaus unterschiedlich großen Layer. Jedem Layer wird proportional zu seiner Größe unterschiedlich viel Redundanz  $n_l - k_l$ , durch die dunklen Flächen dargestellt, zugeordnet. Wichtigere Layer sind durch mehr Redundanz besser geschützt.

Allerdings verändert diese Art der Verteilung der Daten auch die Fehlercharakteristik. Es ist nicht möglich, daß ein Layer mit weniger Redundanz decodiert werden kann, während bei einem Layer mit mehr Redundanz Codiersversagen eintritt. Das Versagen der Codierung ist nun ebenfalls hierarchisch. Diesen Effekt kann man ausnutzen um Gleichung 3.4 auf einen einzigen Layer zu Gleichung 3.5 zu vereinfachen, auch wenn dieser neu geschaffene Layer nun nicht mehr streng hierarchisch ist (man beachte hierzu den letzten Abschnitt des vorherigen Kapitels 3.3.1). Wichtig ist nur, daß die Verzerrung, die durch Wegfall eines Layers entsteht, monoton steigend ist, also  $D_i \geq D_j \quad i < j$  (vereinfachte Schreibweise:  $D_i := D_i^{(1)}$ ).

$$D = \sum_{l=0}^L \Pr(D^{(1)} = D_l) \cdot D_l \quad (3.5)$$

Wie schon in Gleichung 3.4 beschreibt auch hier  $D_l$  die Verzerrung die beobachtet wird, wenn Layer 1 bis  $l$  dargestellt werden, aber Layer  $l + 1$  nicht.  $D_0$  ist die Verzerrung wenn gar keine Daten ankommen und  $D_L$  (vereinfachte Schreibweise:  $L := L^{(1)}$ ) ist die Verzerrung bei fehlerfreier Übertragung (sollte also Null sein).

### 3.3.3 Minimierung der Verzerrung $D$

Ziel ist es nun die mittlere Verzerrung nach Gleichung 3.5 zu minimieren. Als Parameter dafür können wir die Zuordnung der Redundanz  $f_l = n_l - k_l$  zu den jeweiligen Layern variieren, wobei wir die Gesamtdatenmenge von  $N$  Blöcken mit je  $M_B$  Symbolen nicht überschreiten dürfen.

$$\sum_{l=1}^L n_l = N \cdot M_B \quad (3.6)$$

#### Die Anzeigewahrscheinlichkeit

Um nun die mittlere Verzerrung berechnen zu können, wird vor allem die Anzeigewahrscheinlichkeit  $\Pr(D^{(1)} = D_l)$  benötigt. Um diese für einen bestimmten Layer  $l$  zu berechnen, gehen wir von einer Kodierung aus, die bei  $F_l = N - K_l$  redundanten Blöcken auch  $F_l = N - K_l$  Blockauslösungen korrigieren kann, womit man für  $N \rightarrow \infty$  mit der Kapazität des Kanals als Datenrate übertragen könnte. Reed-Solomon-Codes besitzen beispielsweise diese Eigenschaft (vgl. auch [11]) und sind somit hervorragend zur Blockfehler-Korrektur geeignet.

Sei nun  $F$  die Zahl der Fehler pro Codeblock, so gilt für die Anzeigewahrscheinlichkeit des  $l$ -ten Layers unter diesen Bedingungen:

$$\begin{aligned}
\Pr(D^{(1)} = D_l) &= \Pr(F_{l+1} < F \leq F_l) \\
&= \sum_{i=F_{l+1}+1}^{F_l} \binom{N}{i} \cdot p^i (1-p)^{N-i} \\
&= \sum_{i=F_{l+1}+1}^{F_l} B(N; p; i)
\end{aligned}$$

mit

- $N$  : Blöcke pro Codesymbol
- $F$  : Anzahl der Blockauslöschungen
- $p$  : Wahrscheinlichkeit für eine Blockauslöschung
- $F_l = N - K_l$  : redundante Blöcke für Layer  $l$
- $B(N; p; i)$  : Binomialverteilung

und

$$n_l = N \cdot \frac{k_l}{K_l} = N \cdot \frac{k_l}{N - F_l}$$

Die mittlere Verzerrung ergibt sich somit zu Gleichung 3.7 und kann einerseits nach den  $K_l$  optimiert werden, andererseits muß aber auch in Betracht gezogen werden, ob es von Vorteil wäre einige der Layer niedriger Priorität nicht zu übertragen.

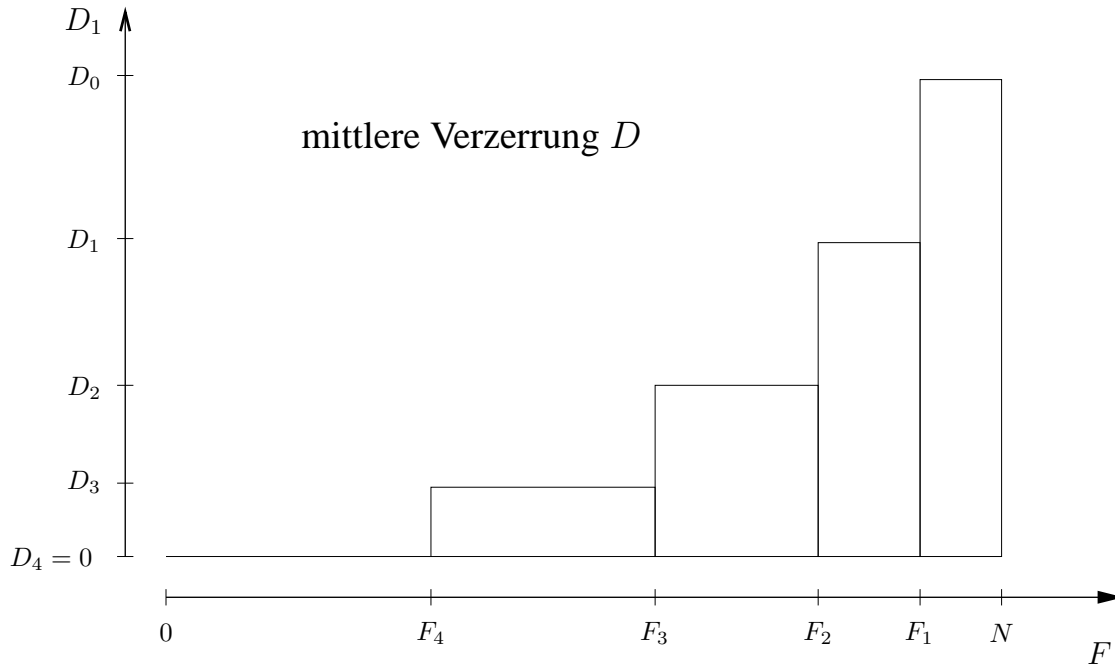
Es werden also zuerst die mittleren Verzerrungen für eine gegebene Anzahl von Layern  $L$  mit ihren Datenmengen  $k_l$   $0 < l \leq L$  optimiert und berechnet und dann werden diese mittleren Verzerrungen unter sich nach dem Minimum durchsucht.

$$D = \sum_{l=0}^L D_l \cdot \sum_{i=F_{l+1}+1}^{F_l} B(N; p; i) \quad (3.7)$$

### Optimierung der gewichteten Binomialverteilung

Die Reihenfolge der Summation in Gleichung 3.7 kann, wie in Abbildung 3.8 verdeutlicht, umgekehrt werden zu Gleichung 3.8.

$$\begin{aligned}
D &= \sum_{l=1}^L (D_{l-1} - D_l) \cdot \sum_{i=F_{l+1}}^N B(N; p; i) \\
&= \sum_{l=1}^L (D_{l-1} - D_l) \cdot \left( 1 - \sum_{i=0}^{F_l} B(N; p; i) \right) \\
&= D_0 - \sum_{l=1}^L (D_{l-1} - D_l) \cdot \sum_{i=0}^{F_l} B(N; p; i)
\end{aligned}$$


 Abbildung 3.8: Berechnung von  $D$  nach Gleichung 3.7

Hierbei sind allerdings einige Nebenbedingungen zu beachten, womit sich Folgendes nach den  $F_1, \dots, F_L \in \mathbb{N}_0$  zu minimierendes Gleichungssystem ergibt:

$$D = D_0 - \sum_{l=1}^L (D_{l-1} - D_l) \cdot \sum_0^{F_l} B(N; p; i) \quad (3.8)$$

$$F_{l+1} \leq F_l \quad \forall l \in \{1, \dots, L-1\} \quad (3.9)$$

$$0 \leq F_l \leq N-1 \quad \forall l \in \{1, \dots, L\} \quad (3.10)$$

$$\sum_{i=1}^{L_1} \frac{k_i}{N-F_i} = M_B \quad (3.11)$$

Gerade diese Nebenbedingungen 3.9 bis 3.11 machen eine analytische Lösung nahezu unmöglich, da das  $L$ -dimensionale Gebiet, auf dem  $D$  minimiert werden soll, kaum erfassbare Ränder hat.

Allerdings erlauben gerade diese eine numerische Lösung mit vertretbarem Aufwand. Die Rechenkomplexität die ohne Einschränkungen proportional zum Faktor  $(N-1)^L$  wäre, kann allein schon durch die Bedingung 3.9 auf den Faktor  $(N-1) \cdot \left(\frac{N-1}{2}\right)^{L-1}$  reduziert werden. Für eine genauere Beschreibung der numerischen Lösung sei auf das Kapitel 4.2.3 verwiesen, hier werden im weiteren die Ergebnisse ausgewertet.

### 3.3.4 Verzerrung für einen Kanal

Für die weiteren Berechnungen wurden die Werte eines realen Videos verwendet. Das Video wurde mit dem am Lehrstuhl entwickelten Algorithmen effizient in 6 verschiedene Dateien

mit Daten unterschiedlicher Wichtigkeit komprimiert. Die Datenmengen  $k_l$  der 6 verschiedenen Layer können durch die jeweiligen Dateigrößen so eingestellt werden, daß sie mit einer vergleichbaren Nachrichtenlänge  $K_e$  ( $e$  wie *equal*) für gleichen Fehlerschutz korrespondieren. Wenn nicht anderes erwähnt wird im weiteren immer eine Vergleichsnachrichtenlänge von  $K_e = 200$  Nachrichtenblöcken bei  $N = 255$  Codewortblöcken verwendet. Die Verzerrungen der jeweiligen Layer wurden durch Messung des PSNR (*Peak Signal to Noise Ratio*) ermittelt und im weiteren auch als PSNR-Werte dargestellt, so daß mit realistischen Daten gerechnet wird.

Abbildung 3.9 stellt nun den PSNR mit *unequal error protection* der mit *equal error protection* gegenüber. Typisch ist die Treppenform der Kurve, die sich aus dem Wegfall bzw. dem Decodierversagen der einzelnen Layer ergibt. Auch hier sieht man wiederum die polarisierende Wirkung der Blockkodierung, wie schon in Kapitel 3.2.1 erwähnt wurde.

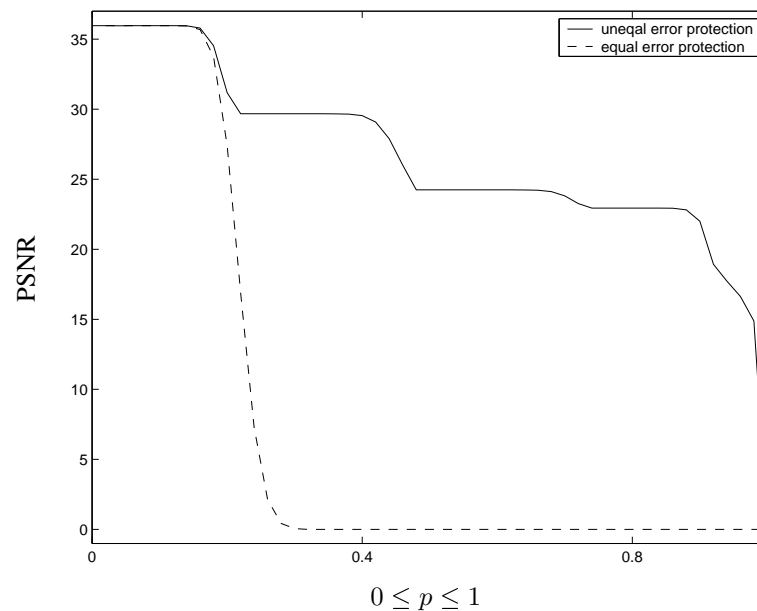


Abbildung 3.9: PSNR mit ungleichem und gleichem Fehlerschutz

### 3.3.5 Zwei Kanäle

Im Kontext dieser Arbeit interessant sind vor allem die Unterschiede, die sich durch *Transcoding* für Datenübertragungen mit ungleichem Fehlerschutz ergeben.

#### Verschobene Stufen

Um die Ergebnisse und die Stufenform der Kurven eindrucksvoller darzustellen, wird in den Abbildungen 3.10 und 3.11 der optimale PSNR für Übertragung mit und ohne *Transcoding* bei einer festen Fehlerrate des Internets  $p_{IP}$  als Funktion der Fehlerrate des Mobilfunkkanals  $p_{Mob}$  dargestellt.

Schon Abbildung 3.10 zeigt deutlich, daß beim *Transcoding* die gesamte Kurve einfach etwas nach hinten geschoben ist, Abfälle an PSNR treten also erst später auf als beim Durchschleifen.

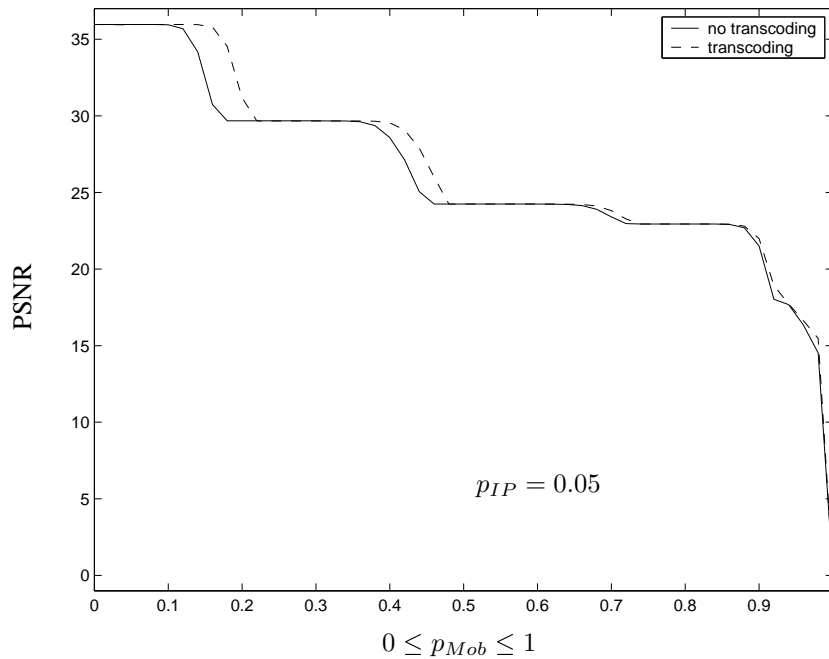


Abbildung 3.10: PSNR mit und ohne *Transcoding* bei  $p_{IP} = 0.05$

*Transcoding* bringt also wie bei gleichem Fehlerschutz aus Kapitel 3.2.2 immer dann Gewinne, wenn die Fehlerrate des Kanals an einer Sprungstelle der Distortion-Rate Kurve liegt. Bei ungleichen Fehlerschutz allerdings, gibt es deutlich mehr als eine Sprungstelle, diese sind zwar von der Größe des Sprungs logischerweise kleiner, aber durch die höhere Anzahl ist es wahrscheinlicher auf eine zu treffen, so daß bei ungleichem Fehlerschutz fast immer Gewinne durch *Transcoding* zu erwarten sind, da die Anzahl der tatsächlichen Fehler schon auf Grund der Varianz der Kanalfehler stark schwankt.

### Größere Unterschiede

Abbildung 3.11 versucht nun einen weiteren Effekt, der bereits aus den informationstheoretischen Betrachtungen von Kapitel 2.2.4 hervorgeht, zu veranschaulichen: Je gleichmäßiger der Fehler auf die beiden Teilkanäle verteilt ist, desto mehr Gewinn ist durch *Transcoding* zu erwarten. Deshalb wird in Abbildung 3.11 die Fehlerrate des Internets auf  $p_{IP} = 0.2$  erhöht. Deutlich zu sehen ist einerseits, daß der maximale PSNR auf Grund der höheren Fehlerrate des Internets kleiner ist, aber andererseits auch, daß der Versatz der Stufen, also die Verschiebung, zwischen den Kurven mit und ohne *Transcoding* deutlich größer geworden ist. Die Regeln aus Kapitel 2.2.4 gelten hier also auch.

### Rate-Distortion-Funktion

Um die Ergebnisse noch aussagekräftiger zu gestalten, wird wie in Kapitel 3.2.2 die Rate-Distortion Funktion ermittelt. Da der Rechenaufwand hier allerdings deutlich höher ist, wurden die Werte aus Kapitel 3.2.2 als Schätzwerte für die zu suchenden Übertragungsraten herangezogen, um dann um diese herum nach den tatsächlichen maximalen Raten  $R$  zu suchen.

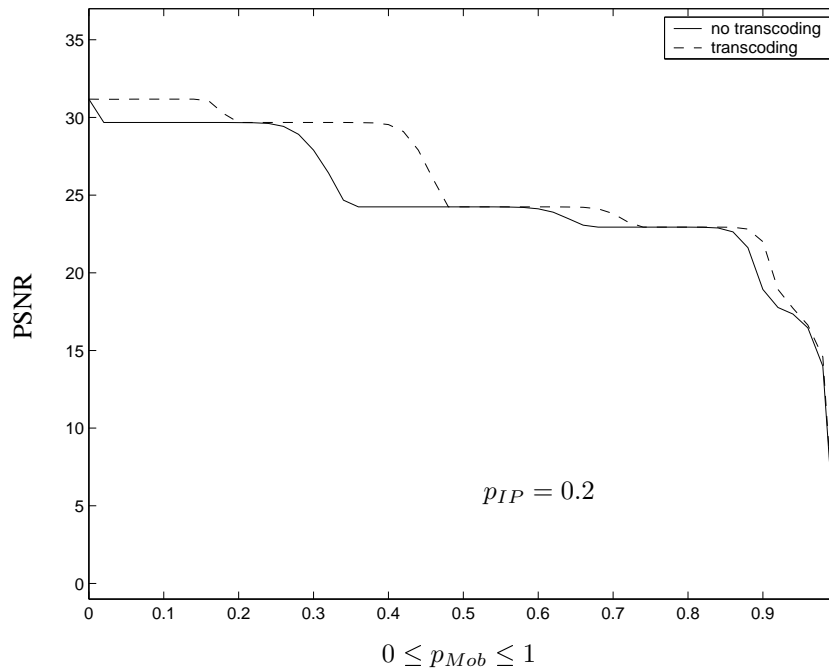


Abbildung 3.11: PSNR mit und ohne *Transcoding* bei  $p_{IP} = 0.2$

Zur Unterscheidung gibt es zwei Plots, einmal in Abbildung 3.12, bei der maximal ein Abfall an PSNR von  $0.3dB$  geduldet wird, es müssen also alle Layer empfangen werden, um  $D_{max} = PSNR_{max} - PSNR = 0.3dB$  sicherzustellen.

Überraschenderweise entsprechen auch für ungleichen Fehlerschutz die Ergebnisse denen aus Kapitel 3.2.2 und somit den informationstheoretischen Betrachtungen aus Kapitel 2.2.4. Die Kapazität, und deren Aussagen zum *Transcoding*, ist also ein relevantes und wichtiges Maß für Betrachtungen der Übertragungsrate  $R$ .

Für Abbildung 3.13 wurden ähnliche Werte wie für Abbildung 3.12 eingesetzt, mit der Ausnahme, daß die Gesamtfehlerrate  $p_G = 0.6$  ist, und daß mehr Verzerrung  $D_{max}$ , also weniger PSNR toleriert wird. Der gemessene PSNR darf vom maximalen Wert um  $6.31dB$  abweichen, was einem Verlust des letzten Layers entspricht. Deutlich zu sehen ist, wie erwartet, der höhere Gewinn durch *Transcoding* auf Grund der größeren Gesamtfehlerrate  $p_G$ .



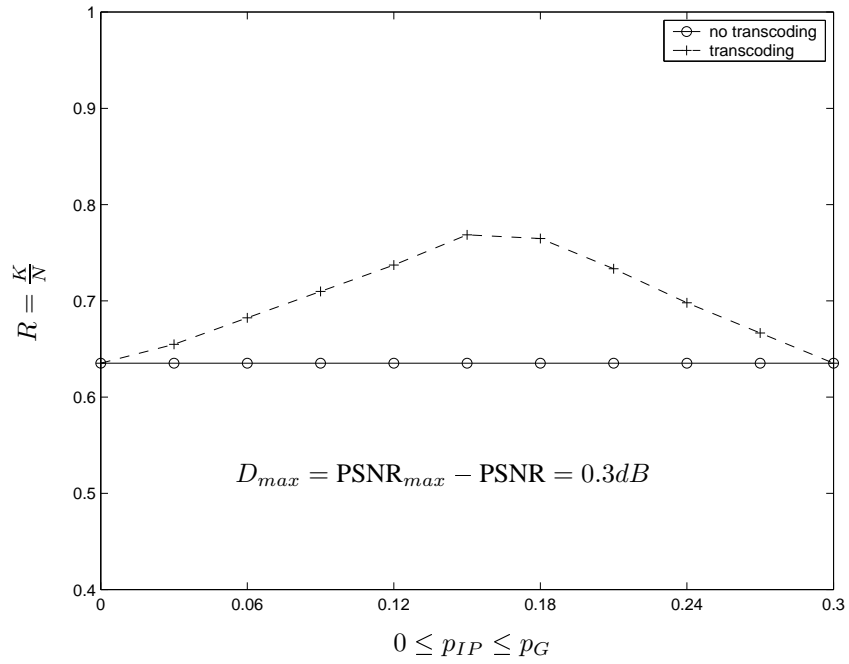


Abbildung 3.12: Übertragungsrate  $R = \frac{K}{N}$  bei tolerierter Verzerrung von  $D_{max} = 0.3dB$

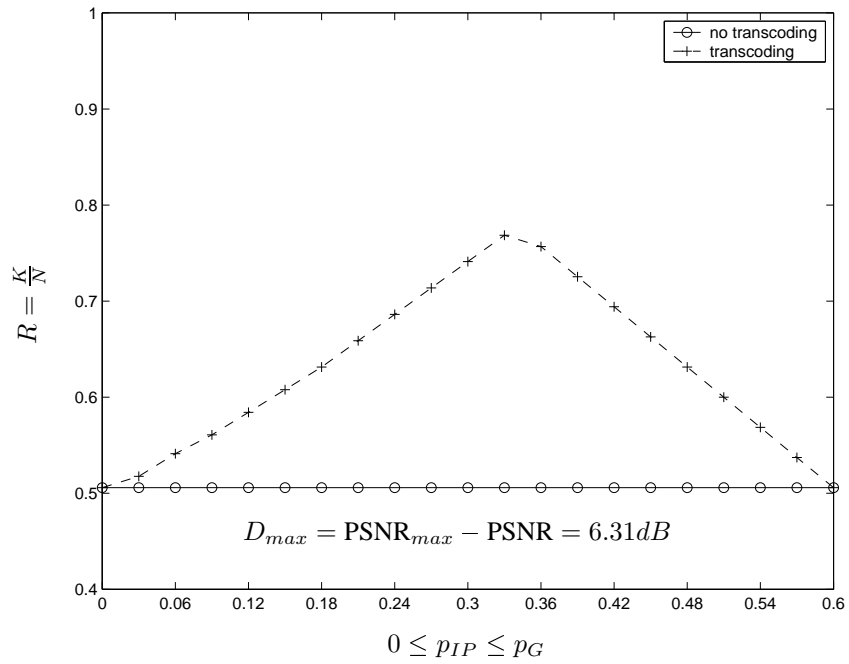


Abbildung 3.13: Übertragungsrate  $R = \frac{K}{N}$  bei tolerierter Verzerrung von  $D_{max} = 6.31dB$

# Kapitel 4

## Verifikation

Als beispielhafte Implementierung eines realen *Streaming*-Szenarios werden in diesem Kapitel die Ergebnisse aus den beiden vorherigen, und insbesondere aus Kapitel 3 verwendet, um ein Video mit ungleichem Fehlerschutz zu versehen und es über zwei Blockfehler-Kanäle zu schicken. Dabei sollen die Ergebnisse und Kurven von Kapitel 3 möglichst gut nachvollzogen werden, um die Brauchbarkeit dieser Arbeit für das *Multimedia-Streaming* zu bestätigen.

### 4.1 Ausgangsmaterial

#### 4.1.1 Das Video

Als Ausgangsmaterial steht zum Beispiel am Lehrstuhl das Testvideo `Vimto` zur Verfügung. Es wurde mit dem am Lehrstuhl entwickelten Quellcodierverfahren effektiv in sechs verschiedene Layer unterschiedlicher Wichtigkeit und Größe komprimiert und in sechs verschiedene Dateien geschrieben. Die von mir verwendeten Dateien wurden ebenfalls nach dem RTP-Standard gemäß [18] paketierte, so daß der Decoder auch bei Paketfehlern die Synchronisation nicht verliert. Um nun die Verzerrung bei der Datenübertragung gemäß Kapitel 3.3.3 zu minimieren wurden dem Video einige Daten entnommen:

- Layerlängen

Zur Optimierung der Redundanzverteilung wird die Anzahl der Datenbytes  $k_l$  pro Codewort für jeden Layer  $l$  benötigt. Dazu wurde eine Vergleichsnachrichtenlänge  $K_e$  festgesetzt, die die Anzahl der Datenblöcke im Codewort der Länge  $N$  ( in unserem Fall immer 255) bei gleichem Fehlerschutz für alle Layer darstellt. Mit den Längen der einzelnen Layer-Dateien konnte somit das passende Verhältnis der  $k_l$  untereinander bestimmt werden und mit der Vergleichsnachrichtenlänge  $K_e$  konnten sie richtig skaliert werden.

$$k_l = \text{Dateilänge} \cdot \frac{K_e \cdot M_B}{\sum_{l=1}^L \text{Dateilängen}}$$

- **Verzerrungen**  
Um die Verzerrung beim Verlust einzelner Pakete zu bestimmen wurde die vorhandene PSNR-Messung des Decoders eingesetzt. Als Variablen  $D_l$  (Definition siehe Kapitel 3.3.2) wurde die Differenz zwischen dem maximalen PSNR-Wert und demjenigen PSNR-Wert herangezogen, der vom Decoder dann berechnet wurde, wenn die entsprechenden Layer zur Decodierung nicht herangezogen wurden. Dies entspricht einer Mittelung über den Verlust aller Pakete und ist wahrscheinlich eine Annahme die nur unzureichend zutrifft, da der Verlust einzelner Pakete sicherlich schwerwiegender ist, als der prozentuale Anteil dieses Pakets am gesamten Layer.
- **Paketgröße**  
Die Paketgröße  $m_B$  ( $[m_B]$  =Byte), die bei der Simulation verwendet wurde, wurde dem Paketiervorgang des Encoders entnommen, der immer einen *Frame* in ein Paket schreibt, solange dieser eine gewisse maximale Paketgröße nicht überschreitet. Ab dieser maximalen Paketgröße schneidet er den Rest ab und sendet ihn erst im nächsten Paket.  $m_B$  wurde also auf die maximale Paketgröße gesetzt.

### 4.1.2 Der Kanal und die Codierung

Als Kanalmodell wurde der Einfachheit halber das in Kapitel 3 behandelte Blockfehlermodell verwendet. Beide Kanäle, Internet und Mobilfunkkanal, transportieren Daten in großen Blöcken mit den Längen  $m_{IP}$  und  $m_{Mob}$  (Einheit jeweils Byte) und kennen nur Blockauslöschungen (Englisch *Erasures*), keine eigentlichen Blockfehler, also falsch empfangene und falsch interpretierte Codewörter.

Wie schon in Kapitel 3.2 beschrieben, sind Reed-Solomon Codes ideal für solche Kanäle und wurden deshalb auch zur Codierung der Videodaten eingesetzt. Als Basis des Codes wurden Byte genommen, weshalb die Codewortlänge  $N = 2^8 - 1 = 255$  beträgt (vgl. [13]). Der Code kann mit Auslöschungen umgehen und bis zu  $N - K$  solcher *Erasures* heben, darüber versagt er allerdings total, weshalb das gesamte Codewort bei mehr als  $N - K$  Fehlern verloren ist.

Die Fehlerquelle wurde auf einfache Weise mit Hilfe des Zufallgenerators des PCs simuliert. Die einzelnen Fehler sind somit stochastisch unabhängig voneinander. Fehler durch das Internet und den Mobilfunkkanal wurden dabei getrennt voneinander ausgeführt.

### 4.1.3 Die PSNR-Messung

Zur Messung des Ergebnisses wurde wiederum die PSNR-Messung des vorhandenen Decoders herangezogen. Das Programm `Coder` erzeugt hierzu aus den ursprünglichen Dateien solche mit fehlenden Paketen. Diese fehlerhaften Dateien werden dann vom Decoder abgespielt, der die PSNR-Werte für jedes Bild in eine Datei schreibt, diese werden wiederum vom Hilfsprogramm `GetPSNR` gemittelt und ausgegeben. Zum Vergleich mit diesen gemessenen Werten, stehen die Werte die das Programm `Coder` selbst ausgibt. Dabei gibt es einerseits die berechneten Werte, die bei der Optimierung der Redundanzverteilung in der Funktion `BestBlock` anfallen und

andererseits die gemessene Verzerrung der Daten, die durch die Anzahl der endgültig richtig empfangenen Layer und deren korrespondierenden Verzerrungen  $D_l$  ermittelt werden kann.

## 4.2 Das Programm und seine Funktionen

### 4.2.1 Der allgemeine Ablauf

Das Programm `Coder` testet und simuliert die Übertragungseigenschaften einer Blockcodierung mit unterschiedlichem Fehlerschutz mit einem Reed-Solomon Code. Dieser befindet sich in den Dateien `ReedSolomon.c` und `ReedSolomon.h` und soll hier nicht weiter erläutert werden.

Es beginnt mit einer Abfrage der Eingabeargumente und entscheidet dann, ob *Transcoding* betrieben wird oder nicht, und, ob eine Videodatei verarbeitet werden soll oder nicht. Je nach gewählter Option, *Transcoding* oder eben nicht, springt es dann in einen der beiden großen Programmabschnitte, die an sich aber gleich ablaufen.

Als erstes werden mit Hilfe der Funktion `BestBlock` die Anzahl der Datenblöcke  $K_l$  der verschiedenen Layer berechnet, dann werden die  $L$  verschiedenen Reed-Solomon Coder initialisiert ( für jede unterschiedliche Anzahl von Datenblöcken wird ein eigener Reed-Solomon Coder benötigt) bevor die eigentliche Übertragungsschleife beginnt.

Die Schleife sammelt die Daten zuerst im  $M \times N$ -Array `code`, je nach Option holt es diese Daten aus den Videodateien oder erzeugt sie. Dann wird das Array `code` mit den  $L$  verschiedenen Reed-Solomon Codern in  $M$  Codewörtern je  $N$  Byte encodiert. Nach der Encodierung wird der Kanal durch zufällig auftretende Fehler simuliert, die ihre Fehlerposition als *Erasure* vermerken. Mit diesen Fehlerpositionen kann der folgende Decoder auch wirklich  $N - K_l$  *Erasures* korrigieren und gibt seinerseits die korrekten Daten dem Array `code` zurück. Gleichzeitig wird in der Variablen `Empfang` vermerkt, wieviele Layer korrekt empfangen wurden. Je nachdem ob *Transcoding* betrieben wird oder nicht, wiederholt sich der bisher beschriebene Vorgang für den Mobilfunkkanal, allerdings wird nun wesentlich mehr Aufwand zur Optimierung der Übertragung betrieben. Es mußte nämlich schon vorher für jede mögliche Anzahl von ankommende Layern die weitere Datenübertragung optimiert werden. Es wird also nun abhängig von der Zahl der empfangenen Layer wieder encodiert und nachher decodiert, bevor die Daten wieder gesichert werden und die aktuelle Verzerrung zur Mittelwertbildung zu den bisherigen addiert wird.

Wenn das Video fertig ist, oder wenn die gewünschte Anzahl an Wiederholungen gemacht wurde, endet das Programm mit der Ausgabe der von der Optimierungsfunktion errechneten Verzerrung und der gemessenen und gemittelten Verzerrung.

### 4.2.2 Bearbeitung eines Videos

Wenn der Name eines Videos in der Kommandozeile angegeben wurde, zum Beispiel `VIMT0`, so werden schon vor der Trennung zum *Transcoding* die entsprechenden Dateinamen der Ein- und Ausgabedateien ergänzt und geöffnet.

Während der simulierten Datenübertragung werden die Videodaten immer in ganzen RTP-Paketen aus den  $L$  Eingabedateien ausgelesen, solange bis kein weiteres Paket mehr in die  $k_l$

Nachrichtenbyte hineinpasst. Auf Grund dieser Aufrundung kann es natürlich gerade bei kleineren Layern dazu kommen, daß noch nicht alle Pakete übertragen worden sind, wähen alle anderen Layer schon durch sind.

Am Ende der Datenübertragung werden dann alle Pakete der Layer, die korrekt empfangen wurden, in die Ausgabedateien geschrieben; diesen können also durchaus einige Pakete fehlen.

### 4.2.3 Die Optimierung der Redundanzverteilung

Das Kernstück des Programmes `Coder` ist die Optimierung der  $K_l$  in der Funktion `BestBlock`. Denn um beide Verfahren, *Transcoding* und Durchschleifen, fair miteinander vergleichen zu können, müssen diese bestmöglich optimiert sein.

#### Die Funktion an sich

`BestBlock` befindet sich in der Header-Datei `Unequal_Optimize.h` und dient eigentlich nur als Rumpf um Vorbereitungen für die eigentliche rekursive Berechnung in der Funktion `Search` zu treffen und dann diese Funktion aufzurufen.

Die wichtigste Vorbereitung ist die Berechnung der Binomialverteilung in der Funktion `BinomSum`, die für jeden verschiedenen Wert von  $p$  neu berechnet werden muß. Danach wird dann nur noch die rekursive Suchfunktion mit den passenden Startwerten gestartet.

#### Berechnung der Binomialverteilung

Der kritische Punkt bei der Optimierung bezüglich der Genauigkeit ist die Berechnung der Binomialverteilung  $\sum_{i=0}^F B(N; p; i)$ , die zur Bildung der Verzerrung nach Gleichung 3.2 benötigt wird.

Im schulischen Mathematikunterricht ist es üblich die Binomialverteilung in Tafelwerken nachzuschlagen, diese sind aber für unsere Zwecke nicht ausreichend, da sie zu wenig und zu kleine Werte für  $N$  beinhalten. Die direkte Berechnung durch ihre Definition nach Gleichung 4.1 (siehe zum Beispiel [21]) beinhaltet leider auch einige rechnerische Tücken.

$$B(N; p; i) = p^i (1 - p)^{N-i} \cdot \binom{N}{i} = p^i (1 - p)^{N-i} \cdot \frac{N!}{(N-i)!i!} \quad (4.1)$$

Gerade für sehr kleine Werte von  $p$  treten Wortlängenfehler auf, die Genauigkeit des Rechners ist nicht gut genug. Andersherum kommt es für große Werte von  $N$  zu Überlauflern bei der Bildung der Fakultät, schon  $200!$  ist mit dem Zahlenformat `double` nicht mehr darstellbar.

Eine Alternative bietet eine Näherungsformel, die bereits in Schulbüchern wie zum Beispiel [21] steht.

$$B(N; p; i) \approx \frac{1}{\sigma} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{i-\mu}{\sigma}\right)^2}$$

mit

$$\begin{aligned}\mu &= N \cdot p \\ \sigma &= \sqrt{N \cdot p(1-p)}\end{aligned}$$

Sie liefert brauchbare Ergebnisse für  $N \cdot p(1-p) > 9$  und wird in der Funktion `BinomSum` eingesetzt um die Binomialverteilung anzunähern.

### Die eigentliche Suchfunktion

Wie schon in Kapitel 3.3.3 beschrieben gilt es die Gleichung 4.2 unter den Nebenbedingungen 4.3 bis 4.5 zu optimieren.

$$D = D_0 - \sum_{l=1}^L (D_{l-1} - D_l) \cdot \sum_0^{F_l} B(N; p; i) \quad (4.2)$$

$$F_{l+1} \leq F_l \quad \forall l \in \{1, \dots, L-1\} \quad (4.3)$$

$$0 \leq F_l \leq N-1 \quad \forall l \in \{1, \dots, L\} \quad (4.4)$$

$$\sum_{i=1}^{L_1} \frac{k_i}{N-F_i} = M_B \quad (4.5)$$

Dies wird von der Funktion `Search` erledigt, wobei gerade die Nebenbedingungen geschickt ausgenutzt werden. Die Funktion benötigt als Eingabeargumente die aktuelle Tiefe `l` in der Layerhierarchie, die Zuweisung von Redundanz an den vorherigen Layer `Fold`, die bisher zu erwartende Verzerrung `Dold`, den noch zur Verfügung stehenden Platz an Bytes im Codewort `Mold` und einen Zeiger auf die Binomialverteilung `*B`.

```
int Search( int l, int Fold, double Dold, int Mold, double *B)
```

Als Rückgabewert liefert sie einen Integer, `success` genannt, der angibt, ob ein neues Minimum gefunden wurde.

Als erstes testet die Funktion, ob der neue Layer mit seiner Größe  $k_l$  überhaupt noch in das Codewort hineinpasst, da ja nur noch `Mold` Byte frei sind, womit Nebenbedingung 4.5 abgefragt wird. Ist dem nicht so, so testet er ob es einen neuen besten Wert für `Dopt` gibt, und sichert diesen mit samt der ihm zugewiesenen Redundanz gegebenenfalls.

Als nächstes wird getestet, ob der letzte Layer erreicht ist, wenn ja, dann wird der verbleibende Platz `Mold` diesem zugewiesen und es wird wiederum getestet, ob ein neues Minimum erreicht wurde.

```
if ( Dold-deltaDmax[l] >= Dopt) return 0;
```

Mit dieser Zeile wird im Folgenden dafür gesorgt, daß sehr schlechte Pfade, die gar nicht mehr zu einem Minimum führen können, nicht weiterverfolgt werden.

Nachdem am Anfang der Funktion die Anzahl der aktuellen Korrekturblöcke `F` auf `Fold` gesetzt wurde, womit Nebenbedingung 4.3 genüge getan wird, wird nun getestet, ob so viele Korrekturblöcke überhaupt noch in den vorhandenen Platz reinpassen und gegebenenfalls wird daraufhin ihre Anzahl verringert.

Von nun an werden in einer Wiederholschleife alle möglichen Werte von `F` bearbeitet und `Search` ruft sich mit den neuen gültigen Werten selbst auf.

### 4.3 Messung

Die Messung sollte vor allem die Verhältnisse zwischen den Verfahren mit *Transcoding* und ohne *Transcoding* bestätigen. Zu diesem Zweck wurden in den Abbildungen 4.1 und 4.2 diese folgenden drei PSNR-Werte, die entweder berechnet oder gemessen wurden, gegenübergestellt.

Wert	Beschreibung
$PSNR_{calc}$	berechneter PSNR-Wert, der von der Funktion <code>BestBlock</code> ausgegeben, bzw. im Falle des <i>Transcoding</i> durch <code>D_adjust</code> berechnet wurde
$PSNR_{data}$	Mittelwert der Verzerrungen der einzelnen Übertragungen, die aus der Zahl der empfangenen Layern ermittelt wurde, und am Schluß in einen PSNR-Wert umgerechnet wurde
$PSNR_{mess}$	echte Messung des PSNR an Hand des Videos <code>vimto</code> mit Hilfe des Video-Decoders

Abbildung 4.1 zeigt diese drei Größen für eine Datenübertragung mit *Transcoding*. Deutlich zu sehen ist dabei bereits die gute Übereinstimmung zwischen den PSNR-Werten, die mit den unterschiedlichsten Methoden ermittelt wurden. Treten Unterschiede in den PSNR's auf, so ist die berechnete Größe  $PSNR_{calc}$  immer größer als die am Video gemessene, da offenbar der Verlust eines Pakets mehr Schaden verursacht als nur der prozentuale Anteil des Pakets am gesamten Layer.

In Abbildung 4.2 sind die gleichen Messungen für die Datenübertragung ohne *Transcoding* durchgeführt worden. Hier ist dagegen deutlich zu merken, daß alle PSNR-Werte immer etwas kleiner sind als die bei der Datenübertragung mit *Transcoding*, die Stufen sind deutlich gegeneinander verschoben und, daß die Unterschiede zwischen den verschiedenen PSNR-Werten geringer ausfallen als bei der Messung mit *Transcoding*, was daran liegt, daß hier nur ein Kanal simuliert werden muß.

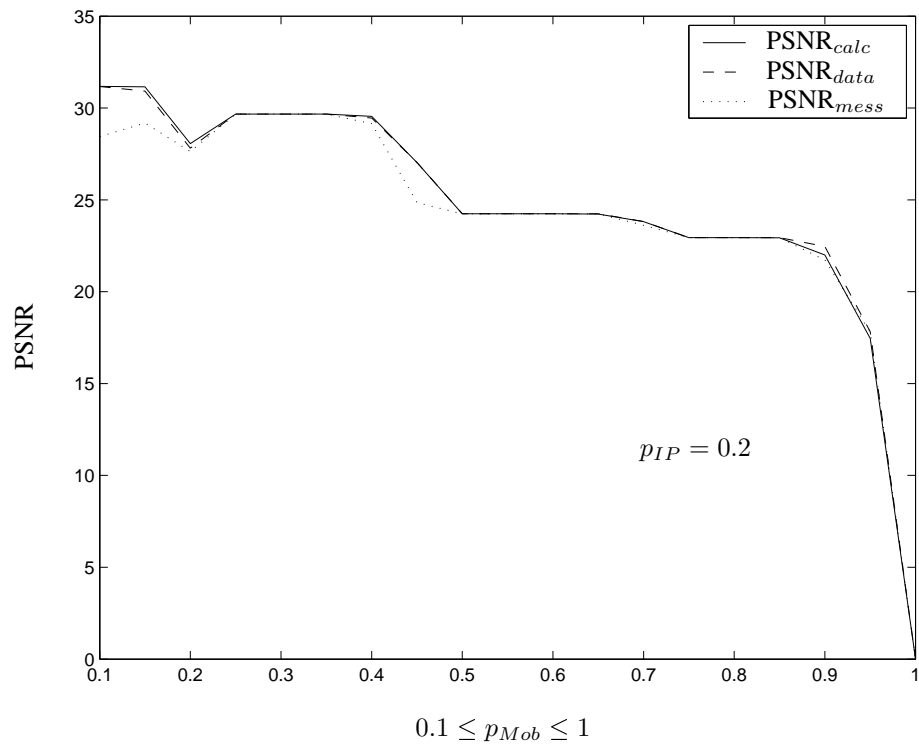


Abbildung 4.1: verschiedene PSNR-Werte, wenn *Transcoding* betrieben wird

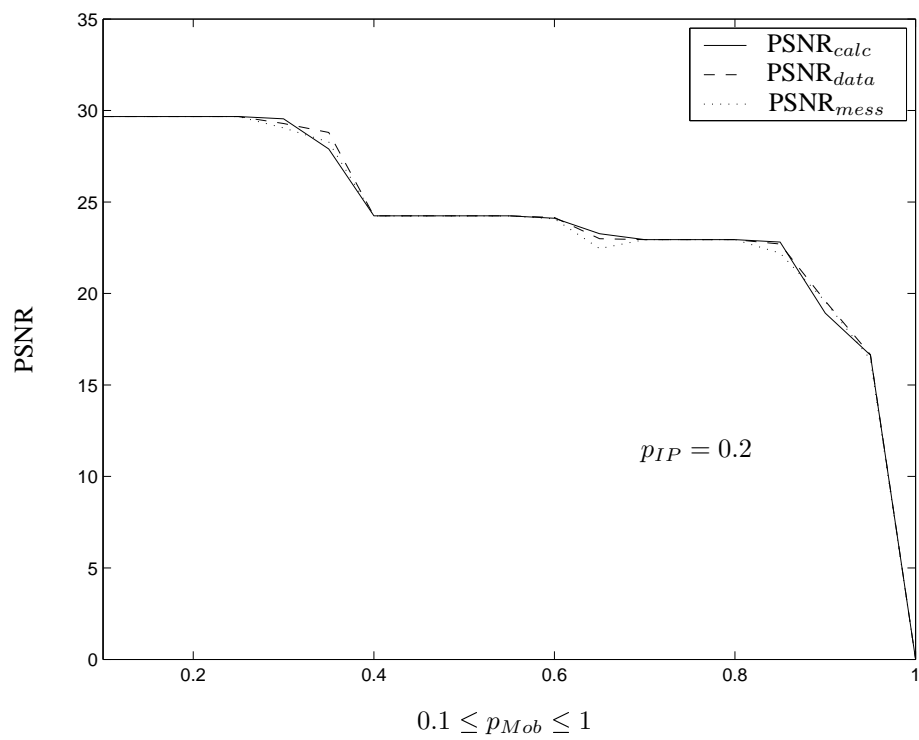


Abbildung 4.2: verschiedene PSNR-Werte, wenn kein *Transcoding* betrieben wird



# Kapitel 5

## Zusammenfassung und Ausblick

In dieser Arbeit wurde das Problem des *streamen* von Multimediainhalten über das Internet und den Mobilfunkkanal auf eine eher theoretische, dafür aber auch allgemeingültige Art und Weise behandelt. Es ging dabei vor allem um die Frage, ob und wieviel ein eingebauter Server an der Nahtstelle zwischen Internet und Mobilfunkkanal, der Basisstation, an Kapazität bzw. Datenrate bringt. Dazu wurden nach einer kurzen Einführung im zweiten Kapitel mehrere Modelle mit den unterschiedlichsten Eigenschaften, wie Unsymmetrie, Blockfehlereigenschaften und Gedächtnis auf informationstheoretischer Basis behandelt, um schließlich zu einem Modell zu führen, das der Realität sehr nahe kommt. Dieser Verbund von zwei verschiedenen Kanalmodellen erlaubte dann eine sehr einfache Abschätzung seiner Kapazität, wodurch jetzt ebenfalls der Gewinn durch *Transcoding* sehr einfach und genau abgeschätzt werden kann, was ein Vergleich zu anderen Autoren bewies.

Im dritten Kapitel wurden die Betrachtungen auf die Effekte der angewandten Codierverfahren ausgedehnt. Es wurde vor allem viel Wert auf die Berechnung der Verzerrung gelegt, die bei der Datenübertragung auftritt. Dabei wurde insbesondere auch *Unequal Error Protection* (ungleicher Fehlerschutz) behandelt. Es wurde durch Simulation und Rechnung gezeigt, daß die Ergebnisse des ersten Kapitels auch hier zutreffen, wenn man nur die Betrachtungsweise umkehrt und zu einer festen Verzerrung die maximal mögliche Datenrate sucht.

Im letzten Kapitel wurde dann ein Beispiel eines Codierverfahrens implementiert und damit die Datenübertragung eines Videos über das Internet und den Funkkanal, mit und ohne *Transcoding*, simuliert. Diese bestätigten durchaus die Ergebnisse des dritten Kapitels, zeigte aber auch, daß gerade bei der Thematik von ungleichem Fehlerschutz, die Suche nach geeigneten Größen zum Messen der Qualität des Videos oder anderer multimedialer Inhalte oft sehr schwer ist.

Einer tatsächlichen Anwendung der Ergebnisse dieser Arbeit steht nichts im Wege, es muß allerdings auch darauf verwiesen werden, daß das angewandten Codierverfahren und das sich ergebende Szenario immer noch einer speziellen Untersuchung bedürfen, da die Eigenarten des Kanalcodes und die Bedingung des Systems, viel ändern können.

# Literaturverzeichnis

- [1] Mordechai Mushkin, Israel Bar-David (1989). **Capacity and Coding for the Gilbert-Elliott Channels**, IEEE Transactions on Information Theory, Vol. IT-35, No. 6, Seite 1277-1290, November 1989
- [2] Andrea J. Goldsmith, Pravin P. Varaiya (1996). **Capacity, Mutual Information, and Coding for Finite-State Markov Channels**, IEEE Transactions on Information Theory, Vol. 42, No. 3, Seite 868-886, Mai 1996.
- [3] Laveebn N. Kanal, A. R. K. Sastry (1978). **Models for Channels with Memory and Their Applications to Error Control**, Proceedings of the IEEE, Vol. 66, No. 7, Seite 724-744, Juli 1978
- [4] Hong Shen Wang (1996). **On Verifying the First-Order Markovian Assumption for a Rayleigh Fading Channel Model**, IEEE Transactions on Vehicular Technology, Vol. 45, No. 2, Seite 353-357, Mai 1996
- [5] Almudena Konrad, Ben Y. Zhao, Anthony D. Joseph, Reiner Ludwig (2001). **A Markov-Based Channel Model Algorithm for Wireless Networks**, Fourth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Juli 2001
- [6] Günther Liebl, Thomas Stockhammer, Frank Burkert (2000). **Modeling and Simulation of Wireless Packet Erasure Channels**, Virginia Tech's Tenth Symposium on Wireless Personal Communications, Juni 2000
- [7] Johannes Huber (2000). **Skript zur Vorlesung Informations-Theorie I. und II.**, Universität Erlangen-Nürnberg, Oktober 2000
- [8] C. E. Shannon, W. Weaver (1949). **The Mathematical Theory of Communication**, University of Illinois Press, 1949
- [9] Uwe Horn, K. Stuhlmüller, M. Link, B. Girod (1999). **Robust Internet Video Transmission Based on Scalable Coding and Unequal Error Protection**, Image Communication, Special Issue on Real-time Video over the Internet, Vol. 15, No. 1-2, Seite 77-94, September 1999
- [10] Wolfgang Sörgel, Klaus Stuhlmüller, Niko Färber, Bernd Girod (2000). **Scalable Video Streaming for Wireless Video-On-Demand Services**, Juni 2000

- [11] Alexander E. Mohr, Eve A. Riskin, Richard E. Ladner (2000). **Unequal Loss Protection: Graceful Degradation of Image Quality over Packet Erasure Channels through Forward Error Correction**, IEEE Journal on Selected Areas in Communication, Vol. 18, No. 7, Seite 819-828, Juni 2000
- [12] Michael Gallant, Faouzi Kossentini (2001). **Rate-Distortion Optimized Layered Coding with Unequal Error Protection for Robust Internet Video**, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, Seite 357-371, März 2001
- [13] T.-W. Angus Lee, S.-H. Gary Chan, Qian Zhang, Wen-Wu Zhu, Ya-Qin Zhang (2001). **Optimal Allocation of Packet-Level and Byte-Level FEC in Video Multicasting over Wired and Wireless Networks**, Proceedings of IEEE Globecom'01, Vol. 3, Seite 1974-1978, November 2001
- [14] Dapeng Wu, Yiwei Thomas Hou, Wenwu Zhu, Ya-Qin Zhang, Jon M. Peha (2001). **Streaming Video over the Internet: Approaches and Directions**, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 3, Seite 282-300, März 2001
- [15] Jozsef Vass, Shelley Zhuang, Xinhua Zhuang (2001). **Scalable, Error-Resilient, and High-Performance Video Communications in Mobile Wireless Environments**, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, No. 7, Seite 833-847, Juli 2001
- [16] International Organisation for Standardisation (1994). **Coding of Moving Pictures and Associated Audio**, ISO/IEC JTC1/SC29/WG11 NO803, November 1994
- [17] International Organisation for Standardisation (2000). **Coding of Moving Pictures and Audio**, ISO/IEC JTC1/SC29/WG11 N3536, Juli 2000
- [18] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson (1996). **RTP: A transport protocol for real-time applications**, RFC 1889. Technical report, IETF, 1996
- [19] GSM-Association (1999). **History of GSM**, <http://www.gsmworld.com/about/history/index.shtml>
- [20] Bronstein, Semendjajew, Musiol, Mühlig (1995). **Taschenbuch der Mathematik - 2. Auflage**, Verlag Harri Deutsch, Frankfurt/Main
- [21] Barth, Mühlbauer, Nikol, Wörle (1985). **Mathematische Formeln und Definitionen - 4. Auflage**, J. Lindauer Verlag, München

# Anhang A

## Kapazität des Gilbert-Elliott Kanals nach Mushkin

### A.1 Modell und allgemeine Definitionen

Das Gilbert-Elliott Kanalmodell beschreibt einen binären symmetrischen Kanal, dessen Bitfehlerwahrscheinlichkeit  $p$  von dem Zustand einer Markovkette mit zwei Zuständen, B für *Bad* und G für *Good*, abhängt.

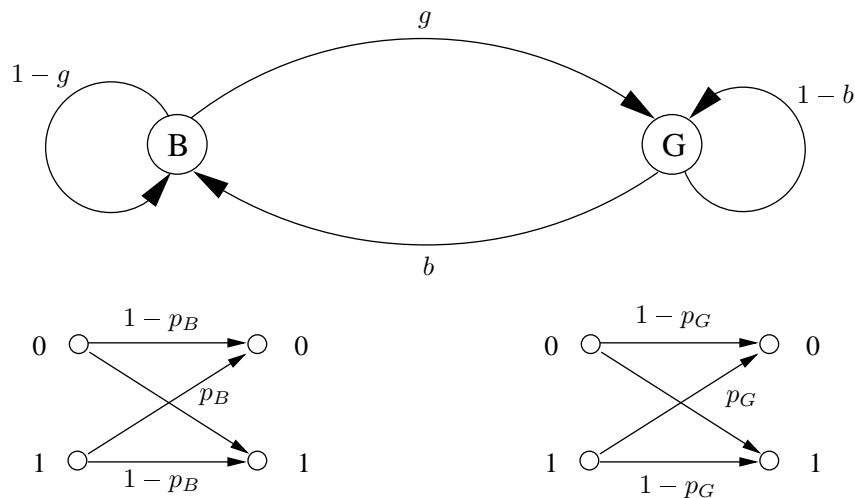


Abbildung A.1: Gilbert-Elliott Kanalmodell

Es ergeben sich demnach die zwei Bitfehlerwahrscheinlichkeiten  $p_B$  und  $p_G$  für die entsprechenden Zustände  $c_1 = B$  und  $c_2 = G$  aus dem Zustandsraum  $\mathcal{C} = \{c_1, c_2\}$  und die beiden Übergangswahrscheinlichkeiten  $b$  und  $g$ . Im weiteren sei außerdem der aktuelle Zustand zum Zeitpunkt  $n$  mit  $S_n$  bezeichnet. Die Eingangsvariablen werden mit  $x$ , die Ausgangsvariablen mit  $y$  notiert. Zusätzlich sei  $z$  Indikator für einen Fehler:

$$z := x \wedge y$$

wobei  $z = 1$  Fehler und  $z = 0$  kein Fehler bedeutet.

Die Variablen  $b$  und  $g$  bilden die Zustandsübergangsmatrix  $P$  mit ihren Elementen  $p_{ij} := \Pr(S_{n+1} = c_j | S_n = c_i)$

$$P = \begin{pmatrix} 1-g & g \\ b & 1-b \end{pmatrix}$$

zudem gibt es für beide Zustände eine Kanalübergangsmatrix mit den Elementen  $k_{ij} := \Pr(y = y_i | x = x_j)$ :

$$\begin{aligned} K_B &= \begin{pmatrix} 1-p_B & p_B \\ p_B & 1-p_B \end{pmatrix} && \text{für den Zustand } S_n = \text{B} \\ K_G &= \begin{pmatrix} 1-p_G & p_G \\ p_G & 1-p_G \end{pmatrix} && \text{für den Zustand } S_n = \text{G} \end{aligned}$$

## A.2 Abgeleitete Größen

Davon abgeleitet besitzt der binäre Markovprozess einige Eigenschaften die auch für den Kanal von entscheidender Bedeutung sind. Die folgenden drei Größen werden auch im weiteren benötigt:

1. mittlere Zustandsaufenthaltswahrscheinlichkeit:  
nach unendlich vielen Durchläufen der Markovkette kann man die mittlere Zustandsaufenthaltswahrscheinlichkeit  $\bar{\pi} = (\pi_B, \pi_G)$  berechnen, sie ergibt sich aber auch durch die Vorschrift (vgl. [7])

$$\bar{\pi} \cdot P = \bar{\pi} \quad \text{mit} \quad \pi_B + \pi_G = 1$$

was nach kurzer Zwischenrechnung zu folgenden Werten führt:

$$\begin{aligned} \pi_B &= \frac{b}{b+g} \\ \pi_G &= \frac{g}{b+g} \end{aligned}$$

2. Zustandsverhältnis:

Als Abkürzung führt man das Zustandsverhältnis  $\rho$  ein

$$\rho := \frac{\Pr(S_n = \text{G})}{\Pr(S_n = \text{B})} = \frac{\pi_G}{\pi_B} = \frac{g}{b} \quad (\text{A.1})$$

was ein gutes Maß für die Güte des Kanals ist, da die durchschnittliche Bitfehlerrate  $\overline{BER}$  nur von den Bitfehlerraten der beiden Zustände *Good* und *Bad* und eben dem Zustandsverhältnis  $\rho$  abhängt.

$$\begin{aligned}\overline{BER} &= \pi_B \cdot p_B + \pi_G \cdot p_G \\ &= p_G + (p_B - p_G) \frac{1}{1 + \rho}\end{aligned}$$

### 3. Gedächtnis:

Die entscheidende Eigenschaft des Markov-Modells, die man sich bei der Kanalmodellierung zu nutze macht, ist das Gedächtnis. Der Zustand  $S_n$  zum Zeitpunkt  $n$  hängt mehr oder weniger von dem vorherigen Zustand  $S_{n-1}$  ab. Dies führt zur Definition von  $\mu$ , dem Kanalgedächtnis:

$$\mu := 1 - g - b \tag{A.2}$$

mit  $\mu = \Pr(S_n = c) - \Pr(S_n \neq c)$  für  $c \in \mathcal{C} = \{G, B\}$  (Beweis nachzulesen in [1]). Dadurch wird offensichtlich, daß der Kanal für  $\mu = 0$  gedächtnislos ist. Ebenfalls offensichtlich ist, daß für  $\mu = 1$  der Kanal immer im selben Zustand bleibt, also deterministisch bestimmt ist. Für  $\mu = -1$  oszilliert der Kanal zwischen den beiden Zuständen, sein Zustand ist also auch deterministisch bestimmbar. Nach A.1 und A.2 kann  $\mu$  also folgende Werte annehmen:

$$\max \left\{ -\rho, -\frac{1}{\rho} \right\} \leq \mu \leq 1$$

Desweiteren kann man mit folgender Schreibweise

$$\mathbf{z}_n := (z_1, z_2, z_3, \dots, z_n),$$

die zur Berechnung der Kanalkapazität benötigte Variable  $q$  als Wahrscheinlichkeit für einen Übertragungsfehler  $z_n = 1$  bei der  $n$ -ten Benutzung des Kanals unter der Bedingung aller zuvor aufgetretenen Übertragungsfehler  $\mathbf{z}_{n-1}$  und des Anfangszustandes  $s_0$  definieren.

$$q_n(\mathbf{z}_{n-1}, s_0) := \Pr(z_n = 1 | \mathbf{z}_{n-1}, s_0) \tag{A.3}$$

## A.3 Rekursive Berechnung von $q_n$

Die Berechnung der Kanalkapazität nach Mushkin (vgl. [1]) beschränkt sich im wesentlichen auf die Berechnung der Werte von  $q_n$  für möglichst große Werte von  $n$ , sowie auf die Bestimmung der Wahrscheinlichkeit für die speziellen Werte von  $q_n$  (vgl. auch das folgende Kapitel

A.4). Daher wird in diesem Abschnitt eine rekursive Berechnung der Werte von  $q_n$  eingeführt. Prinzipiell kann es bei jedem Schritt  $n$  entweder einen Fehler geben,  $z_n = 1$  oder eben auch nicht,  $z_n = 0$ . Dabei sind die Wahrscheinlichkeiten für einen Fehler bereits in der Variablen  $q_n$  selbst festgelegt, es gilt:

$$\Pr(q_{n+1} = \alpha | q_n = \beta) = \begin{cases} 1 - \beta & \text{für } z_n = 0 \\ \beta & \text{für } z_n = 1 \end{cases} \quad (\text{A.4})$$

mit  $\alpha$  und  $\beta$  gemäß A.3

$$\begin{aligned} \alpha &= \Pr(z_{n+1} = 1 | (z_n, \mathbf{z}_{n-1}), s_0) \\ \beta &= \Pr(z_n = 1 | \mathbf{z}_{n-1}, s_0) \end{aligned}$$

Die Berechnung von  $\alpha$  in Abhängigkeit von  $\beta$  und  $z_n$  gestaltet sich deutlich aufwendiger und sei hier nur vorgestellt, genauer nachzulesen in [1]:

$$\alpha \stackrel{z_n=0}{=} \begin{cases} p_G + b(p_B - p_G) + \mu \frac{(\beta - p_g)(1 - p_B)}{1 - \beta} & \text{für } p_B \neq 1 \\ (1 - b)p_G + b & \text{für } p_B = 1 \end{cases}$$

$$\alpha \stackrel{z_n=1}{=} \begin{cases} p_G + b(p_B - p_G) + \mu \frac{p_B(\beta - p_g)}{\beta} & \text{für } p_G \neq 0 \\ (1 - g)p_B & \text{für } p_G = 0 \end{cases}$$

Als Startwerte für die Rekursion nutzt man nun noch die mittlere Zustandsverteilung  $\pi_B$  und  $\pi_G$  als Wahrscheinlichkeit für den bisherigen Verlauf und entsprechend die Kanalübergangswahrscheinlichkeiten  $p_B$  und  $p_G$  als  $q_0$ .

$$\Pr(q_0 = \alpha) = \begin{cases} \frac{b}{b + g} & \text{für } \alpha = p_B \\ \frac{g}{b + g} & \text{für } \alpha = p_G \end{cases} \quad (\text{A.5})$$

## A.4 Die Kapazität des Gilbert-Elliott Kanals und ihre Eigenschaften

Da  $q_n$  nun rekursiv berechnet werden kann, kann nunmehr die Kapazität des Gilbert-Elliott Kanals durch eine einfache Erwartungswertbildung bestimmt werden ( Beweis wieder gemäß [1]).

$$C = 1 - \lim_{n \rightarrow \infty} \mathcal{E} \{e_2(q_n)\} \quad (\text{A.6})$$

mit der binären Entropiefunktion  $e_2(\cdot)$  :

$$e_2(q) := -q \log_2(q) - (1 - q) \log_2(1 - q)$$

Zur Bildung des Erwartungswertes benötigt man die in Kapitel A.3 beschriebenen Anfangs- und Übergangswahrscheinlichkeiten, Gleichung A.5 und A.4, womit sich die Berechnung der Kapazität in Gleichung A.6 wie folgt darstellt:

$$C = 1 - \lim_{n \rightarrow \infty} \sum_{\forall \mathbf{z}_n} \Pr(\mathbf{z}_n) \cdot e_2[q_n(\mathbf{z}_{n-1}, s_0)] \quad (\text{A.7})$$

Ein charakteristischer Verlauf der Kapazität  $C$  für einen Gilbert-Elliott Kanal nach Gleichung A.7 ist in Abbildung A.2 zu sehen. Angetragen ist die Kapazität  $C$  des Kanals für ein festes Zustandsverhältnis  $\rho$  und für feste Bitfehlerwahrscheinlichkeiten  $p_B$  und  $p_G$  gegenüber dem Kanalgedächtnis  $\mu$ .

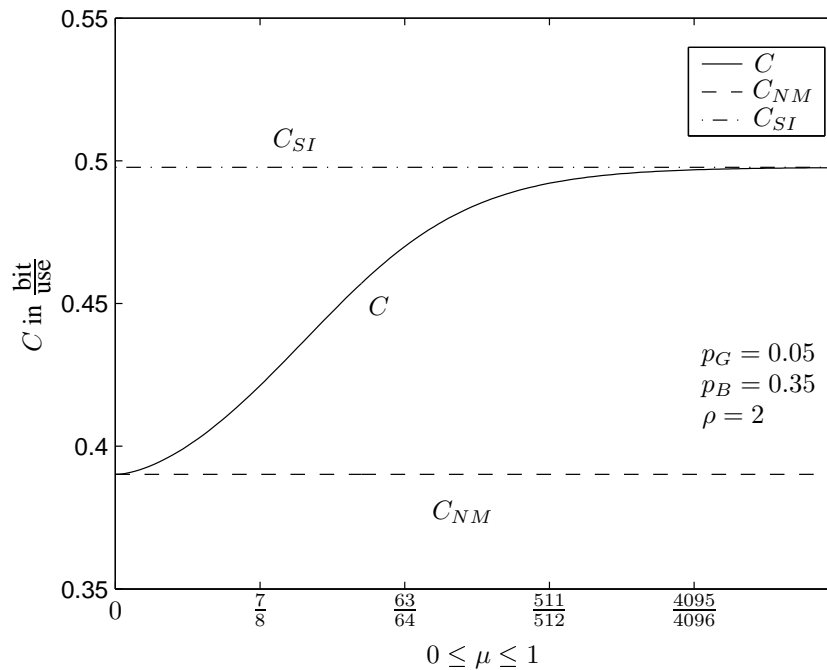


Abbildung A.2: Kapazität  $C$  eines Gilbert-Elliott Kanals

Der Grenzwert  $C_{NM}$  ist dabei die Kapazität des Kanals wenn sein Gedächtnis nicht berücksichtigt wird ( *No Memory* ) oder durch sogenanntes *interleaving* unterdrückt wird.  $C_{SI}$  hingegen ist die Kapazität die man erreichen würde, wenn man perfekte Kenntnis über den Zustand des Kanals hat ( *Side Information* ).



# Anhang B

## Kapazität zweier gekoppelter Gilbert-Elliott Kanäle

### B.1 Modell und Vorgehensweise

Das in Kapitel 2.3.2 beschriebene Modell hat nun vier verschiedene Zustände und beschreibt somit auch vier verschiedene Bitfehlerwahrscheinlichkeiten. Abbildung B.1 verdeutlicht das Szenario nochmals, wobei die vier Bitfehlerwahrscheinlichkeiten sich aus denen der einzelnen Kanäle bilden.

$$\begin{aligned} p_{BB} &= (1 - p_{B1})p_{B2} + p_{B1}(1 - p_{B2}) \\ p_{BG} &= (1 - p_{B1})p_{G2} + p_{B1}(1 - p_{G2}) \\ p_{GB} &= (1 - p_{G1})p_{B2} + p_{G1}(1 - p_{B2}) \\ p_{GG} &= (1 - p_{G1})p_{G2} + p_{G1}(1 - p_{G2}) \end{aligned}$$

Zur Berechnung der Kapazität dieses Kanals gibt es keine speziellen Vorlagen mehr, man kann aber allgemeinere Veröffentlichungen heranziehen und auf diesen speziellen Fall anwenden. In dieser Arbeit geschehen mit [2], mit der die Berechnung der Kapazität von *Finite-State Markov Channels* möglich ist, die in diesem Abschnitt auf den Fall der Abbildung B.1 angewendet wird.

### B.2 Rekursive Berechnung der bedingten Zustandsverteilung

Wie auch in Kapitel A erfolgt nach [2] die Berechnung der Kapazität von *Finite-State Markov Channels* durch die rekursive Ermittlung spezieller Zufallsgrößen. Hier werden deshalb die beiden bedingten Zustandsverteilungen  $\rho_n$  und  $\pi_n$ , sowie ihre allgemeine rekursive Berechnung und ihre Übergangswahrscheinlichkeiten nach [2] eingeführt.

Mit

$$\begin{aligned} \rho_n(k) &= \Pr(S_n = c_k | \mathbf{y}_{n-1}) \\ \pi_n(k) &= \Pr(S_n = c_k | \mathbf{x}_{n-1}, \mathbf{y}_{n-1}) \end{aligned}$$

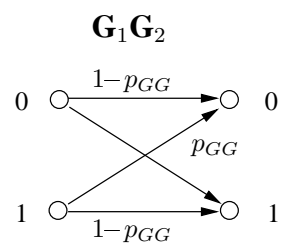
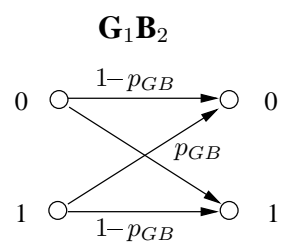
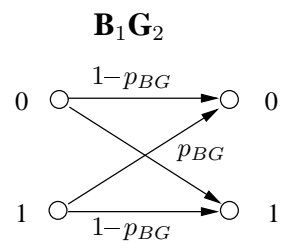
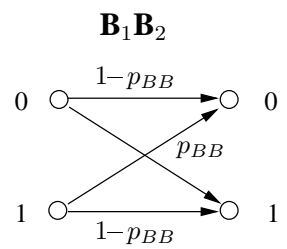
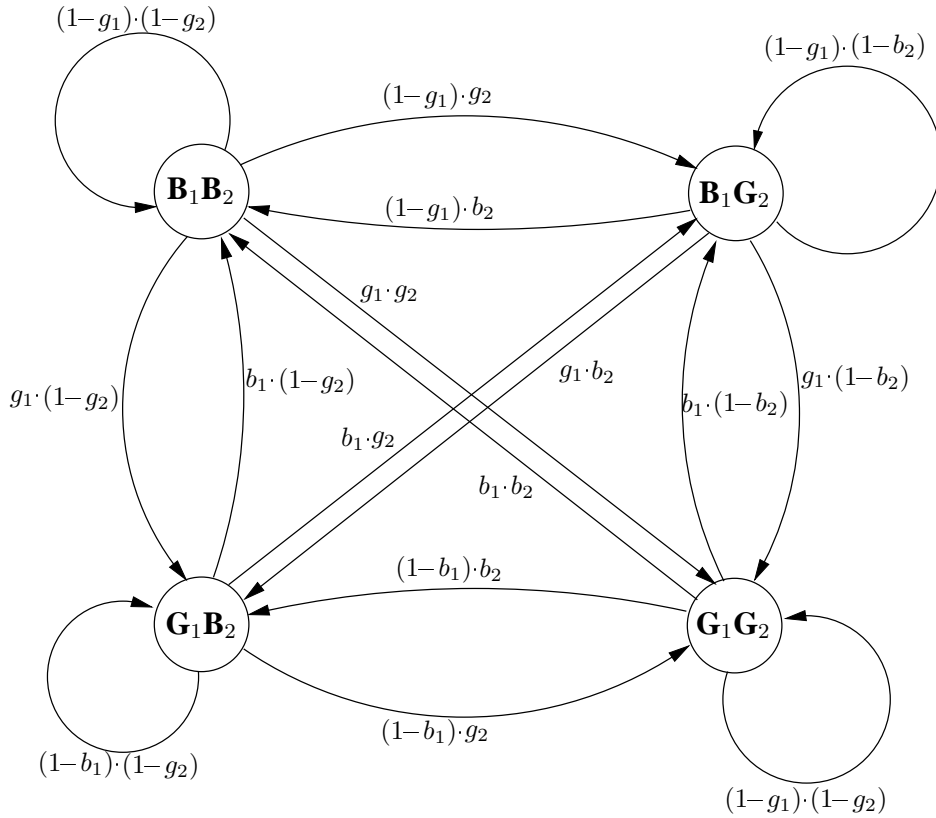


Abbildung B.1: Kanalmodell mit vier Zuständen

werden die bedingten Zustandsverteilungen  $\rho_n$  und  $\pi_n$  wie folgt definiert

$$\begin{aligned}\rho_n &= (\rho_n(1), \rho_n(2), \dots, \rho_n(K)) \\ \pi_n &= (\pi_n(1), \pi_n(2), \dots, \pi_n(K))\end{aligned}$$

wobei  $c_k \in \mathcal{C}$  und  $K = \|\mathcal{C}\|$  für  $\mathcal{C} := \{c_1, c_2, \dots, c_K\}$  gilt.

Die rekursive Berechnung von  $\rho_n$  geschieht folgendermaßen:

$$\rho_{n+1} = \frac{\rho_n B(y_n) P}{\rho_n B(y_n) \underline{1}}$$

mit  $B(y_n)$  als  $K \times K$  Diagonalmatrix mit den Elementen  $\Pr(y_n | S_n = c_k)$ , der Zustandsübergangsmatrix  $P$  gemäß Kapitel A.1 und dem einfachen stehenden  $K \times 1$  Vektor  $\underline{1} := (1, 1, \dots, 1)^T$ . Als Startwert nutzt man die mittlere Zustandsaufenthaltswahrscheinlichkeit  $\bar{\pi}$  nach Kapitel A.2, womit  $\rho_0 = \bar{\pi}$ .

Die Übergangswahrscheinlichkeiten errechnen sich wie folgt

$$\Pr(\rho_{n+1} = \alpha | \rho_n = \beta) = \sum_{\forall y_n | \beta \xrightarrow{y_n} \alpha} \Pr(y_n | \rho_n = \beta)$$

Für  $\pi_n$  gestaltet sich die rekursive Berechnung sehr ähnlich:

$$\pi_{n+1} = \frac{\pi_n D(x_n, y_n) P}{\pi_n D(x_n, y_n) \underline{1}}$$

wobei  $D(x_n, y_n)$  in analoger Weise eine  $K \times K$  Diagonalmatrix ist, nun aber mit den Elementen  $\Pr(y_n | x_n, S_n = c_k)$ .  $P$  und  $\underline{1}$  sind definiert wie oben, der Startwert ist ebenfalls identisch,  $\pi_0 = \bar{\pi}$  und die Übergangswahrscheinlichkeiten werden folgendermaßen ermittelt:

$$\Pr(\pi_{n+1} = \alpha | \pi_n = \beta) = \sum_{x_n, y_n | \beta \xrightarrow{x_n, y_n} \alpha} \Pr(y_n | x_n, \pi_n = \beta) \cdot \Pr(x_n)$$

### B.3 Ermittlung der Kapazität und Unterschiede zu Mushkin

Da die bedingten Zustandsverteilungen  $\rho_n$  und  $\pi_n$  nun rekursiv berechnet werden können, kann mit ihnen die Kapazität ermittelt werden. In der allgemeinen Form geschieht dies nach Gleichung B.1.

$$\begin{aligned}C = \lim_{n \rightarrow \infty} \max_{\forall \mathbf{x}_n} \frac{1}{n} \sum_{i=1}^n \left[ \mathcal{E} \left\{ -\log_2 \sum_{k=1}^K \Pr(y_i | S_i = c_k) \rho_i(k) \right\} - \right. \\ \left. - \mathcal{E} \left\{ -\log_2 \sum_{k=1}^K \Pr(y_i | x_i, S_i = c_k) \pi_i(k) \right\} \right] \quad (\text{B.1})\end{aligned}$$

Der erste Erwartungswert in Gleichung B.1 ist für gleichverteilte Ausgangssymbole immer konstant, da die Summe über alle  $\rho_i(k)$  eins ist, und ist speziell für binäre Ausgangssymbole eins.

Die Mittelwertbildung aus Gleichung B.1 ist nötig, da das Eingangssignal nicht zwingend stationär sein muß, und man somit nicht von stationären Werten in der eckigen Klammer ausgehen kann.

Die Transformation wird bei gleichmäßig streuenden und sammelnden Kanälen durch gleichverteilte, stochastisch unabhängige Eingangssymbole erreicht (vgl. [7]). Für solche Kanäle kann demnach die Maximierung in Gleichung B.1 entfallen, wenn man gleichverteilte und stochastisch unabhängige Eingangssymbole wählt. Gleichzeitig kann auch die Mittelwertbildung entfallen, da stochastisch unabhängige Eingangssignale stationär sind. Die vereinfachte Berechnung der Kapazität geht nun nach Gleichung B.2.

$$C = 1 - \lim_{n \rightarrow \infty} \mathcal{E} \left\{ -\log_2 \sum_{k=1}^K \Pr(y_n | x_n, S_n = c_k) \pi_n(k) \right\} \quad (\text{B.2})$$

Für binäre Ein- und Ausgangssymbole kann man desweiteren den Fehlerindikator  $z = x \wedge y$  und somit auch die Definition  $q_n := \Pr(z_n = 1 | \mathbf{z}_{n-1}, s_0)$  aus Kapitel A.1 verwenden. Und da die bedingte Wahrscheinlichkeit  $\Pr(y_n | x_n)$  nur zwei Zustände, nämlich  $z = 1$  und  $z = 0$  annehmen kann, vereinfacht sich die Kapazitätsberechnung für Kanäle mit binären Ein- und Ausgangssymbolen zu Gleichung B.3.

$$C = 1 - \lim_{n \rightarrow \infty} \mathcal{E} \{ -\log_2 q_n - \log_2(1 - q_n) \} \quad (\text{B.3})$$

Und für den weiteren vereinfachten Fall eines Gilbert-Elliott Kanals kann die Erwartungswertbildung so aufgelöst werden, daß sich die unten nochmals dargestellte Gleichung A.7 ergibt (man beachte, daß hier der Erwartungswert auch über alle möglichen Paare von  $x_n$  und  $y_n$  gebildet wurde, für Gleichung A.7 waren nur alle Möglichkeiten von  $z$  zu berücksichtigen).

$$C = 1 - \lim_{n \rightarrow \infty} \sum_{\forall \mathbf{z}_n} \Pr(\mathbf{z}_n) \cdot e_2 [q_n(\mathbf{z}_{n-1}, s_0)]$$

Beide Methoden sind demnach ziemlich ähnlich in ihrer Vorgehensweise, die eben Beschriebene ist gegenüber der nach [1] aus dem vorherigen Kapitel nur allgemeiner.