

Institut für Informatik  
der Technischen Universität München  
Lehrstuhl für Informatik VIII

**Verfahren zur Übertragung von  
Dienstgüteaussagen**

*Marco Hoffmann*

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Rüdiger Westermann

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Eike Jessen, em.
2. Univ.-Prof. Dr. Uwe Baumgarten

Die Dissertation wurde am 07.07.2005 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 01.12.2005 angenommen.



# Verfahren zur Übertragung von Dienstgüteaussagen

**Marco Hoffmann**

Institut für Informatik  
Lehrstuhl für Rechnerkommunikation  
Technische Universität München, Germany  
hoffmmar@t-online.de

## Stichwörter

Quality of Service, Differentiated Services, Integrated Services, NS2, Simulation, Extrapolation, Gigabit Wissenschaftsnetz, Warteschlangentheorie

## Kurzdarstellung

Die stetige Weiterentwicklung von Telekommunikationsnetzen bringt unter anderem die Einführung neuer Realzeit-Anwendungen mit sich. Diese Anwendungen können sowohl delay-, jitter- als auch durchsatz-sensitiv sein. Daher ist es unerlässlich jeder dieser Anwendungen eine angemessene Dienstgüte (*Quality of Service*, QoS) für ihre Datenübertragung bereitzustellen, um ihre Funktionen im vollen Umfang nutzen zu können. Zu diesem Zweck wurden von der *Internet Engineering Task Force* (IETF) verschiedenste QoS Technologien standardisiert. Das Projekt des Deutschen Forschungsnetzes (DFN) *Labor-Experiment zum Testen von systemischer QoS* (LETS QoS), in dessen Rahmen die vorliegenden Ergebnisse entstanden, hatte die Bewertung bestehender QoS Technologien hinsichtlich Realisierbarkeit in Zusammenhang mit Kosten und Nutzen, sowie eine Prognose über die QoS Technologieentwicklung im Internet zum Ziel. Diese Dissertation entwickelt basierend darauf und in Abstimmung dazu ein Verfahren zur quantitativen Vorhersage von QoS Eigenschaften von Netzen für vorgegebene QoS Technologien, Verkehrsmischungen, Konfigurationen und resultierende Lastverteilungen. Dieses Verfahren zeichnet sich durch die Fähigkeit aus, in bekannten Umgebungen erzielte Meßergebnisse auf unbekannte Umgebungen übertragen zu können. In dieser Arbeit wird die im Mittelpunkt der Untersuchung stehende Umgebung die des Gigabit-Wissenschaftsnetzes (G-WiN) des DFN sein. Ziel ist es, die unterschiedlichen QoS Technologieansätze für das G-WiN zu untersuchen und anhand der Ergebnisse eine QoS Technologieempfehlung auszusprechen.

## Danksagung

An aller erster Stelle möchte ich meinem Doktorvater Prof. Jessen von der TU-München für seine große Unterstützung bei der Erstellung dieser Arbeit danken, desweiteren Prof. Baumgarten, der sich freundlicherweise als Zweitkorrekteur zur Verfügung stellte, sowie Dr. Jobmann, der mir stets durch Rat und Tat zur Seite stand. Außerdem danke ich Prof. Feldmann und ihrem Lehrstuhl für die dauerhafte Unterstützung. Zum Abschluß bedanke ich mich noch bei meiner Familie, insbesondere bei meiner Frau, die auf Grund dieser Arbeit einiges über sich ergehen lassen mußte.

# Abkürzungen

AC	Admission Control, 16	JoBS	Joint Buffer management and Scheduling, 29
AdSpec	Advertising Specification, 13	LETs QoS	Labor-Experiment zum Testen von systemischer QoS, 1
AF	Assured Forwarding, 15	LRZ	Leibniz Rechenzentrum, 176
ALTQ	Alternate Queueing, 29	LS	Lastspektrum, 31
AON	All Optical Networks, 191	MTU	Maximum Transfer Unit, 23
ASON	Automatically Switched Optical Networks, 191	nexp	negativ exponentielle Verteilung, 53
BB	Bandwidth Broker, 17	NK	Netzwerkkonfiguration, 31
BE	Best Effort, 11	NS2	Network-Simulator (Version 2), 26
BSD	Berkely Software Design, 29	OTcl	objektorientierte Erweiterung der Skriptsprache Tcl, 26
CBQ	Class-Based Queueing, 15	P2P	Peer to Peer, 24
CBR	Constant Bit Rate, 25	PHB	Per-Hop Behaviour, 14
CNM	Customer Network Management, 174	PQ	Priority Queueing, 15
Codecs	Coder-decoder compression schemes, 22	PSTN	Public Switched Telephone Network, 22
DFN	Deutsches Forschungsnetz, 1	QBSS	QBone Scavenger Service, 16
DiffServ	Differentiated Services, 14	QoS	Quality of Service, 1
DS	Differentiated Services, 14	QS	QoS Spektrum, 32
DSCP	DiffServ Codepoint, 14	QT	QoS Technologien, 32
EF	Expedited Forwarding, 15	RAM	Random Access Memory, 29
erl	Erlang Verteilung, 55	RED	Random Early Detection, 27
ES	Extrapolations-Szenario, 31	RIO	RED with IN and OUT profile, 29
ETS	Extrapolations-Testszenario, 32	RSpec	Reserve Specification, 13
FCFS	First Come First Serve, 11	RSVP	Reservation Protocol, 12
FIFO	First In First Out, 28	SDH	Synchronous Digital Hierarchy, 22, 173
G-WiN	Gigabit Wissenschaftsnetz, 8	SLA	Service Level Agreement, 14
GML	Graph Modeling Language, 24	TB	Token Bucket, 40
GPS	Global Positioning System, 29	TCA	Traffic Conditioning Agreement, 14
H-FSC	Hierarchical Fair Service Curve, 29	TclCl	Tcl-C-Linkage Layer, 26
hyp2	hyperexponentielle Verteilung, 53	TCP	Transmission Control Protocol, 68
IETF	Internet Engineering Task Force, 1	TFF	Topology File Format, 24
IntServ	Integrated Services, 11	TOS	Type of Service, 14
IP	Internet Protocol, 173	tpt	Truncated Powertail Verteilung, 53
IPv4	Internet Protocol version 4, 14	TSpec	Traffic Specification, 13
IS	Integrated Service, 12	TTL	Teleteaching / Telelearning, 20
ISDN	Integrated Services Digital Network, 22	UDP	User Datagram Protocol, 25
ITU	International Telecommunication Union, 22		

---

VBR	Variable Bit Rate, 22
VoIP	Voice over IP, 19
WDM	Wavelength Division Multiplexing, 173
WFQ	Weighted Fair Queueing, 42
WRR	Weighted Round Robin, 15, 49
WWW	World Wide Web, 20

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Überblick</b>	<b>8</b>
<b>2</b>	<b>QoS Technologien</b>	<b>11</b>
2.1	Best Effort (BE)	11
2.2	Overprovisioning	11
2.3	Integrated Services (IntServ)	11
2.3.1	IS Model	12
2.3.1.1	Guaranteed Service	12
2.3.1.2	Controlled Load Service	12
2.3.2	Framework	12
2.3.2.1	Reservierungsprotokoll (Reservation Setup Protocol)	12
2.3.2.2	Weitere Framework Komponenten	13
2.4	Differentiated Services (DiffServ)	14
2.4.1	Expedited Forwarding PHB	15
2.4.2	Assured Forwarding PHB	15
2.4.3	Default PHB	15
2.4.4	Unterschiedliche DiffServ Ansätze	16
2.4.4.1	Default Ansatz	16
2.4.4.2	Olympic Ansatz	16
2.4.4.3	Admission Control (AC)	16
<b>3</b>	<b>Anwendungen im Hinblick auf Dienstgüte</b>	<b>18</b>
3.1	Auswirkungen der Dienstgütemetriken auf QoS-sensitive Anwendungen	19
3.2	Einteilung der Anwendungen nach Metriken	20
<b>4</b>	<b>Charakteristik und Generierung von Verkehr</b>	<b>22</b>
4.1	Verkehr QoS-sensitiver Anwendungen	22
4.2	Verkehrsgenerierung	23
<b>5</b>	<b>Szenario-Generator</b>	<b>24</b>
5.1	Topologie-Generator	24
5.2	Last-Generator	24

---

<b>6</b>	<b>Netzwerk-Simulator</b>	<b>26</b>
6.1	NS2 - Netzwerk-Simulator . . . . .	26
6.2	Anbindung an den Szenario-Generator . . . . .	26
6.3	IntServ Modul . . . . .	27
6.4	DiffServ Modul . . . . .	27
<b>7</b>	<b>Testbett</b>	<b>29</b>
7.1	Beschreibung . . . . .	29
7.2	Alternate Queueing Package (ALTQ) . . . . .	29
<b>8</b>	<b>Extrapolationsverfahren</b>	<b>31</b>
8.1	Definition . . . . .	31
8.2	Extrapolation . . . . .	32
8.2.1	Simulationen . . . . .	33
8.2.2	Berechnung . . . . .	35
<b>9</b>	<b>Modellierung und analytischer Ansatz</b>	<b>37</b>
9.1	Architekturen und Best/Worst Case Analyse . . . . .	37
9.1.1	Best Effort . . . . .	37
9.1.2	IntServ . . . . .	38
9.1.3	DiffServ . . . . .	44
9.1.3.1	Olympic Ansatz . . . . .	44
9.1.3.2	Default Ansatz . . . . .	48
9.2	Stochastisches Verfahren . . . . .	51
9.2.1	Wartezeitformel . . . . .	51
9.2.2	Extrapolation mit stochastischen Modellen . . . . .	63
<b>10</b>	<b>Extrapolations-Testszenario</b>	<b>67</b>
10.1	Spezifizierung des Ausgangsszenarios . . . . .	67
10.2	Sensitivitätsanalyse und Extrapolation durch Simulation . . . . .	68
10.2.1	Simulationen mit Best Effort . . . . .	69
10.2.2	Simulationen mit IntServ . . . . .	80
10.2.3	Simulationen mit DiffServ (Olympic Ansatz) . . . . .	89
10.2.4	Simulationen mit DiffServ (Default Ansatz) . . . . .	102
10.2.5	Overprovisioningexperimente . . . . .	115
10.2.6	QoS Fallstudie . . . . .	122
10.2.7	Validierung . . . . .	125
10.2.8	Bewertung des simulativen Extrapolationsverfahrens . . . . .	125

10.3	Best/Worst Case Analyse . . . . .	125
10.3.1	Best Effort . . . . .	125
10.3.2	Olympic Ansatz . . . . .	133
10.3.3	Default Ansatz . . . . .	141
10.3.4	Bewertung der Best/Worst Case Extrapolation . . . . .	149
10.4	Stochastisches Verfahren . . . . .	151
10.4.1	Variation der QoS Technologie . . . . .	151
10.4.2	Variation des Lastspektrums . . . . .	155
10.4.3	Variation der Netzkonfiguration . . . . .	158
10.4.4	Bewertung des stochastischen Verfahrens . . . . .	163
10.5	Allgemeine Extrapolationsdurchführung . . . . .	164
<b>11</b>	<b>G-WiN Extrapolations-Szenario</b>	<b>172</b>
11.1	G-WiN Topologie und Auslastung . . . . .	172
11.1.1	Beschreibung . . . . .	172
11.1.2	Die G-WiN Kernnetztopologie . . . . .	172
11.1.3	Lastsituation des G-WiN . . . . .	174
11.1.3.1	Kernnetzauslastung . . . . .	174
11.1.3.2	Auslastung der Zugangsleitungen . . . . .	175
11.1.4	Verkehrszusammensetzung . . . . .	176
11.2	Ausgangsszenario- und Zielszenarienspezifikation für das G-WiN . . . . .	176
11.3	Prüfung der G-WiN Voraussetzungen . . . . .	179
11.4	Extrapolationsdurchführung . . . . .	179
11.4.1	Best/Worst Case Analyse . . . . .	179
11.4.2	Simulationen . . . . .	183
11.4.3	Stochastisches Verfahren . . . . .	186
11.4.4	Zusammenfassung und Empfehlung . . . . .	188
<b>12</b>	<b>Zusammenfassung und Ergebnisse</b>	<b>189</b>
<b>13</b>	<b>Offene Probleme und Weiterentwicklungsmöglichkeiten</b>	<b>191</b>
<b>A</b>	<b>Definitionen und Bezeichnungen</b>	<b>192</b>
A.1	Definitionen . . . . .	192
A.2	Variablenbezeichnungen . . . . .	195

---

<b>B</b>	<b>Singlequeue Simulator Ergebnisse</b>	<b>198</b>
B.1	Ergebnisse der Heavy Traffic Formel . . . . .	198
B.2	Ergebnisse zusätzlicher Variationen der Variationskoeffizienten . . . . .	203
<b>C</b>	<b>Simulationsergebnisse des Extrapolations-Testszenarios</b>	<b>220</b>
<b>D</b>	<b>G-WiN Kernnetztopologie</b>	<b>272</b>
	<b>Literaturverzeichnis</b>	<b>288</b>

## Kapitel 1

# Einleitung und Überblick

Die stetige Weiterentwicklung von Telekommunikationsnetzen bringt unter anderem die Einführung neuer Realzeit-**Anwendungen** mit sich. Diese Anwendungen können sowohl **delay** -, **jitter** - als auch **durchsatz-sensitiv** sein. Daher ist es unerlässlich jeder dieser Anwendungen eine angemessene **Dienstgüte** (*Quality of Service*, QoS) für ihre Datenübertragung bereitzustellen, um ihre Funktionen im vollen Umfang nutzen zu können. Zu diesem Zweck wurden von der *Internet Engineering Task Force* (IETF) verschiedenste **QoS Technologien** standardisiert. Das Projekt des Deutschen Forschungsnetzes (DFN) *Labor-Experiment zum Testen von Systemischer QoS* (LETS QoS), in dessen Rahmen die vorliegenden Ergebnisse entstanden, hatte die Bewertung bestehender QoS Technologien (QT) hinsichtlich Realisierbarkeit in Zusammenhang mit Kosten und Nutzen, sowie eine Prognose über die QoS Technologieentwicklung im Internet zum Ziel. Um dies zu erreichen, wurden sowohl bestehende, als auch Ansätze für neue QoS Technologien untersucht und nach Realisierbarkeit für das Gigabit Wissenschaftsnetz (G-WiN) des DFN ausgewählt. Eine Beschreibung der entsprechenden nach deren Kriterien ausgewählten QoS Technologien wird in Kapitel 2 vorgenommen. Um diese QoS Technologien Untersuchungen von möglichst realen Bedingungen zu unterziehen, wurde eine Implementation in einem Testbett (siehe Kapitel 7), sowie in einem parallel zum Testbett mit identischer Topologie und Verkehr speisbaren Simulator (siehe Kapitel 6) durchgeführt. Die Topologie und der Verkehr werden durch einen innerhalb des Projektes entwickelten Szenariogenerator (siehe Kapitel 5) erzeugt.

Basierend auf und in Abstimmung zu den Untersuchungen des Projektes entwickelt diese Dissertation ein neuartiges Verfahren, das es ermöglicht durch Beobachtung von Verkehrsgütemetriken deren neue Werte vorauszusagen, die sich bei Variation der Quantität und Qualität der Last, der Komponenten und der Topologie oder der QoS Technologien ergeben. Dieses Verfahren wird im Anschluß als *Extrapolation* bezeichnet. Hierbei sollen in einem Netz näherungsweise **strombezogene** Dienstgütevorsagen unter Rückgriff auf paketbezogene Kenntnisse aus ähnlichen Situationen gemacht werden. Die ähnliche Situation kann durch folgende Freiheitsgrade gegeben sein:

- “anderes Netz“ - Die Kenntnis über die Werte der Verkehrsgütemetriken ist in einem realen oder modellierten Netz (Ausgangsnetz) gewonnen worden und wird auf das Netz übertragen, in dem sie gesucht wird (Zielnetz).
- “andere Last“ - Die Kenntnis über die Werte der Verkehrsgütemetriken ist in einem identischen Netz mit einer anderen Last gewonnen worden (Ausgangsnetz) und wird auf das Netz übertragen, in dem sie gesucht wird (Zielnetz). Dieser Fall wird zunächst darauf eingeschränkt, dass der Lasttyp gleichbleibt und sich lediglich der Durchsatz oder das Mischungsverhältnis verschiedener Lasttypen ändert, wird dann aber auch auf Kombinationen anderer Lasttypen erweitert.
- “andere QoS Technologie“ - Die Kenntnis über die Werte der Verkehrsgütemetriken ist in einem identischen Netz mit einer anderen QoS Technologie (siehe Definition und Kapitel 2) gewonnen worden (Ausgangsnetz) und wird auf das Netz übertragen, in dem sie gesucht wird (Zielnetz).

Beim Schluss aus der ähnlichen Situation auf das Zielnetz können daher grundsätzlich 8 Fälle vorliegen (siehe Tabelle 1.1), die sich dadurch unterscheiden, bezüglich welcher Freiheitsgrade die Situation, aus

	anderes Netz	andere Last	andere QT
1	nein	nein	nein
2	nein	nein	ja
3	nein	ja	nein
4	nein	ja	ja
5	ja	nein	nein
6	ja	nein	ja
7	ja	ja	nein
8	ja	ja	ja

Tabelle 1.1: Variationsmöglichkeiten

der die Kenntnis über die Werte der Verkehrsgütemetriken stammt, anders ist.

Im Einzelnen haben die unterschiedlichen Fälle folgende Relevanz:

- Fall 1 ist ohne Interesse.
- Fall 2 bis 4 beziehen sich auf Änderungen der Last oder der QoS Technologie. Diese Fälle sind wichtig, um Informationen über die Auswirkungen der Lastveränderungen und der Vor- und Nachteile der unterschiedlichen QoS Technologien in einem gleichbleibenden Netz (bezieht sich auf Topologie, Routing, Komponenten) zu erlangen. Die Kenntnis wird aus dem Zielnetz selbst oder einem zugehörigen, das Netz nachbildenden Simulator gewonnen.
- Fall 5 bis 8 untersuchen Änderungen des Netzes auch mit zusätzlicher Veränderung (bei Fall 7, 8) der Last und/oder (bei Fall 6, 8) der QoS Technologie. Diese Fälle spielen die Hauptrolle innerhalb der Extrapolation auf das G-WiN.

Die Extrapolation von dem Ausgangsnetz auf das Zielnetz wird sich somit auf die Fälle 2 bis 8 beschränken. Diese wird entweder durch mathematische Analyse oder durch Simulation durchgeführt. Die allgemeine Vorgehensweise bei beiden Verfahren ist wie folgt:

- Mathematische Analyse:
  - Bestimmung eines zur Untersuchung ausgewählten Übertragungsweges und aller Ströme, die über den gesamten oder ein Teilstück dieses Pfades innerhalb des Zielnetzes übertragen werden.
  - Bestimmung der **Knoten-** und **Linkauslastung**.
  - Bestimmung von Delay und Jitter pro Paket für jeden Strom pro Knoten und Link basierend auf der QoS Technologie des Zielnetzes durch Extrapolation der Messungen aus dem Ausgangsnetz.
- Simulation:
  - Modellierung des Ausgangsnetzes in einem Simulator und Definition einer Menge von QoS Technologien und Lasttypen.
  - Kalibrierung und Validierung des Modells durch Messungen im Ausgangsnetz.
  - Aufbau eines Zielmodells im Simulator und Implementation der unterschiedlichen Fälle.
  - Simulation des Zielmodells.

Die Lösung dieser Aufgaben und die Entwicklung und der Einsatz der dazu erforderlichen Technologien und Komponenten wird wie folgt in den aufgeführten Kapiteln der Arbeit beschrieben:

Kapitel 2 beschreibt die für die Untersuchungen eingesetzten QoS Technologien, Kapitel 3 die Einteilung der QoS-sensitiven Anwendungen nach den QoS Metriken, Kapitel 4 die Verkehrsgenerierung, Kapitel 5 den Szenario-Generator, Kapitel 6 den verwendeten Netzwerk-Simulator und Kapitel 7 das Testbett. In Kapitel 8 wird ein Extrapolations-Szenario definiert und die Extrapolationsdurchführung und -verfahren allgemein beschrieben. Kapitel 9 stellt den analytischen Ansatz zur mathematischen Analyse und Extrapolation vor. In Kapitel 10 wird die Extrapolation durch die entwickelten Extrapolationsverfahren von einem spezifizierten Extrapolations-TestszENARIO auf Zielszenarien, die den 7 zu untersuchenden Extrapolationsfällen entsprechen, durchgeführt und die einzelnen Extrapolationsverfahren hinsichtlich des Extrapolations-Testszenarios bewertet, diese Bewertung auf beliebige Extrapolationsszenarien erweitert und eine allgemeine Anleitung zur Durchführung von Extrapolationen angegeben. Kapitel 11 beschreibt die Durchführung einer Extrapolation unter Verwendung der Extrapolationsverfahren anhand dieser Anleitung auf das G-WiN. Zum Abschluß folgt eine Zusammenfassung der Ergebnisse in Kapitel 12 und die Diskussion noch offener Probleme und Weiterentwicklungsmöglichkeiten der untersuchten Ansätze in Kapitel 13. Den Anhang bilden Definitionen und Bezeichnungen, Informationen zur G-WiN Kernnetztopologie und aus den Kapiteln aus Übersichtlichkeitsgründen ausgelagerte Plots von Simulationsergebnissen und Ergebnistabellen.

## Kapitel 2

# QoS Technologien

In diesem Kapitel werden die in dieser Arbeit zu untersuchenden QoS Technologien erläutert. Es wird jedoch keine umfassende Beschreibung sämtlicher existierender QoS technologischen Ansätze gegeben. Eine Bestandsaufnahme der zu diesem Zeitpunkt relevanten QoS Technologien und eine Begründung für die getroffene Auswahl der zu untersuchenden QoS Technologien ist [SCHMITT ET AL. 02] zu entnehmen. Die Ausführlichkeit der Beschreibung gewisser Teilbereiche der betrachteten Technologien begründet sich in dem erforderlichen Verständnis dieser in Verbindung mit den analytischen Ansätzen von Kapitel 9.

### 2.1 Best Effort (BE)

Der Best Effort Dienst ist keine QoS Technologie, sondern der Standarddienst des Internets, der den Ausgangspunkt für die im Weiteren beschriebenen QoS Technologien bildet. Er verwendet keinerlei Reservierung, sondern überträgt die Pakete mit einer First Come First Serve (FCFS) Strategie (in der Eintreffreihenfolge) in allen Knoten ohne dabei eine quantitative Zusicherung über eine Dienstgüte zu liefern.

### 2.2 Overprovisioning

Unter Overprovisioning versteht man die Erlangung einer geforderten Dienstgüte mit Hilfe des Best Effort Dienstes in Verbindung mit einer Unterauslastung des Übertragungs**kanals**. Dabei wird der **Grenzdurchsatz** so dimensioniert, daß die Auslastung des Übertragungskanals so klein ist, daß die gewünschte Dienstgüte erreicht wird. Die akzeptable Auslastung hängt dabei sensibel von der Lastcharakteristik und der Anzahl der gleichzeitigen Verkehrsströme ab.

Der Vorteil dieser Technologie liegt in ihrer Einfachheit: es findet keine Einführung neuer Techniken innerhalb des Netzes statt. Jedoch entstehen zum einen durch den Ausbau des Netzes Mehrkosten und zum anderen existiert keine Bevorzugung kritischer Anwendungen und keine Garantie für Zeit- und Verlustverhalten bei unvorhersehbarer Last [JESSEN 01].

### 2.3 Integrated Services (IntServ)

In den frühen 90er Jahren entstand die Idee der Entwicklung einer Technologie namens Integrated Services, die es ermöglichen sollte, eine große Vielfalt von Diensten auf einem Medium unter garantierter Dienstgüte zu integrieren, die sich vom einfachen Datentransfer bis hin zu Realzeit-Multimediaanwendungen erstrecken. Dabei sollten Ende-zu-Ende Dienste auf der Basis von Anwendungsströmen zur Verfügung gestellt werden. Unter einem Strom versteht man hierbei eine identifizierbare Sequenz zusammengehörender Datagramme einer einzelnen Benutzeraktivität, deren Datagramme

dieselbe Dienstgüte erfordern. IntServ basiert auf der Reservierung von Ressourcen. Dies bedeutet, daß zunächst auf Grund der von einer Applikation geforderten Dienstgüte und des Bedienanspruchs ein Pfad gewählt und Ressourcen reserviert werden, bevor eine Datenübertragung stattfindet. Aus diesen Überlegungen heraus wurde es ermöglicht, die traditionelle Best Effort Architektur des Internets durch die Integrated Service Architektur zu erweitern.

Die IntServ Architektur besteht aus zwei Elementen [BRADEN ET AL. 94]:

Erstens, dem Integrated Service (IS) model, einem Dienstmodell, das das äußerlich sichtbare Verhalten der Dienste definiert, und zweitens, einem Framework, der die Grundlagen und die Hilfsmittel zur Realisierung des IS Models zur Verfügung stellt.

### 2.3.1 IS Model

Momentan beinhaltet das IS Model zwei Dienstarten, den Guaranteed Service und den Controlled Load Service.

#### 2.3.1.1 Guaranteed Service

Guaranteed Service [SHENKER ET AL. 97] wird von "intoleranten" (delay-sensitiven) Anwendungen verwendet. Er stellt einen Dienst mit einer festgelegten oberen Grenze für einen Ende-zu-Ende Delay zur Verfügung, bei Einhaltung eines maximalen Durchsatzes seitens der Last. Dies beinhaltet, daß Pakete innerhalb einer vorbestimmten Zeit ihr Ziel erreichen und nicht u.a. wegen eines Pufferüberlaufes verworfen werden. Diese Garantien können jedoch nur gegeben werden, wenn jeder Knoten innerhalb des Netzwerkes auf dem Weg von der Quelle zur Senke diesen Dienst unterstützt.

#### 2.3.1.2 Controlled Load Service

Controlled Load Service [WROCLAWSKI 97] kann nur bei "toleranten" (delay-unempfindlichen) Anwendungen verwendet werden. Er stellt eine Dienstgüte mit einer Qualität vergleichbar zu derer des Best Effort Dienstes in unbelasteten (keine Last im Wettbewerb) Netzen zur Verfügung. Bei einer auftretenden Überlast (entspricht je nach Konfiguration einem bestimmten oder mehreren Dienstgütemängeln, wie z.B. auftretendem **Loss**) wird jedoch im Gegensatz zu dem Best Effort Dienst versucht, im Netzwerk diese Qualität durch Zugangssteuerung (admission control) beizubehalten.

### 2.3.2 Framework

Bei der Verwendung von IntServ muß ein Router jedem Flow eine angemessene Dienstgüte im Bezug auf das Dienstmodell zur Verfügung stellen. Um dies zu gewährleisten, wird ein Framework bestehend aus folgenden Komponenten implementiert: Reservierungsprotokoll (reservation setup protocol), Zugangssteuerung (admission control), **Policy** Steuerung (policy control), Paket Scheduler (packet scheduler), Paket Klassifizierer (packet **classifier**).

#### 2.3.2.1 Reservierungsprotokoll (Reservation Setup Protocol)

Als Reservierungsprotokoll [WROCLAWSKI 97] wird RSVP (ReSerVation Protocol) verwendet, das für ein Internet mit integriertem IntServ entworfen wurde. Es handelt sich dabei um ein Netzwerk- und

Signalisierungs-, jedoch um kein Routingprotokoll. Es wurde aber so konzipiert, daß es mit einer Vielzahl von Routing-Protokollen verwendet werden kann. RSVP ist verantwortlich für den Verbindungsaufbau und die Verbindungssteuerung. Die Reservierung ist hierbei empfangerbasiert. Best Effort Knoten werden akzeptiert (Netzwerkheterogenität), gefährden aber die gegebene Dienstgütegarantie. RSVP ermöglicht sowohl uni- als auch multicast Übertragungen auf einem unidirektionalen Weg. RSVP kann automatisch erforderliche Routenveränderungen durch das periodische Versenden der Reservierungsnachrichten erreichen. Bleiben diese eine vordefinierte Zeitspanne aus, erfolgt ein Timeout, und die Reservierung wird gelöscht. RSVP arbeitet daher mit einem "soft state". Durch den Einsatz von RSVP haben Endgeräte (hosts) die Möglichkeit, eine bestimmte Dienstgüte vom Netzwerk anzufordern. RSVP propagiert diese QoS Anforderung an alle Router entlang der gewählten Pfade und aktualisiert zusätzlich Zustandsinformationen innerhalb der Router und Endgeräte, um den angeforderten Dienst zur Verfügung stellen zu können.

Die Reservierung eines Pfades läuft im Einzelnen folgendermaßen ab:

In RSVP wird ein Datenstrom als einfacher Verbreitungsbaum modelliert, nach dem die Daten von der Wurzel zu sämtlichen Empfängern geroutet werden. Dabei werden alle RSVP Nachrichten wie gewöhnliche Pakete im Internet als IP Pakete versendet (es wird jedoch empfohlen den Signalisierungsverkehr über einen entsprechend gewichteten, reservierten Kanal zu priorisieren). Zunächst sendet die Anwendungsquelle eine sogenannte PATH-Message, die ein Sender Template, eine Traffic Specification (TSpec) und eine Advertising Specification (AdSpec) enthält, zur Zielortmenge, die entweder vom Typ uni- oder multicast sein kann. Das Sender Template beschreibt das Format der Datenpakete, die ein Sender schicken will. Die TSpec definiert die Verkehrscharakteristik des Flows, den ein Sender schicken will. Sie wird den Zwischen- wie auch dem Empfängerknoten mitgeteilt und von diesen nicht verändert. Die AdSpec enthält allgemeine Parameter, wie RSVP-Fähigkeit des Knotens etc., und dienstspezifische Parameter, die auf ihrem Weg vom Sender zum Empfänger an jedem Knoten aktualisiert werden. Die AdSpec sammelt von jedem Knoten des Pfades für die Reservierung wichtige Informationen. Die PATH-Message erfüllt somit folgende drei Aufgaben: Erstens, die Verbreitung der Traffic Specifications von den Sendern zu den Empfängern, zweitens die Speicherung von Pfadzustandsinformationen in den Knoten vom Sender zu dem Empfänger (diese enthalten zum Beispiel die IP Adresse des Vorgängerknotens, die wichtig für die Wegfindung der Reservierungsnachricht ist), drittens, die Zustandsinformation sämtlicher Knoten entlang des Übertragungspfades. Der Inhalt einer PATH-Message wird im Empfängerknoten über eine RSVP Schnittstelle an die Anwendung weitergegeben. Diese interpretiert die eingehenden Daten und benutzt diese um geeignete Reservierungsparameter zu wählen. Diese beinhalten den gewünschten QoS Service (Guaranteed oder Controlled Load), die Traffic Specification, die hier den Verkehr beschreibt für den eine Reservierung durchgeführt werden soll, und die Reserve Specification (RSpec), die den gewünschten Service beschreibt. Diese Informationen werden in einem Flowspec Object in eine sogenannte RESV-Message integriert. Diese nimmt die identische Route in entgegengesetzter Richtung zurück zum Sender. In jedem Knoten, den die RESV-Message passiert, wird eine Reservierung auf Grund der in der RESV-Message gespeicherten Information durchgeführt und das Flowspec Objekt mit den Reservierungsdaten aktualisiert. Dieses Objekt wird dann im Senderknoten wieder der Anwendung bekannt gemacht. Diese wird somit über die durchgeführte Reservierung unterrichtet. Anschließend können Daten über den reservierten Pfad versendet werden.

### 2.3.2.2 Weitere Framework Komponenten

Die weiteren Komponenten des Frameworks haben folgende Aufgaben:

Der Packet Classifier bestimmt anhand des Vergleichs von benutzerspezifischen Paketheaderinformationen mit benutzerspezifischen Dienstgütevereinbarungen die QoS **Klasse** für jedes einzelne Paket. Der Packet Scheduler gibt die Pakete nach einem die geforderte Dienstgüte realisierenden Algorithmus an die versendende Schnittstelle weiter. Die Admission Control kontrolliert in jedem Knoten, ob

der Sender seinen vereinbarten Durchsatz einhält und somit die bereitgestellten Ressourcen nicht überschreitet. Die Policy Control hingegen kontrolliert, ob der Benutzer die administrative Zulassung für eine solche Reservierung besitzt.

Eine genauere Betrachtung der Funktionsweisen dieser Komponenten wird in Kapitel 9 vorgenommen, um die für die Entwicklung eines analytischen Ansatzes zur Bestimmung von Dienstgütemetriken notwendigen Formeln herleiten zu können.

## 2.4 Differentiated Services (DiffServ)

Da Befürchtungen bestanden, daß der Einsatz von Reservierungsprotokollen, wie bei RSVP/IntServ, sowohl auf Grund eines hohen Signalisierungsaufwandes durch die PATH und RESV Nachrichten, aber auch durch die reservierungsbedingten Zusatzbelastungen der Router für eine Vielzahl von Strömen nicht angemessen skaliert und daß dadurch die derzeitigen Router überfordert wären, wurde Ende der 90er Jahre das Konzept der Differentiated Services (DiffServ) entwickelt.

Bei DiffServ wird im Gegensatz zu IntServ keine Reservierung von Ressourcen, sondern eine Klassifizierung, Priorisierung und/oder Quotenfestlegung von aggregiertem Verkehr durchgeführt. Die DiffServ Architektur wird zum Großteil durch die Implementation folgender Komponenten in den **Edge-Routern** realisiert: Packet Classifier, **Meter**, **Marker**, **Shaper**, Policer bzw. **Dropper** und eine geringe Anzahl von **Per-Hop Behaviours** (PHB).

Der an den Edge-Routern in die DiffServ **Domäne** eingehende Verkehr und dessen geforderte Dienstgüte werden auf Grund von Paketheaderinformationen vom Classifier identifiziert. Er gibt diese Informationen an den Meter weiter, der den Verkehr mit den in einem Service Level Agreement (SLA) vereinbarten und mit den in einem Traffic Conditioning Agreement (TCA) festgeschriebenen Daten vergleicht. SLAs beschreiben Vereinbarungen über den Charakter des Datenstroms und über die Dienstgüte, die diesem zugesichert wird. Sie können sowohl zwischen Endsystem und Anbieter des Dienstes als auch zwischen zwei Dienstanbietern existieren. In einer TCA sind die Regeln für die Anwendung von Classifier, Meter, Marker, Shaper und Dropper auf den vom Classifier ausgewählten Strom und die Dienstanforderungen enthalten. Entspricht der vom Meter gemessene Verkehr dem in der TCA vorgegebenen Profil, wird er als in-profile ansonsten als out-of-profile gekennzeichnet. Ist der Verkehr in-profile wird er vom Marker mit einem DS (Differentiated Services) Codepoint (wird auch als DSCP bezeichnet) entsprechend der in den SLA ausgehandelten Klasse markiert. Ist er out-of-profile kann er je nach Vereinbarung entweder vom Marker mit einem anderen (vorher vereinbarten) DS Codepoint versehen werden (re-marking), vom Shaper „zurecht geschnitten“ werden oder vom Policer bzw. Dropper verworfen werden. Das Shaping und Policing wird jeweils durch Token Buckets realisiert. Das Markieren mit einem DS Codepoint findet in den Edge-Routern auf der IP Ebene im Type of Service Feld (TOS Feld, im IPv4 Protokoll) des IP Headers statt. Hierbei gibt die Markierung Auskunft über das PHB, das für die Art von Service steht, die für das entsprechende Paket gewählt wurde, und somit, wie dieses Paket weitergegeben werden soll. Auf Grund dieser Markierung der Pakete kann die Komplexität in den Core-Routern sehr niedrig gehalten werden, indem lediglich Tabellen vorhanden sein müssen, die die Markierungen auf entsprechende PHBs abbilden (Klassifizieren) und somit den Core-Routern mitteilen, wie sie den mit Hilfe der Edge-Router in die DiffServ Domäne eingebrachten Verkehr weiterleiten sollen. Unter einem PHB versteht man allgemein eine Beschreibung des nach außen hin sichtbaren Forwarding Behaviour. Im speziellen Fall von DiffServ entspricht dies eines Forwarding Behaviour eines DiffServ unterstützenden Knotens, angewandt auf ein DiffServ Aggregat (Pakete mit dem selben Codepoint). Dies bedeutet, daß je nach DS Codepoint ein PHB gewählt wird, das eine bestimmte Art von Bedienung realisiert. Die so realisierbare Serviceunterscheidung ermöglicht die Erfüllung unterschiedlicher Anwendungsanforderungen, Anwendererwartungen und ermöglicht

eine dienstspezifische Tarifierung. Ein Service definiert signifikante Charakteristiken für unidirektionale Paketübertragungen entlang eines oder mehrerer Pfade innerhalb eines Netzwerks. Dabei handelt es sich um quantitative Angaben von Durchsatz, Delay, Jitter, und/oder Loss oder spezifizierte Angaben über relative Prioritäten für den Netzwerkressourcenzugriff [BLAKE ET AL. 98]. Heute sind drei PHB von der IETF spezifiziert: das Expedited Forwarding (EF) PHB, das Assured Forwarding (AF) PHB und das Default PHB.

### 2.4.1 Expedited Forwarding PHB

Dieses PHB kann dazu verwendet werden, um einen Ende-zu-Ende Service innerhalb einer DiffServ Domäne mit geringem Loss, geringem Delay und geringem Jitter zur Verfügung zu stellen. Dieser Service wird mehrfach auch als Premium Service bezeichnet. Um einen solchen Service zu realisieren, dürfen nur geringe Verzögerungen durch Warteschlangenaufenthalte auftreten, was bedeutet, daß die Ankunftsrate (Anzahl der eingehenden Pakete pro Zeiteinheit) des Verkehrs eines DiffServ Aggregates unterhalb der minimalen Ausgangsrate (Anzahl der abgehenden Pakete pro Zeiteinheit) dieses Aggregates an jedem DiffServ Knoten innerhalb einer DiffServ Domäne liegen sollte. Der DS Codepoint des EF PHB ist 101110 (= 46 dezimal). Mechanismen zur Realisierung dieses PHB sind folgende Scheduling Verfahren: Priority Queueing (PQ), Weighted Round Robin (WRR), Class-Based Queueing (CBQ) [DAVIE ET AL. 02].

### 2.4.2 Assured Forwarding PHB

Bei dem AF PHB handelt es sich um eine PHB Gruppe bestehend aus 4 unabhängigen AF Klassen, die in jedem Knoten der DS Domäne bestimmte Forwarding Ressourcen (Puffer, **Bandbreite**) zugeweiht bekommen. In jeder Klasse kann nochmals zwischen 3 Drop Precedences unterschieden werden, die darüber entscheiden, welche Pakete bei einer Überlastsituation als erste verworfen werden. Je höher die Drop Precedence, umso eher wird das Paket verworfen. Die AF PHB Gruppe dient dazu, verschiedene Garantieniveaus für die Weiterleitung zur Verfügung zu stellen. Das Niveau hängt hierbei von den allokierten Bandbreiten, der momentanen Auslastung der Klasse und der klassenlokalen Drop Precedence ab. Ein Paket, das zu der AF Klasse  $i$  gehört und die Drop Precedence  $j$  hat, besitzt einen DS Codepoint AF $ij$ , mit  $1 \leq i \leq N$ ,  $N = 4$  und  $1 \leq j \leq M$ ,  $M=3$ . Im Speziellen sind den einzelnen Klassen und Drop Precedence Kombinationen folgende DS Codepoints zugeweiht (siehe Tabelle 2.1) [HEINANEN ET AL. 99]:

Drop Precedence	AF Klasse 1	AF Klasse 2	AF Klasse 3	AF Klasse 4
Niedrige Drop Precedence	001010 (10)	010010 (18)	011010 (26)	100010 (34)
Mittlere Drop Precedence	001100 (12)	010100 (20)	011100 (28)	100100 (36)
Hohe Drop Precedence	001110 (14)	010110 (22)	011110 (30)	100110 (38)

Tabelle 2.1: AF DS Codepoints - binär (dezimal)

### 2.4.3 Default PHB

Das Default PHB entspricht dem traditionellen Best Effort Forwarding. Es hat als DS Codepoint den Wert 000000 (= 0 dezimal) [HEINANEN ET AL. 99].

## 2.4.4 Unterschiedliche DiffServ Ansätze

Durch die allgemeine Definition von PHBs in [BLAKE ET AL. 98] ist es möglich, selbst PHBs zu entwickeln oder aber Ansätze zu präsentieren, die aus Kombinationen der bereits vorgestellten PHBs bestehen. Die beiden Ansätze, die innerhalb dieser Arbeit untersucht werden, sind zum einen der Default Ansatz und zum anderen der Olympic Ansatz.

### 2.4.4.1 Default Ansatz

Unter dem Default Ansatz versteht man die Kombination der von der IETF spezifizierten PHBs. Dies bedeutet, man verwendet die EF, AF und die BE (Best Effort) Klassen. In dieser Arbeit wird folgende Ausprägung dieses Ansatzes verwendet:

Es wird das EF PHB für den hochprioreren Verkehr verwendet. Zusätzlich existieren drei AF Klassen, die sich durch ihre Zugangsbestimmungen in den Policies voneinander unterscheiden. Die AF1 Klasse wird nicht überbucht, die AF2 Klasse wird bis zu maximal 100 % überbucht und die AF3 Klasse weist keine Pakete ab und entspricht in der Behandlung der Pakete der BE PHB, aber wegen der schwankenden **Kapazität** nicht derselben, sondern ungünstigeren **Wartezeit**- und Verlustverhältnissen, als bei einer BE Bedienung gleicher (mittlerer) Auslastung. Alle drei AF Klassen verwenden die niedrige Drop Precedence.

### 2.4.4.2 Olympic Ansatz

Der Olympic Ansatz benutzt ebenfalls das EF PHB, das Default PHB (BE) und das neudefinierte QBone Scavenger Service (QBSS) PHB. Dieses PHB stellt einen Service zur Verfügung der nur dann die Pakete, die ihn benutzen, überträgt, wenn Ressourcen frei zur Verfügung stehen. Auf die zur Realisierung beider DiffServ Ansätze verwendeten Bedienstrategien wird in Kapitel 9 noch detailliert eingegangen.

### 2.4.4.3 Admission Control (AC)

Allgemein prüft eine DiffServ AC die Lastsituation in einer DiffServ Domäne und entscheidet anhand dieser, ob ein neuer an einem Edge-Knoten der DiffServ Domäne ankommender Flow zugelassen und weiter übertragen wird oder nicht. Bei DiffServ kann man drei Arten von AC unterscheiden:

- keine AC
- lokale, dezentrale AC
- globale, zentrale AC

Wenn keine AC verwendet wird, werden alle SLA-konformen Pakete für die jeweilige Klasse akzeptiert, was folglich bei einem ansteigenden Verkehrsaufkommen der einzelnen in den Klassen aggregierten Verkehrsströmen zu einem Warteschlangenüberlauf und somit zu Loss führt. Außerdem können bei Warteschlangen mit hoher Kapazität für die hochprioreren Klassen die niederprioreren Klassen „ausgehungert“ werden.

Die lokale, dezentrale AC arbeitet in den Edge-Knoten der DiffServ Domäne und überprüft lediglich, ob der Edge-Knoten bzw. seine Ausgangsleitungen die Kapazität haben den ankommenden Flow weiterzusenden. Für die ausgewählten DiffServ Ansätze können folgende Ausprägungen der lokalen AC verwendet werden:

Für den Default Ansatz wird eine AC implementiert, die für die EF, die AF1 und AF2 Klasse einen über Parameter einstellbaren Anteil der gesamten Bandbreite eines Ausgangslinks eines DiffServ Edge-Knotens in die DiffServ Domäne zur Verfügung stellt. Jede Bandbreitenanforderung einer einzelnen Anwendung für die von dieser Anwendung gewählten Klasse wird überprüft. Ist diese Anforderung in dem Bandbreitenanteil, der für diese Klasse zur Verfügung gestellt wird, noch realisierbar, wird die Anwendungsanforderung akzeptiert, andernfalls für diese Klasse abgewiesen. Eine Abweisung für die EF Klasse bedeutet, daß die Anforderung verworfen wird, da eine Herabstufung der Dienstgüte die Pakete einer EF Anwendung zu sehr verzögern könnte und ein bestimmter Anteil des hochpriorien Verkehrs am Gesamtverkehr durch die Admission Control eingehalten und nicht überschritten werden soll. Für die AF1 und AF2 Klasse wird die Anwendungsanforderung an die nächst „schlechtere“ Klasse weitergereicht (AF1 an AF2, AF2 an AF3). Für die AF3 Klasse werden alle Anforderungen akzeptiert.

Für den Olympic Ansatz wird eine AC implementiert, die für die EF und die BE Klasse einen über Parameter einstellbaren Anteil der Gesamtbandbreite (siehe Default Ansatz) zur Verfügung stellt. Im Gegensatz zum Default Ansatz werden sowohl für die EF, als auch für die BE Klasse nicht realisierbare Anwendungsanforderungen verworfen. Nur für die QBSS Klasse werden alle Anforderungen akzeptiert, da für diese Klasse, im Gegensatz zur EF Klasse, nicht versucht wird Dienstgüteparameter einzuhalten und die Pakete dieser Klasse lediglich versandt werden, wenn keine Pakete der anderen beiden Klassen vorhanden sind (siehe Kapitel 9).

Die lokale AC kann allerdings nur Überlastungen der Edge-Knoten bzw. deren Ausgangslinks für jede Klasse (wenn gewollt) verhindern. Überlagerungen von Strömen innerhalb der Domäne, dadurch höhere Auslastungen einzelner Links und daraus resultierende Dienstgütemängel können dadurch nicht verhindert werden.

Hierzu ist eine globale, zentrale AC, auch Bandwidth Broker (BB) genannt, notwendig. Der BB prüft nicht nur die Auslastungssituation der Edge-Knoten und deren Ausgangsleitungen, sondern die Lastsituation auf allen Links des Übertragungspfades des neuankommenden Flows innerhalb der DiffServ Domäne. Für die Entscheidung, ob ein Flow zur Übertragung zugelassen wird oder nicht, ist das Wissen über die aktuelle Lastsituation der Domäne für alle DiffServ Klassen und die Routen der Flows Voraussetzung. Ebenfalls eignet sich der BB zu einer unter mehreren BBs verschiedener, verbundener DiffServ Domänen abgestimmten Admission Control. Bei beiden Varianten werden jedoch wieder mehr Aufgaben an die Core-Knoten (und natürlich auch Edge-Knoten) übergeben und es entsteht erheblicher Signalisierungsverkehr. Dieser Ansatz bewegt sich näher in Richtung IntServ, was in dieser Arbeit nicht gewünscht wird, weshalb er nicht eingesetzt wird. Im Vergleich zu IntServ müssen allerdings auch bei der zentralen AC nicht die Zustände aller einzelnen Flows verwaltet werden, sondern nur die der einzelnen Klassen. Außerdem werden diese von einer zentralen Stelle verwaltet und nicht, wie bei IntServ, von jedem einzelnen Knoten. Diese zentrale Verwaltungsstelle kann entsprechend der geforderten Kapazitäten dimensioniert werden.

## Kapitel 3

# Anwendungen im Hinblick auf Dienstgüte

Der Delay von Daten auf ihrem Weg vom Sender zum Empfänger besteht aus einem konstanten, vorhersehbaren und einem unbeständigen, unvorhersehbaren Zeitanteil.

Der konstante Anteil setzt sich zum einen aus dem Zeitaufwand für das Erfassen, Digitalisieren und Komprimieren der Daten, etwa bei der Erzeugung eines Videorahmens oder eines Samples von Sprachdaten, zusammen und zum anderen aus der konstanten Verzögerung, die durch das Paketieren und den Transport der Daten vom Sender zum Empfänger über das Netzwerk entsteht. Die reine Transportzeit über das Netzwerk beinhaltet ebenfalls einen konstanten Delay, den sogenannten Propagation- oder Ausbreitungsdelay. Dieser Delay ist definiert als die Zeit, die ein Signal bei der Übertragung zwischen zwei Netzwerkknoten auf Grund der Leitungsbeschaffenheit benötigt. Auf der Empfängerseite besteht der konstante Delay dann noch aus der Fehlerüberprüfung, dem Dekomprimieren und dem Abspielen einer z.B.: Videokonferenz.

Ein unvorhersehbarer Teil des Delays entsteht auf Grund von unterschiedlicher Nutzung und Auslastung der Netzwerkelemente innerhalb der Datenübertragung, verursacht durch die Verwendung des Standardinternetdienstes (Best Effort) bzw. durch die innerhalb diesen Standards verwendeten Bedienelemente und -strategien, die eine schwankende Dienstgüte zur Folge haben. Die meßbaren Metriken, um Dienstgüte zu beschreiben, sind:

- Bandbreite
- Durchsatz
- Jitter
- Delay
- Loss

Die Definition dieser Metriken ist Anhang A zu entnehmen. Die Einteilung von Anwendungen erfolgt allerdings meist durch drei dieser Metriken: Durchsatz, Delay und Jitter.

Nach einem, vielen Arbeiten in diesem Forschungsbereich als Grundlage dienenden, Draft der ITU-T (International Telecommunication Union, Telecommunication Standardization Sector) gibt es folgende Grenzwerte für diese drei Dienstgütemetriken, die je nach Anwendungsart bzw. -klasse nicht überschritten werden sollten (siehe Tabelle 3.1). Die ITU bezeichnet Jitter hierbei als Delay Variation, deren Definition in Appendix II von [ITU 00] von der in dieser Arbeit gewählten Jitterdefinition insofern abweicht, daß dort nicht das absolute Delaymaximum eines Intervalls genommen wird und daß das betrachtete Intervall eine Dauer von 1 Minute besitzt. Da diese Intervalldauer für die Berücksichtigung von kurzzeitigen Delayschwankungen von Realzeitanwendungen zu groß ist, wird diese (siehe Kapitel 2) auf einen Wert von 3 Sekunden verkürzt, dafür aber der minimale vom maximalen Delaywert subtrahiert. Der Delay entspricht dem in dieser Arbeit definierten Ende-zu-Ende Delay zuzüglich der Datenverarbeitung bzw. Aufbereitung an beiden Endpunkten der Übertragung, als deren Dauer ein Mittelwert von 50 ms angenommen wird. Die in Tabelle 3.1 aufgezeigten Grenzwerte gelten für die

meisten, wie auch für die im Anschluß untersuchten, Netzwerke. Es gibt jedoch einige Ausnahmen, die [ITU 00] entnommen werden können.

Dienstgütemetrik	statistische Beschreibung	Realzeitanwendung
Delay	Mittelwert des Delays	150 ms
Jitter	Differenz des $1 - 10^{-3}$ Quantils des Delays und dem minimalen Delay	50 ms
Loss	Paketverlustwahrscheinlichkeit für einen Flow	$10^{-3}$

**Tabelle 3.1:** Obere Grenzwerte der Dienstgütemetriken

Die Nichteinhaltung dieser Grenzwerte hat folgende Auswirkungen auf die zu untersuchenden Anwendungen, Videokonferenz und VoIP. Der Auswahl der Anwendungen liegen Überlegungen zugrunde, die Kapitel 4 zu entnehmen sind. Sie bilden, da es sich bei beiden um interaktive Anwendungen handelt, Extremfälle, deren Anforderungen keinesfalls für alle QoS-sensitiven Anwendungen gelten (siehe Kapitel 4).

### 3.1 Auswirkungen der Dienstgütemetriken auf QoS-sensitive Anwendungen

#### Bandbreite:

Wenn sich der Durchsatz der Bandbreite (dem Grenzdurchsatz) eines Übertragungsmediums annähert, verursachen entstehende Delays und Loss eine Verminderung der Dienstgüte. Im Allgemeinen führt eine geringere Bandbreite bei gleichbleibender Last zu verminderter Dienstgüte.

#### Durchsatz:

Der Durchsatz kann durch eine Bandbreitenaufteilung unter verschiedenen Anwendungen, soweit eine ausreichende Nachfrage besteht, stark vom Grenzdurchsatzwert abweichen. Durch ihn können momentane Leistungsfähigkeiten von Systemen oder Komponenten ausgedrückt werden.

#### Jitter:

Jitter ist nur für Realzeitanwendungen eine kritische Metrik (siehe Abschnitt 3.2). Eine zu hohe Differenz der Delaywerte der für die Jitterberechnung betrachteten Pakete, hat sichtbare Auswirkungen auf Videokonferenzen und führt zu einer „stotternden“ Ton-Wiedergabe.

#### Delay:

Typische Effekte von längeren Delays sind unnatürlich lange Pausen und eine schwierige Dialogsynchronisation innerhalb von Sprach- und Videokommunikationen. Beides führt zu einer schlechten und unnatürlichen Kommunikation. Die Auswirkungen einer Delaysteigerung sind in [HOLLECZEK ET AL.]

an Beispielen gut zu erkennen.

Loss:

Loss führt bei VoIP und Videokonferenzen zu Störungen, die eine “stotternde“ Wiedergabe zur Folge haben.

### 3.2 Einteilung der Anwendungen nach Metriken

Nach der Stärke des Einflusses, den eine Änderung der Werte dieser Metriken auf die Anwendungen hat, lassen sich diese in QoS Sensitivitätsklassen, wie in Tabelle 3.2 und Abbildung 3.1 gezeigt wird, unterteilen.

QoS Sensitivität	Anwendung
Durchsatz	- Video & Audio Download - Abgleich von Servern - Asynchrones TTL (kein simultanes Teleteaching / Telelearning)
Delay	- Anspruchsvolle WWW-Nutzung (World Wide Web)
Jitter	- Realzeit-Audio
Durchsatz, Delay	- Realzeitmessungen - GRID (Gitter von Rechnern und Datenbanken zur gemeinsamen Problemlösung)
Durchsatz, Jitter	- Synchrones TTL (simultanes Teleteaching / Telelearning) - TV - Video on Demand
Delay, Jitter	- Sprachkonferenzen - Fernsprechen (z.B.: VoIP (Voice over IP))
Durchsatz, Delay, Jitter	- Virtual Reality - Videokonferenzen - Synchrones und interaktives TTL

**Tabelle 3.2:** Einteilung von Anwendungen nach QoS Sensitivität

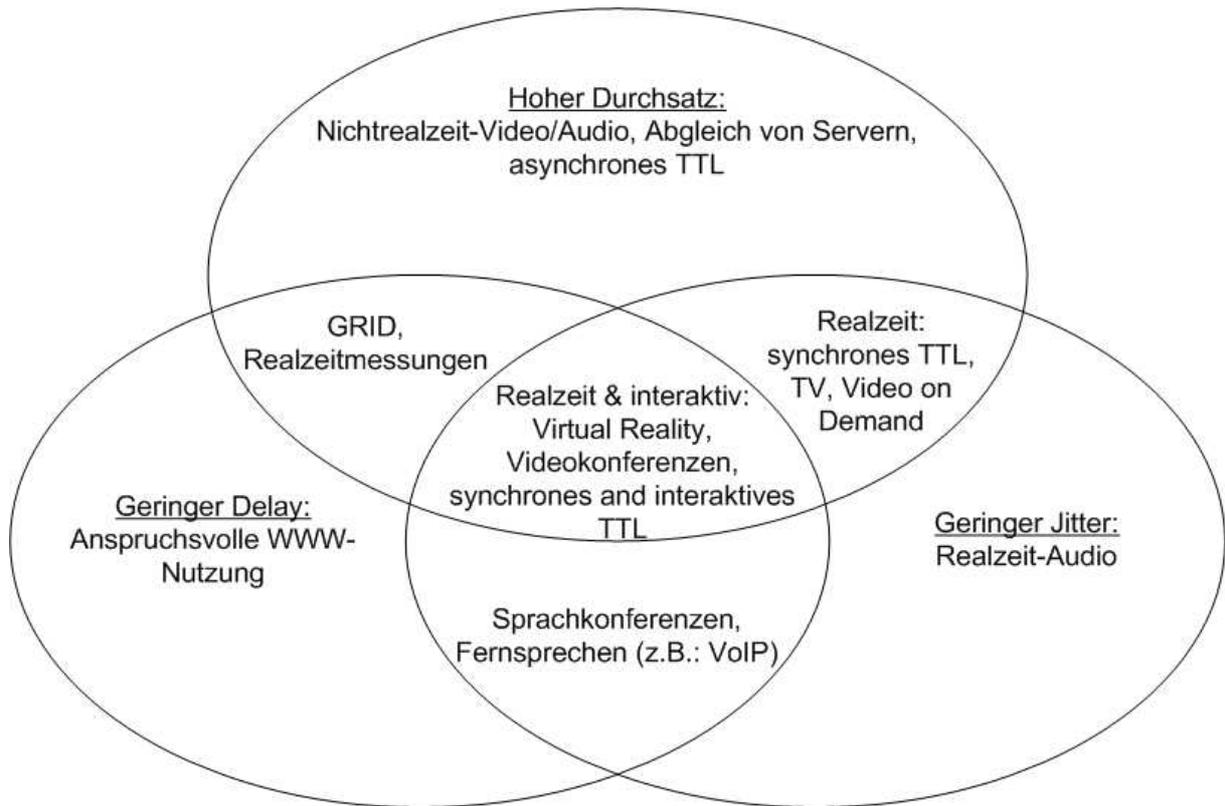


Abbildung 3.1: Einteilung von Anwendungen nach QoS Sensitivität

## Kapitel 4

# Charakteristik und Generierung von Verkehr

Die Einteilung der Anwendungen nach QoS Sensitivitätsklassen, wie in Kapitel 3 beschrieben, führt zu dem Schluß, daß die problematischsten Anwendungen in Bezug auf die Garantie der Dienstgüte der Klasse (Durchsatz, Delay, Jitter) angehören, da diese Anwendungen sensitiv bezüglich aller drei Metriken sind. Aus diesem Grund ist die Untersuchung dieser Anwendungen von größtem Interesse.

Für unsere weiteren Untersuchungen haben wir daher eine Anwendung aus dieser Klasse, die Videokonferenz, gewählt. Für diese Anwendung spricht außerdem die laufende Realisierung durch das DFN und andere Netzbetreiber. Als zweite Anwendung wird Voice over IP (VoIP) ausgewählt.

Im Anschluß folgt nun eine kurze Beschreibung dieser beiden Anwendungen und deren Verkehrscharakteristiken.

### 4.1 Verkehr QoS-sensitiver Anwendungen

#### Voice over IP (VoIP) Verkehr:

Voice over IP ist die allgemeine Bezeichnung für einen Bereich, der sich mit der Übertragung von interaktiven Sprachinformationen über das Internet Protokoll beschäftigt. Sprachdaten werden im Gegensatz zum traditionellen leitungsvermittelten Public Switched Telephone Network (PSTN, hier sind Vorläufer von Integrated Services Digital Network (ISDN) / Synchronous Digital Hierarchy (SDH) gemeint) in digitalisierter Form in diskreten Paketen übertragen. Sprachpakete werden in periodischen Zeitintervallen von sogenannten Codecs (Coder-decoder compression schemes) erzeugt. Kodierungstechniken für Telephonie und Sprache wurden von der ITU-T (International Telecommunication Union) in ihren Empfehlungen der G-Serie standardisiert. Beispiele für Codec-Standards sind ITU-T G.723.1, der für Videokonferenzen und Telephonie entwickelt wurde und für Codieren und Decodieren in Realzeit optimiert ist und ITU-T G.711 (siehe Videokonferenz Sprachcodecs). Bei diesen werden Daten mit Spitzenbitraten von 6.4 und 64 Kbit/s mit konstanten Paketlängen übertragen.

#### Videokonferenz Verkehr:

Unter einer Videokonferenz versteht man eine Konferenz zwischen zwei oder mehreren Personen, die sich an verschiedenen Orten befinden und zur Übertragung der Audio- und Videodaten ein Computernetzwerk verwenden. Die ITU hat einen internationalen Standard namens H.323 für Sprach- und Videoübertragungen über IP definiert. Es handelt sich bei H.323 um einen weltweiten Standard für Videokonferenzen in IP Netzwerken [CONSTABLE & PRICE 01, CISCO SYSTEMS 02]. H.323 soll es selbst Benutzern unterschiedlicher Videokonferenzsysteme ermöglichen eine Videokonferenz miteinander aufzubauen. Als Anwendungsbeispiel wählen wir Netmeeting von Microsoft aus, da dies die am meisten verwendeten Videokonferenzanwendung ist und auch beim DFN eingesetzt wird (Stand: 01.01.2003). Netmeeting verwendet einen proprietären Videocodec, bei dem es sich um eine typische Variable Bit Rate Quelle (VBR) ohne Ratenfestlegung handelt. Die Senderate ändert sich

bei Veränderungen der Bewegungs- und Lichtverhältnisse. Bei größeren Bildern tritt ein kleinerer Frameabstand auf. Bildinhalt und -qualität haben nur Einfluß auf die Framegröße. Als Parameter können Bildgröße und -qualität geändert werden. Für die Audioübertragung werden die beiden zuvor erwähnten Sprachcodecs G.723.1 und G.711 verwendet. Auf Grund der Bildsenderate hat Video eine höhere und sehr schwankende Paketsenderate mit einer weitaus höheren Maximum Transfer Unit (MTU). Abbildung 4.1 zeigt eine typische Paketgrößenverteilung eines Verkehrs einer Videokonferenz [CISCO SYSTEMS 02].

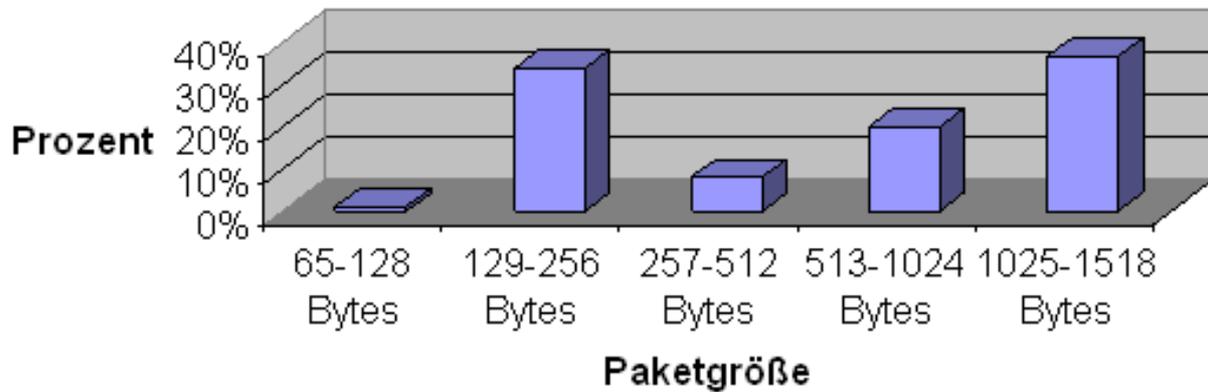


Abbildung 4.1: Video Paketgrößenverteilung

## 4.2 Verkehrsgenerierung

Die Verkehrsgenerierung der beiden zur Untersuchung herangezogenen und zuvor beschriebenen Anwendungen wird von dem in Kapitel 5 beschriebenen Szenariogenerator bzw. dessen integriertem Last-Generator übernommen und ist daher dessen Beschreibung zu entnehmen.

## Kapitel 5

# Szenario-Generator

Im Rahmen des LETS QoS Projektes wurde am Lehrstuhl KOM Multimedia Communications der TU Darmstadt ein Szenario-Generator entwickelt. Dieser ist eine Sammlung von Werkzeugen, die zwei Hauptbestandteile enthält, den Topologie- und den Last-Generator, deren Funktionsweise in den folgenden Abschnitten kurz erläutert wird und in ([SCHMITT ET AL. 03-1]) präzise nachvollzogen werden kann.

Der Szenario-Generator hat die Aufgabe, das Testbett und den Netzwerksimulator mit identischen Topologien und Verkehren zu speisen, um identische Voraussetzungen für durchzuführende Testbett- und Simulationsläufe zu schaffen. Er erzeugt aus den mit Verkehr versehenen Topologien zum einen Konfigurationsdateien für das Testbett und zum anderen Inputdateien für den NS2 Simulator. Dies soll die Validierung der Simulationsergebnisse durch die Testbettergebnisse sicherstellen.

### 5.1 Topologie-Generator

Der Topologie-Generator ermöglicht zum einen über eine grafische Benutzeroberfläche eine Topologie selbst "per Hand" zu generieren und zum anderen bereits mit Hilfe anderer Topologiegeneratoren erzeugte Topologien einzulesen, zu verändern und in dem eigens für diesen Topologiegenerator entwickelten Format, dem Topology File Format (TFF), abzuspeichern. Dieses Format ist eine auf die in dem LETS QoS Projekt anfallenden Anforderungen eingehende Erweiterung des Graph Modeling Language (GML) Formats. Der Topologie-Generator kann dabei die Formate der verbreitetsten Topologiegeneratoren (TIERS - Random Network Topology Generator, BRITE - Boston University Representative Internet Topology Generator, GT-ITM - Georgia Tech Internetwork Topology Models und Inet - AS Level Network Topology Generator) konvertieren. Dieser Auswahl lag eine Analyse der vorhandenen Topologiegeneratoren zu Grunde. Die Erzeugung einer neuen Topologie kann auch über eine Generierungsdatei automatisiert werden, um bei großen Topologien nicht jeden einzelnen Knoten per Hand generieren zu müssen.

### 5.2 Last-Generator

Der Last-Generator erzeugt aus ausgewählten Lastgenerierungsmodellen Verkehr und speist diesen in eine ausgewählte Topologie ein. Momentan existieren folgende einfache Verkehrsmodelle: zwei Webverkehrsmodelle, ein peer to peer (P2P) Modell, ein IP Telefoniemodell und ein Modell für elastischen und unelastischen Verkehr. Die Verkehre der beiden in Kapitel 4 beschriebenen und zur Untersuchung ausgewählten Anwendungen Videokonferenz und VoIP, sowie die als Hintergrundverkehre in den Untersuchungen verwendbaren WWW und P2P Verkehre werden wie folgt generiert:

- Videokonferenz: Verwendet Trace-Files. Diese Trace-Files wurden durch Mitschnitte von Videokonferenzen am Lehrstuhl KOM der TU Darmstadt erzeugt. Als Videokonferenzanwendung wurde Netmeeting verwendet.

- VoIP: Verwendet das IP Telefoniemodell. Dabei handelt es sich um Constant Bit Rate (CBR) Verkehr, der aus User Datagram Protocol (UDP) Paketen besteht. Die Sprechdauer ist exponentiell verteilt und zwei Teilnehmer sprechen abwechselnd. Dies entspricht einem ON/OFF Markov Modell für jeden Teilnehmer.
- WWW: Verwendet ein Webverkehrsmodell, das WWW Sitzungen eines einzelnen Users nachbildet. Dabei sind die Dokumentgrößen Pareto verteilt und die Zwischenankunftszeiten der Dokumente exponentiell verteilt.
- P2P: Verwendet das P2P Modell. Dabei sind die Dokumentgrößen Pareto verteilt und die Zwischenankunftszeiten der Dokumente exponentiell verteilt.

Exakte Beschreibungen der WWW, VoIP und P2P Verkehrsmodelle sind [SCHMITT ET AL. 03-1], sowie die Erzeugung der Videotraces [SCHMITT ET AL. 04-1] zu entnehmen.

## Kapitel 6

# Netzwerk-Simulator

### 6.1 NS2 - Netzwerk-Simulator

Im Rahmen dieser Arbeit wird für die durchzuführenden Simulationen als Grundgerüst der Netzwerksimulator NS2, hierbei die Version 2.1b9a, verwendet.

NS2 ist ein Simulator für diskrete Ereignisse, der aus einer Zusammenarbeit der University of California Berkeley (UCB), Lawrence Berkeley National Laboratory (LBNL), University of Southern California (USC) / Information Science Institute (ISI) und Xerox Palo Alto Research Center (PARC) innerhalb des Virtual InterNetwork Testbed (VINT) Projekts entstanden ist. Im Bereich der QoS Untersuchungen stellt der NS2 das Standard-Simulationstool dar. Viele neue QoS Simulationsbausteine sind nur für den NS2 erhältlich. Der Quellcode und die Schnittstellen sind *open source*. Der NS2 ist in C++ mit einem OTcl (objektorientierte Erweiterung der Skriptsprache Tcl) *Frontend* implementiert. Die Verwendung zweier Programmiersprachen beruht auf unterschiedlichen Aufgaben, die der Simulator zu erfüllen hat. Einerseits muß er Protokolle effizient implementieren und ausführen, wofür sich eine kompilierbare Sprache wie C++ sehr gut eignet. Andererseits soll er eine einfach zu bedienende Schnittstelle für den Benutzer zur Verfügung stellen, damit dieser unterschiedlichste Netzwerktopologien schnell aufbauen und testen kann. Dies wird durch die Skriptsprache OTcl realisiert. Zwischen beiden Sprachen besteht eine Verbindung durch eine spezielle Tcl-C-Linkage Layer (TclCl). Weitere Informationen zu NS2 können [FALL & VARADHAN 02, GREIS 02] entnommen werden.

Im Folgenden werden die Anbindung an den Szenario-Generator, die durchgeführten Anpassungen des Simulators (Erweiterung der IntServ und DiffServ Module) an die gestellten Anforderungen, sowie die Realisierung der einzelnen QoS Technologien im Simulator beschrieben.

### 6.2 Anbindung an den Szenario-Generator

Wie in Kapitel 5 aufgeführt, wird der Simulator vom Szenario-Generator mit einer Topologie und darauf angesetztem Verkehr gespeist (siehe Grafik 6.1). Diese Informationen sind in einer sogenannten *run.tcl*-Datei, in dem für NS2 lesbaren *tcl*-Format gespeichert. Diese Datei ist komplett losgelöst von den zu untersuchenden QoS Technologien, um somit die *run.tcl*-Datei für die Untersuchung sämtlicher QoS Technologien nutzen zu können. Die Spezifikation von QoS Technologien findet in sogenannten *header.tcl*-Dateien statt, die alle QoS-technologiespezifischen Implementierungen (Policer, Marker, Classifier, Meter, Scheduler, etc.) für die von der *run.tcl*-Datei vorgegebene Topologie innerhalb des Simulators durchführen. Die *run.tcl*-Datei liest je nach gewählter QoS Technologie eine *header.tcl*-Datei ein und speist dann den NS2 Simulator mit dem sich aus beiden Dateien zusammensetzenden Startskript.

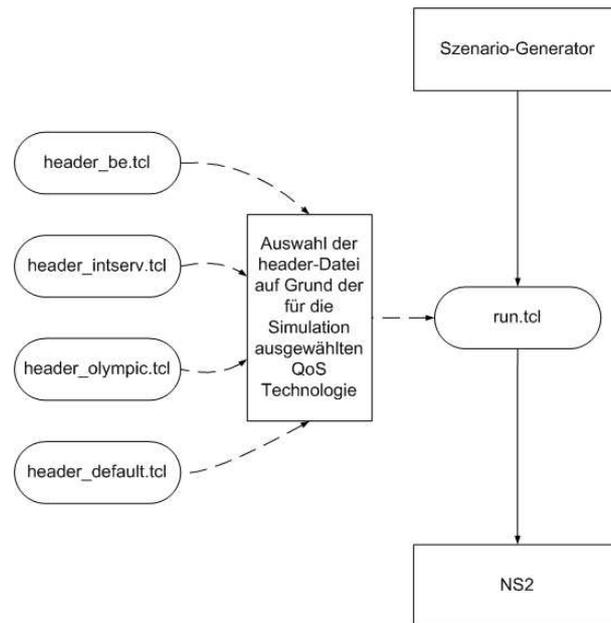


Abbildung 6.1: Anbindung des NS2 an den Szenario-Generator

### 6.3 IntServ Modul

Um IntServ so exakt wie möglich im NS2 nachzubilden, wird die von Martin Karsten am Lehrstuhl KOM Multimedia Communications der TU Darmstadt entwickelte RSVP Engine Release 3.0e im Rahmen des LETS QoS Projektes von den Projektteilnehmern der TU Darmstadt implementiert. Weitere Informationen über die RSVP Engine sind [KARSTEN 03] zu entnehmen.

Zusätzlich wird unter der Leitung des Projektpartners in Darmstadt die Weighted Fair Queuing (WFQ) (siehe [KESHAV 98]) Bedienstrategie zur Realisierung der Bedienung mehrerer parallel vorhandener IntServ Reservierungen in den C++ Code des NS2 implementiert und eine Admission Control hinzugefügt, die auf Grund von Bandbreitenreserven auf dem angeforderten Übertragungspfad entscheidet, ob eine Reservierung eines Users zugelassen oder abgewiesen wird. Hierzu sind stets aktuelle Informationen über die momentanen Reservierungen von Nöten.

### 6.4 DiffServ Modul

Um die innerhalb des LETS QoS Projektes und dieser Arbeit zur Untersuchung ausgewählten Ansätze exakt nachbilden zu können, wird das quellen- und senkenbasierte automatische Routing auf ein flowbasiertes manuelles Routing umprogrammiert und die Tracefilegenerierung durch zusätzliche DiffServ spezifische Informationen im C++ Code des NS2 erweitert.

Die beiden DiffServ Ansätze sind wie folgt in einer header.tcl-Datei realisiert:

Allgemein ist jede DiffServ Klasse im NS2 durch zwei virtuelle Warteschlangen innerhalb einer physikalischen Warteschlange realisiert. Die virtuellen Warteschlangen sind einzelne RED (Random Early Detection) Warteschlangen im RIO-C Modus. Dies bedeutet, daß Pakete, die durch den Policier (ist wählbar, in dieser Arbeit stets ein Token Bucket) als konform (in-profile) markiert werden in die erste virtuelle Warteschlange ihrer gewählten Klasse gelangen und ihr Wegwerfen dort von der gewichteten,

mittleren Warteschlangenlänge dieser virtuellen Warteschlange abhängt. Die vom Token Bucket als nicht konform (out-of-profile) markierten Pakete werden heruntergestuft und gelangen in die zweite virtuelle Warteschlange und ihr Wegwerfen hängt dort von den gewichteten, mittleren Warteschlangenlängen beider virtuellen Warteschlangen der physikalischen Warteschlange ab ([FALL & VARADHAN 02]). Die Parameter der RED Warteschlangen können jedoch auch so eingestellt werden, daß diese einfachen FIFO (First In First Out) Warteschlangen entsprechen. Zwischen den einzelnen physikalischen Warteschlangen entscheidet eine wählbare Bedienstrategie über die Reihenfolge der Bedienung.

Detailliert wird die Implementierung im NS2 für den Olympic und den Default Ansatz (vergleiche Kapitel 2 und 9) wie folgt realisiert:

Beim Olympic Ansatz werden die beiden virtuellen Warteschlangen einer physikalischen Warteschlange als FIFO Warteschlangen implementiert. Die aus den ersten virtuellen Warteschlangen heruntergestuften Pakete der EF, BE und QBSS Klasse werden in der zweiten virtuellen Warteschlange der jeweiligen Klasse unabhängig von den Warteschlangenlängen verworfen. Als Bedienstrategie wird ein striktes Priority Queueing (PQ) gewählt.

Beim Default Ansatz werden in der EF und AF3 Klasse ebenfalls für beide virtuelle Warteschlangen FIFO Warteschlangen verwendet und die vom Token Bucket aus der ersten virtuellen Warteschlange heruntergestuften Pakete werden in der zweiten virtuellen Warteschlange beider Klassen unabhängig von der Warteschlangenlänge verworfen. Die AF1 und AF2 Klasse verwenden jedoch den oben beschriebenen RIO-C Modus und werfen die Pakete nach dessen Schema. Als Bedienstrategie wird WRR (Weighted Round Robin) verwendet.

Die Realisierung dieser Komponenten in der Skriptsprache Tcl ist dem NS Manual ([FALL & VARADHAN 02]) und den header.tcl-Dateien ([HOFFMANN 04]) zu entnehmen.

Zusätzlich wird eine einfache, zuschaltbare lokale Admission Control, wie in Kapitel 2 beschrieben, hinzugefügt, die an den Edge-Routern der DiffServ Domänen überprüft, ob die gestellte Bandbreitenanforderung einer Anwendung für eine bestimmte Dienstgüteklasse erfüllt werden kann. Ist die geforderte Bandbreite nicht mehr vorhanden, wird die Anforderung abgewiesen. Um dies entscheiden zu können, muß in den Edge-Router über die momentanen Bandbreitensituationen in den einzelnen Klassen buchgeführt werden.

# Kapitel 7

## Testbett

### 7.1 Beschreibung

Das im Rahmen dieser Arbeit zur Untersuchung von QoS Technologien verwendete Testbett wurde in der TU Darmstadt am Lehrstuhl KOM Multimedia Communications von Ralf Steinmetz im Rahmen des DFN Projekts LETS QoS aufgebaut. Es besteht aus 16 PCs, die mit einem Intel Pentium 850 MHz Prozessor, 256 MB RAM (Random Access Memory), einer 20 GB Festplatte und 4 Netzwerkinterfacekarten ausgestattet sind. Es wird für jeden Router ein Global Positioning System (GPS) Receiver zur Synchronisation der einzelnen PC-Router (in Abständen von einer Mikrosekunde) verwendet. Als Betriebssystem wird FreeBSD 4.6 verwendet, da es sich in der Vergangenheit in diesem Einsatzbereich als sehr zuverlässig erwiesen hat. Ein zusätzlicher Router dient als Verbindung des Testbetts zur Außenwelt. Als Warteschlangenframework wird das Programm ALTQ (Alternate Queueing) verwendet.

### 7.2 Alternate Queueing Package (ALTQ)

Im Testbett steht das Verkehrsmanagement-Paket ALTQ zur Verfügung, das von Kenjiro Cho entwickelt wurde [CHO 01]. ALTQ ist ein Warteschlangenframework, der eine Vielzahl von Warteschlangenstrategien zur Verfügung stellt und darüber hinaus es dem Benutzer ermöglicht, einfach selbst neue Warteschlangenstrategien zu implementieren ohne genauere Kenntnis über den Kernel des Systems zu haben. Desweiteren ermöglicht ALTQ die Realisierung des Setzens von Zeitstempeln, der Verkehrsfilterung und -markierung, des Warteschlangen Managements und Scheduling, von RSVP und der Bandbreitenskalierung. Es läuft auf PC-basierten Routern unter einem BSD (Berkeley Software Design) Betriebssystem. Es kann unabhängig von der zu untersuchenden QoS Technologie eingesetzt werden. Es arbeitet für IntServ als Verkehrskontrollmodul und kann zum Aufbau eines DiffServ Netzwerkes eingesetzt werden. Momentan sind folgende Warteschlangenstrategien im offiziellen AltQ Release enthalten:

- FIFO: *First In First Out Queueing*
- WFQ: *Weighted Fair Queueing*
- RED: *Random Early Detection*
- RIO: *RED with IN and Out profile*
- H-FSC: *Hierarchical Fair Service Curve*
- CBQ: *Class Based Queueing*
- JoBS: *Joint Buffer management and Scheduling*

Die detaillierten Funktionserläuterungen einzelner Warteschlangenstrategien sind Kapitel 9 zu entnehmen. Umfassende Beschreibungen aller Warteschlangenstrategien sind in [SCHMITT ET AL. 03-1] zusammengefasst.

## Kapitel 8

# Extrapolationsverfahren

Ziel dieser Arbeit ist es ein Verfahren zu entwickeln, das es ermöglicht durch Beobachtung von Verkehrsgütemetriken deren neue Werte vorauszusagen, die sich bei der Variation der Quantität und Qualität der Last, der Komponenten und der Topologie oder der QoS Technologie ergeben. Dieses Verfahren wird als Extrapolation bezeichnet. Die Extrapolation soll Netzbetreibern bei der Entscheidung helfen, welche QoS Technologie sie in ihr bereits bestehendes oder geplantes Netzwerk zur Bereitstellung einer gewünschten Dienstgüte für eine gewählte Anwendung implementieren sollen. Sie soll weiterhin einen zeit- und kostenintensiven Testnetzaufbau und -betrieb zur Analyse der QoS Technologien vermeiden. In diesem Kapitel wird zunächst ein Extrapolations-Szenario definiert, das den Grundbaustein für die durchzuführende Extrapolation bildet, die im Anschluß daran beschrieben wird.

### 8.1 Definition

In dieser Arbeit ist ein Extrapolations-Szenario  $ES$  definiert als ein Quadrupel der im folgenden beschriebenen Komponenten:

Netzwerkconfiguration (NK):

Eine Netzwerkconfiguration  $NK=(T,B,LV,R)$  besteht aus einer Topologie  $T$ , einer Menge von Bandbreiten  $B$  von Netzwerkleitungen, einer Menge von Leitungsverzögerungen  $LV$  und Routinginformationen  $R$ . Diese Bestandteile sind wie folgt definiert:

- $T=(ID,N,E)$  ist eine Topologie. Ihre Komponenten sind  $ID$ =Identifizierer der Topologie, die Menge der Knoten (Host, Router)  $N=\{n_0, \dots, n_m\}$  und die Menge der Kanten (Verbindungen)  $E=\{e_{i,j}|e_{i,j}=(n_i, n_j) \in N \times N\}$ , mit  $m \in \mathbb{N}$ ,  $i, j \in \{0, \dots, m\}$  und  $i \neq j$ .
- $B = \{B_{i,j}|e_{i,j} \in E\}$  ist die Menge von Bandbreiten der Kanten  $e_{i,j} \in E$ .
- $LV = \{LV_{i,j}|e_{i,j} \in E\}$  ist die Menge von Leitungsverzögerungen der Kanten  $e_{i,j} \in E$ .
- $R$  sind Routinginformation.

Lastspektrum (LS):

Das Lastspektrum eines Extrapolationsverfahrens  $LS = \{VLS_i | \text{für einige } i \in VT\}$  ist eine Menge von Lastspektren verschiedener Verkehrstypen aus  $VT = \{VT_1, \dots, VT_k\}$ , mit  $k \in \mathbb{N}$  der Menge existierender Verkehrstypen. Diese verkehrstypenspezifische Lastspektren sind wie folgt definiert:

- $VLS_i = (SD_i, TT_i)$  ist das Lastspektrum des Verkehrstyps  $i$ .  $SD_i = \{(Q, S)|Q, S \in N, (Q, S) \text{ ist eine Kantenfolge von Kanten } e \in E \text{ mit einer gerichteten Verbindung von } Q \text{ nach } S; Q = \text{Verkehrsquelle, } S = \text{Verkehrssenke}\}$  ist eine Menge von Verkehrsquellenknoten und -senkenknoten des Verkehrstyps  $i$ .  $TT_i$  ist eine Beschreibung des Verkehrstyps  $i$ .

### QoS Technologien (QT):

Innerhalb eines Extrapolations-Szenarios werden die jeweils verwendeten QoS Technologien wie folgt definiert. Hierbei wurde die Menge der zur Verfügung stehenden QoS Technologien in Kapitel 2 bestimmt.

Die in einem Extrapolationsverfahren verwendeten QoS Technologien  $QT = \{QT_j | QT_j \in T\}$ , mit  $j \in N$  bilden eine Teilmenge der für die Extrapolation zur Verfügung stehenden QoS Technologien  $T$ . Jede QoS Technologie  $QT_j$  ist durch ihre Knoteninformationen  $KI$  und Bewertungsmetriken  $BM$  definiert. In einfachen Extrapolations-Szenarien, wie dem in Kapitel 10 untersuchten Extrapolations-Testszenario (ETS), wird jeweils nur eine QoS Technologie pro Extrapolations-Szenario verwendet. Kombinationen mehrerer QoS Technologien sind jedoch in anderen Zusammenhängen, wie auch in der Anwendung innerhalb des G-WiN, realisierbar (Hierbei sind Kombinationen mit IntServ auf Grund der notwendigen Ende-zu-Ende Implementierung auszuschließen).

- Die Knoteninformationen  $KI = \{(n, WM, ST) | \text{Knoten } n \in N, n \text{ benutzt QoS Technologie } QT_j\}$  sind eine Menge von Tripel. Jedes Tripel setzt sich aus einem Knoten  $n$ , seinem Warteschlangenmanagement  $WM$ , einer Menge von Tupeln bestehend aus Warteschlangentyp und Warteschlangenlänge (angegeben in Anzahl von Paketen, unabhängig von der Paketgröße), und einem Scheduler aus einer Menge von Schemulern  $ST$  zusammen. Aus diesen Informationen lassen sich die für eine Delaybestimmung notwendigen bedienstrategiebedingten Routerverzögerungen bestimmen.
- $BM = \{\text{Zeitskala, Garantie, Differenzierung, Preisgestaltung und Abrechnung, Aufstellung, Policing, Reaktivität, Heterogenität, Skalierbarkeit, Intelligenz}\}$  sind die Bewertungsmetriken einer QoS Technologie  $QT_j$ . Die Bedeutung der einzelnen Metriken ist in [SCHMITT ET AL. 02] beschrieben.

### QoS Spektrum (QS):

Das QoS Spektrum eines Extrapolationsverfahrens  $QS = \{VQS_i | \text{für einige } i \in VT\}$  ist eine Menge von QoS Spektren verschiedener Verkehrstypen aus  $VT = \{VT_1, \dots, VT_k\}$  der Menge existierender Verkehrstypen.

- Das QoS Spektrum  $VQS_i = \{Loss, Delay, Jitter\}$  vom Verkehrstyp  $i$  besteht aus drei meßbaren QoS Metriken für QoS-sensitive Anwendungen (siehe Kapitel 3). In diesem Fall beschreiben diese Metriken die Bedürfnisse des Verkehrstyps bzw. des Anwendungstyps. Richtwerte für die zu untersuchenden Anwendungen sind Kapitel 3 Tabelle 3.1 zu entnehmen.

## 8.2 Extrapolation

Das Ziel der QoS Extrapolation ist ein Transfer der in einem Ausgangsszenario erzielten Ergebnisse auf ein anderes, diesem ähnlichen Zielszenario, das sich entweder durch eine andere Netzwerkkonfiguration, ein anderes Lastspektrum, eine andere QoS Technologie oder eine Kombination dieser drei Freiheitsgrade vom Ausgangsszenario unterscheidet. Wie in Kapitel 1 beschrieben ergeben sich daraus sieben für die Untersuchungen relevante Fälle (Fall 2 - 8), die hier nochmals kurz in Tabelle 8.1 aufgezeigt werden. Der Netzbetreiber erhält Aussagen darüber:

- welche Netzkonfiguration für welche QoS Technologien und welches Lastspektrum (Anwendungen) erforderlich ist

Fall	andere Netzkonfiguration	anderes Lastspektrum	andere QoS Technologie
1	nein	nein	nein
2	nein	nein	ja
3	nein	ja	nein
4	nein	ja	ja
5	ja	nein	nein
6	ja	nein	ja
7	ja	ja	nein
8	ja	ja	ja

**Tabelle 8.1:** Auftretende Fälle durch Variation der 3 Freiheitsgrade

- welches Lastspektrum (Anwendungen) bei gegebenen QoS Technologien und Netzkonfiguration zugelassen werden kann
- welche QoS Technologien bei gegebenem Lastspektrum (Anwendungen) und Netzkonfiguration einsetzbar sind

Im Einzelnen kann die QoS Technologie in dieser Arbeit zwischen BE, IntServ, DiffServ Olympic und Default Ansatz variiert werden. Bei der Netzkonfiguration können sich die Topologie und die Bandbreiten im Zielszenario ändern und beim Lastspektrum die Verkehrsintensität und die Verkehrszusammensetzung.

Die in dieser Arbeit entwickelte Extrapolation läßt sich mit Hilfe unterschiedlicher Extrapolationsverfahren durchführen. Die Extrapolationsverfahren lassen sich in zwei Gruppen aufspalten, in Simulationen und in Berechnungen. Diese Verfahren bedienen sich der in den Kapiteln 5, 6 und 7 und in Abbildung 8.1 vorgestellten Komponenten: Szenario-Generator, Netzwerk-Simulator und Testbett. Diese müssen einer Kalibrierung unterzogen werden. Zusätzlich müssen die Simulations- und Berechnungsergebnisse validiert werden. Die Kalibrierung und Validierung bilden jedoch keine Teilverfahren der Extrapolation. Die allgemeine Durchführung der Extrapolation bzw. der einzelnen Extrapolationsverfahren wird im folgenden Abschnitt kurz beschrieben und in Kapitel 10 anhand eines Extrapolations-Testszenarios (siehe Abbildung 8.1) ausführlich getestet, bewertet und daraus eine allgemeine Durchführungsanleitung für Extrapolationen für beliebige Ausgangs- und Zielszenarien entwickelt. In Kapitel 11 wird diese auf das G-WiN angewendet. Die einzelnen Extrapolationsverfahren der beiden Gruppen werden wie folgt durchgeführt:

### 8.2.1 Simulationen

Das erste Extrapolationsverfahren ist die Simulation: Hierbei wird ein Ausgangsszenario im Simulator modelliert, durch den Szenario-Generator kalibriert, simuliert und durch Vergleichsmessungen im

Testbett validiert. Dann werden die zu untersuchenden Zielszenarien modelliert, simuliert und die zu untersuchenden QoS Metriken gemessen. Die Simulationen sehen im Einzelnen wie folgt aus:

Es wird ein Ausgangsszenario definiert. Durch einzelne Gruppen von Simulationsläufen mit sensitiven Variationen (Variation eines einzelnen Parameters bei Beibehaltung aller anderen Parametereinstellungen) der Ausgangsszenarioparameter, können Simulationen für jedes durch die drei Freiheitsgrade realisierbare Zielszenario durchgeführt werden. Diese können eine Variation der QoS Technologien zwischen BE, IntServ und DiffServ, wobei für DiffServ die beiden in Kapitel 2 kurz beschriebenen Ausprägungen verwendet werden, beinhalten. Für diese Variationen können jeweils die Auslastung, wie auch die Verkehrsanteile und -zusammensetzung variiert werden. Außerdem kann die Topologie des Ausgangsszenarios im Zielszenario erweitert werden und erneute Variationen der Extrapolationsszenarioparameter durchgeführt werden. Exakte Simulationsplanungen sind der Anwendung der Extrapolationsverfahren auf das Extrapolations-Testszenario in Kapitel 10 und der Anwendung auf das G-WiN in Kapitel 11 zu entnehmen.

Es werden bei jedem Simulationslauf Messungen des Ende-zu-Ende Delays, des Jitters und des Loss' der Pakete sämtlicher Verkehrstypen durchgeführt, deren Ergebnisse statistisch ausgewertet, über alle Simulationsläufe gleicher Konfigurationen mit verschiedenen Seeds gemittelt und grafisch aufbereitet. Außerdem werden sowohl Teilergebnisse dieser Größen als auch Queueing und Scheduling-Daten aller Knoten gemessen, die als Input für den analytischen Ansatz (Kapitel 9) unverzichtbar sind und diese wiederum über alle Simulationsläufe gemittelt. Diese Messungen sehen im Einzelnen wie folgt aus (die Definitionen der Meßgrößen können Anhang A entnommen werden):

- Ende-zu-Ende Delay: Für jeden Simulationslauf wird für jedes einzelne Paket jeder Verkehrsart der Delay gemessen. Für jede Verkehrsart werden der minimale, maximale und der mittlere Delay aller Pakete ermittelt. Über alle Simulationsläufe wird das 90 % Vertrauensintervall für den minimalen, maximalen und mittleren Delay jeder Verkehrsart gebildet.
- Jitter: Für jeden Simulationslauf wird der in einem vorgegebenen Zeitintervall zwischen den darin am Senkenknoten ankommenden Paketen auftretende Jitter gemessen. Für jede Verkehrsart werden der minimale, maximale und der mittlere Jitter aller Pakete ermittelt. Über alle Simulationsläufe wird das 90 % Vertrauensintervall für den minimalen, maximalen und mittleren Jitter jeder Verkehrsart gebildet.
- Loss: Für jeden Simulationslauf wird jeder Paketverlust zum Zeitpunkt seines Auftretens für jede Verkehrsart aufgezeichnet. Zusätzlich wird der prozentuale Paketverlust bezüglich aller am Senkenknoten angekommenen Pakete für alle Verkehrsarten, sowie für den Gesamtverkehr berechnet. Über alle Simulationsläufe wird das 90 % Vertrauensintervall für den prozentualen Loss jeder Verkehrsart, sowie für den Gesamtverkehr gebildet.
- Warteschlangenlänge: Für jeden Simulationslauf werden die Variationen der Warteschlangenlängen zum Zeitpunkt deren Auftretens für jede Warteschlange aufgezeichnet. Außerdem wird der prozentuale zeitliche Anteil einer auftretenden Warteschlangenlänge an der gesamten Aufzeichnungsdauer berechnet. Über alle Simulationsläufe wird das 90 % Vertrauensintervall für diesen prozentualen Anteil für alle Warteschlangen gebildet. Die Warteschlangenlänge wird sowohl in Bytes, als auch in Paketen angegeben. Die Warteschlangenlänge enthält auch das momentan bediente Paket.
- Wartezeit in einzelner Warteschlange: Für jeden Simulationslauf wird die Summe der Wartezeit in einer Warteschlange und der anschließenden Bedienzeit durch den Scheduler für jedes Paket und für jede Verkehrsart gemessen und diese Dauer für alle Pakete einer Verkehrsart gemittelt. Über alle Simulationsläufe wird für jedes Warteschlangen-Scheduler Paar das 90 % Vertrauensintervall dieser Zeitdauern gebildet.

- Virtuelle Warteschlangenlängen: Die Messungen entsprechen den zuvor beschriebenen Messungen der Warteschlangenlängen, jedoch werden diese für die nach DSCP unterschiedenen virtuellen Warteschlangen bei Verwendung der DiffServ Technologie eingesetzt.
- Auslastung: Für jeden Simulationslauf wird die Auslastung für einen oder mehrere vorgegebene Links, die von jeder Verkehrsart verursacht wird, sowie die Gesamtauslastung für ein vorgegebenes Zeitintervall gemessen und die Intervallwerte gemittelt. Über alle Simulationsläufe wird das 90 % Vertrauensintervall dieser Intervallwerte gebildet.

Diese Ergebnisse werden mit Hilfe von entwickelten Perl-Skripten aus den durch den Simulator erzeugten Trace-Dateien gewonnen und durch Gnuplot visualisiert. Die Skripten sind zusätzlich in der Lage, die Testbettdaten zu verarbeiten und auszuwerten.

#### **Validierung:**

Parallel werden einigen Simulationsläufen exakt entsprechende Testläufe im Testbett mit identischer Definition des Extrapolations-Szenarios durchgeführt. Die identische Implementation wird durch den Szenario-Generator (siehe Kapitel 5) und die verwendete Hard- und Software zu realisieren versucht (siehe Kapitel 7, 5). Die aus den Testbettläufen gewonnenen und aufbereiteten Ergebnisse der zu den Simulationsmessungen identischen Meßgrößen validieren die mit Hilfe des Simulators erzielten Ergebnisse und liefern konkrete Gültigkeitsbereiche für die Simulationsergebnisse. Leider konnte auf Grund der späten Fertigstellung der QoS Technologieimplementierung im Testbett nicht jedes im Simulator untersuchte Zielszenario nachgebildet werden. Die punktuell durchgeführten Validierungen lassen jedoch auf eine exakte Simulatorimplementation schließen ([SCHMITT ET AL. 04-1]). Generell ist das Modellieren und somit das Validieren im Testbett durch die vorhandene Hardware begrenzt (z.B. durch 16 PC Router, siehe Kapitel 5).

### **8.2.2 Berechnung**

Zusätzlich wird eine analytische Extrapolation durchgeführt. Hierbei werden zum einen Best und Worst Case Werte für Ende-zu-Ende Delay und Worst Case Werte für den Jitter für alle zu untersuchenden Zielszenarien bestimmt und zum anderen wird versucht, einen Ansatz zur einfachen Bestimmung von stochastischen Mittelwerten für den Ende-zu-Ende Delay eines Zielszenarios durch Extrapolation ausgehend von einem bekannten Ausgangsszenario und dessen Simulationsergebnissen aufzustellen. Die Formeln für die sogenannte Best/Worst Case Analyse basieren auf der Analyse der QoS Architekturen. Die Herleitung dieser Formeln, wie auch der des stochastischen Ansatzes, wird in Kapitel 9 detailliert beschrieben und die Anwendung der beiden Extrapolationsverfahren, ebenfalls wie die Simulationen, an einem Beispiel, dem Extrapolations-Testszenario, in Kapitel 10 getestet und bewertet und in Kapitel 11 auf das G-WiN angewendet.

#### **Validierung:**

Hierbei werden die Formeln, die die Grundlage für die Berechnung der stochastischen Mittelwerte bilden, mit Hilfe eines Single Queue Simulators validiert (siehe Kapitel 9).

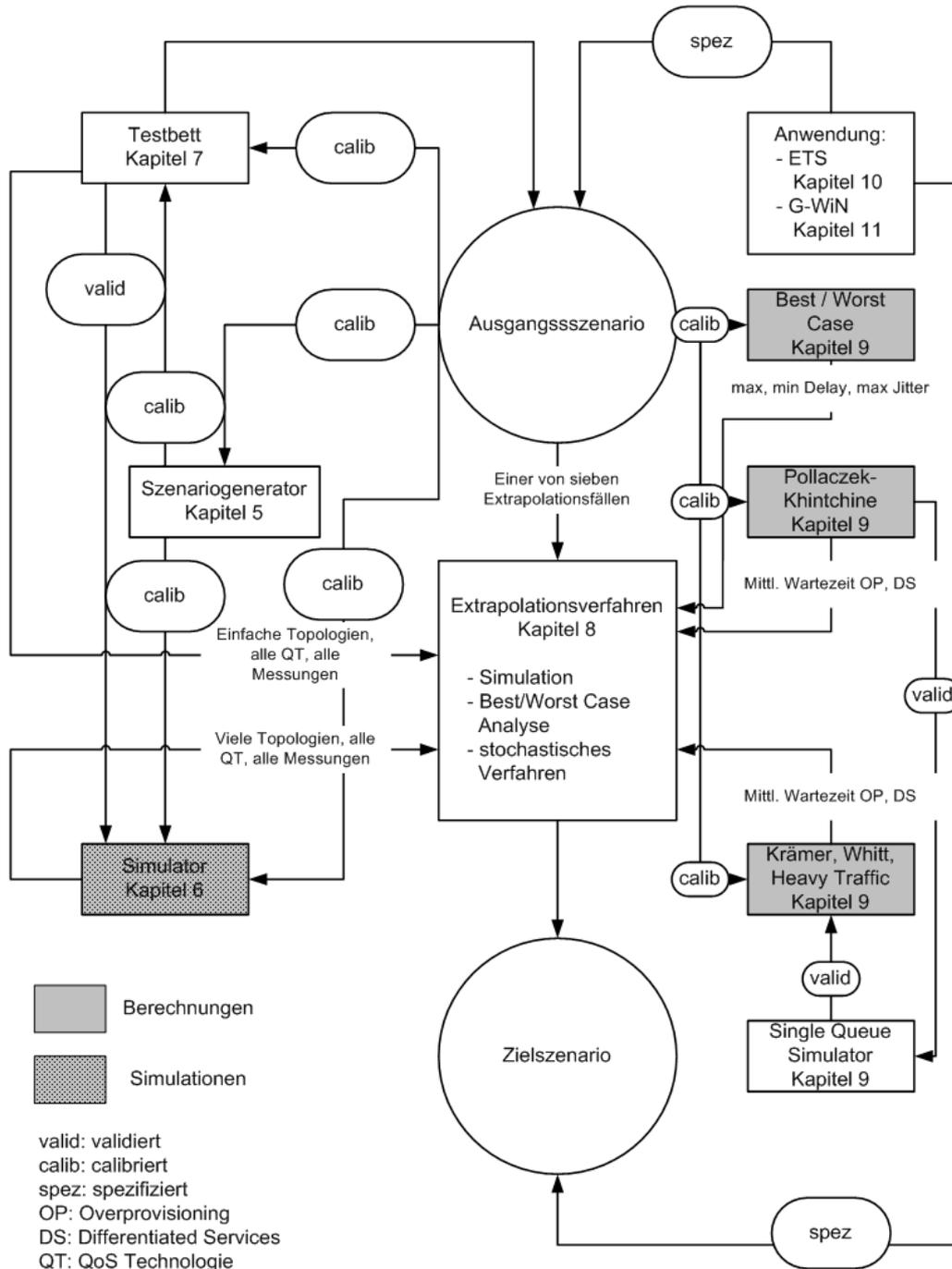


Abbildung 8.1: Extrapolationsschema

## Kapitel 9

# Modellierung und analytischer Ansatz

In diesem Kapitel wird eine der beiden Gruppen von Extrapolationsverfahren, die Berechnungen, vorgestellt. Dabei handelt es sich um einen analytischen Extrapolations-Ansatz, der parallel zu den in Simulations- und Testbettläufen erzielten Ergebnissen quantitative Vorhersagen für QoS Eigenschaften von beliebigen Netzen treffen soll. Zunächst werden die detaillierten Architekturen des Best Effort Ansatzes und der zu untersuchenden QoS Technologien, sowie der sich aus diesen Architekturen ergebenden Bedienstrategien und Formeln zur deterministischen Berechnung der Best und Worst Case Ende-zu-Ende Delay- und Jitterwerte beschrieben. Diese Formeln ermöglichen die Berechnung dieser Best und Worst Case Werte für sämtliche Zielnetze einer Extrapolation. Danach wird durch ein stochastisches Verfahren versucht, Mittelwerte für den Ende-zu-Ende Delay aus erzielten Simulationsergebnissen (oder Meßergebnissen) eines Ausgangsszenarios durch Extrapolation für ähnliche Zielnetze zu erzeugen. Werte für den Jitter können aus den Mittelwerten des Ende-zu-Ende Delays nicht berechnet werden.

### 9.1 Architekturen und Best/Worst Case Analyse

Da sich der Ende-zu-Ende Delay, wie in Kapitel 3 beschrieben, aus einem konstanten und einem schwankenden Anteil zusammensetzt, wobei der schwankende Anteil mitunter aus den Warteschlangen- und Scheduling-Verzögerungen (wird im Anschluß definiert) innerhalb der Router besteht, müssen die zu untersuchenden QoS Architekturen bis auf diese Ebene heruntergebrochen und analysiert werden, um eine Aufstellung von Best/Worst Case Formeln zur Extrapolation auf die zu untersuchenden Extrapolationsfälle zu ermöglichen. Es werden Formeln für die unterschiedlichen QoS Technologien gesucht, die dann je nach Netzwerkkonfiguration und Lastspektrum des Zielszenarios durch unterschiedliche Parametrisierung angepaßt werden können. Um einen Vergleich zum Best Effort Standard Internet Ansatz und dem Overprovisioning Ansatz zu ermöglichen, wird auch dessen Architektur vorweg kurz diskutiert. Die detaillierten Beschreibungen der Architekturen beziehen sich auf die bereits in Kapitel 2 durchgeführten Erläuterungen.

#### 9.1.1 Best Effort

Beim Best Effort Dienst und somit auch beim verwendeten Overprovisioning Ansatz wird für jede Ausgangsleitung im Router eine Warteschlange in Verbindung mit der First Come First Serve (FCFS) Bedienstrategie verwendet. Dies hat zur Folge, daß die Pakete genau in der Reihenfolge ihres Eintreffens in der Ausgangswarteschlange vom Scheduler ausgewählt und auf den ausgehenden Link gesetzt werden. Dies bedeutet, daß QoS-sensitive Anwendungen exakt die gleiche Behandlung erfahren wie Anwendungen, die sich gegenüber den QoS Metriken robust verhalten, welche durch ihre Gleichbehandlung bei bestimmten Auslastungsverhältnissen eine Unbrauchbarkeit der QoS-sensitiven Anwendungen verursachen können. Dies kann durch lange Warteschlangenwartezeiten oder sogar durch ein Wegwerfen von Paketen durch Warteschlangenüberläufe verursacht werden. Um dies zu verhindern, greifen die für die Untersuchungen ausgewählten QoS Technologien innerhalb ihrer Ansätze auf Bedienstrategien zurück,

die von der gerade beschriebenen Bedienstrategie abweichen.

Auf Grund der Best Effort Bedienstrategie setzt sich der Best Effort Worst Case Ende-zu-Ende Delay  $D_{max}$  für jedes Netzwerkelement aus dem Propagation Delay  $D_{prop}$ , der Wartezeit in der Warteschlange, der Restbedienzeit eines Paketes im Scheduler und der Bedienzeit des betrachteten Paketes im Scheduler zusammen (jeweils das Setzen des Paketes auf den Link). Die Wartezeit in der Warteschlange entsteht durch die maximal in dieser vor dem betrachteten Paket gepufferten Pakete  $Q_{max} - 1$  (maximale Warteschlangenlänge abzüglich eines Platzes für das zu betrachtende Paket) mit maximaler Größe  $MTU$  (Maximum Transmission Unit), die jeweils auf den Link der Kapazität  $\gamma$  gebracht werden müssen. Die Restbedienzeit besteht aus der Bedienzeit eines Paketes maximaler Größe  $MTU$  das ebenso wie das betrachtete Paket der Größe  $M$  auf den Link der Kapazität  $\gamma$  gebracht werden muß. Hieraus ergibt sich Formel 9.1 für den maximalen Ende-zu-Ende Delay. Der verwendete Index gibt die Position eines Netzwerkelementes auf dem Übertragungspfad von Sender zu Empfänger an. Unter einem Netzwerkelement versteht man hier einen sendenden Host oder einen Router zusammen mit der Ausgangsleitung zum auf dem Pfad folgenden Host oder Router. In allen folgenden Formeln steht  $n$  für die Anzahl der Netzwerkelemente, die der Pfad, für den die Berechnungen durchgeführt werden, enthält, und  $i$  für die Position eines Netzwerkelementes entlang des Pfades. Die Position 0 entspricht dem sendenden Netzwerkelement.

$$D_{max} = \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{(Q_{max,i} - 1) * MTU_i}{\gamma_i} + \frac{MTU_i}{\gamma_i} + \frac{M}{\gamma_i} \right) \quad (9.1)$$

Der minimale Ende-zu-Ende Delay  $D_{min}$  der bei Best Effort auftreten kann, unterscheidet sich vom maximalen dadurch, daß keine Wartezeit in der Warteschlange und keine Restbedienzeit vorliegt:

$$D_{min} = \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{M}{\gamma_i} \right). \quad (9.2)$$

Somit entsteht folgende Worst Case Formel für den maximalen Jitter  $J_{max}$  (siehe Formel 9.3), der die Differenz des maximalen und minimalen Ende-zu-Ende Delays unter der Berücksichtigung der maximal auftretenden Paketgrößendifferenz  $\Delta M$  zwischen zwei für die Jitterberechnung betrachteten Pakete darstellt. Bei der maximalen Wartezeit in der Warteschlange muß berücksichtigt werden, daß der Jitter im Gegensatz zum Ende-zu-Ende Delay nicht über die Gesamtdauer einer Untersuchung betrachtet wird, sondern lediglich in einem physiologisch vorgegebenen Zeitintervall  $\Delta t$ . Daher muß geprüft werden, ob für die auftretenden Paketgrößen bei der verwendeten Topologie die Warteschlange ganz oder nur zum Teil in diesem Zeitintervall geleert oder gefüllt werden kann und sich die maximale Wartezeit und somit der maximale Jitter verringert. Dies bedeutet, wenn  $Q_{max,i} \cdot MTU_i > \Delta t \cdot \gamma_i$  verringert sich die maximale Wartezeit, bei  $Q_{max,i} \cdot MTU_i \leq \Delta t \cdot \gamma_i$  hat die Intervallgröße keinen Einfluß auf die maximale Wartezeit.

$$J_{max} = \sum_{i=0}^{n-1} \left( \frac{(Q_{max,i} - 1) * MTU_i}{\gamma_i} + \frac{MTU_i}{\gamma_i} + \frac{\Delta M_i}{\gamma_i} \right) \quad (9.3)$$

### 9.1.2 IntServ

Die in Abbildung 9.1 [BRADEN ET AL. 97] dargestellte, feingranulare IntServ Architektur greift nochmals die einleitenden Beschreibungen von Kapitel 2 auf. Der in die IntServ Domäne eingehende Verkehr wird an einem Edge-Router dieser Domäne klassifiziert, wenn nötig auf Grund dieser Klassifikation von einem Flußsteuermechanismus zugeschnitten und in entsprechende Warteschlangen geschoben.

Für IntServ sind momentan 2 Services spezifiziert: Der Guaranteed Service und der Controlled Load

Service. Da in dieser Arbeit QoS-sensitiver Verkehr bzw. Applikationen untersucht werden, ist hier lediglich die Behandlung des hochpriorigen Guaranteed Service, der für diese Art von Anwendungen mit einer Garantie für eine obere Grenze des Ende-zu-Ende Delays und einer Garantie für Verlustlosigkeit für konforme Paketdurchsätze entwickelt wurde, von Interesse. Dies bedeutet, es wird nur die Verzögerung, die der hochpriorige Verkehr innerhalb des Guaranteed Services erfährt, analysiert. Daher liegt das Interesse in Konfigurationen, bei denen die Bandbreite eines Netzwerks in reservierte Bandbreite für Anwendungen, die den Guaranteed Service benutzen, und einen verbleibenden Teil von Bandbreite, der sich aus der Differenz der Link Bandbreite und der Summe der reservierten Bandbreiten ergibt und für Best Effort Anwendungen Verwendung findet, aufgespalten wird.

### IntServ - GS

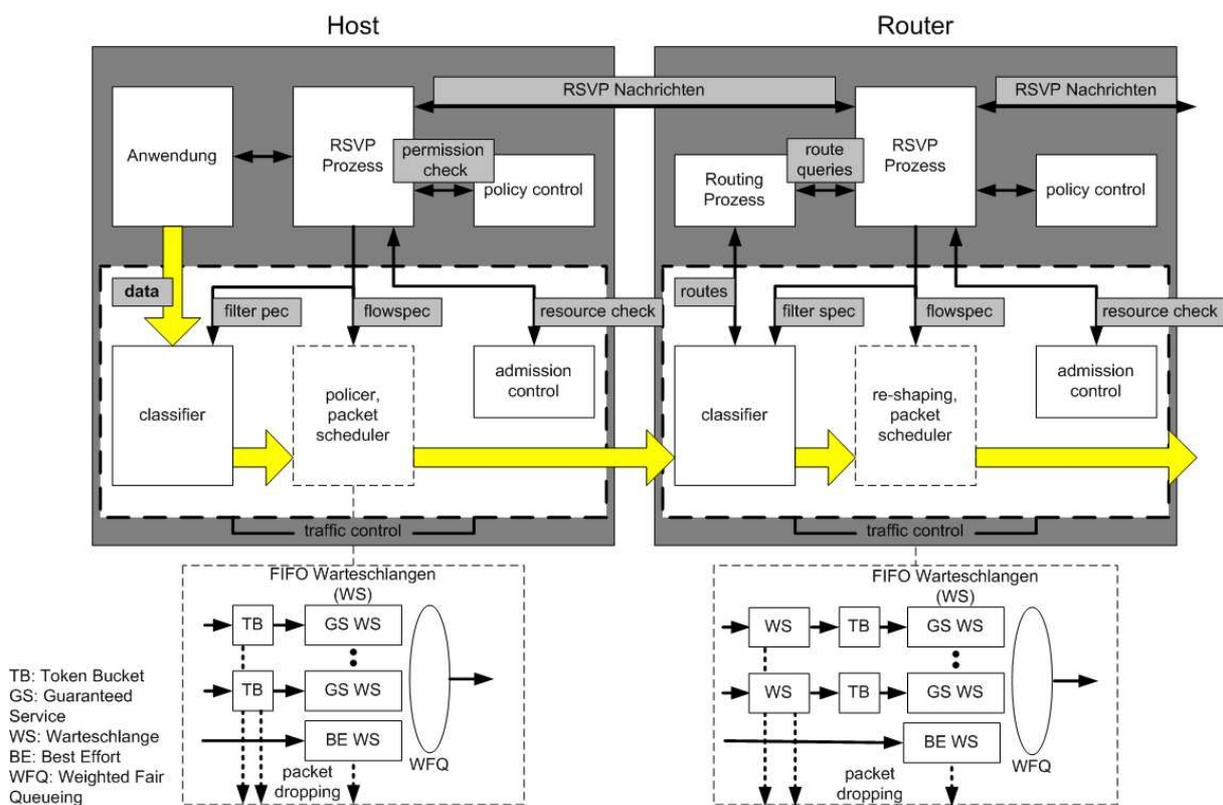


Abbildung 9.1: IntServ Architektur

Pakete, die dem Guaranteed Service nicht entsprechen, werden nicht auf Best Effort abgestuft, sondern verworfen. Dies begründet sich darin, daß wegen Nicht-Einhaltung der Übertragungsvereinbarungen oder wegen überlaufender Warteschlange herabgestufte Pakete, die dann mit Hilfe des Best Effort Service übertragen werden, eine so hohe Verzögerung erfahren, daß sie von einer QoS-sensitiven Anwendung wie einer Videokonferenz nicht mehr zu gebrauchen sind. Der Guaranteed Service garantiert jedem einzelnen Paket der Anwendung, die für diesen klassifiziert wurde, einen maximalen Ende-zu-Ende Delay ohne Verlust. Dies kann dadurch realisiert werden, daß nicht nur in den Edge-Routern, sondern auch in den Core-Routern ein Reshaping des Verkehrs mit Hilfe eines Flußsteuermechanismus, dem Token Bucket (TB), stattfindet. Die Verkehrsregelung wird hierbei mit dem Anwender zuvor ausgehandelt und die Token Bucket Parameter demnach gewählt. Über die Einstellung der Token Bucket Parameter können die zur Delay und Loss Garantierrealisierung benötigten und zuvor auf Grund der Delayanforderungen der Anwendung berechneten Puffer- und Übertragungsratenwerte einer jeden Anwendung implementiert werden. Die Token Bucket Parameter sind innerhalb zweier Parameterpaare  $(r, b)$ ,  $(M, p)$  definiert.  $r$  steht für die Tokengenerierungsrate und  $b$  für die konstante Buckettiefe;  $M$  ist die maximale Paketgröße der zu regelnden Pakete und  $p$  die maximal erlaubte Senderate, die sogenannte Peakrate. Die Verwendung des zweiten Parameterpaars  $(M, p)$  ist optional. Wenn  $p \rightarrow \infty$ , dann ist die maximale Senderate lediglich durch den maximalen Burst und die maximal übertragbare Rate  $R$  begrenzt. Angaben dieser beiden Werte dienen jedoch einer exakteren Bestimmung des maximalen Ende-zu-Ende Delays, wie den anschließenden Formeln zu entnehmen ist. Die detaillierte Aushandlung und Verbreitung dieser Parameter auf den für die entsprechende Anwendung gewählten Übertragungspfaden mit Hilfe des in Kapitel 2 beschriebenen Reservierungsprotokolls ist [SHENKER ET AL. 97, WROCLAWSKI 97] zu entnehmen. Durch diese Art von Verkehrsregelung ist es möglich, Formeln zur Berechnung des maximalen Ende-zu-Ende Delays  $D_{max}$  anzugeben [SHENKER ET AL. 97]. Der verwendete Index entspricht dem des Best Effort Ansatzes:

$$D_{max} = \frac{b-M}{R} \cdot \frac{p-R}{p-r} + \frac{M}{R} + \sum_{i=0}^{n-1} \left( \frac{C_i}{R} + E_i \right) \quad , p > R \quad (9.4)$$

$$D_{max} = \frac{b}{R} + \sum_{i=0}^{n-1} \left( \frac{C_i}{R} + E_i \right) \quad , p \rightarrow \infty \quad (9.5)$$

$$D_{max} = \frac{M}{R} + \sum_{i=0}^{n-1} \left( \frac{C_i}{R} + E_i \right) \quad , p \leq R \quad (9.6)$$

Die Formeln enthalten folgende Parameter:

$r$	Token Generierungsrate	[Bytes/s]
$b$	Token Buckettiefe	[Bytes]
$p$	Peakrate	[Bytes/s]
$M$	maximale Paketgröße	[Bytes]
$C_i$	ratenabhängiger Fehlerterm des Netzwerkelements $i$	[Bytes]
$E_i$	ratenunabhängiger Fehlerterm des Netzwerkelements $i$	[s]
$R$	reservierte Rate	[Bytes/s]

**Tabelle 9.1:** Parameter zur Ende-zu-Ende Delayberechnung

Die Formeln sind am einfachsten aus der Abbildung 9.2 abzuleiten, die auf dem “fluid model“ Ansatz von [ZHANG 95, PAREKH & GALLEGER, WHITE & CROWCROFT, GLASMANN ET AL.] beruht. Bei diesem Ansatz wird das betrachtete System (in dieser Arbeit ein Pfad) als Blackbox gesehen, in die am Systemeingang Bits einfließen, im System bedient werden und am Systemausgang ankommen. Hierbei gestaltet sich die Ankunftscurve (Zahl der bislang angekommenen Pakete) von Bits folgendermaßen: Durch die Verkehrsregelung durch Token Buckets wird im “Worst Case“ (der hier von Interesse ist) zunächst ein Burst von  $b$  Bits, gefolgt von einer Folge von Bits gesendet mit der Tokengenerierungsrate  $r$  (Funktionsweise des Token Bucket [TANENBAUM 98]) oder es wird zunächst ein Paket mit maximaler Paketgröße  $M$  gesendet, gefolgt von einer Folge von Bits mit der maximalen Peakrate  $p$ , die nach komplettem Aufbrauchen der vorhandenen Tokens in die geringere Tokengenerierungsrate übergeht. Die Bedienkurve (Zahl der bislang abgegangenen Pakete) des Systems gestaltet sich folgendermaßen: Innerhalb des Systems erfahren die Pakete einen konstanten Gesamtdelay  $T$ , den man in der Blackboxansicht auch zusammengesetzt direkt an den Anfang der Bedienung setzen kann. Dieser Delay besteht aus einem ratenbasierenden Teil  $C$  und einem ratenunabhängigen Teil  $E$ , die sich je nach Bedienstrategie des Systems unterschiedlich zusammensetzen. Desweiteren werden die Pakete mit einer reservierten Rate  $R$  durch das System geschickt. Für ein beliebiges Bit kann der Zeitpunkt des Abschickens und der Ankunft graphisch (siehe Abbildung 9.2) dadurch bestimmt werden, daß eine Waagerechte von der y-Achse auf Höhe des betrachteten Bits gezogen wird und die Schnittpunkte dieser mit der Ankunfts- (Punkt  $X$ ) und Bedienkurve (Punkt  $Y$ ) bestimmt werden. Diese Punkte werden senkrecht auf die x-Achse projiziert. Dadurch erhält man die Punkte  $t'$  und  $t''$  auf der Zeitachse. Die zeitliche Differenz dieser beiden Punkte gibt den maximalen Ende-zu-Ende Delay  $D_{max}$  an.

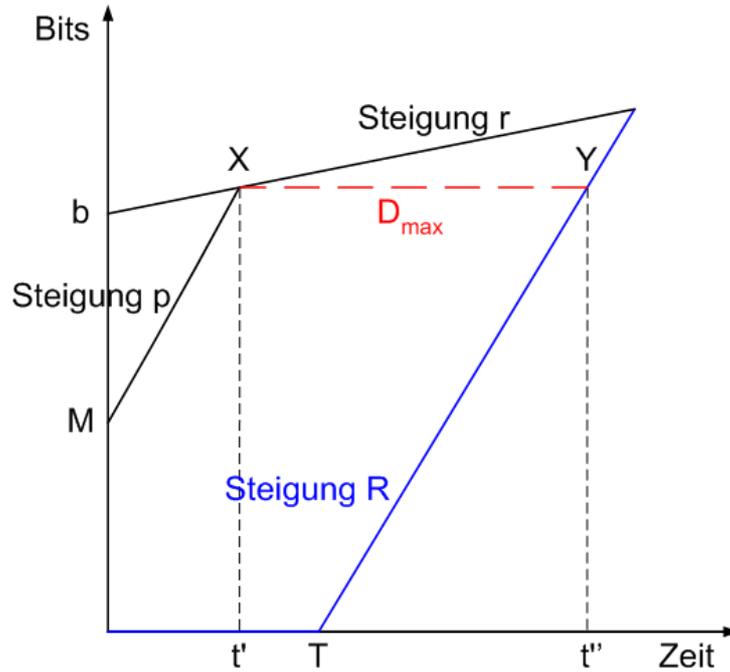


Abbildung 9.2: IntServ Calculus Grafik: Kumulativer Fluß bei IntServ als Funktion der Zeit

Die Formeln des maximalen Ende-zu-Ende Delays (Formeln 9.1-9.3) lassen sich wie folgt aus der Abbildung ableiten.

Punkt X:

$$M + p \cdot t' = b + r \cdot t'$$

$$M + p \cdot t' - b - r \cdot t' = 0$$

$$t' \cdot (p - r) = b - M$$

$$\Rightarrow t' = \frac{b-M}{p-r}$$

Punkt Y:

$$R \cdot (t'' - T) = M + p \cdot t'$$

$$t'' - T = \frac{M}{R} + \frac{p \cdot t'}{R}$$

$$\Rightarrow t'' = \frac{M}{R} + \frac{p \cdot (b-M)}{R \cdot (p-r)} + T$$

$$D_{max} = t'' - t' = \frac{M}{R} + \frac{p \cdot (b-M)}{R \cdot (p-r)} + T - \frac{b-M}{p-r}$$

Für den Fall, daß wie in dieser Arbeit als Schedulingverfahren WFQ (Weighted Fair Queueing), das typische Schedulingverfahren für IntServ in Verbindung mit Guaranteed Service verwendet wird, ergeben sich nach [ZHANG 95, PAREKH & GALLEGGER, WHITE & CROWCROFT, GLASMANN ET AL.] für die Fehlerterme  $C$  und  $E$  folgende Werte für jedes Netzwerkelement:

$$C = M \tag{9.7}$$

$$E = B_r + D_{prop}, \quad \text{mit } B_r = \frac{MTU}{\gamma} \tag{9.8}$$

$C$  entspricht der maximalen Paketgröße eines Stroms, die in den Delayformeln 9.4-9.6 noch durch die reservierte Rate  $R$  geteilt werden muß. Der Bruch bezeichnet den Zeitaufwand, um ein Paket der Größe  $M$  bei einer reservierten Rate  $R$  vom Scheduler bedienen zu lassen. Dieses Bedienen umfaßt nicht nur die Ablaufplanung, sondern auch das Verschicken der Pakete über die Ausgangsleitung. Die WFQ Bedienstrategie berücksichtigt, um eine faire Behandlung gleichpriorer Flows  $n$  (die den Guaranteed Service GS verwenden) zu ermöglichen, zur Festlegung der Bedienreihenfolge der Pakete die Paketgröße, den Ankunftszeitpunkt und das Gewicht der Reservierung  $x$ , das den Bandbreitenanteil (Rate)  $R$  der Linkbandbreite  $\gamma$  bestimmt. Abbildung 9.3 zeigt beispielhaft eine Aufteilung der Linkbandbreite. Die Reservierungsgewichte erfüllen hierbei folgende Gleichung:  $\sum_{j=1}^n x_j + y = 1$ . Die detaillierte Funktionsweise des WFQ Algorithmus ist [KESHAV 98] zu entnehmen.

Der Parameter  $E$  setzt sich aus der Restbedienzeit  $B_r$  und dem Propagation Delay  $D_{prop}$  einer ausgehenden Leitung zusammen. Unter der Restbedienzeit versteht man die Zeit, die ein Paket warten muß, bis das aktuell gesendete Paket durch eine Ausgangsschnittstelle vollständig auf die Ausgangsleitung gelangt ist. Die Paketgröße entspricht in einer Worst Case Untersuchung der Maximum Transmission Unit  $MTU$ . Diese ist die maximale Paketgröße aller Ströme dieses Links.

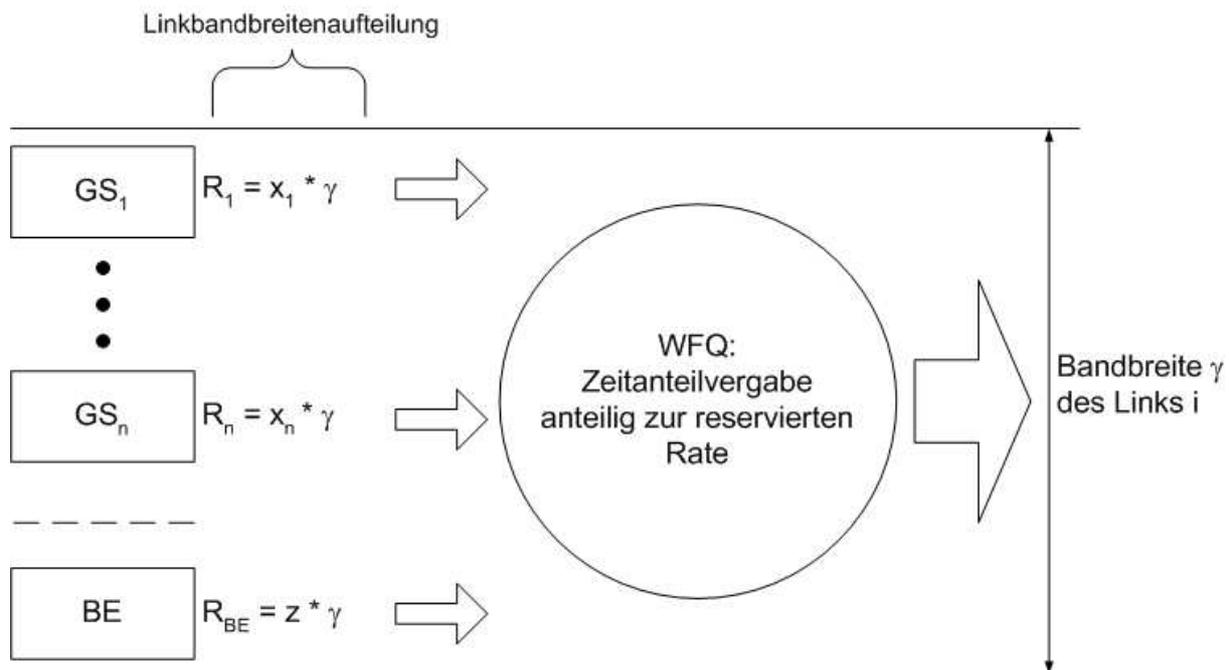


Abbildung 9.3: IntServ - Scheduling

Auf Grund der verwendeten Jitterdefinition ist für die Worst Case Berechnung des Jitters neben dem maximalen auch der minimale Ende-zu-Ende Delay von Nöten. Für diese Worst Case Berechnung muß der kleinstmögliche Ende-zu-Ende Delay berechnet werden. Der sogenannte „Best Case“ dieses Delays,  $D_{min}$ , tritt auf, wenn keine Restbedienzeiten auftreten, also der Scheduler beim Ankommen eines Paketes nicht belegt ist. Außerdem ist die Peakrate  $p$  kleiner als die reservierte Rate  $R$ . Konkurrierende Reservierungen spielen bei IntServ keine Rolle, da der Strom stets durch das Shaping mit Hilfe der Token Buckets begrenzt wird, also zusätzlich freie Ressourcen nur im Rahmen der Token Bucket Parameter zur Nutzung hinzugezogen werden können. Somit unterscheidet sich der minimale vom maximalen Delay lediglich durch die Summe der Restbedienzeiten und die Paketgrößendifferenz der für den Jitter be-

trachteten Pakete (siehe Formeln 9.9 und 9.10). Bei der maximalen Wartezeit in der Warteschlange muß berücksichtigt werden, daß der Jitter im Gegensatz zum Ende-zu-Ende Delay nicht über die Gesamtdauer einer Untersuchung betrachtet wird, sondern lediglich in einem vorgegebenen Zeitintervall. Daher muß geprüft werden, ob für die auftretenden Paketgrößen bei der verwendeten Topologie die Warteschlange ganz oder nur zum Teil in diesem Zeitintervall geleert oder gefüllt werden kann und sich die maximale Wartezeit und somit der maximale Jitter  $J_{max}$  verringert. Hieraus ergibt sich folgende Worst Case Jitter Formel.

$$D_{min} = \frac{M}{R} + \sum_{i=0}^{n-1} D_{prop,i}, \quad p \leq R \quad (9.9)$$

$$J_{max} = \sum_{i=0}^{n-1} B_{r,i} + \frac{\Delta M_i}{R}, \quad p \leq R \quad (9.10)$$

### 9.1.3 DiffServ

Wie in Kapitel 2 beschrieben, werden in dieser Arbeit zwei unterschiedliche Ansätze untersucht. Der Olympic und der Default Ansatz.

#### 9.1.3.1 Olympic Ansatz

Der Olympic Ansatz benutzt das EF PHB, das Default PHB (Best Effort, BE) und das neudefinierte QBone-Scavenger-Service PHB (QBSS). Die exakte Funktionalität der PHBs ist Kapitel 2 zu entnehmen. Die Klasse, die das EF PHB verwendet, ist für QoS-sensitive Anwendungen geeignet, die BE Klasse für Webverkehr und die QBSS Klasse für niederpriorere Anwendungen, die keinen Dienstgütereanforderungen, insbesondere keinen Realzeitanforderungen, unterliegen. Die am Edge-Router in die DiffServ Domäne einfließenden Pakete werden, wie bereits bekannt, vom Traffic Conditioner, wenn sie SLA-konform sind, der für sie bestimmten Klasse zugewiesen, markiert und in die für jede einzelne Klasse vorgesehene Warteschlange geschoben (siehe Abbildung 9.4) und dort nach dem der Klasse zugewiesenen PHB behandelt.

### DiffServ - Edge Knoten (Olympic Ansatz)

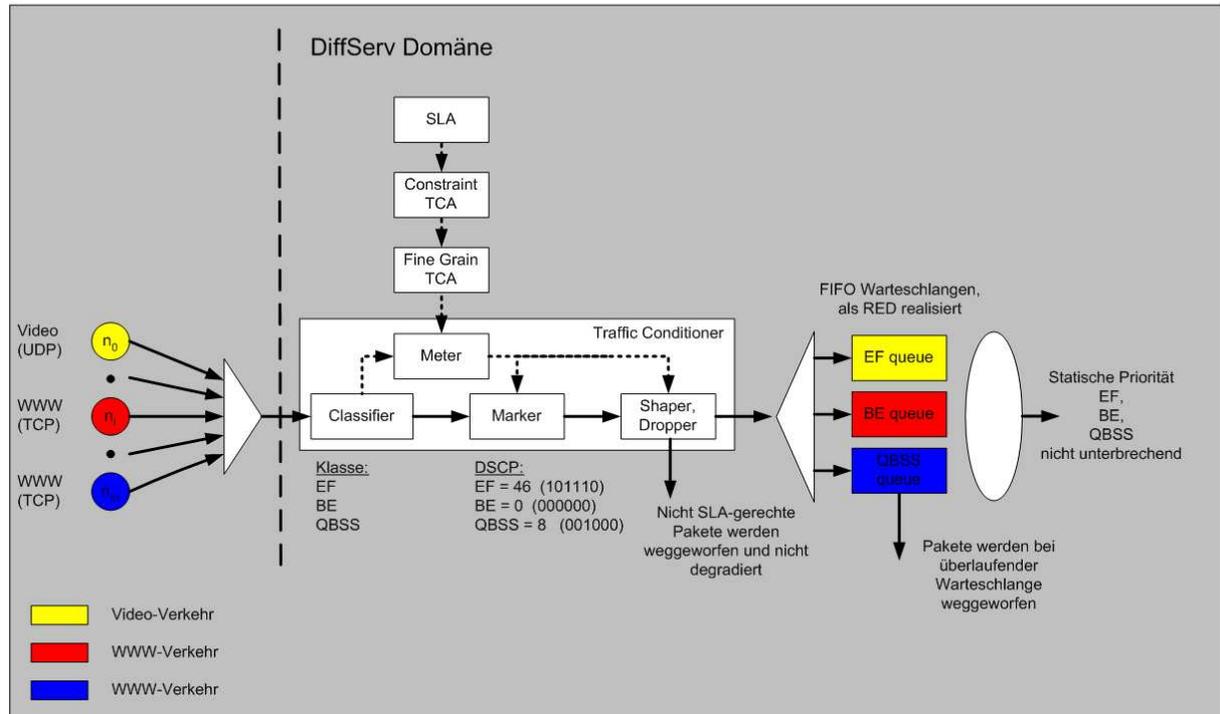


Abbildung 9.4: DiffServ Edge-Knoten Architektur nach dem Olympic Ansatz

Die Klassifizierung in den Core-Routern ist darauf reduziert, die Pakete auf Grund ihrer Markierungen im ToS-Byte des IP-Headers den entsprechenden Warteschlangen der einzelnen Klassen und somit einer bestimmten PHB zuzuordnen. Die Klassifizierung muß in jedem einzelnen Router (siehe Abbildung 9.5) auf dem Weg der Pakete durch die DiffServ Domäne wieder durchgeführt werden, um die in einem PHB zugesagte Behandlung garantieren zu können.

Die Bedienstrategie ohne eine Admission Control, deren anwendbare Varianten in Kapitel 2 beschrieben sind, für die drei Klassen bzw. deren Warteschlangen ist in allen Routern innerhalb der DiffServ Domäne gleich. Bei jeder der drei Warteschlangen handelt es sich um sogenannte First In First Out (FIFO) Warteschlangen. Dies bedeutet, die eingehenden Pakete werden der Eingangsreihenfolge nach die Warteschlange auch wieder verlassen. Um nun die Funktionalität der einzelnen PHBs zu realisieren, herrscht zwischen den drei Warteschlangen eine strikte Priorität, die durch den Scheduler realisiert wird (siehe Abbildung 9.6). Die höchstpriorie Klasse ist die mit dem zugewiesenen EF PHB, dann kommt die Klasse mit dem BE PHB und zuletzt die mit dem QBSS PHB. Ein Downgraden der nicht SLA-konformen Pakete entfällt in diesem Ansatz. Die EF Pakete würden durch ein Downgraden und durch die Behandlungsweise in einer untergeordneten Klasse zu lange verzögert und somit unbrauchbar. Sie werden daher genau wie die Best Effort Pakete bei Nichtkonformität weggeworfen (QBSS Verkehr unterliegt normalerweise keinen Einschränkungen, die Pakete sind also immer konform). Überlaufende Warteschlangen innerhalb der Domäne führen ebenfalls stets zu Paketverlusten in den betroffenen Klassen.

Durch diese Bedienstrategie wird ein priorisiertes Paket an jedem Router im schlimmsten Fall durch die maximal vor diesem in die priorisierte Warteschlange aufnehmbaren, vor diesem Paket angekommenen  $Q_{max} - 1$  Pakete der maximalen Paketgröße der priorisierten Ströme,  $MTU_{EF}$ , (siehe

## DiffServ - Core Knoten (Olympic Ansatz)

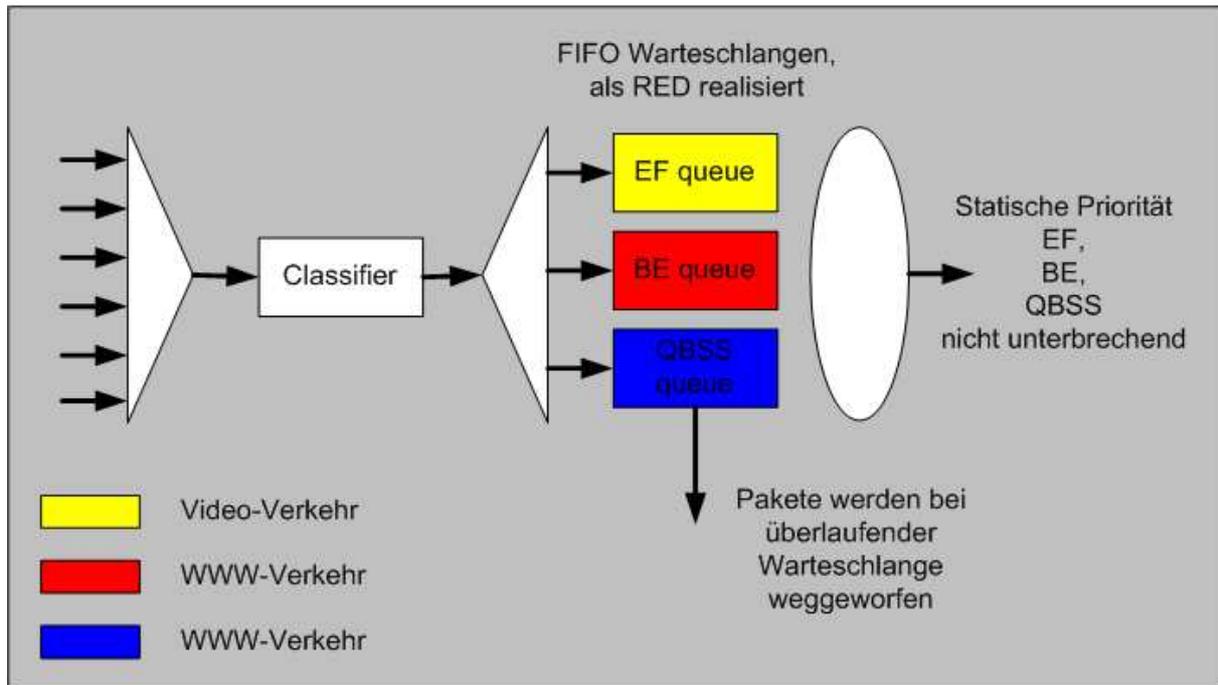


Abbildung 9.5: DiffServ Core-Knoten Architektur nach dem Olympic Ansatz

Abbildung 9.6) und durch ein auf den Link der Bandbreite  $\gamma$  abgehendes Paket mit der maximalen Paketgröße aller bedienten Klassen  $MTU$  (Maximum Transmission Unit), das bis zum vollständigen Versenden  $\frac{MTU}{\gamma}$  Zeit benötigt, verzögert. Letzteres ist der Fall, da die Bedienung durch den Scheduler nicht unterbrechend ist. Zusätzlich erfährt jedes hochpriorere Paket, das selbst in der Zeit  $\frac{M}{\gamma}$  vollständig auf die Ausgangsleitung gebracht wird, auf den Übertragungsleitungen einen Propagation Delay  $D_{prop}$ . Auf Grund dieser Delays läßt sich für den Olympic Ansatz folgende Worst Case Formel für den maximalen Ende-zu-Ende Delay  $D_{max}$  der hochprioreren Pakete entwickeln:

$$D_{max} = \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{(Q_{max,i} - 1) * MTU_{EF,i}}{\gamma_i} + \frac{MTU_i}{\gamma_i} + \frac{M}{\gamma_i} \right) \quad (9.11)$$

Der minimale Delay  $D_{min}$  enthält keinen Warteschlangendelay und keine Restbedienzeit im Scheduler (siehe Formel 9.12).

$$D_{min} = \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{M}{\gamma_i} \right) \quad (9.12)$$

Der maximale Jitter  $J_{max}$  (Formel 9.13) kann aus der Differenz des maximalen und minimalen Delays unter Berücksichtigung der Differenz ( $\Delta M$ ) der maximalen und minimalen Paketgrößen der Pakete berechnet werden. Bei der maximalen Wartezeit in der Warteschlange muß berücksichtigt werden, daß der

Jitter im Gegensatz zum Ende-zu-Ende Delay nicht über die Gesamtdauer einer Untersuchung betrachtet wird, sondern lediglich in einem vorgegebenen Zeitintervall. Daher muß geprüft werden, ob für die auftretenden Paketgrößen bei der verwendeten Topologie die Warteschlange ganz oder nur zum Teil in diesem Zeitintervall geleert oder gefüllt werden kann und sich somit die maximale Wartezeit verringert.

$$J_{max} = \sum_{i=0}^{n-1} \left( \frac{(Q_{max,i} - 1) * MTU_{EF,i}}{\gamma_i} + \frac{MTU_i}{\gamma_i} + \frac{\Delta M_i}{\gamma_i} \right) \quad (9.13)$$

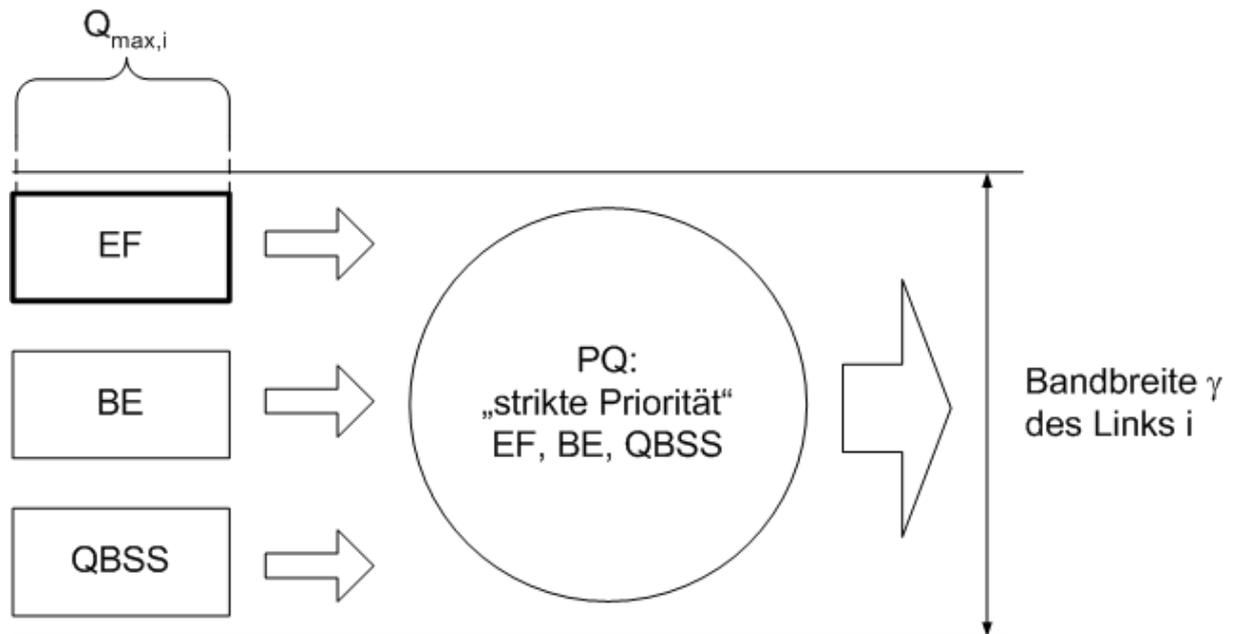
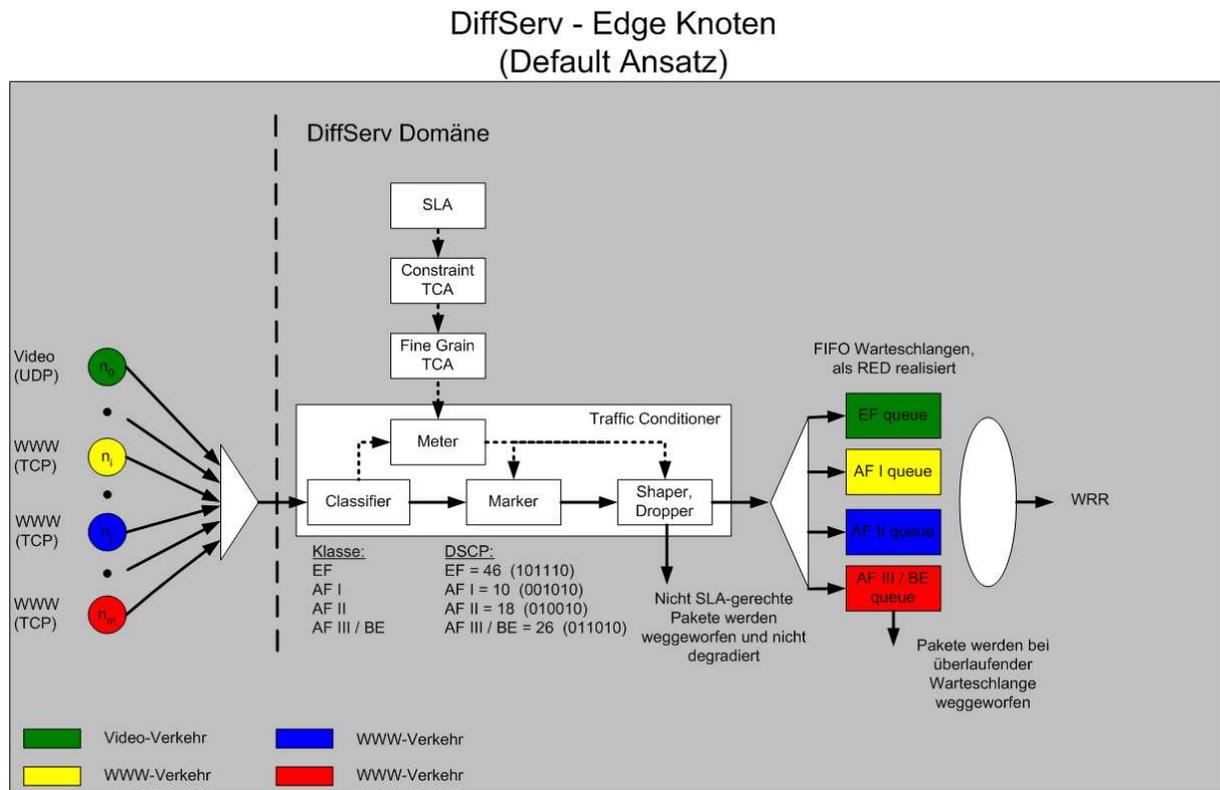


Abbildung 9.6: DiffServ Olympic Ansatz - Scheduling

### 9.1.3.2 Default Ansatz

Der Default Ansatz benutzt zwei PHBs, das EF PHB und das AF PHB. Das AF PHB erfährt hierbei drei unterschiedliche Ausprägungen, die auf drei einzelne Klassen angewendet werden, wie der exakten Beschreibung in Kapitel 2 zu entnehmen ist. Die Abbildungen 9.7 und 9.8 veranschaulichen die Aufteilung in die einzelnen Warteschlangen der somit vier Klassen im Rahmen der DiffServ Architektur.



**Abbildung 9.7:** DiffServ Edge-Knoten Architektur nach dem Default Ansatz

## DiffServ - Core Knoten (Default Ansatz)

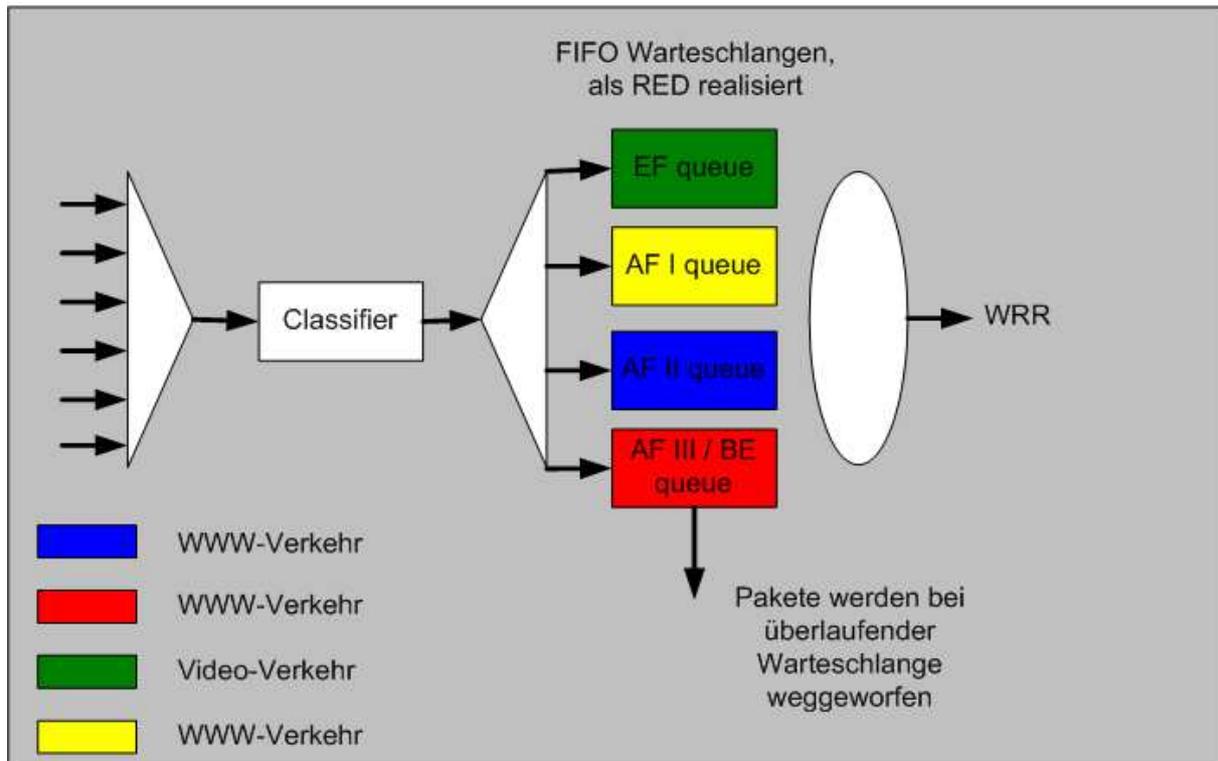


Abbildung 9.8: DiffServ Core-Knoten Architektur nach dem Default Ansatz

Im Unterschied zum Olympic Ansatz verwendet der Default Ansatz die Weighted Round Robin (WRR, [KESHAV 98]) Bedienstrategie (siehe Abbildung 9.9). Beim WRR bekommt jede Klasse auf Grund ihrer Priorität Anteile (Gewichte) einer Schedulerrunde zugeteilt. Somit werden Klassen mit höheren Gewichten öfter vom Scheduler bedient. Hier entspricht ein Anteil einem Paket (nicht einem Bit) das pro Schedulerrunde übertragen werden darf. Die Reihenfolge der Bearbeitung der Anteile ist implementierungsabhängig. Es besteht die Möglichkeit alle Anteile einer Klasse sequentiell nach allen Anteilen einer anderen Klasse zu bedienen oder jeweils nach einem bedienten Anteil die zu bedienende Klasse zu wechseln.

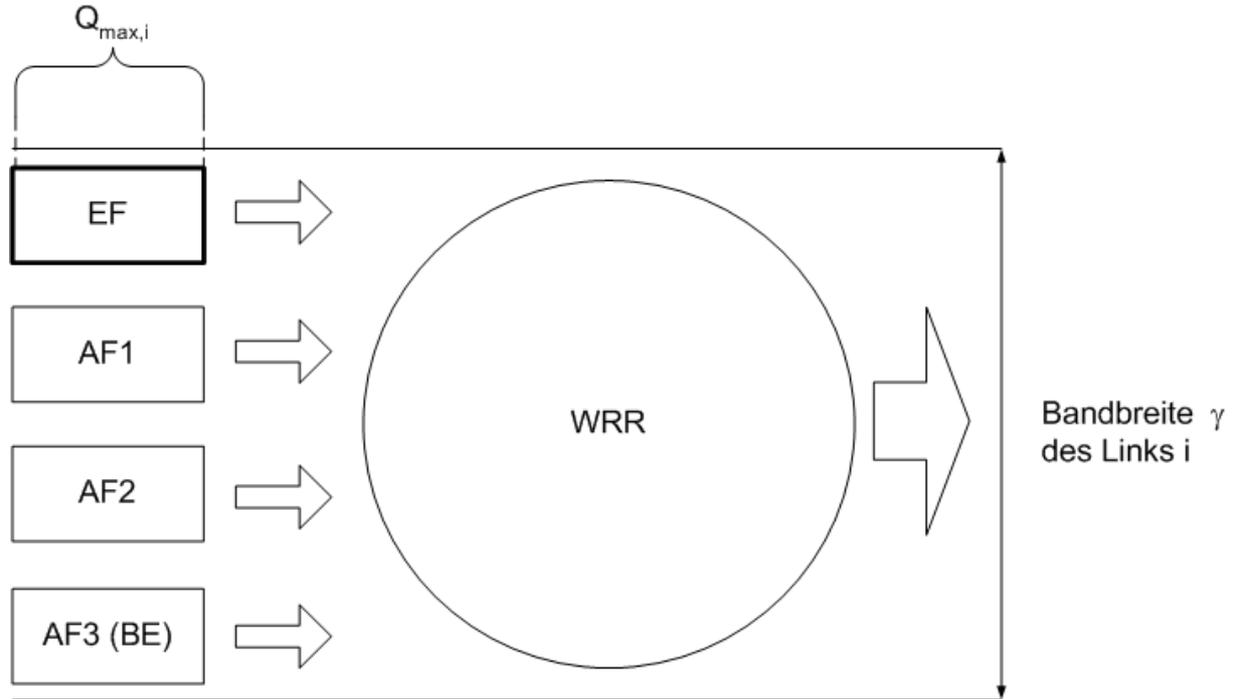


Abbildung 9.9: DiffServ Default Ansatz - Scheduling

Die Worst Case Ende-zu-Ende Delay Formel, die sich hieraus für den Default Ansatz ergibt, entspricht bis auf die Restbedienzeit der des Olympic Ansatzes. Es müssen im Worst Case Fall nicht ein Paket sondern alle Anteile (Pakete) aller AF Klassen ( $AF1$ ,  $AF2$  und  $AF3$ ), die maximal in einer Reihe vor dem nächsten EF Paket bedient werden können, für die Restbedienzeit berücksichtigt werden. Diese können für die Pakete einer AF Klasse jeweils deren maximale Paketgröße ( $MTU_{AF1}$ ,  $MTU_{AF2}$  und  $MTU_{AF3}$ ) haben. Für den minimalen Ende-zu-Ende Delay und den Jitter gilt wiederum dasselbe wie beim Olympic Ansatz, woraus sich folgende Best und Worst Case Formeln für den Ende-zu-Ende Delay und den Jitter ergeben:

$$\begin{aligned}
 D_{max} = & \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{(Q_{max,i} - 1) * MTU_{EF,i}}{\gamma_i} \right) + \\
 & \sum_{i=0}^{n-1} \left( \frac{(MTU_{AF1,i} * AF1_i) + (MTU_{AF2,i} * AF2_i) + (MTU_{AF3,i} * AF3_i)}{\gamma_i} \right) + \\
 & \sum_{i=0}^{n-1} \left( \frac{M}{\gamma_i} \right)
 \end{aligned} \tag{9.14}$$

$$D_{min} = \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{M}{\gamma_i} \right) \tag{9.15}$$

$$\begin{aligned}
J_{max} = & \sum_{i=0}^{n-1} \left( \frac{(Q_{max,i} - 1) * MTU_{EF,i}}{\gamma_i} \right) + \\
& \sum_{i=0}^{n-1} \left( \frac{(MTU_{AF1,i} * AF1_i) + (MTU_{AF2,i} * AF2_i) + (MTU_{AF3,i} * AF3_i)}{\gamma_i} \right) + \\
& \sum_{i=0}^{n-1} \left( \frac{\Delta M_i}{\gamma_i} \right)
\end{aligned} \tag{9.16}$$

## 9.2 Stochastisches Verfahren

Ziel dieses stochastischen Ansatzes ist es, durch Extrapolation von Simulations- oder Messergebnissen eines Ausgangsszenarios (Parameterindex A) den stochastischen Mittelwert des Ende-zu-Ende Delays für ein ähnliches Zielszenario (Parameterindex Z) zu gewinnen. Der mittlere Ende-zu-Ende Delay  $E[D_{e2e}]$  setzt sich aus den mittleren Ende-zu-Ende Delays  $E[D_Z]_i$  der betrachteten Pakete in den  $n$  einzelnen Netzwerkelementen (Knoten + Ausgangslink) entlang des untersuchten Pfades der betrachteten Pakete zusammen. Der Index der Netzwerkelemente gibt die Position entlang des Sendepfades an. Der mittlere Ende-zu-Ende Delay für ein Netzwerkelement  $i$  des Zielszenarios berechnet sich aus der mittleren Wartezeit  $E[W_Z]_i$ , der mittleren Bedienzeit  $E[B_Z]_i$  und dem Propagation Delay  $D_{prop,Z,i}$ :

$$E[D_{e2e}] = \sum_{i=0}^{n-1} E[D_Z]_i = \sum_{i=0}^{n-1} (E[W_Z]_i + E[B_Z]_i + D_{prop,Z,i}) \tag{9.17}$$

Um diese Formel anwenden zu können, muß zunächst eine geeignete Formel zur Wartezeitberechnung gefunden werden.

### 9.2.1 Wartezeitformel

Ziel ist es, für die Delayberechnung eine Wartezeitformel zu finden. Probleme entstehen hierbei dadurch, daß man weder über den Ankunfts- noch über den Bedienprozess Aussagen über z.B.: Streuung und Korrelation der Zwischenankunfts-/Bedienzeiten treffen kann. Daher muß versucht werden eine GI/G/1 Wartezeitformel zu finden. Aus diesem Grund werden zunächst Untersuchungen der Wartezeit in einzelnen M/G/1 [JESSEN 01, KLEINROCK 75, ALLEN 90] und GI/G/1 Modellen mit Hilfe eines Single Queue Simulators für verschiedene Verteilungen für den Ankunfts- als auch den Bedienprozess durchgeführt. Diese Simulationsergebnisse werden mit den Ergebnissen verglichen, die mit Hilfe dreier bereits bekannter Formeln, der M/G/1, der Krämer [ALLEN 90, STOYAN 83] und der Heavy Traffic Formel [ALLEN 90] erzielt werden können. Diese Formeln setzen voraus, daß die Bedien- und die Zwischenankunftszeit unabhängig sind und beide Prozesse ohne Gedächtnis durchlaufen.

Die erste Formel ist die Wartezeitformel für M/G/1 Systeme (siehe Formel 9.18) die sogenannte Pollaczek-Khintchine Mittelwertsformel, die exakte Ergebnisse für M/G/1-Systeme liefert und daher zur Validierung des Simulators dient.  $\rho$  ist hierbei die Auslastung des Systems,  $C_B^2$  der Variationskoeffizient der Bedienzeit und  $E(B)$  die Bedienzeit.

$$E[W] = \frac{\rho \cdot E(B)}{2 \cdot (1 - \rho)} \cdot (1 + C_B^2) \tag{9.18}$$

Auf dieser Formel basieren die beiden Ansätze zur Abschätzung der mittleren Wartezeiten für GI/G/1 Systeme. Die beiden Annäherungsformeln sind die Heavy-Traffic Formel (siehe Formel 9.19) und die Formel von Krämer und Langenbach-Belz (siehe Formel 9.20). Die Krämer Formel reduziert sich bei einem Variationskoeffizienten des Ankunftsprozess gleich 1 auf die Pollaczek-Khintchine Formel. Die Heavy Traffic Formel, soll wie der Name schon sagt, lediglich für hohe Auslastungen geeignet sein, im Gegensatz zur Krämer Formel, die Annäherungen gleicher Güte für sämtliche Auslastungsbereiche liefern soll.

Heavy Traffic:

$$E[W] \approx \frac{\rho \cdot E[B]}{2 \cdot (1 - \rho)} \cdot \left( \frac{C_A^2}{\rho^2} + C_B^2 \right) \quad (9.19)$$

Krämer:

$$E[W] \approx \frac{\rho \cdot E[B]}{2 \cdot (1 - \rho)} \cdot (C_A^2 + C_B^2) \cdot f(\rho, C_A^2, C_B^2)$$

$$f(\rho, C_A^2, C_B^2) = \begin{cases} e^{-\frac{(1-\rho) \cdot \frac{C_A^2 - 1}{C_A^2 + 4 \cdot C_B^2}}{e}}, & C_A^2 \geq 1 \\ e^{-\frac{2 \cdot (1-\rho) \cdot (1 - C_A^2)^2}{3 \cdot \rho \cdot (C_A^2 + C_B^2)}}, & C_A^2 < 1 \end{cases}, \quad (9.20)$$

Es sei im Folgenden:

$E[W]$	mittlere Wartezeit
$\rho$	Auslastung
$E[B]$	mittlere Bedienzeit
$C_A^2$	Variationskoeffizient des Ankunftsprozesses
$C_B^2$	Variationskoeffizient des Bedienprozesses

**Tabelle 9.2:** Parameter der Wartezeitformeln

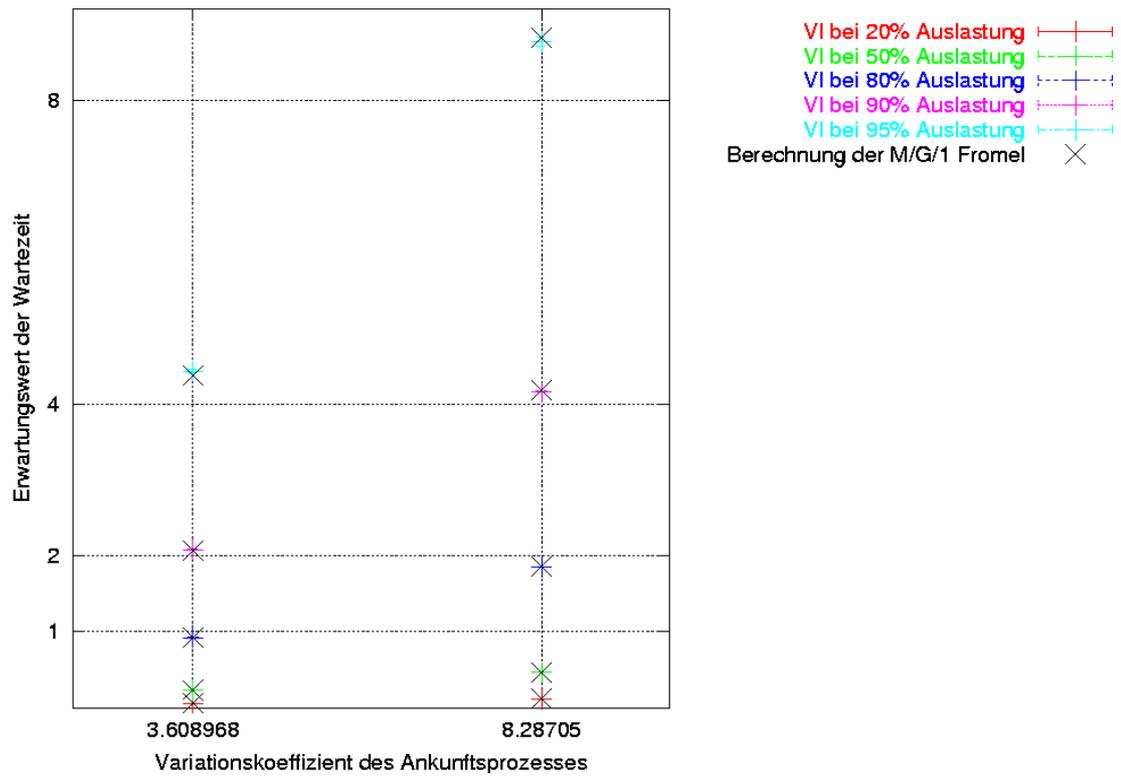
Um die Grenzen für die Gültigkeitsbereiche der gewählten Formeln testen zu können, werden für ausgesuchte Verteilungen für den Ankunfts- und Bedienprozess die Formelberechnungen mit Simulationsläufen des Single Queue Simulators, entwickelt von Hans-Peter Schwefel [LIPSKY ET AL.], verglichen. Der Simulator erlaubt es unter anderem, ein Ein-Stationen-Wartesystem bestehend aus einer Bedieneinheit mit einer unendlichen Warteschlangenkapazität und frei wählbaren Verteilungen für die Zwischenankunfts- und Bedienzeiten, zu simulieren. Hierbei können für den Ankunfts- und Bedienprozess folgende Verteilungen gewählt werden:

- Pareto Verteilung
- Negativ Exponentielle Verteilung
- Truncated Powertail Verteilung
- Erlang-k Verteilung
- Hyperexponentiell-2 Verteilung
- Uniforme Verteilung
- Deterministische Verteilung

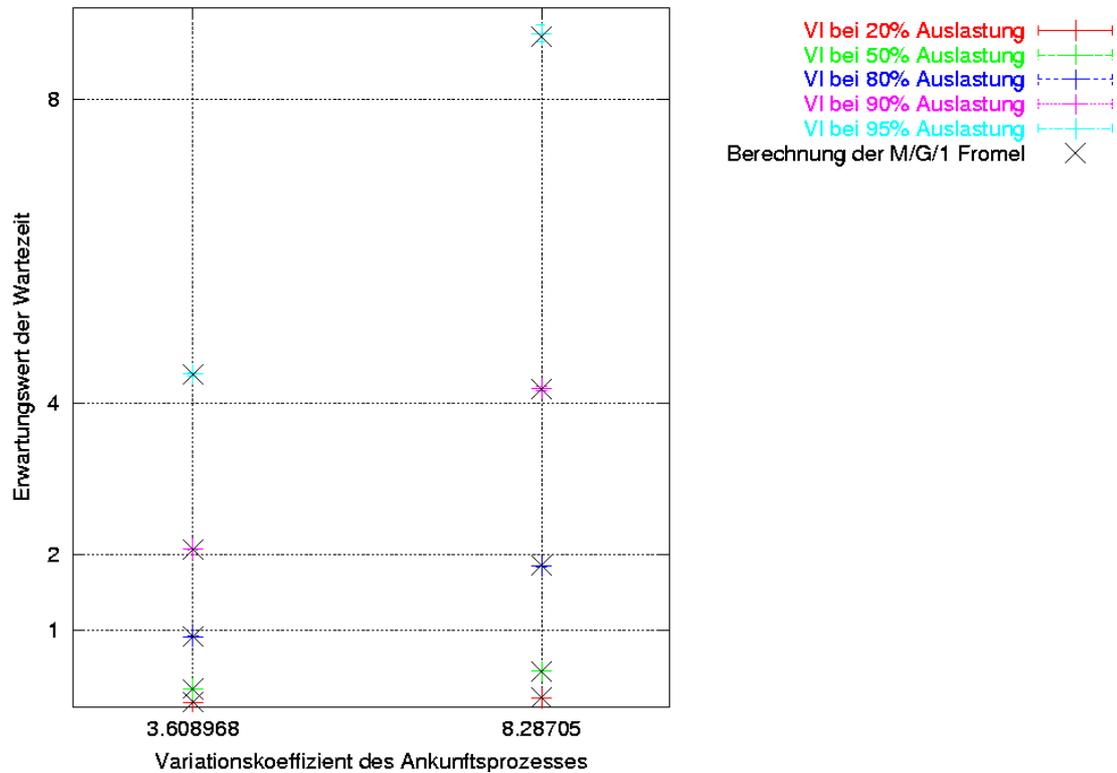
Zunächst wird die Gültigkeit des Simulators untersucht, in dem Simulationen mit einer negativ exponentiellen Verteilung (kurz: nexp) für den Ankunftsprozess und einer hyperexponentiellen (kurz: hyp2) und einer Truncated Powertail Verteilung (kurz: tpt) für den Bedienprozess mit den in Tabelle 9.3 angegebenen Variationskoeffizienten durchgeführt werden. Eine exakte Beschreibung der Truncated Powertail Verteilung ist [LIPSKY ET AL.] zu entnehmen. Die nicht ganzzahligen Werte der Variationskoeffizienten entstehen durch die Auswahl eines ganzzahligen  $T$  Wertes für die Powertail Verteilung. Für alle Simulationen werden die mittleren Wartezeiten (und die Standardabweichung) gemessen und über 20 Simulationsläufe das 90 % Vertrauensintervall gebildet. Es werden Auslastungen von 20 %, 50 %, 80 %, 90 % und 95 % untersucht. Es werden in allen Simulationsläufen  $10^7$  Pakete bedient. Es wird verglichen, ob die mit Hilfe der Pollaczek-Khintchine Formel (Formel 9.18) exakt berechneten Werte innerhalb der durch Simulation gewonnenen Vertrauensintervalle für die mittlere Wartezeit liegen und die Ergebnisse in den Abbildungen 9.10 und 9.11 dargestellt.

Verteilung	Parameter	Werte
nexp-hyp2	$C_A^2 / C_B^2$	1/3.608968, 1/8.28705
nexp-tpt	$C_A^2 / C_B^2 / T$	1/3.608968/7, 1/8.28705/10

**Tabelle 9.3:** Verteilungen und Variation der Variationskoeffizienten innerhalb der M/G/1 Untersuchungen



**Abbildung 9.10:** Vergleich der Berechnungsergebnisse der M/G/1 Formel mit den Simulationen des Single Queue Simulators bei der Verteilungskombination nex-p-hyp2



**Abbildung 9.11:** Vergleich der Berechnungsergebnisse der M/G/1 Formel mit den Simulationen des Single Queue Simulators bei der Verteilungskombination nexpt-tp

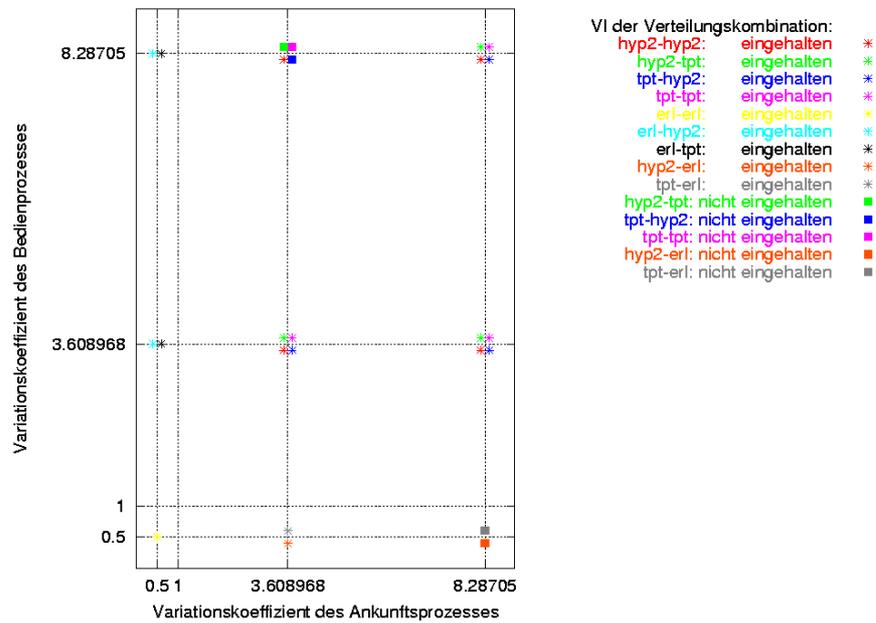
Wie sich aus diesen Abbildungen erkennen lässt, entsprechen die Simulationsergebnisse sowohl für die Kombination nexpt-hyp2, als auch für nexpt-tp den exakten Berechnungen der M/G/1 Formel und bestätigen somit die korrekte Arbeitsweise des Simulators.

Anschließend untersuchen wir folgende Verteilungskombinationen und Variationen der Variationskoeffizienten (siehe Tabelle 9.4, zusätzlich untersuchte Verteilung: Erlang Verteilung (kurz: erl)) der Krämer und Heavy Traffic Formel für die Abschätzung von GI/G/1 Warteschlangensystemen. Zusätzlich wird ein weiterer Ansatz basierend auf [WHITT 88], ab jetzt mit Whitt Ansatz bezeichnet, mit den beiden anderen Formeln verglichen. Dieser Ansatz verwendet für die Krämer Formel für den Fall, daß der Variationskoeffizient des Ankunftsprozesses größer als 1 ist ( $C_A^2 > 1$ ), den Wert 1 für die f-Funktion der Formel. Hier wird die Verwendbarkeit des Ansatzes auch für  $C_A^2 < 1$  getestet. Dies bedeutet, daß die Reduktion der Wartezeit durch die f-Funktion (Wertebereich ]0;1]), die bei dominantem  $C_A^2$  stärker und bei dominantem  $C_B^2$  weniger als bei identischen Variationskoeffizienten reduziert, wegfällt. Für alle Simulationen von Tabelle 9.4 werden die mittleren Wartezeiten (und die Standardabweichung) gemessen und über 20 Simulationsläufe das 90 % Vertrauensintervall gebildet. Es werden Auslastungen von 20 %, 50 %, 80 %, 90 % und 95 % untersucht. Es werden in allen Simulationsläufen  $10^7$  Pakete bedient. Es wird verglichen, ob die mit Hilfe der Näherungsformeln (Formeln 9.19 und 9.20) berechneten Werte innerhalb der durch Simulation gewonnenen Vertrauensintervalle für die mittleren Wartezeiten liegen und die Ergebnisse der Krämer Formel und des Whitt Ansatzes in den Abbildungen 9.12 - 9.16 und die Ergebnisse der Heavy Traffic Formel in den Abbildungen B.1 - B.5 im Anhang B dargestellt. Zusätzlich wurden Untersuchungen anderer Variationen der Variationskoeffizienten durchgeführt. Da

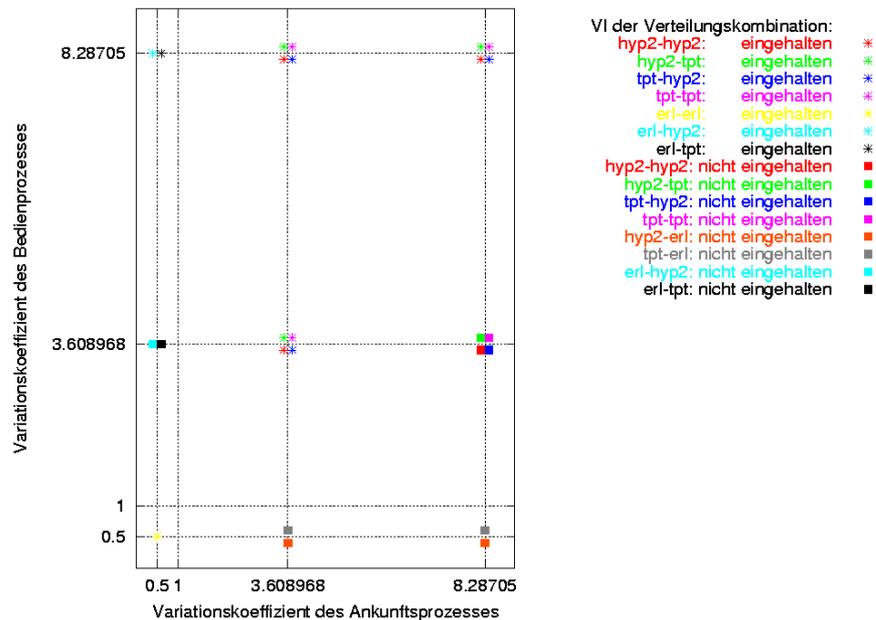
diese Ergebnisse den in den folgenden Abbildungen dargestellten entsprechen, und zur Bewahrung der Übersichtlichkeit, sind die untersuchten Variationen und deren Ergebnisse Anhang B zu entnehmen. Diese zeigen die gemittelten Simulationsergebnisse, die Berechnungsergebnisse und die prozentuale Abweichung dieser Ergebnisse voneinander.

Verteilungs-kombination	Parameter	Werte
hyp2-hyp2	$C_A^2 / C_B^2$	3.608968/3.608968, 8.28705/8.28705, 3.608968/8.28705, 8.28705/3.608968
hyp2-tpt	$C_A^2 / C_B^2 / T$	3.608968/3.608968/7, 8.28705/8.28705/10, 3.608968/8.28705/10, 8.28705/3.608968/7
tpt-hyp2	$T / C_A^2 / C_B^2$	7/3.608968/3.608968, 10/8.28705/8.28705, 7/3.608968/8.28705, 10/8.28705/3.608968
tpt-tpt	$T / C_A^2 / C_B^2 / T$	7/3.608968/3.608968/7, 10/8.28705/8.28705/10, 7/3.608968/8.28705/10, 10/8.28705/3.608968/7
erl-hyp2	$C_A^2 / C_B^2$	0.5/3.608968, 0.5/8.28705
erl-tpt	$C_A^2 / C_B^2 / T$	0.5/3.608968/7, 0.5/8.28705/10
hyp2-erl	$C_A^2 / C_B^2$	3.608968/0.5, 8.28705/0.5
tpt-erl	$T / C_A^2 / C_B^2$	7/3.608968/0.5, 10/8.28705/0.5

**Tabelle 9.4:** Verteilungen und Variation der Variationskoeffizienten innerhalb der GI/G/1 Untersuchungen

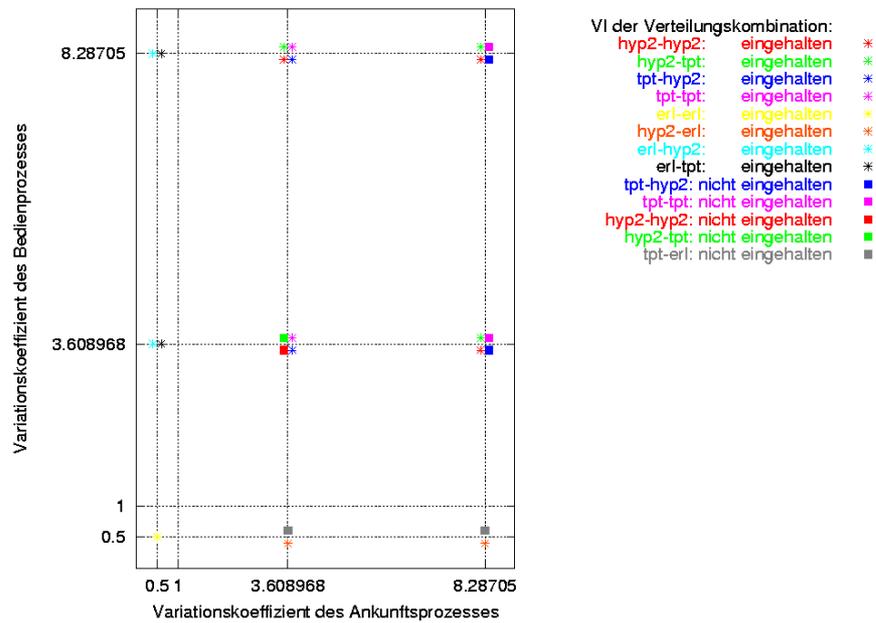


(a) Berechnungen mit der Krämer Formel

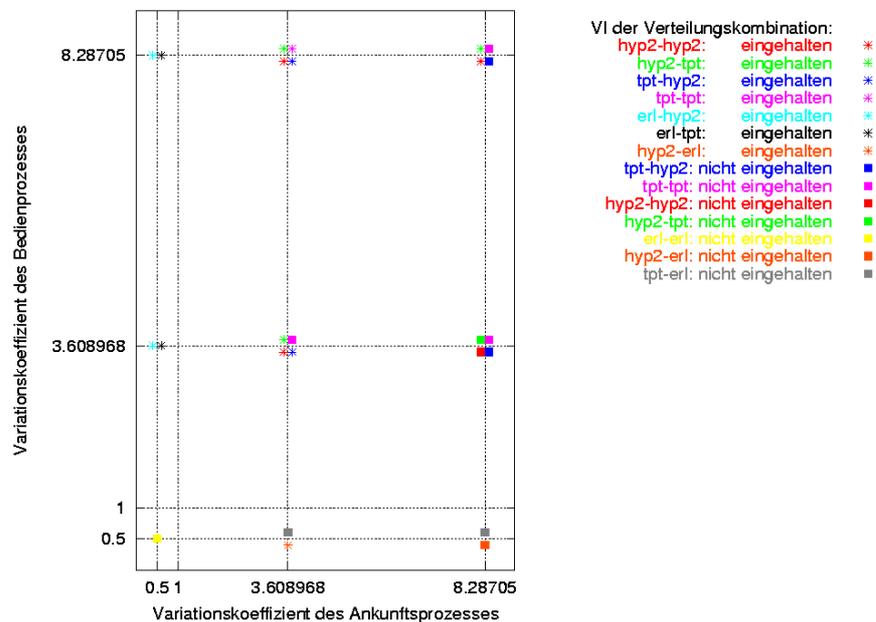


(b) Berechnungen mit dem Whitt Ansatz

**Abbildung 9.12:** Bewertung der Krämer Formel und des Whitt Ansatzes bei 20 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen

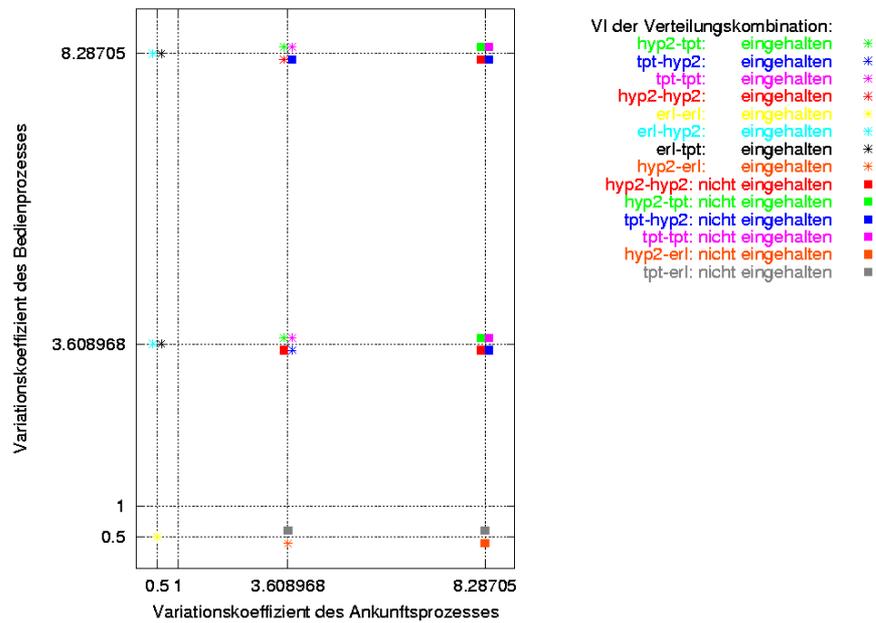


(a) Berechnungen mit der Krämer Formel

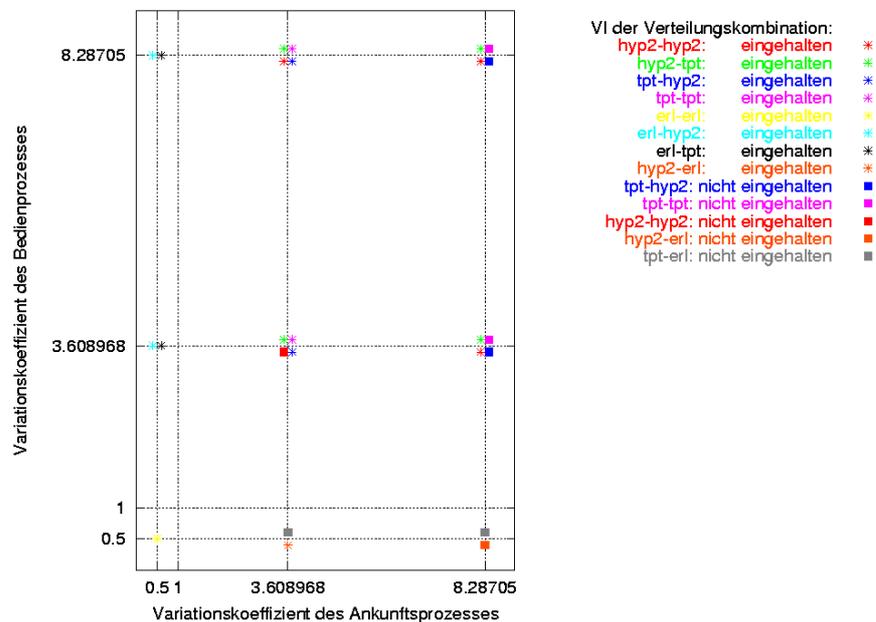


(b) Berechnungen mit dem Whitt Ansatz

**Abbildung 9.13:** Bewertung der Krämer Formel und des Whitt Ansatzes bei 50 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen

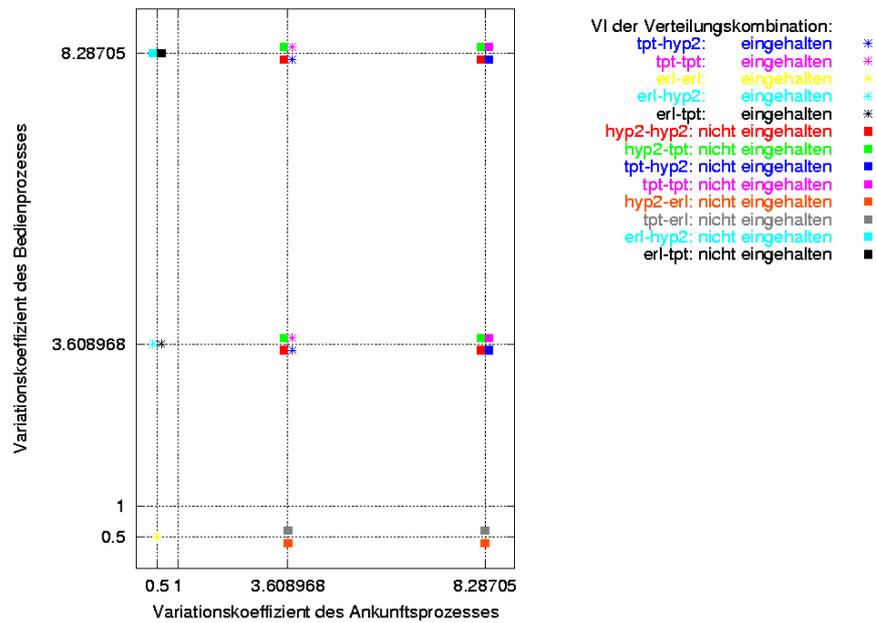


(a) Berechnungen mit der Krämer Formel

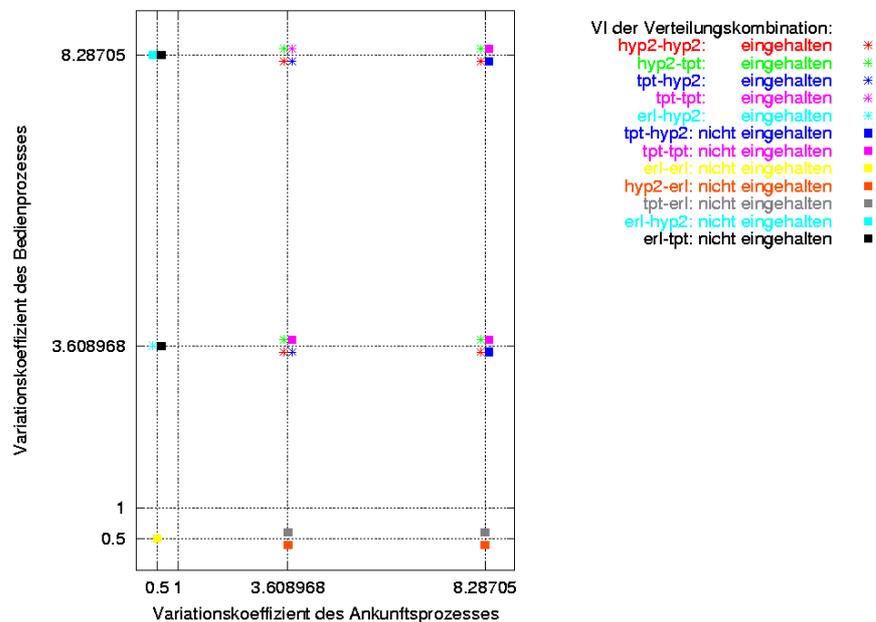


(b) Berechnungen mit dem Whitt Ansatz

**Abbildung 9.14:** Bewertung der Krämer Formel und des Whitt Ansatzes bei 80 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen

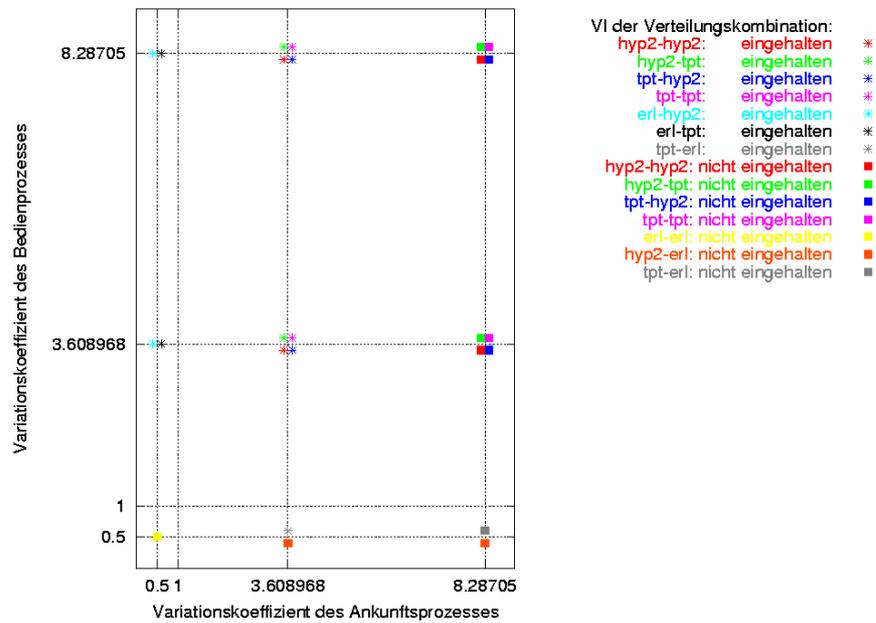


(a) Berechnungen mit der Krämer Formel

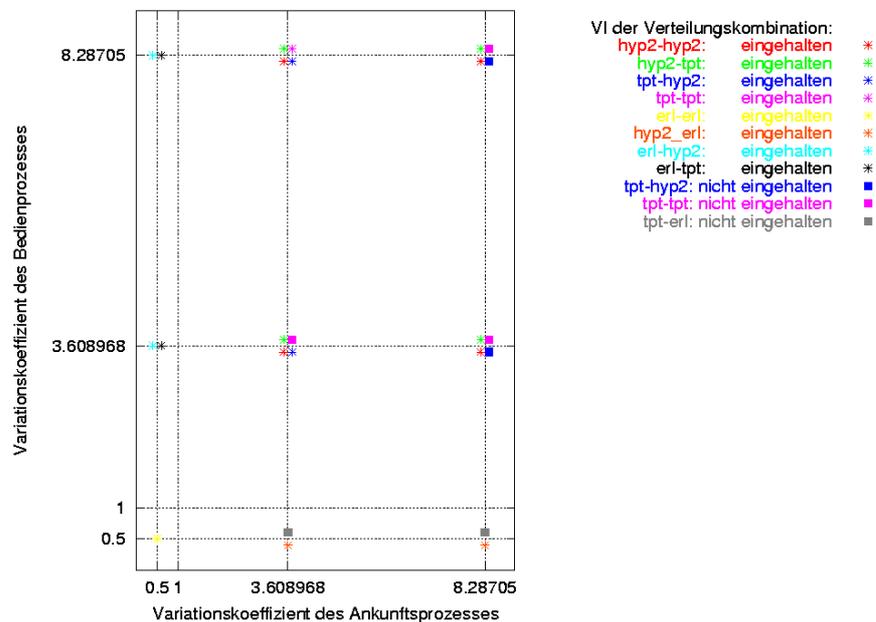


(b) Berechnungen mit dem Whitt Ansatz

**Abbildung 9.15:** Bewertung der Krämer Formel und des Whitt Ansatzes bei 90 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen



(a) Berechnungen mit der Krämer Formel



(b) Berechnungen mit dem Whitt Ansatz

**Abbildung 9.16:** Bewertung der Krämer Formel und des Whitt Ansatzes bei 95 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen

Die durchgeführten Simulationen führen zu folgenden Ergebnissen:

Die Heavy Traffic Formel eignet sich, wie auch vorgesehen, lediglich für sehr hohe Auslastungsbereiche, da für  $\rho \rightarrow 0$ ,  $E[W] \rightarrow \infty$  folgt. Selbst bei Auslastungen von 95 % liegen die Ergebnisse aber lediglich in zwei Fällen innerhalb der angestrebten Vertrauensintervalle und dies für einen Variationskoeffizienten des Ankunftsprozesses  $C_A^2 < 1$ .

Die beiden anderen Formeln, die Krämer Formel und der Whitt Ansatz, schneiden im Vergleich bei den unterschiedlichen Verhältnissen der Variationskoeffizienten zueinander und unterschiedlichen Verteilungskombinationen für den Ankunfts- und Bedienprozess wie folgt ab:

- Abhängigkeit vom Verhältnis der Variationskoeffizienten zueinander:
 

Bei beiden Formeln ist eine deutliche Abhängigkeit der Exaktheit der Berechnungen von dem Verhältnis der Variationskoeffizienten zueinander auszumachen, die sich mit steigenden Auslastungen noch intensiviert.

  - $C_A^2 \gg C_B^2$  ( $C_A^2$  ist dominant):  
Bei dominantem  $C_A^2$  steigen die Abweichungen von den simulierten Werten deutlich an.
  - $C_A^2 \ll C_B^2$  ( $C_B^2$  ist dominant): Bei dominantem  $C_B^2$  ist keine Veränderung der Ergebnisse feststellbar.
- Abhängigkeit von der Verteilungskombination:
 

Die Dominanz von  $C_A^2$  spiegelt sich auch in der Abhängigkeit von der Verteilungskombination wieder. Eine Variation der Verteilung für den Bedienprozess bei unveränderter Verteilung für den Ankunftsprozess hat keine gravierende Veränderung der durch Simulation gewonnenen Vertrauensintervalle zur Folge. Bei entsprechender (nächstem Punkt zu entnehmen) Verteilungskombination und Variationskoeffizienten für den Ankunftsprozess liefern beide Formeln stets korrekte Berechnungen.

Eine Variation der Verteilung des Ankunftsprozesses führt für beide Formeln zu folgenden Ergebnissen:

  - Erlang-k Verteilung für den Ankunftsprozess ( $C_A^2 < 1$ ):  
Für  $C_A^2 < 1$  liefert die Krämer Formel Ergebnisse innerhalb der geforderten Vertrauensintervalle. In diesem Bereich ist sie dem Whitt Ansatz überlegen, der die Vertrauensintervalle mehrmals (knapp) verfehlt. Dies ist allerdings durch die eigene Adaption des Ansatzes im Bereich  $C_A^2 < 1$  begründet. Ohne diese Adaption wären die Formeln und somit die erzielten Ergebnisse für  $C_A^2 < 1$  identisch.
  - Truncated Powertail und hyperexponentielle Verteilung für den Ankunftsprozess ( $C_A^2 > 1$ ):  
Bei einer Variation der Verteilungskombination für  $C_A^2 > 1$  stellt sich heraus, daß die Ergebnisse beider Formeln bereits ab Auslastungen von 50 % die Vertrauensintervalle bei einer verwendeten Truncated Powertail Verteilung (unabhängig von der Größe des Variationskoeffizienten des Ankunftsprozesses) nicht mehr erfüllen. Eine zu kurze Simulationsdauer bzw. eine zu geringe Anzahl untersuchter Pakete als Grund für die schlechten Ergebnisse wegen der besonderen Eigenschaften einer Truncated Powertail Verteilung konnte durch Simulationen mit einer erhöhten Anzahl von Paketen ausgeschlossen werden. Bei einer verwendeten hyperexponentiellen Verteilung liefert der Whitt Ansatz fast durchwegs korrekte Werte, wohingegen die Ergebnisse der Krämer Formel bereits ab einer Auslastung von 20 % Abweichungen aufweisen. Die Größe des Variationskoeffizienten spielt hier erst ab extremen (nicht mehr relevanten) Werten von ca. 289 eine Rolle. Ab diesem Wert weichen ebenfalls die Ergebnisse des Whitt Ansatzes erheblich von den geforderten Vertrauensintervallen ab.

Fazit:

Die Untersuchung der drei Formeln zeigt, daß keine Formel für alle Verteilungskombinationen und Variationskoeffizienten geeignet ist. Die Heavy Traffic Formel wird auf Grund der schlechten Resultate in allen Testbereichen für die Extrapolation nicht eingesetzt. Die beiden anderen Formeln berücksichtigen beide ein dominantes  $C_A^2$  zu wenig und liefern hierfür unzureichende Ergebnisse. Für den Whitt Ansatz war dies auf Grund der Gleichbehandlung der Variationskoeffizienten in der Formel zu erwarten. Für die Krämer Formel sollte dies eigentlich durch die f-Funktion in die Berechnung mit einfließen. Die Krämer Formel hat ihre Vorteile bei  $C_A^2 < 1$ , hat aber Schwächen bei  $C_A^2 > 1$  sowohl für eine hyperexponentielle, als auch eine Truncated Powertail Verteilung für den Ankunftsprozess, da auch hier die f-Funktion eine Reduktion durchführt, die zu einem zu geringen Wartezeitergebnis führt. Der Whitt Ansatz zeigt Abweichungen für  $C_A^2 < 1$ , die jedoch minimal waren und kam am besten mit Werten von  $C_A^2 > 1$  zu recht. Außerdem spricht die Einfachheit für den Whitt Ansatz, die für den Einsatz in dem zu entwickelten stochastischen Extrapolationsverfahren mitentscheidend ist. Daher wird der Whitt Ansatz für die weiteren Untersuchungen verwendet.

Allgemein müssen bei dem folgenden Einsatz des Whitt Ansatzes innerhalb des stochastischen Verfahrens zur Extrapolation folgende Punkte berücksichtigt werden:

- Es wurden nur Ein-Stationen-Wartesysteme und keine Ketten von Wartesystemen (Pfad durch ein Netzwerk) untersucht. Bei der Berechnung der Wartezeit eines Knotens auf einem Pfad durch ein Netzwerk hängt dessen Ankunftsprozess vom Bedienprozess des Vorgängerknotens oder der Vorgängerknoten ab. D.h.  $C_A^2$  nähert sich um so mehr  $C_B^2$  am Vorgängerknoten an, je höher die Auslastung ist. Der Ankunftsprozess ist im Grenzfall der Bedienprozess des Vorgängerknotens. Damit ist  $C_A^2$  am Pfadeneingang (erster Knoten des Pfades, Quelle) anders als innerhalb des Pfades. Zusätzlich treten bei mehreren Vorgängerknoten Überlagerungen auf, die ebenfalls Veränderungen hervorrufen können. Hierzu gibt es ebenfalls in [WHITT 88] einen Ansatz, um diese Abhängigkeit mit in GI/G/1 Formeln einfließen zu lassen, der jedoch auf Grund seiner Komplexität für den Einsatz innerhalb einer Extrapolation nicht in Frage kommt.
- Es wurde ein Ein-Stationen-Wartesystem mit einer Bedieneinheit mit unendlicher Warteschlangenkapazität untersucht. Bei dem Einsatz im stochastischen Extrapolationsverfahren werden die Warteschlangenkapazitäten des Zielszenarios aber begrenzt sein.

## 9.2.2 Extrapolation mit stochastischen Modellen

Basierend auf den vorangegangenen Untersuchungen wird für das stochastische Extrapolationsverfahren der Whitt Ansatz als Wartezeitformel eingesetzt. Im Folgenden wird die Untersuchung der Extrapolation mit stochastischen Modellen wiederum in die drei Freiheitsgrade und die 8 sich daraus ergebenden Extrapolationsfälle unterteilt. Das Ausgangsszenario bildet ein Netzwerkelement das BE verwendet.

Für ein einzelnes, beliebiges Netzwerkelement führt der Whitt Ansatz eingesetzt in Formel 9.17 für ein Zielszenario, das ebenso wie das Ausgangsszenario den BE Dienst verwendet, zur Delay Formel 9.21 und der Restbedienzeitformel 9.22. Da das Ausgangs- und Zielszenario für BE in diesem simplen Fall identisch sind, können die Zielszenarioparameter durch die Ausgangsszenarioparameter ersetzt werden.

$$\begin{aligned}
E[D_Z]_{be} &= E[W_Z] + E[B_Z] + D_{prop,Z} \\
&= \frac{\rho_Z}{1 - \rho_Z} \cdot E[B_Z] \cdot \frac{C_{A,Z}^2 + C_{B,Z}^2}{2} + E[B_Z] + D_{prop,Z}
\end{aligned} \tag{9.21}$$

$$\begin{aligned}
E[RBZ_Z]_{be} &= E[B_Z] \cdot \frac{C_{A,Z}^2 + C_{B,Z}^2}{2} = E[B_A] \cdot \frac{C_{A,A}^2 + C_{B,A}^2}{2} \\
&= \frac{1 - \rho_A}{\rho_A} \cdot (E[D_A]_{be} - D_{prop,A} - E[B_A])
\end{aligned} \tag{9.22}$$

Dies ist der erste der acht Extrapolationsfälle. Die sieben weiteren, interessanten Fälle werden in der Folge untersucht.

### **Variation der QoS Technologie (Fall 2, siehe Kapitel 8):**

Wird im Zielszenario die QoS Technologie geändert, muß die Berechnung der Restbedienzeiten auf Grund der abweichenden Bedienstrategien (siehe Abschnitt 9.1) angepaßt werden. Die Formeln 9.23 und 9.24 zeigen die unterschiedlichen Restbedienzeitformeln für die beiden DiffServ Ansätze (Olympic, Default). Problematisch ist jedoch, daß die DiffServ Restbedienzeitformeln auf M/G/1 Prioritätsformeln ([JESSEN 01, JAISWAL 68]) beruhen und daher für GI/G/1 Systeme ungeeignet sind. Die Exaktheit der Berechnungen hängt somit direkt von der Charakteristik des Ankunftsprozesses des untersuchten Verkehrs ab. IntServ wird wiederum aus den Untersuchungen ausgeschlossen, da der gewünschte maximale Delay bei IntServ zur Einstellung der Bedienelemente vorgegeben wird. Die Bedeutung der einzelnen Parameter ist Anhang A zu entnehmen.

$$\begin{aligned}
E[RBZ_Z]_{olympic} &= \rho_{A,ef} \cdot \frac{1 - \rho_{A,tcp}}{\rho_{A,tcp}} \cdot (E[D_{A,tcp}]_{be} - D_{prop,A} - E[B_{A,tcp}]) \\
&\quad + (\rho_{A,be} + \rho_{A,qbss}) \cdot \frac{1 - \rho_{A,ef}}{\rho_{A,ef}} \cdot (E[D_{A,ef}]_{be} - D_{prop,A} \\
&\quad - E[B_{A,ef}])
\end{aligned} \tag{9.23}$$

$$\begin{aligned}
E[RBZ_Z]_{default} &= (\rho_{A,ef} \cdot w_{ef}) \cdot \frac{1 - \rho_{A,tcp}}{\rho_{A,tcp}} \cdot (E[D_{A,tcp}]_{be} - D_{prop,A} - E[B_{A,tcp}]) \\
&\quad + ((\rho_{A,af1} \cdot w_{af1}) + (\rho_{A,af2} \cdot w_{af2}) + (\rho_{A,af3} \cdot w_{af3})) \\
&\quad \cdot \frac{1 - \rho_{A,ef}}{\rho_{A,ef}} \cdot (E[D_{A,ef}]_{be} - D_{prop,A} - E[B_{A,ef}])
\end{aligned} \tag{9.24}$$

Wie man sieht, sind sämtliche Parameter entweder durch die Eigenschaften des Zielszenarios gegeben oder können durch Simulationsergebnisse des Ausgangsszenarios ersetzt werden ( $E[D_A] - D_{prop,A} - E[B_A]$ , siehe Formel 9.22 - 9.24).

**Variation des Lastspektrums (Fall 3, siehe Kapitel 8):**

Der zweite Freiheitsgrad des Zielszenarios betreffend ist das Lastspektrum. Das Lastspektrum kann auf zwei Arten variiert werden, durch die Variation der Verkehrszusammensetzung und durch die Variation der Verkehrsintensität. Für eine Variation der Verkehrsintensität um den Faktor  $p$  sieht die Parameterersetzung wie folgt aus. Hierbei kann sich die Variation der Verkehrsintensität auch auf unterschiedliche Verkehrsarten aufspalten und somit  $p \cdot \rho_Z = (p_1 \cdot \rho_{A,1} + \dots + p_m \cdot \rho_{A,m})$  mit  $m = \text{Anzahl der unterschiedlichen Verkehrstypen}$ , entsprechen.

$$\begin{aligned}
 E[B_Z] &= E[B_A] \\
 \rho_Z &= p \cdot \rho_A \\
 C_{B,Z}^2 &= C_{B,A}^2 \\
 C_{A,Z}^2 &\neq C_{A,A}^2
 \end{aligned}
 \tag{9.25}$$

Für eine Variation der Verkehrsintensität um den Faktor  $p$  ergibt sich folgende Formel für den Erwartungswert des Delays:

$$\begin{aligned}
 E[D_Z]_{be} &= E[W_Z] + E[B_Z] + D_{prop,Z} \\
 &= \frac{\rho_Z}{1 - \rho_Z} \cdot E[B_Z] \cdot \frac{C_{A,Z}^2 + C_{B,Z}^2}{2} + E[B_Z] + D_{prop,Z} \\
 &= \frac{p \cdot \rho_A}{1 - (p \cdot \rho_A)} \cdot E[B_A] \cdot \frac{x \cdot C_{A,A}^2 + C_{B,A}^2}{2} + E[B_A] + D_{prop,Z} \\
 &= \frac{1 - \rho_A}{1 - (p \cdot \rho_A)} \cdot p \cdot \frac{\rho_A}{1 - \rho_A} \cdot E[B_A] \cdot \frac{x \cdot C_{A,A}^2 + C_{B,A}^2}{2} + E[B_A] + D_{prop,Z}
 \end{aligned}
 \tag{9.26}$$

Auf Grund der folgenden Beziehung kann das Ergebnis der vorangegangenen Formel mit Hilfe von sich aus der Topologie des Zielszenarios ergebenden Parametern und mit Hilfe von Simulationsergebnissen des Ausgangsszenarios berechnet werden.

$$\frac{\rho_A}{1 - \rho_A} \cdot E[B_A] \cdot \frac{C_{A,A}^2 + C_{B,A}^2}{2} = E[D_A]_{be} - E[B_A] - D_{prop,A}
 \tag{9.27}$$

Bei dieser Variation kann die Extrapolation allerdings abweichende Ergebnisse liefern, da sich der Variationskoeffizient des Ankunftsprozesses des Zielszenarios  $C_{A,Z}^2$  im Vergleich zum Variationskoeffizienten des Ausgangsszenarios  $C_{A,A}^2$  in unbekannter Weise um den Faktor  $x$  ändert. Eine Extrapolation auf ein Zielszenario mit einer Variation der Verkehrszusammensetzung ist für BE, als auch für die unterschiedlichen DiffServ Ansätze mit den angegebenen Formeln durchführbar, unterliegt allerdings auch den Ungewissheiten der Ergebniserzeugung mit Hilfe von M/G/1 Formeln für GI/G/1 Fälle.

**Variation der Netzkonfiguration (Fall 5, siehe Kapitel 8):**

Eine Extrapolation auf ein Netz mit einer anderen Netzkonfiguration kann man wiederum in drei

Fälle unterteilen, erstens in eine Extrapolation auf ein Zielszenario mit einer anderen Topologie und somit unterschiedlichen Anzahl von Knoten und Kanten, zweitens eine Extrapolation auf ein Zielszenario mit unterschiedlichem Propagationdelay und drittens einer unterschiedlichen Linkkapazität. Eine Veränderung der Anzahl der Knoten und Kanten zeichnet sich lediglich durch eine mögliche Veränderung der Anzahl der Netzwerkelemente entlang des zur Berechnung des Ende-zu-Ende Delays ausgewählten Sendepfades aus. Die Formeln zur Delayberechnung der einzelnen Netzwerkelemente bleiben unverändert. Eine konstante Variation des Propagationdelays aller Links eines untersuchten Pfades bewirkt lediglich, je nach Variationsart, eine Reduktion oder Aufstockung des Ende-zu-Ende Delays. Bei der Variation des Propagationdelays einzelner Links wird jedoch die Auslastung der Links beeinflusst. Dies kann durch die bei der Veränderung der Linkkapazität angegebenen Formeln für alle QoS Technologien untersucht werden. Für eine Veränderung der Linkkapazität um den Faktor  $k$  werden die Zielszenarioparameter wie folgt durch die Ausgangsszenarioparameter präsentiert:

$$\begin{aligned}
 E[B_Z] &= \frac{1}{k} \cdot E[B_A] \\
 \rho_Z &= \frac{1}{k} \cdot \rho_A \\
 C_{B,Z}^2 &= C_{B,A}^2 \\
 C_{A,Z}^2 &= C_{A,A}^2
 \end{aligned} \tag{9.28}$$

Die ersetzten Parameter führen für eine Variation der Linkkapazität um den Faktor  $k$  zu folgenden Delayformeln:

$$\begin{aligned}
 E[D_Z]_{be} &= E[W_Z] + E[B_Z] + D_{prop,Z} \\
 &= \frac{\rho_Z}{1 - \rho_Z} \cdot E[B_Z] \cdot \frac{C_{A,Z}^2 + C_{B,Z}^2}{2} + E[B_Z] + D_{prop,Z} \\
 &= \frac{\frac{\rho_A}{k}}{1 - \frac{\rho_A}{k}} \cdot \frac{1}{k} \cdot E[B_A] \cdot \frac{C_{A,A}^2 + C_{B,A}^2}{2} + \frac{1}{k} \cdot E[B_A] + D_{prop,Z} \\
 &= \frac{\rho_A}{k - \rho_A} \cdot \frac{1}{k} \cdot E[B_A] \cdot \frac{C_{A,A}^2 + C_{B,A}^2}{2} + \frac{1}{k} \cdot E[B_A] + D_{prop,Z} \\
 &= \frac{\frac{\rho_A}{k - \rho_A}}{\frac{\rho_A}{1 - \rho_A}} \cdot \frac{1}{k} \cdot \frac{\rho_A}{1 - \rho_A} \cdot E[B_A] \cdot \frac{C_{A,A}^2 + C_{B,A}^2}{2} + \frac{1}{k} \cdot E[B_A] + D_{prop,Z}
 \end{aligned} \tag{9.29}$$

Zur Durchführung der Extrapolation auf die Fälle, die sich aus Kombinationen der drei Freiheitsgrade ergeben (Fall 4,6,7,8) werden zur stochastischen Ermittlung des Ende-zu-Ende Delays die zuvor vorgestellten Formeln je nach Extrapolationszielszenarioveränderung kombiniert.

## Kapitel 10

# Extrapolations-Testszenario

Die zuvor entwickelten Extrapolationsverfahren, die Simulation, die Best/Worst Case Analyse und das stochastische Verfahren werden auf ihre Anwendbarkeit auf die 7 zu untersuchenden Extrapolationsfälle an einem Extrapolations-Testszenario, *ETS*, ausgehend von einem in Abschnitt 10.1 spezifizierten Ausgangsszenario getestet und bewertet. Die Bewertung wird auf beliebige Extrapolationsszenarien verallgemeinert. Hinzu kommt eine Bewertung der ausgewählten QoS Technologien (BE, IntServ, DiffServ) im Bezug auf das *ETS*. Diese Bewertung wird durch eine simulative Sensitivitätsanalyse realisiert und ebenfalls auf beliebige Extrapolationsszenarien verallgemeinert. Zum Abschluß wird eine auf den erzielten Ergebnissen basierende Anleitung für eine Extrapolationsdurchführung für die typischen Einsatzgebiete bzw. Extrapolationsszenarien präsentiert.

### 10.1 Spezifizierung des Ausgangsszenarios

Ein Extrapolations-Testszenario *ETS* ist ein Extrapolations-Szenario *ES* (vgl. Abschnitt 8.1) mit einer minimalen Netzwerkkonfiguration *NK*, um die Untersuchung des QoS Spektrums *QS* von QoS-sensitiven Anwendungen mit Variationen des Lastspektrums *LS* in Kombination mit unterschiedlichen QoS Technologien *QT* und Netzkonfigurationsänderungen zu vereinfachen. Das *ETS* ist das Ausgangsszenario für die Extrapolation. Es besteht aus folgender Netzwerkkonfiguration, Lastspektrum, QoS Technologie und QoS Spektrum:

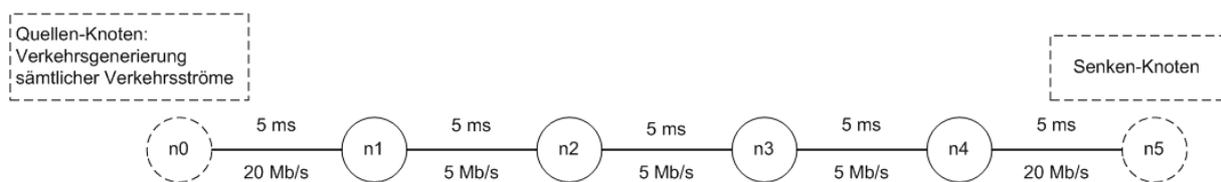


Abbildung 10.1: ETS Topologie

Abbildung 10.1 zeigt die Topologie der Netzwerkkonfiguration *NK* des *ETS*:  $NK = \{T, B, LV, R\}$  bestehend aus:

- der Topologie  $T = \{ID, N, E\}$ , mit  $ID = ETS$ ,  $N = \{n_0, n_1, n_2, n_3, n_4, n_5\}$  Knoten und  $E = \{e_{0,1}, e_{1,2}, e_{2,3}, e_{3,4}, e_{4,5}\}$  Kanten
- der Menge von Bandbreiten  $B = \{B_{0,1}, B_{1,2}, B_{2,3}, B_{3,4}, B_{4,5}\}$ , mit  $B_{0,1} = B_{4,5} = 20Mb/s$  und  $B_{1,2} = B_{2,3} = B_{3,4} = 5Mb/s$
- der Menge von Leitungsverzögerungen  $LV = \{LV_{0,1}, LV_{1,2}, LV_{2,3}, LV_{3,4}, LV_{4,5}\}$ , mit  $LV_{0,1} = LV_{1,2} = LV_{2,3} = LV_{3,4} = LV_{4,5} = 5ms$

- der Routinginformation  $R = \{\text{IP Routing}\}$

Zunächst wird für das **Lastspektrum** ( $LS$ ) eine Mischung aus CBR und TCP Verkehr gewählt:

$$LS = \{VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = (SD_{TCP}, TT_{TCP})$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologie** ( $QT$ ) wird zunächst keine ausgewählt. Es wird der BE Dienst verwendet. Alle Knoten benutzen FIFO Warteschlangen mit einer Länge von 50 Paketen unabhängig von der Paketgröße. Die Länge der Warteschlange beruht auf Werten, die im G-WiN eingesetzt werden.

$$QT = \{\}$$

Für den für die Untersuchung ausgewählten Verkehr sieht das **QoS Spektrum** ( $QS$ ) wie folgt aus:

$$QS = \{VQS_{CBR}\}$$

$$VQS_{CBR} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Von diesem Ausgangsszenario aus werden Extrapolationen durchgeführt, die die 7 zu untersuchenden Extrapolationsfälle abdecken. Dies wird durch die drei entwickelten Verfahren in den folgenden Abschnitten versucht.

## 10.2 Sensitivitätsanalyse und Extrapolation durch Simulation

Unter einer Sensitivitätsanalyse versteht man hier die Variation einzelner Extrapolations-Testszenarioparameter um vorgegebene Werte, um die isolierte Wirkung eines einzelnen dieser Parameter auf die zu messenden Größen Ende-zu-Ende Delay, Jitter und Loss (siehe Kapitel 3 und Kapitel 9) innerhalb des Extrapolations-Testszenarios zu bestimmen. Diese Sensitivitätsanalyse dient zum einen der Bewertung der QoS Technologien hinsichtlich ihrem Einsatz im *ETS* und einer Verallgemeinerung dieser Bewertung auf den Einsatz der QoS Technologien in anderen Netzen, zum anderen zur Extrapolation ausgehend vom zuvor spezifizierten Ausgangsszenario auf die 7 zu untersuchenden Extrapolationsfälle. Allgemein ergeben sich durch die mit Hilfe der Sensitivitätsanalyse erzielten Ergebnisse für den Entwerfer eines Systems Hinweise, mit welchen Parameteränderungen er eine gewünschte Änderung des Systemverhaltens am leichtesten herbeiführen kann, bzw. welche Auswirkungen ungewünschte Parameteränderungen auf das System haben werden. Bei den durchzuführenden Untersuchungen wird ausgehend vom Ausgangsszenario die Netzwerkkonfiguration und das Lastspektrum für jede QoS Technologie variiert, d.h. die Extrapolationsfälle 1, 3, 5 und 7 (siehe Kapitel 8) werden für jede QoS Technologie untersucht. Die Variation der QoS Technologien deckt die Extrapolationsfälle 2, 4, 6 und 8 ab. Das Simulationsexperiment wird nach QoS Technologien in folgende Telexperimente aufgeteilt:

- Simulationen mit Best Effort
- Simulationen mit IntServ

- Simulationen mit DiffServ (Olympic und Default Ansatz)
- Simulationen mit Overprovisioning

Für die ersten drei Teilbereiche wird eine Sensitivitätsanalyse nach dem folgenden Schema durchgeführt. Für jede QoS Technologie wird die Verkehrslast des Gesamtverkehrs, die Verkehrsanteile der einzelnen Verkehrstypen am Gesamtverkehr und die Verkehrszusammensetzung des Gesamtverkehrs aus den einzelnen Verkehrstypen variiert. Außerdem wird eine Erhöhung der Linkkapazitäten für BE, bzw. Overprovisioning, simuliert und die Ergebnisse mit den zuvor mit den Bandbreiten des Ausgangsszenarios erzielten Ergebnissen von IntServ und DiffServ (Olympic und Default Ansatz) verglichen. Zudem werden innerhalb der DiffServ Simulationsexperimente Untersuchungen durchgeführt, ab welchen Auslastungen unterschiedlicher Verkehrszusammensetzungen für beide DiffServ Ansätze eine Implementation einer (lokalen) Admission Control unumgänglich ist. Es werden für jedes Simulationsexperiment jeweils 20 Simulationläufe mit verschiedenen Startpunkten desselben Extrapolations-Szenarios durchgeführt und die 90 % Vertrauensintervalle (wie in Kapitel 8 beschrieben) über die 20 erzielten Werte des Ende-zu-Ende Delays, Jitters und Loss' gebildet und grafisch dargestellt. Repräsentativ für alle untersuchten QoS Technologien wird für BE für alle untersuchten Verkehrszusammensetzungen die Auslastung grafisch dargestellt. Jeder einzelne Flow hält seine ausgehandelten Policies ein (Policing ist jedoch für IntServ und DiffServ im Simulator implementiert und kann bei Bedarf zugeschaltet werden) und es wird zunächst lediglich bei IntServ eine Admission Control angewendet. Alle Simulationsergebnisse der Experimente mit einem CBR Flow (einem Video Trace oder jeweils einem von beiden Verkehrstypen) und 10 CBR Flows (10 Video Traces oder jeweils 5 von beiden Verkehrstypen) werden für jede Meßgröße (Auslastung, Delay, Jitter, Loss) in getrennten Abbildungen dargestellt. Für jeden Wert der x-Achse werden die zugehörigen y-Werte nach steigender Anzahl der TCP Flows des Simulationsexperiments von links nach rechts angegeben (siehe Abbildung 10.2).

### 10.2.1 Simulationen mit Best Effort

Startpunkt ist das Ausgangsszenario. Zunächst werden die Last und die prozentualen Verkehrsanteile der einzelnen festgelegten Verkehrstypen des Ausgangsszenarios verändert. Dies wird anschließend für verschiedene Verkehrstypenkombinationen durchgeführt.

- **CBR-Verkehr mit TCP Hintergrundverkehr:**

Es werden Simulationsexperimente mit folgenden Verkehrszusammensetzungen durchgeführt:

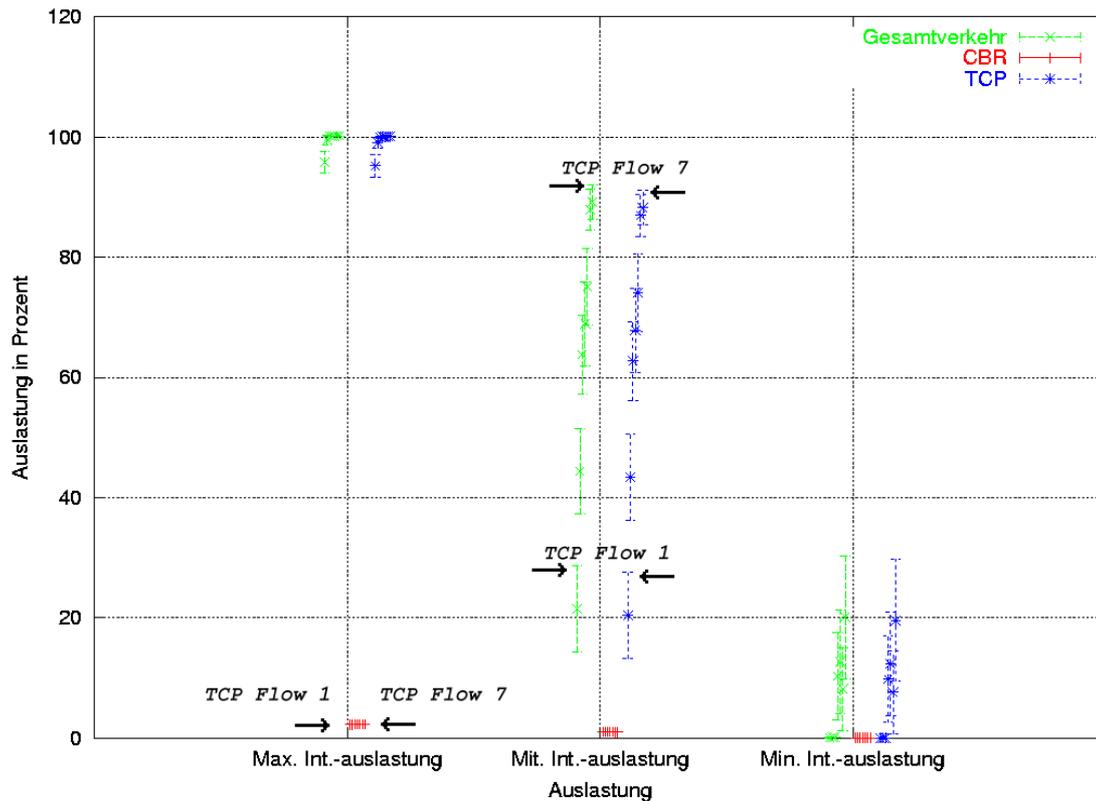
CBR Flows	TCP Flows
1	1, 2, ..., 7
10	1, 2, ..., 7

**Tabelle 10.1:** Variation der Anzahl von CBR und TCP Flows innerhalb der Simulationsexperimente

### Simulationsergebnisse:

#### Auslastung:

Wie die Abbildungen 10.2 und 10.3 zeigen, variiert die Auslastung anhand der Anzahl der generierten Flows bzw. bei Videosimulationen der Anzahl der verwendeten Trace Files. Die mittleren Gesamtauslastungen werden zunächst am Bottleneck von Knoten n1 nach n2 im Bereich von 20 % bis 90 % untersucht. Der CBR Verkehr verursacht dabei eine mittlere Auslastung von 1 % bis 10 %. Die maximale Gesamtauslastung bei steigender Anzahl von TCP Flows erreicht die 100 % Marke.

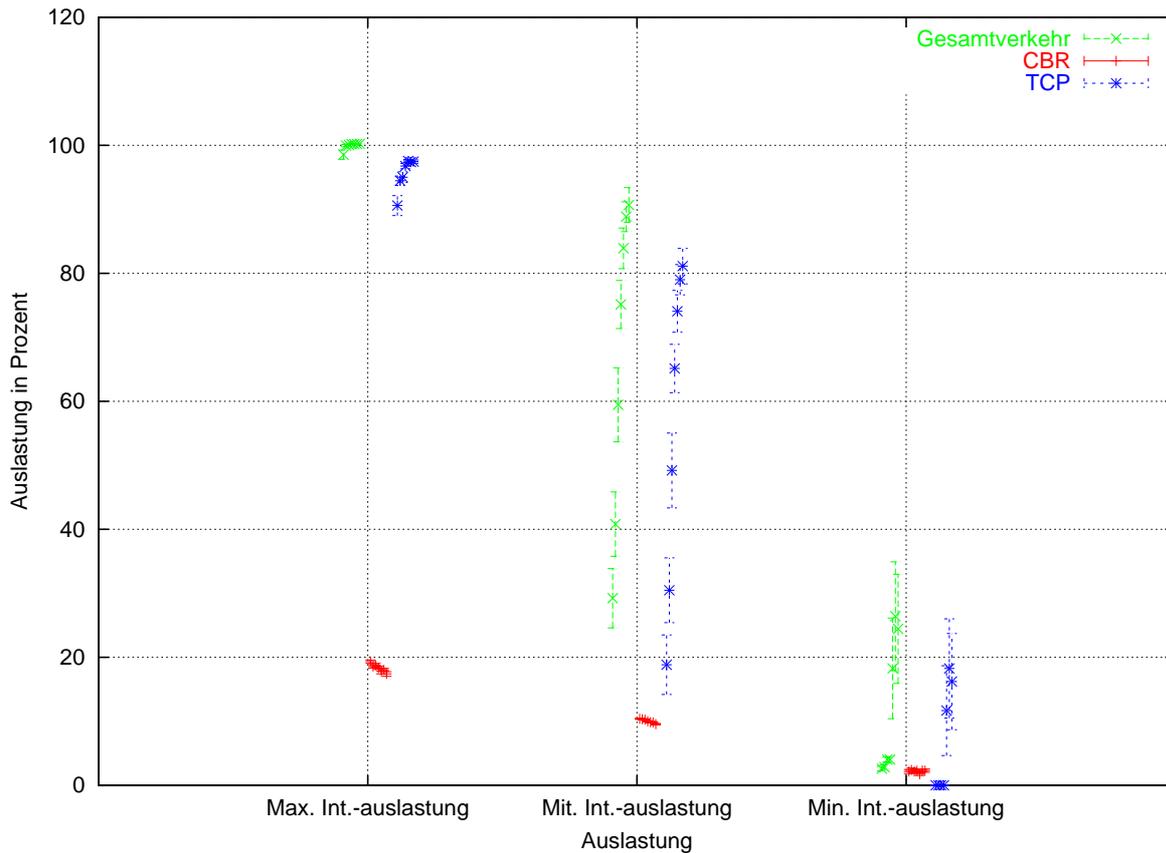


**Abbildung 10.2:** Best Effort: Auslastung bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) innerhalb eines 1 s Intervalls (Int.)

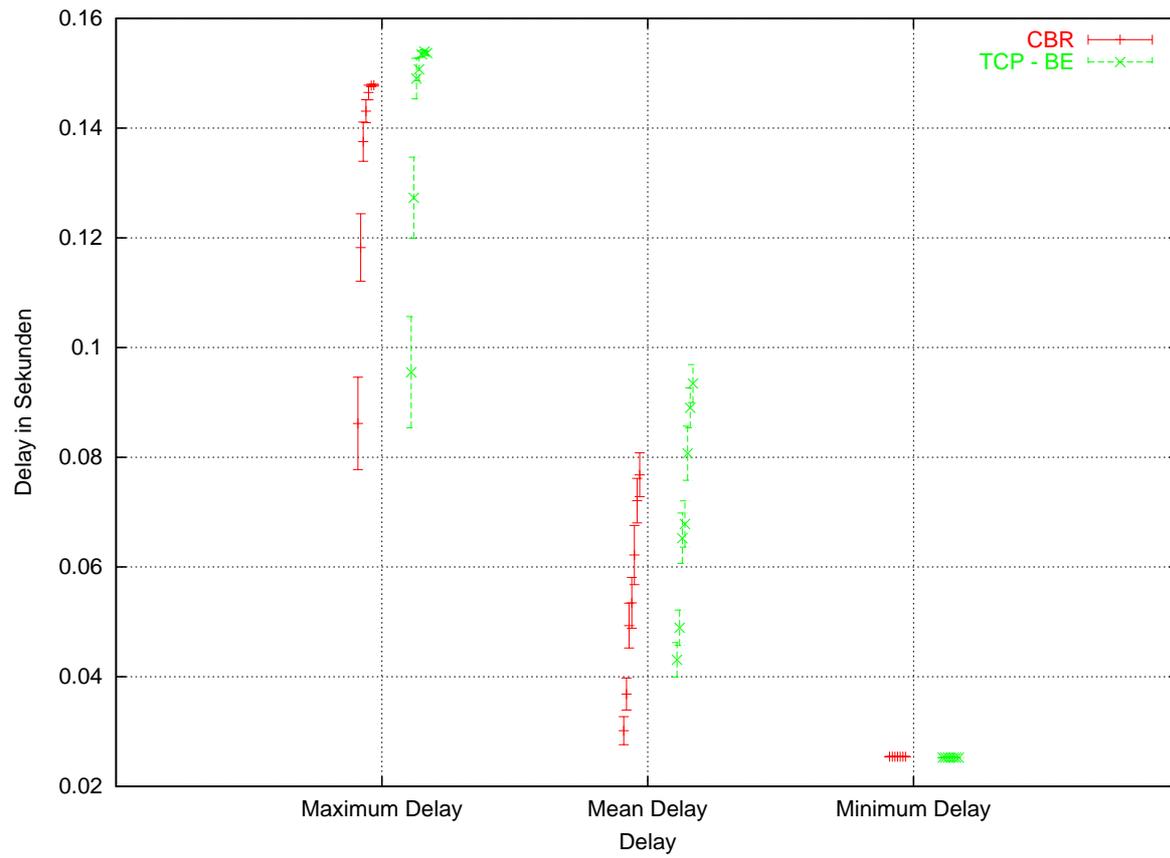
#### Delay, Jitter und Loss:

Der maximale Delay erreicht ab mittleren Auslastungen von ca. 30 % für beide Verkehrsarten schnell die 100 ms Grenze (siehe Abbildungen 10.4 und 10.5). Er schmiegt sich bei hoher mittlerer Auslastung von ca. 80 % an ein Maximum von ca. 150 ms an. Dieser Wert ist durch die verwendeten Queuelängen gegeben. Der mittlere Delay bleibt unter 100 ms. Der im Vergleich zum CBR höhere TCP Delay entsteht durch die größeren TCP Pakete. Der maximale Jitter überschreitet stets die 50 ms Grenze und hat bei ca. 120 ms verursacht durch die Delaybeschränkung seinen Maximalwert (siehe Abbildungen 10.6 und 10.7). Auch der mittlere Jitter überschreitet bei

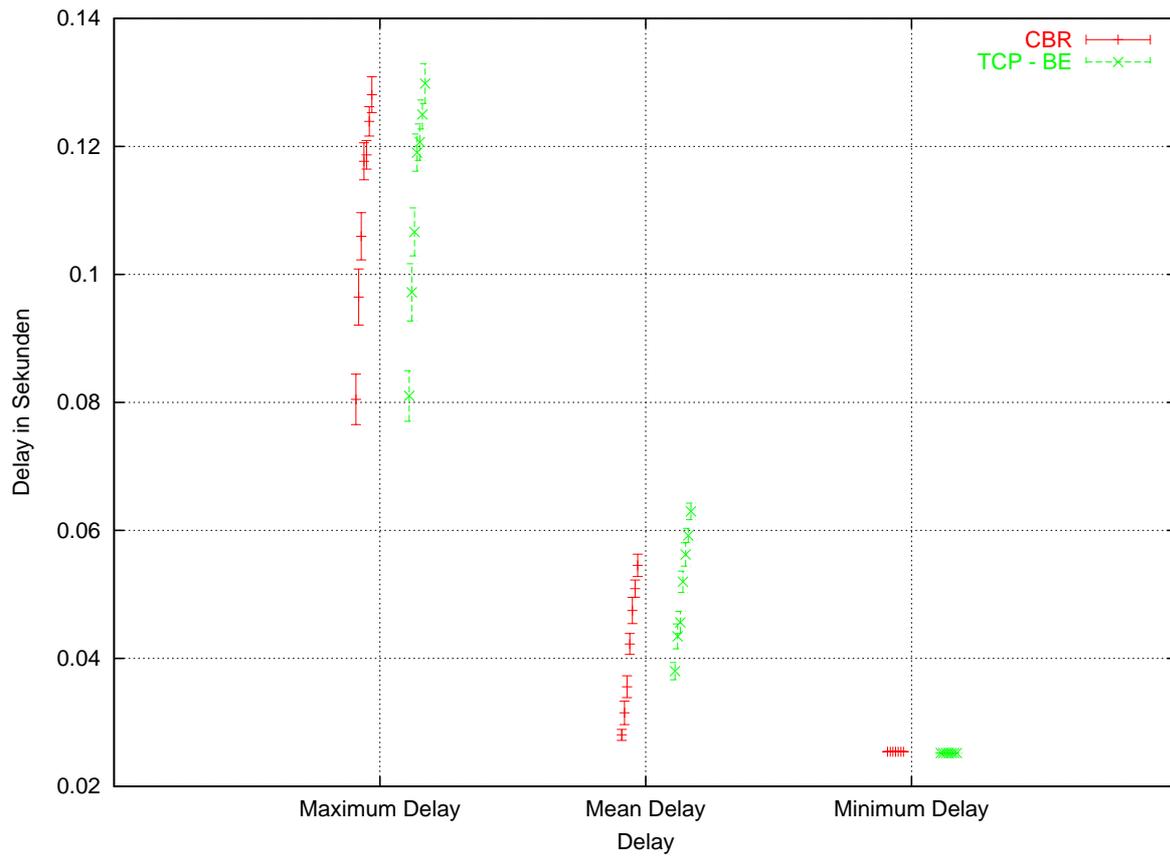
höheren mittleren Auslastungen von ca. 70 % die 50 ms Grenze. Der Loss hält sich bei mittleren Auslastungen bis ca. 50 % für beide Verkehrsarten in akzeptablen Bereichen unter 0.1 % auf (siehe Abbildungen 10.8 und 10.7). Allerdings steigt er dann schnell auf für den CBR Verkehr inakzeptable Werte von bis ca. 9 % an. Dies liegt daran, daß bei höheren Auslastungen die Warteschlangen überlaufen. Hier erkennt man deutlich den Zusammenhang des steigenden Delays bis zu einer oberen Grenze (bis die Warteschlange voll ist) und der dann extrem ansteigenden Prozentzahl von Loss.



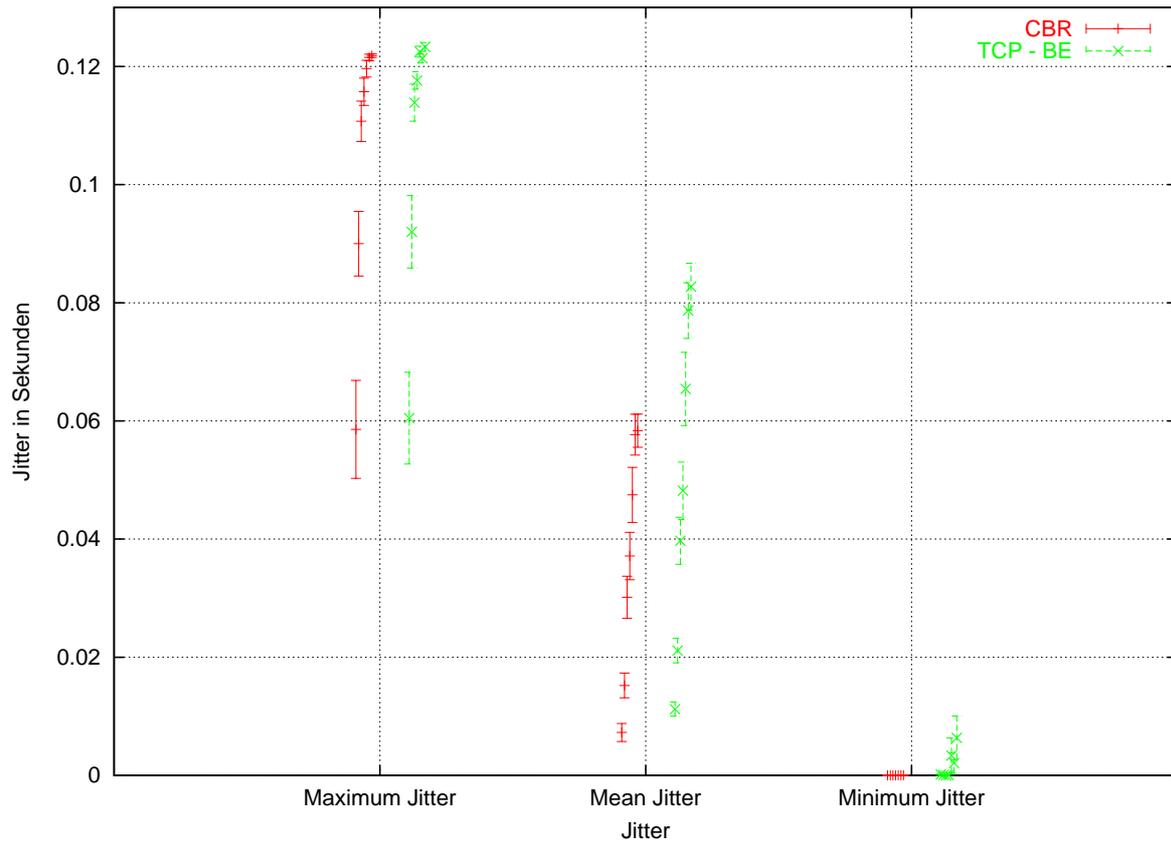
**Abbildung 10.3:** Best Effort: Auslastung bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) innerhalb eines 1 s Intervalls (Int.)



**Abbildung 10.4:** Best Effort: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.5:** Best Effort: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ...,7)



**Abbildung 10.6:** Best Effort: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ...,7)

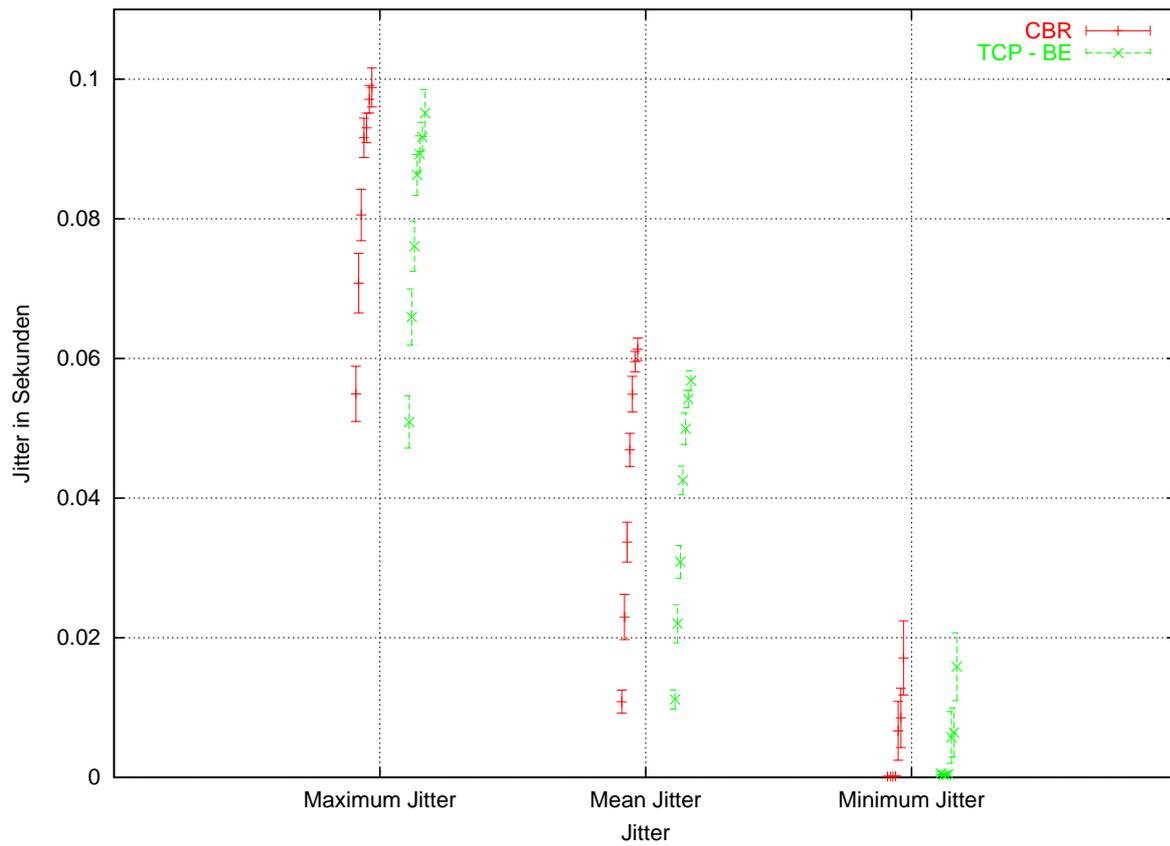
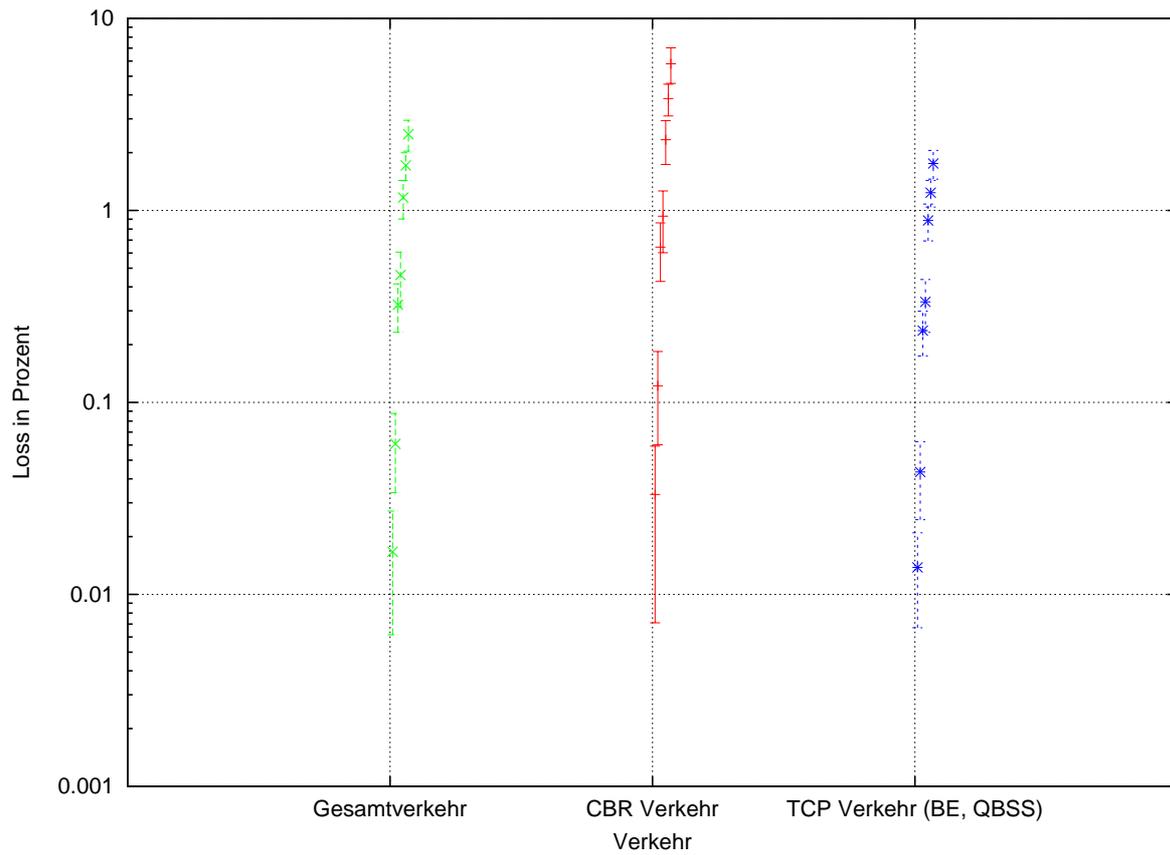
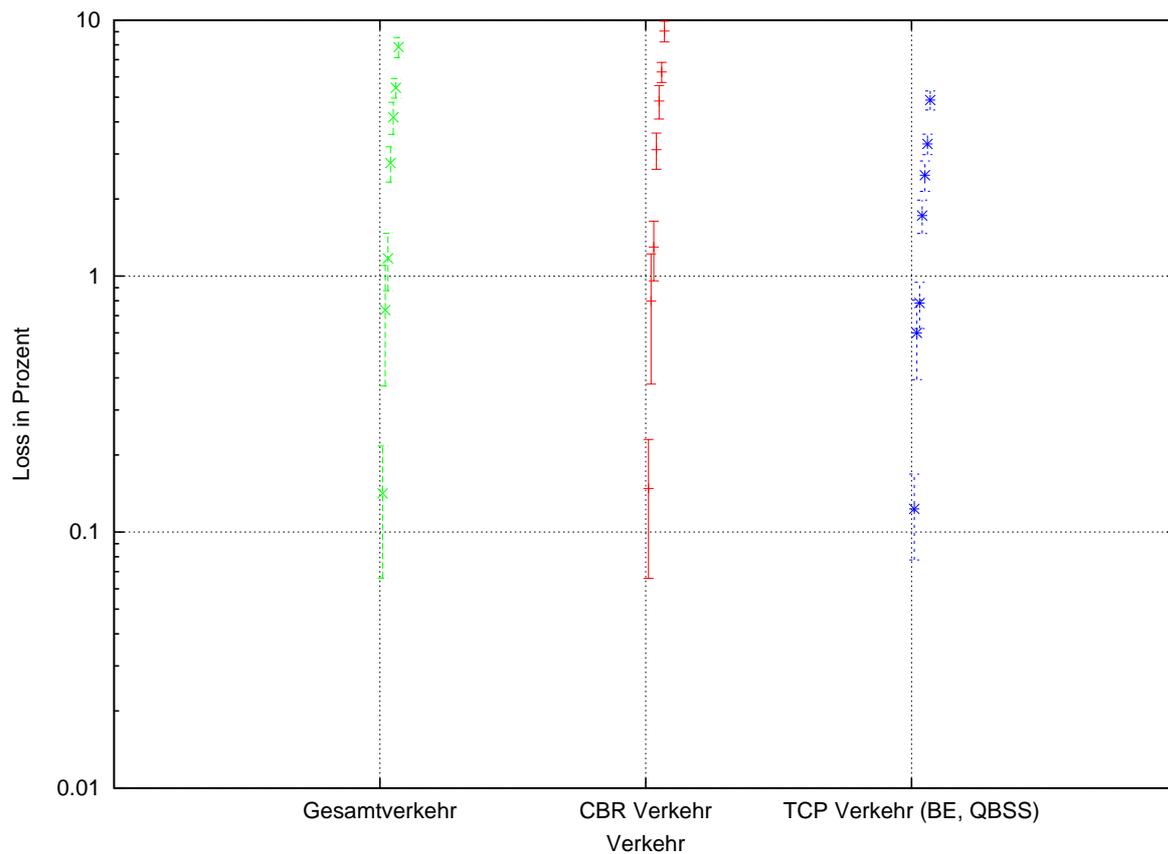


Abbildung 10.7: Best Effort: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ...,7)



**Abbildung 10.8:** Best Effort: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ...,7)



**Abbildung 10.9:** Best Effort: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ...,7)

- **Videoverkehr mit TCP Hintergrundverkehr:**

**Lastspektrum ( $LS$ ):**

$$LS = \{VLS_{VIDEO}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien ( $QT$ ):**

$$QT = \{\}$$

**QoS Spektrum ( $QS$ ):**

$$QS = \{VQS_{VIDEO}\}$$

$$VQS_{VIDEO} = (0.1\%, 100\text{ ms}, 50\text{ ms})$$

Video Traces	TCP Flows
1	1, 7
10	1, 7

**Tabelle 10.2:** Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente

### Simulationsergebnisse:

#### Auslastung:

Bei den Simulationen mit hochpriorrem Videoverkehr werden mittlere Gesamtauslastungen des Bottleneck Links ( $n_1 - n_2$ ) von ca. 34 % bis 94 % untersucht. Der Videoverkehr verursacht dabei eine mittlere Auslastung von 9 % bis 43 %. Die maximale Gesamtauslastung bei steigender Anzahl von TCP Flows erreicht die 100 % Marke (siehe Abbildungen C.1 und C.2 in Anhang C).

#### Delay, Jitter und Loss:

Der maximale Delay zeigt die gleiche Tendenz wie beim CBR Verkehr, ist aber auf Grund der durch Videoverkehr höheren Verkehrslast und der höheren Paketgrößen bereits bei einem Hintergrundverkehr von einem TCP Flow über der 100 ms Grenze. Dies entspricht einer mittleren Auslastung von ca. 30 %. Der mittlere Delay bleibt unter 100 ms (siehe Abbildungen C.3 und C.4). Der max. Jitter überschreitet stets die 50 ms und zeigt somit ein identisches Verhalten wie der Jitter bei simuliertem CBR und TCP Verkehr, auf Grund der variierenden Paketgrößen innerhalb des Videoverkehrs ergeben sich aber höhere Werte für den Jitter (siehe Abbildungen C.5 und C.6). Der auftretende Loss bleibt bei mittleren Auslastungen von bis ca. 50 % unter 0.1 %. Allerdings explodiert der Loss förmlich bei der Überschreitung der Warteschlangenkapazitäten (siehe Abbildungen C.7 und C.8).

### • Videoverkehr, CBR Verkehr mit TCP Hintergrundverkehr:

#### **Lastspektrum ( $LS$ ):**

$$LS = \{VLS_{VIDEO}, VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

#### **QoS Technologien ( $QT$ ):**

$$QT = \{\}$$

**QoS Spektrum (QS):**

$$QS = \{VQS_{VIDEO}, VQS_{CBR}\}$$

$$VQS_{VIDEO} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

$$VQS_{CBR} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Video Traces	CBR Flows	TCP Flows
1	1	1, 7
5	5	1, 7

**Tabelle 10.3:** Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente

Simulationsergebnisse:Auslastung:

Bei den Simulationen des hochpriorigen Verkehrsmixes aus CBR und Videoverkehr werden mittlere Gesamtauslastungen des Bottleneck Links (n1 - n2) von ca. 28 % bis 92 % untersucht. Der Videoverkehr verursacht dabei eine mittlere Auslastung von 9 % bis 20 %, der CBR Verkehr von 1 % bis 3 %. Die maximale Gesamtauslastung bei steigender Anzahl von TCP Flows erreicht die 100 % Marke (siehe Abbildungen C.9 und C.10 in Anhang C).

Delay, Jitter und Loss:

Delay und Jitter zeigen identische Tendenzen wie die zuvor mit den QoS-sensitiven Einzelverkehren durchgeführten Simulationen. Sie unterscheiden sich lediglich durch die auf Grund der Verkehrscharakteristik entstehenden Abweichungen (Auslastung, Paketgrößen) (siehe Abbildungen C.11, C.12 und C.13, C.14). Der Loss wird durch die auslastungsintensivere Videoanwendung dominiert (siehe Abbildungen C.15 und C.16).

Zusammenfassung der Best Effort Simulationsergebnisse:

Allgemein läßt sich sagen, daß der Best Effort Dienst, wie erwartet, bereits ab mittleren Auslastungen des Bottlenecks von ca. 30 % für QoS-sensitiven Verkehr inakzeptable Werte für den maximalen Ende-zu-Ende Delay für alle untersuchten Verkehrstypen liefert. Bei niedrigeren Auslastungen reichen die mit Hilfe des BE erzielten Dienstgüten aus. Hohe mittlere Auslastungen des Bottlenecks von bis ca. 50 %, verursacht meist lediglich durch Hintergrundverkehr, führen für den QoS-sensitiven Verkehr zu einem akzeptablen Loss von ca. 0.1 %. Höhere mittlere Auslastungen von bis ca. 94 % führen zu einem Loss von bis zu ca. 58 %. Somit ist die Einführung einer QoS Technologie oder die Erhöhung der Bandbreiten für den BE Dienst (Overprovisioning) bei den untersuchten Verhältnissen unumgänglich!

## 10.2.2 Simulationen mit IntServ

### • CBR-Verkehr mit TCP Hintergrundverkehr:

#### Lastspektrum ( $LS$ ):

$$LS = \{VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

#### QoS Technologien ( $QT$ ):

$$QT = \{EQT_{IntServ}\}$$

$$EQT_{IntServ} = \{NI, EM\}$$

$NI = \{(n_0, (\text{FIFO, 10 pro Flow}), \text{WFQ}), (n_1, (\text{FIFO, 10 pro Flow}), \text{WFQ}), (n_2, (\text{FIFO, 10 pro Flow}), \text{WFQ}), (n_3, (\text{FIFO, 10 pro Flow}), \text{WFQ}), (n_4, (\text{FIFO, 10 pro Flow}), \text{WFQ}), (n_5, (\text{FIFO, 10 pro Flow}), \text{WFQ})\}$

$EM = (\text{Paket, per Flow / Ende-zu-Ende / deterministisch / Loss / Delay, infinit, extrem komplex, Router / Endsysteme, Edge / Router, proaktiv, nein, schlecht, Netzwerk})$

#### QoS Spektrum ( $QS$ ):

$$QS = \{VQS_{CBR}\}$$

$$VQS_{CBR} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

CBR Flows	TCP Flows
1	1, 2, ..., 7
10	1, 2, ..., 7

**Tabelle 10.4:** Variation der Anzahl von CBR Flows und TCP Flows innerhalb der Simulationsexperimente

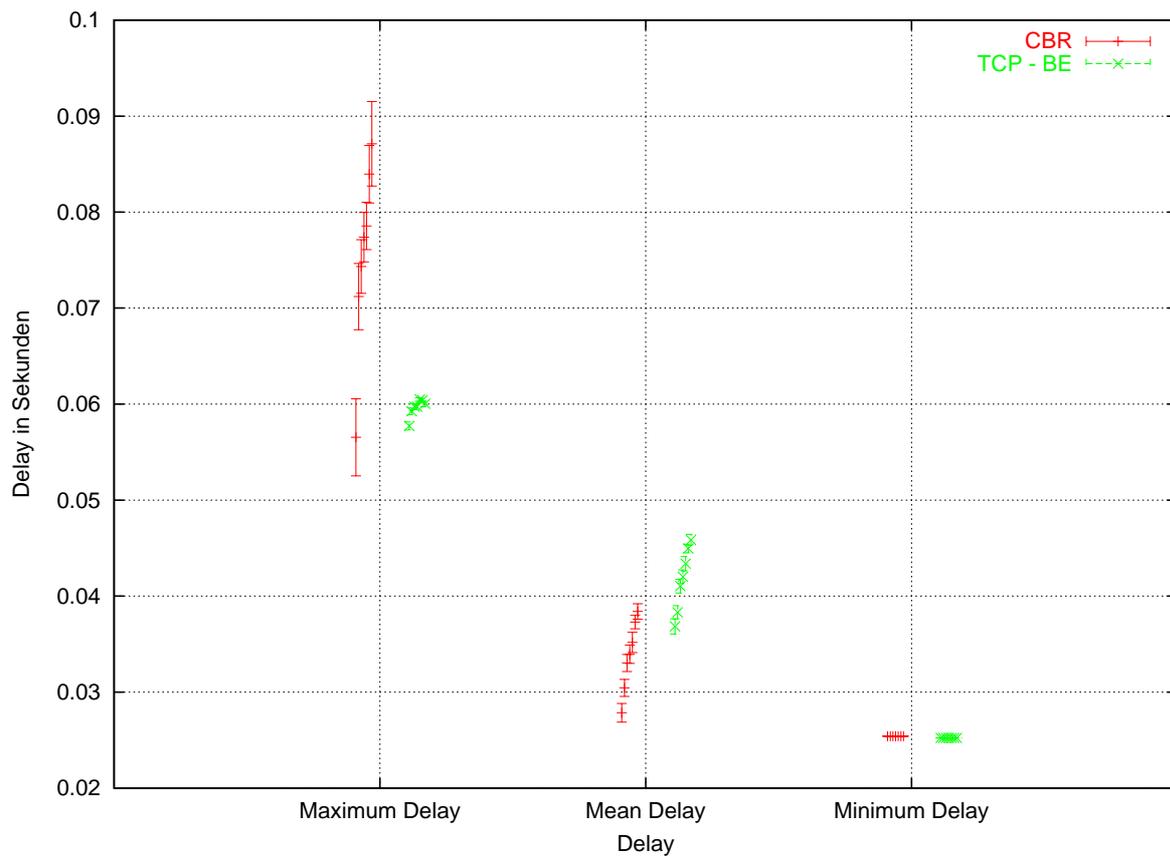
#### Simulationsergebnisse:

Bei IntServ wurden auf Grund der Anforderungen der Anwendungen die reservierten Raten und Puffergrößen so eingestellt, daß ein maximaler Ende-zu-Ende Delay von 100 ms nicht überschritten wird. IntServ benutzt im Gegensatz zu den beiden DiffServ Ansätzen bereits bei den ersten Simulationsläufen eine Admission Control, die überprüft, ob eine Reservierung auf einem Pfad zum gewünschten Empfänger noch durchgeführt werden kann (vergleiche Kapitel 2). Es werden Reservierungen bis zu 80 % der zu Verfügung stehenden Bandbreite durchgeführt. Bandbreitenanforderungen darüber hinaus werden von der Admission Control abgelehnt.

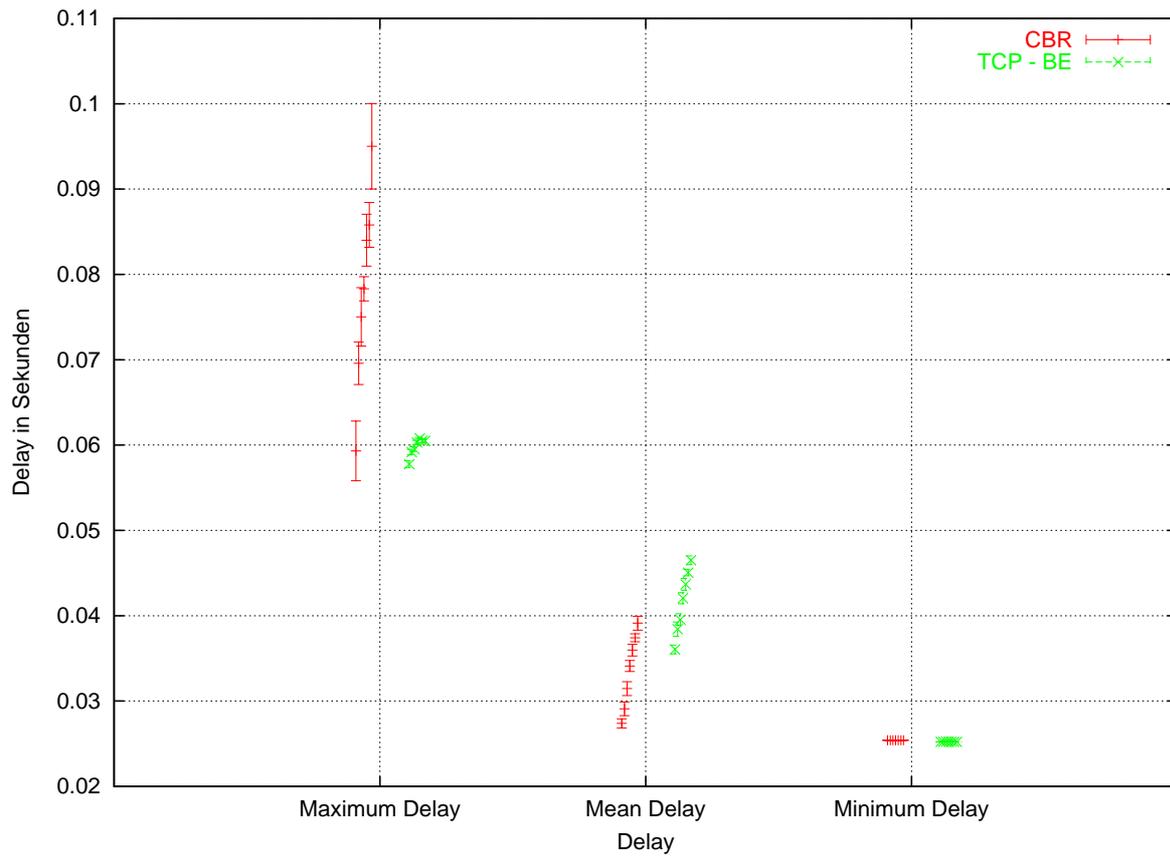
#### Delay, Jitter und Loss:

Wie die Abbildungen 10.10 und 10.11 zeigen, steigt der Delay des CBR Verkehrs kontinuierlich bei zunehmendem Verkehr (egal welcher Art) an, was auf den verwendeten WRR Scheduler zurückzuführen ist (siehe Kapitel 9). Er hält aber die vorgegebene Grenze von 100 ms ein. Der

TCP Verkehr erfährt bei diesen Simulationen einen geringeren Delay, da die eingestellte Puffergröße basierend auf den Eigenschaften des CBR Verkehrs auf eine für diesen minimale Größe zur Erbringung der erforderlichen Dienstgüte (100 ms Delay) eingestellt wurde (siehe Abbildungen 10.10 und 10.11). Diese Puffergröße bezieht sich in der gewählten DiffServ Realisierung ebenfalls auf den TCP Verkehr, dessen Pakete allerdings größer sind, so daß nur eine geringere Anzahl dieser gespeichert werden kann. Dadurch verringert sich der Delay, aber der Loss steigt an (siehe Abbildungen 10.14 und 10.15), da TCP keiner Reservierung unterliegt. Diese Puffergrenze ist bei TCP schnell erreicht. Dies sieht man daran, daß sich der maximale Jitter bei mittleren Auslastungen von 30 % - 85 % (höchste untersuchte Auslastung) nicht mehr verändert (siehe Abbildungen 10.12 und 10.13). Bei einer im Fall von unterschiedlichen Reservierungsanforderungen notwendigen Entkopplung der Puffergrößen kann der Puffer individuell vergrößert werden. Dies hat einen Anstieg des Delays, aber auch eine Verringerung des Losses zur Folge. Der CBR Loss liegt, wie zugesichert, und durch die berechnete Puffergröße und reservierte Rate realisiert bei 0 %. Der Jitter verhält sich entsprechend dem Delay im Bezug auf die unterschiedlichen Verkehrstypen. Es ist jedoch zu beachten, daß der maximale Jitter sehr hoch bzw. zu hoch ausfällt. Er überschreitet schon bei mittleren Auslastungen von ca. 50 % die 50 ms Grenze. IntServ gibt allerdings auch keine Garantie für Jitter und ist daher für extrem jitteranfällige Anwendungen nicht geeignet. Der Jitter kann aber durch eine Reduktion des maximalen Ende-zu-Ende Delays verringert werden (entspricht einer reduzierten maximalen Ende-zu-Ende Delayangabe beim Aushandeln der Übertragungsbedingungen). Hierfür wäre im Vorfeld eine Worst Case Jitter Analyse, wie in Kapitel 9 beschrieben, sehr hilfreich.



**Abbildung 10.10:** IntServ: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ...,7)



**Abbildung 10.11:** IntServ: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)

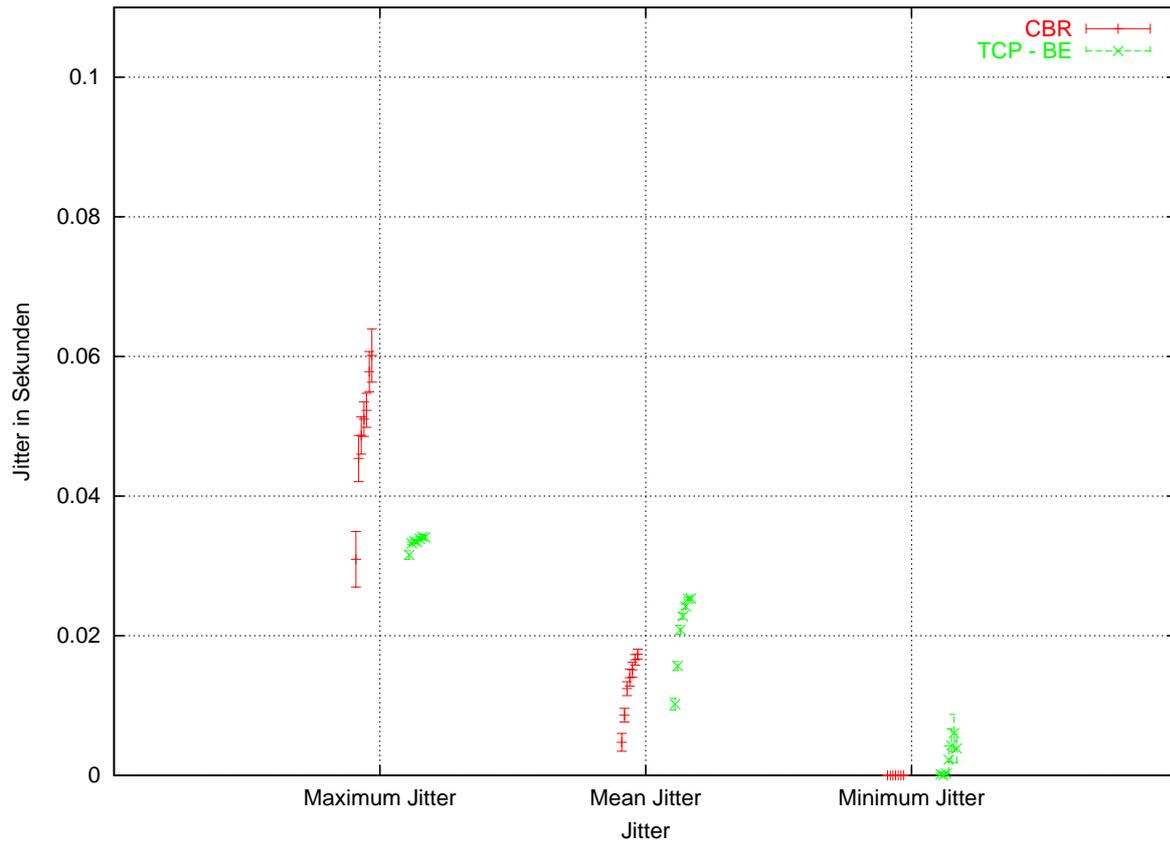


Abbildung 10.12: IntServ: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)

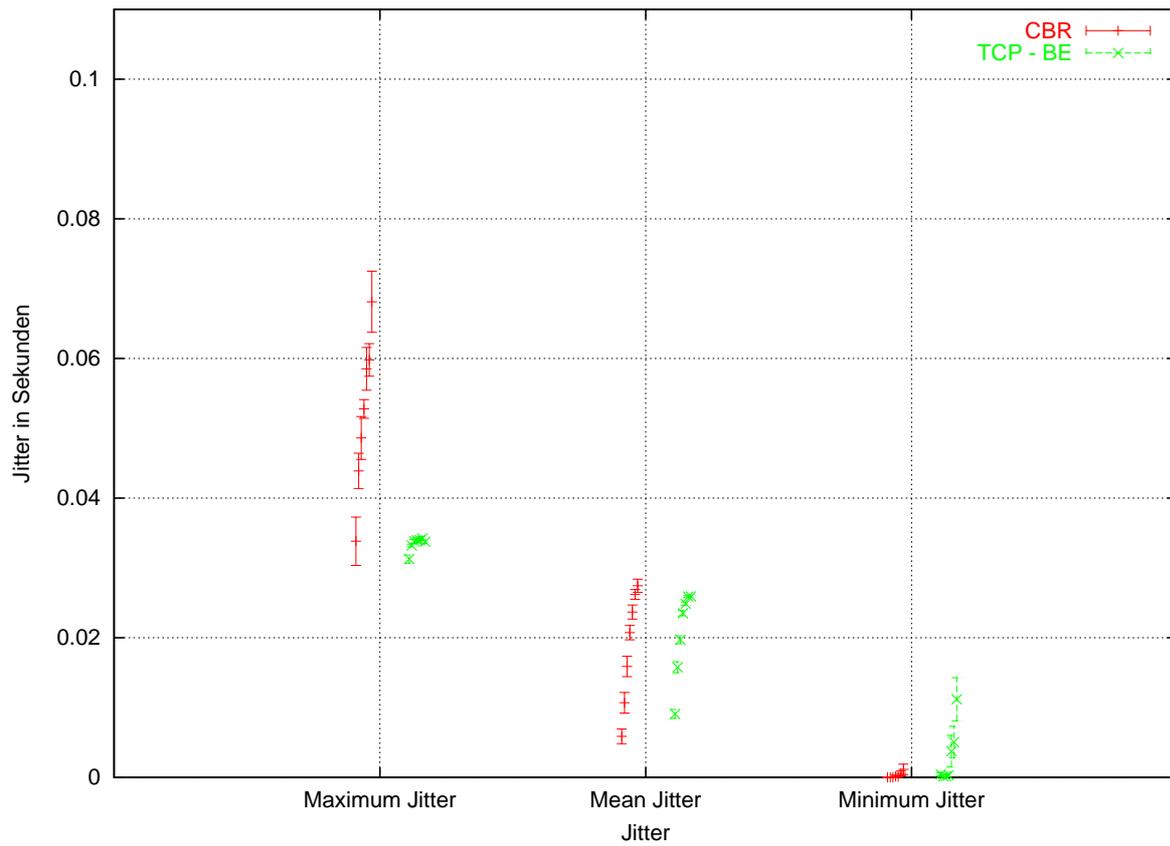
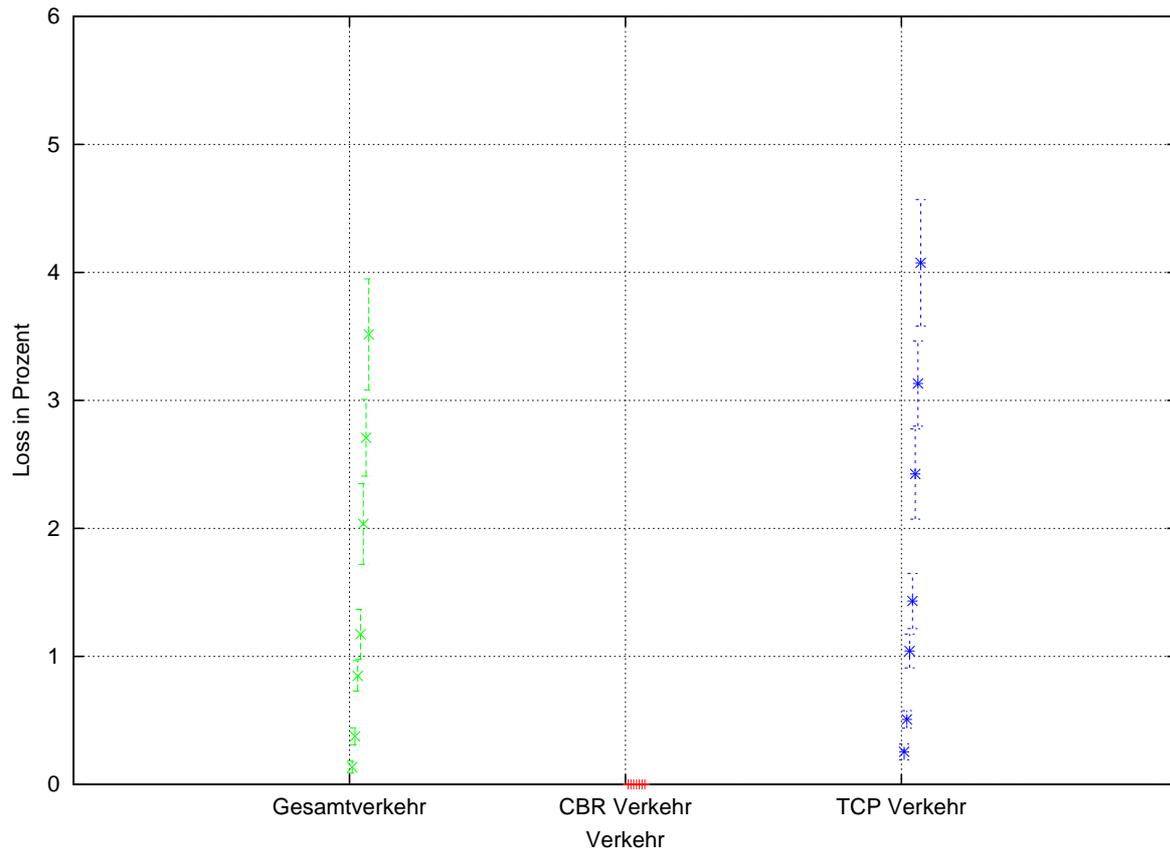


Abbildung 10.13: IntServ: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.14:** IntServ: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)

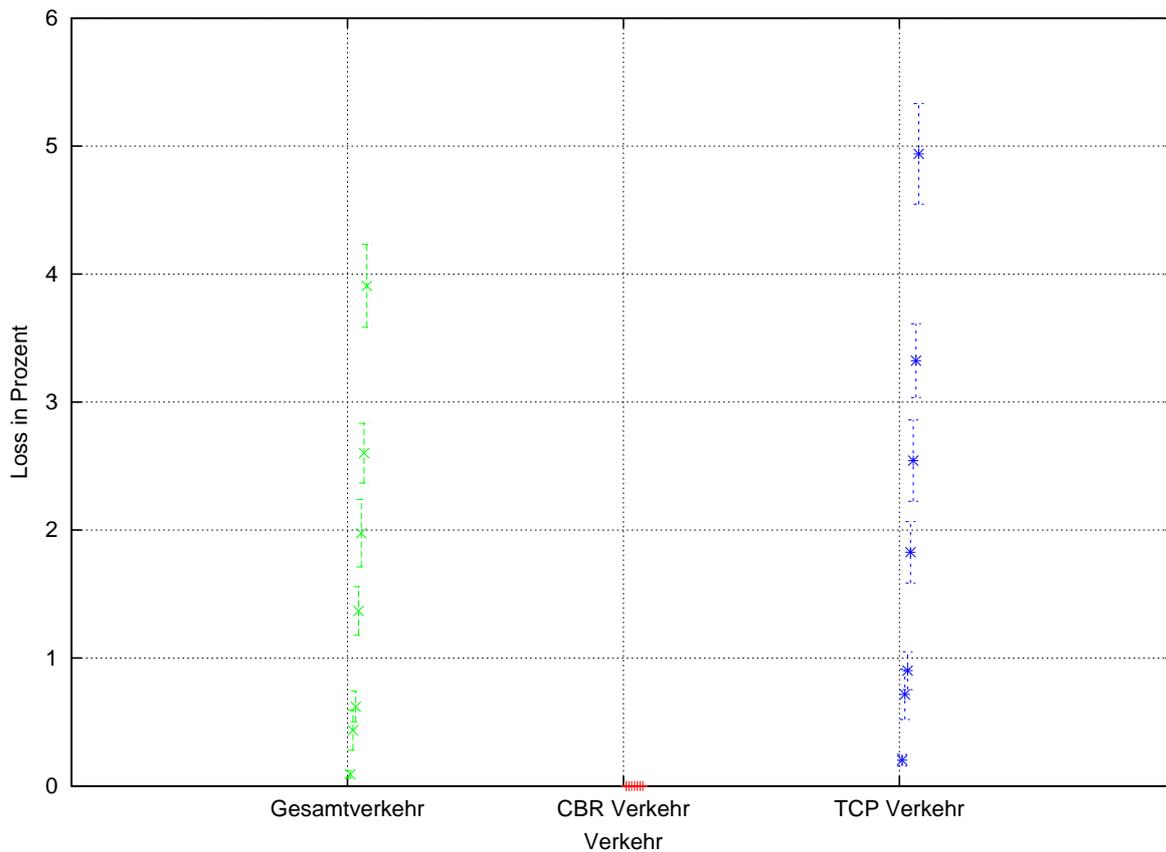


Abbildung 10.15: IntServ: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)

- **Videoverkehr mit TCP Hintergrundverkehr:**

**Lastspektrum ( $LS$ ):**

$$LS = \{VLS_{VIDEO}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien ( $QT$ ):**

$$QT = \{EQ_{IntServ}\}$$

$$EQ_{IntServ} = \{NI, EM\}$$

$$NI = \{(n_0, (\text{FIFO}, 20), \text{WFQ}), (n_1, (\text{FIFO}, 20), \text{WFQ}), (n_2, (\text{FIFO}, 20), \text{WFQ}), (n_3, (\text{FIFO}, 20), \text{WFQ}), (n_4, (\text{FIFO}, 20), \text{WFQ}), (n_5, (\text{FIFO}, 20), \text{WFQ})\}$$

$$EM = (\text{Paket, per Flow / Ende-zu-Ende / deterministisch / Loss / Delay, infinit, extrem komplex, Router / Endsysteme, Edge / Router, proaktiv, nein, schlecht, Netzwerk})$$

**QoS Spektrum ( $QS$ ):**

$$QS = \{VQS_{VIDEO}\}$$

$$VQS_{VIDEO} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Video Traces	TCP Flows
1	1, 7
10	1, 7

**Tabelle 10.5:** Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente

Simulationsergebnisse:

Die Puffer und Ratenreservierung wurde den Anforderungen des Videoverkehrs angepaßt. Es werden Reservierungen bis zu 80 % der zu Verfügung stehenden Bandbreite durchgeführt. Bandbreitenanforderungen darüber hinaus werden von der Admission Control abgelehnt.

Delay, Jitter und Loss:

Delay, Jitter und Loss verhalten sich von der Tendenz wie bei CBR Verkehr. Die Delaygrenze von 100 ms, sowie 0 % Loss werden eingehalten (siehe Abbildungen C.17, C.18 und C.21, C.22). Der max. Jitter überschreitet bei allen untersuchten Auslastungsverhältnissen die 50 ms Grenze und stellt ein Problem für Jitter-sensitive Anwendungen dar (siehe Abbildungen C.19 und C.20). Der TCP Delay steigt, auf Grund längerer Warteschlangen (siehe CBR Ergebnis) und eines höheren Verkehrsaufkommens des höher gewichteten Videoverkehrs und der daher vermehrt auftretenden Reduzierung des TCP Bandbreitenanteils auf das garantierte Minimum, an. Der Loss steigt auch bei mittleren Auslastungen von ca. 94 % bis zu inakzeptablen ca. 9 % an. Hier ist zu überlegen, ob eine Admission Control für TCP nicht sinnvoll wäre, um zumindest einigen TCP Flows eine akzeptable Dienstgüte zu gewährleisten. Die zweite Möglichkeit dies zu erreichen ist eine extreme Puffergröße für den TCP Verkehr zu wählen, die den Loss reduzieren würde, allerdings den Delay sehr stark ansteigen ließe. Die dritte Möglichkeit ist den prozentualen Anteil, der für reservierte Ströme von der gesamten zur Verfügung stehenden Bandbreite verwendet werden darf, zu reduzieren. Die Wahl des geeigneten Mittels hängt von den speziellen Anforderungen des TCP Verkehrs und dem Verkehrsanteil des QoS-sensitiven Verkehrs ab.

• Videoverkehr, CBR Verkehr mit TCP Hintergrundverkehr:

**Lastspektrum (LS):**

$$LS = \{VLS_{VIDEO}, VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien (QT):**

$$QT = \{EQT_{IntServ}\}$$

$$EQT_{IntServ} = \{NI, EM\}$$

$$NI = \{(n_0, (\text{FIFO}, 20), \text{WFQ}), (n_1, (\text{FIFO}, 20), \text{WFQ}), (n_2, (\text{FIFO}, 20), \text{WFQ}), (n_3, (\text{FIFO}, 20), \text{WFQ}), (n_4, (\text{FIFO}, 20), \text{WFQ}), (n_5, (\text{FIFO}, 20), \text{WFQ})\}$$

$EM = (\text{Paket, per Flow / Ende-zu-Ende / deterministisch / Loss / Delay, infinit, extrem komplex, Router / Endsysteme, Edge / Router, proaktiv, nein, schlecht, Netzwerk})$

#### QoS Spektrum (QS):

$$QS = \{VQS_{VIDEO}, VQS_{CBR}\}$$

$$VQS_{VIDEO} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

$$VQS_{CBR} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Video Traces	CBR Flows	TCP Flows
1	1	1, 7
5	5	1, 7

**Tabelle 10.6:** Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente

#### Simulationsergebnisse:

##### Delay, Jitter und Loss:

Die Untersuchungen des QoS-sensitiven Verkehrsmixes unterstreichen die mit CBR und Videoverkehr im Einzelnen gewonnenen Ergebnisse. Der Delay des sensitiven Verkehrs überschreitet die 100 ms Grenze nicht. Der Jitter überschreitet für alle untersuchten Auslastungsverhältnisse für Videoverkehr die 50 ms Grenze. Der Loss beträgt bei beiden Verkehrsarten 0 %. Der TCP Loss steigt bei mittleren Auslastungen von 94 % auf inakzeptable 7 %, die durch die zuvor angesprochenen Maßnahmen umgangen werden können.

##### Zusammenfassung der IntServ Simulationsergebnisse:

IntServ erfüllt seine durch die Reservierungstechnik ermöglichten Garantien vollständig. Der QoS-sensitive Verkehr hält die für ihn spezifizierte Delaygrenze von 100 ms und die Garantie für Verlustfreiheit auf Grund der berechneten Puffergrößen und Übertragungsraten ein. Problematisch ist jedoch die Höhe des auftretenden Jitters, der durch ein Heruntersetzen des Delays aber indirekt geregelt werden kann, indem der maximale Jitter vorweg exakt berechnet wird (Worst Case Analyse). Außerdem sollte der Anteil der Reservierungen an der gesamt zur Verfügung stehenden Bandbreite gut durchdacht sein, um den TCP Verkehr nicht durch zu hohen Loss unbrauchbar zu machen. Weitere Möglichkeiten dies in den Griff zu bekommen, wäre eine Admission Control für TCP Verkehr oder eine Reduzierung des Losses auf Kosten des Delays, was aber nur für delay-unempfindliche Anwendungen in Frage kommt.

### 10.2.3 Simulationen mit DiffServ (Olympic Ansatz)

Für die DiffServ Simulationen werden in der Topologie (siehe Abbildung 10.16) DiffServ Edge-Knoten angegeben, an denen das Policing, die Admission Control (hier deaktiviert) und das Markieren der Pakete stattfindet. Es entsteht somit eine DiffServ Domäne, die die Knoten n1, n2, n3 und n4 umfaßt.

Die Warteschlangenlängen werden wie folgt gewählt:  $EF = 10$ ,  $BE = 40$ ,  $QBSS = 50$ . Um eine Vergleichsmöglichkeit zum reinen Best Effort Dienst zu haben, wird die dort gewählte Warteschlangenlänge von 50 Paketen (für alle Größen) aufgeteilt auf die EF und BE Klasse. Die QBSS Klasse erhält, da die Anwendungen, die sich für diese Klasse entschieden haben, nicht sensitiv auf Delay und Jitter reagieren, und auf Grund der niedrigsten Priorität eine größere Warteschlange, um übermäßigen Loss zu vermeiden.

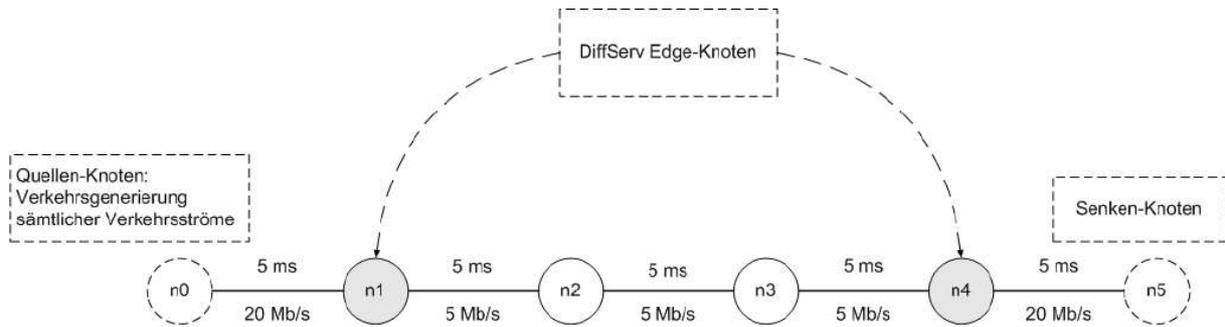


Abbildung 10.16: ETS DiffServ Topologie

- **CBR-Verkehr mit TCP Hintergrundverkehr:**

**Lastspektrum ( $LS$ ):**

$$LS = \{VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien ( $QT$ ):**

$$QT = \{EQT_{DiffServO}\}$$

$$EQT_{DiffServO} = \{NI, EM\}$$

$$NI = \{(n_1, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ}), (n_2, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ}), (n_3, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ}), (n_4, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ})\}$$

$$EM = (\text{Paket, per Hop / Aggregat / deterministisch / Loss / Delay, n Klassen, komplex, Router, Edge, proaktiv, ja, gut, Netzwerk})$$

**QoS Spektrum ( $QS$ ):**

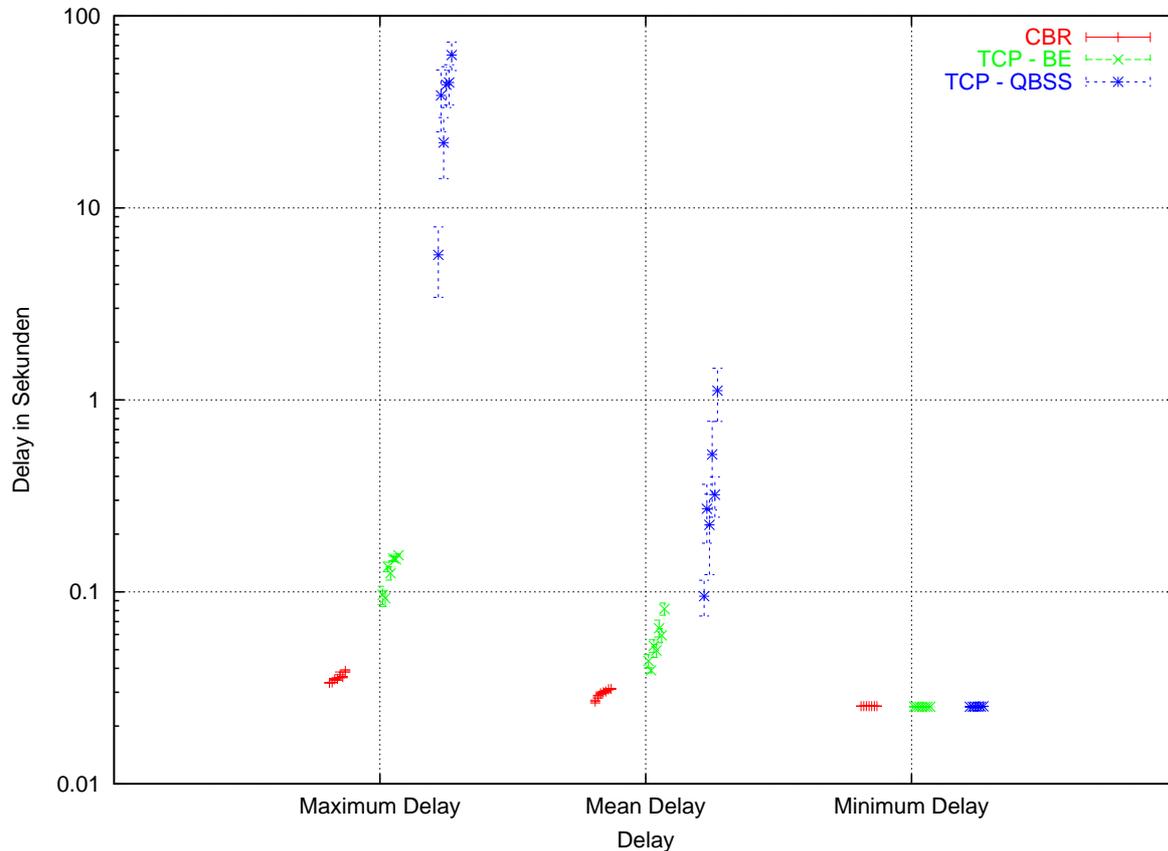
$$QS = \{VQS_{CBR}\}$$

$$VQS_{CBR} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

**Simulationsergebnisse:**

CBR Flows	TCP Flows
1	1, 2, ..., 7
10	1, 2, ..., 7

**Tabelle 10.7:** Variation der Anzahl von CBR Flows und TCP Flows innerhalb der Simulationsexperimente

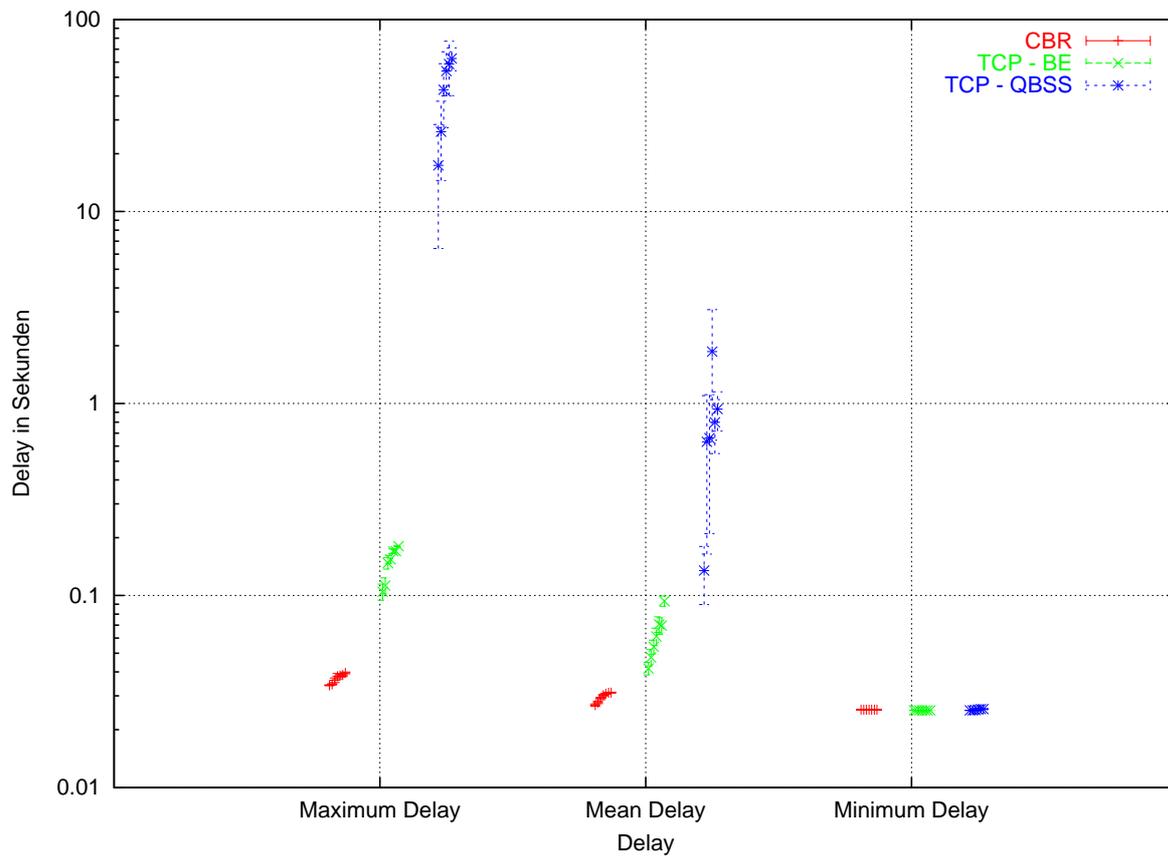


**Abbildung 10.17:** DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)

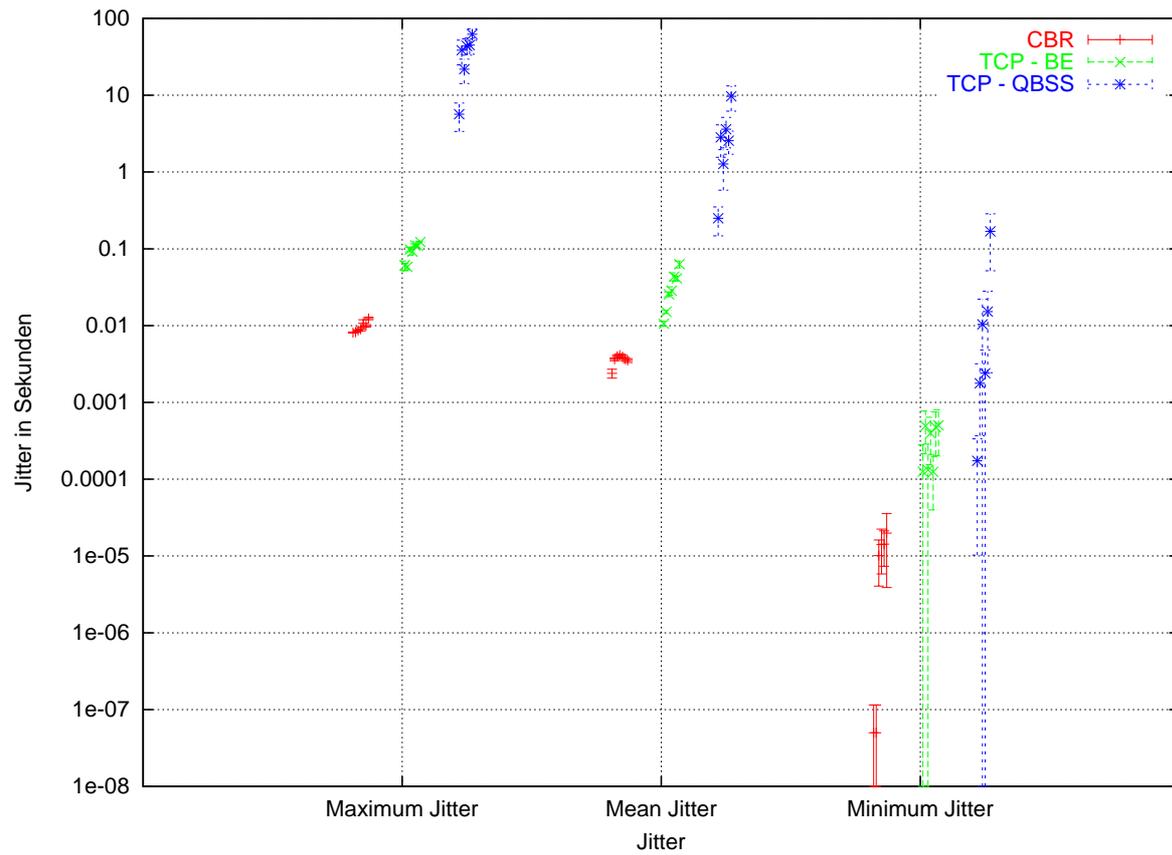
#### Delay, Jitter und Loss:

Der Delay des CBR Verkehrs ist bei den untersuchten mittleren Auslastungen von ca. 22 % bis 92 % des CBR Verkehrs minimal (siehe Abbildungen 10.17 und 10.18). Der maximale Delay unterscheidet sich kaum vom mittleren Delay. Auch für die BE Klasse ergeben sich ebenfalls für TCP Verkehr gute Delayverhältnisse. Für die QBSS Klasse hingegen werden bei einer mittleren Auslastung von ca. 92 % maximale Delaywerte von bis zu ca. 63 s erreicht. Der mittlere Delay hingegen liegt bei dieser Auslastung bei ca. 1 s. Da aber die QBSS Klasse nur Anforderungen enthalten soll, die Delay und Jitter unempfindlich sind und als Lückenfüller angesehen wird, stellen diese hohen Werte kein Problem dar. Die Jitterwerte haben identische Tendenzen wie die Delaywerte, wobei der CBR Jitter durch die geringe Schwankung des CBR Delays extrem niedrig ist (siehe Abbildungen 10.19 und 10.20). Beim CBR Verkehr entsteht bei der untersuchten Verkehrssituation überhaupt kein Loss. Auch der Loss der BE Klasse liegt bei einer mittleren

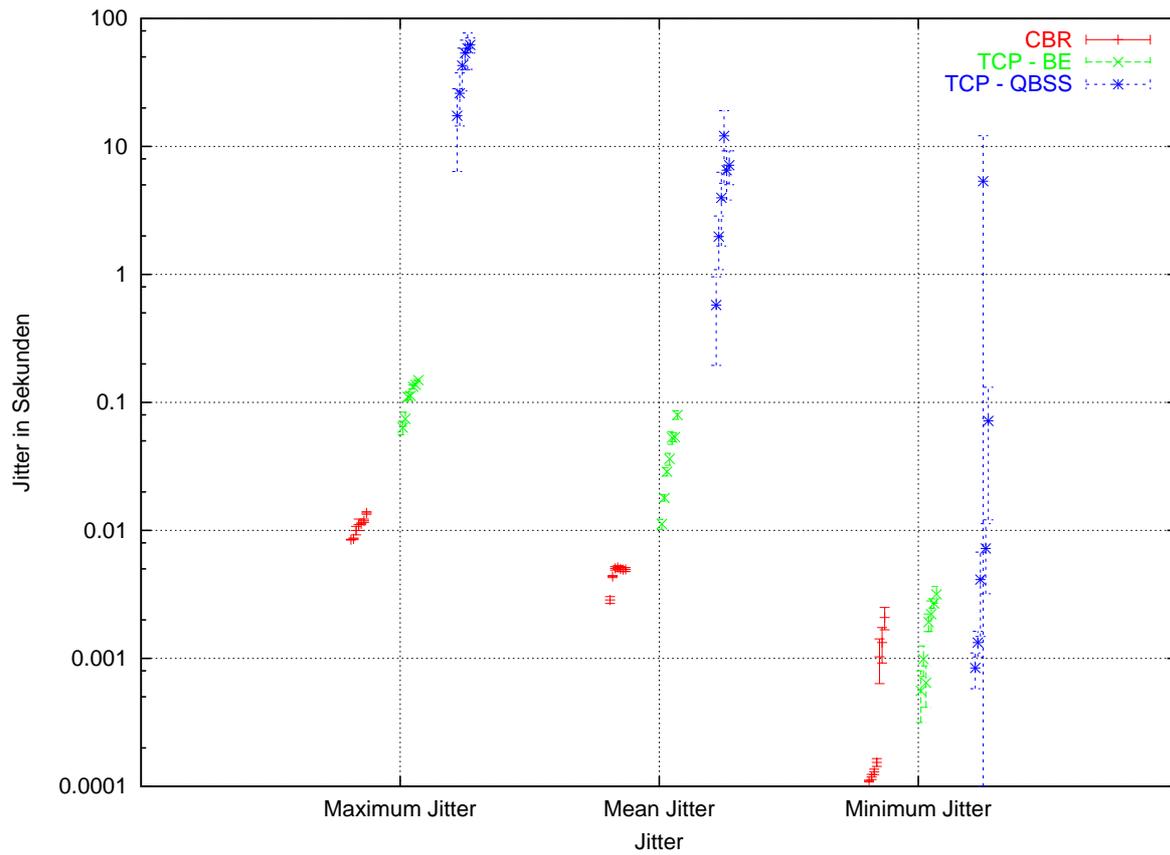
Auslastung von 92 % beim Maximum von ca. 1 %. Der Loss der QBSS Klasse wächst bei dieser Auslastung durch TCP Verkehr allerdings auf Werte über 10 % (siehe Abbildungen 10.21 und 10.22). Hier gäbe es die Möglichkeit durch eine Warteschlangenvergrößerung für die QBSS Klasse den Loss zu reduzieren und damit den Delay nochmals zu erhöhen, was aber, wie bereits erwähnt, für den Verkehr, der sich für diese Klasse entschieden hat, keine Rolle spielen darf.



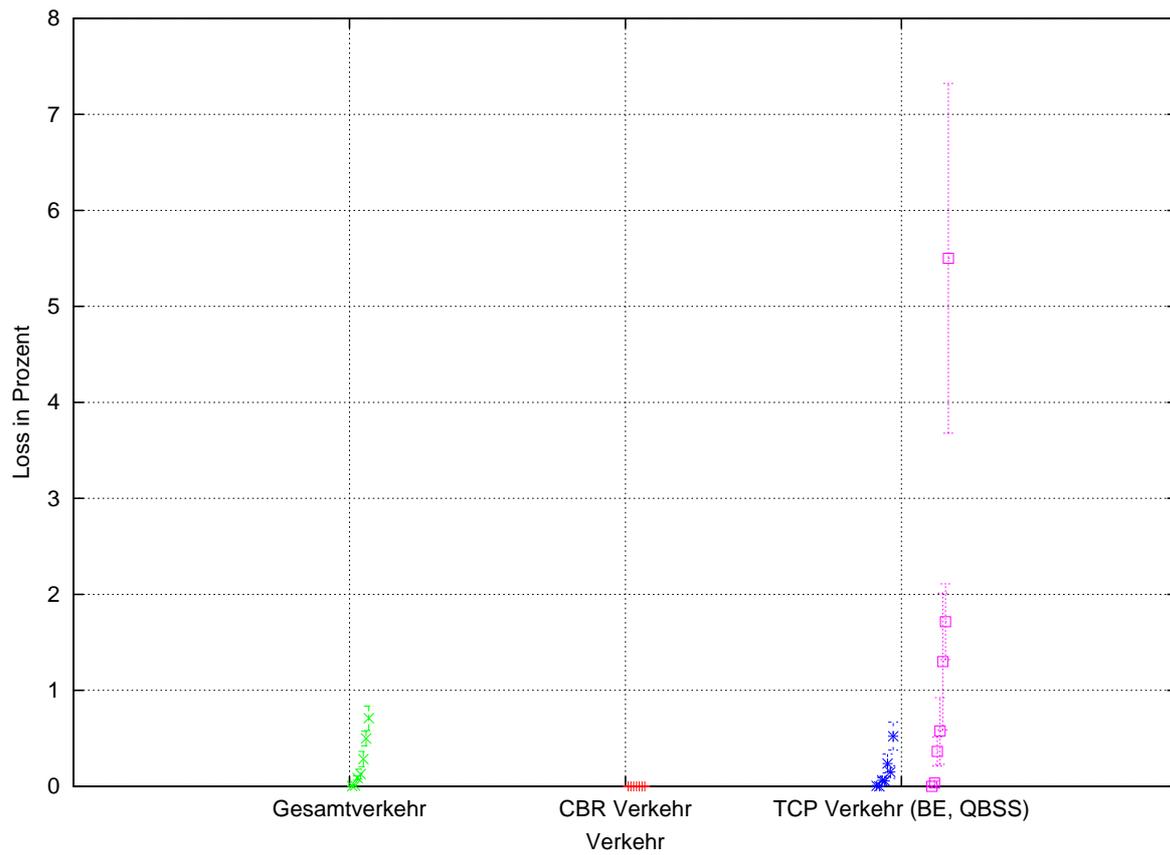
**Abbildung 10.18:** DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)



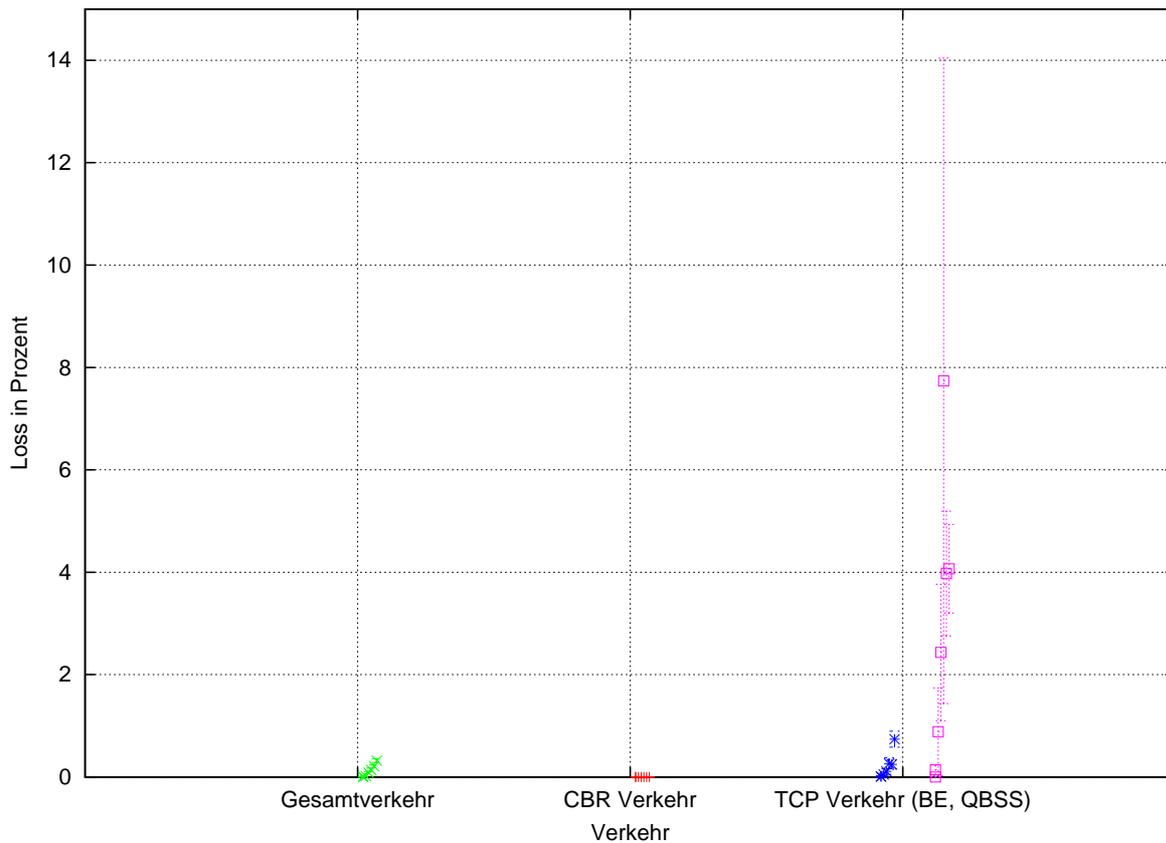
**Abbildung 10.19:** DiffServ - Olympic Ansatz: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.20:** DiffServ - Olympic Ansatz: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.21:** DiffServ - Olympic Ansatz: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.22:** DiffServ - Olympic Ansatz: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)

- **Videoverkehr mit TCP Hintergrundverkehr:**

**Lastspektrum (LS):**

$$LS = \{VLS_{VIDEO}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien (QT):**

$$QT = \{EQT_{DiffServO}\}$$

$$EQT_{DiffServO} = \{NI, EM\}$$

$$NI = \{(n_1, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ}), (n_2, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ}), (n_3, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ}), (n_4, ((\text{FIFO}, 10), (\text{FIFO}, 40), (\text{FIFO}, 50)), \text{PQ})\}$$

$$EM = (\text{Paket, per Hop / Aggregat / deterministisch / Loss / Delay, n Klassen, komplex, Router, Edge, proaktiv, ja, gut, Netzwerk})$$

**QoS Spektrum (QS):**

$$QS = \{VQS_{VIDEO}\}$$

$$VQS_{VIDEO} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Video Traces	TCP Flows
1	1, 7
10	1, 7

**Tabelle 10.8:** Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente

Simulationsergebnisse:

Delay, Jitter und Loss:

Im Gegensatz zum CBR Verkehr wurden hier durch QoS-sensitiven Verkehr hervorgerufene mittlere Auslastungen des Bottleneck Links von bis zu maximal 40 % untersucht und somit die Grenze überschritten unterhalb derer Verkehr bei einer DiffServ Technologie ohne Admission Control geregelt werden kann. Der Delay des Videoverkehrs wird auf Grund der gewählten Warteschlangenlänge für die EF Klasse begrenzt. Der Grenzwert liegt wiederum bei ca. 100 ms. Der TCP Verkehr der BE Klasse erfährt ebenfalls für diese Verkehrsart einen sehr niedrigen bzw. akzeptablen Delay (siehe Abbildungen C.29 und C.30). Hingegen sollte die QBSS Klasse nur von Anwendungen genutzt werden, die unempfindlich auf Delay und Jitter reagieren. Der Jitter erreicht durch den geringen Minimaldelay und den jetzt an seinen Grenzwert gelangten maximalen Delay einen zu hohen Wert von ca. 75 ms (siehe Abbildungen C.31 und C.32). Dies entspricht den Problemen bei den IntServ Untersuchungen. Durch die hohen, vom Videoverkehr verursachten, Auslastungen laufen die Warteschlangen der hochpriorigen EF Klasse über, und es entsteht ein sehr hoher, inakzeptabler Anteil an Loss von bis zu ca. 50 % (siehe Abbildungen C.33 und C.34). Dies liegt daran, daß die Warteschlangen sich kaum mehr leeren können und somit alle weiteren ankommenden Pakete weggeworfen werden. Dies kann nur mit Hilfe einer Admission Control verhindert werden. Der Loss des TCP Verkehrs bewegt sich für die BE Klasse bei einer mittleren Auslastung von 97 % in Bereichen von ca. 1.5 %. Bei der QBSS Klasse steigert sich dieser aber auf knapp 7 %. Dieser könnte wie bei den CBR Untersuchungen angesprochen durch Warteschlangenvergrößerungen auf Kosten eines wachsenden Delays verringert werden.

• Videoverkehr, CBR Verkehr mit TCP Hintergrundverkehr:

**Lastspektrum (LS):**

$$LS = \{VLS_{VIDEO}, VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien (QT):**

$$QT = \{EQTDiffServO\}$$

$$EQTDiffServO = \{NI, EM\}$$

$$NI = \{(n_1, ((FIFO, 10), (FIFO, 40), (FIFO, 50)), PQ), (n_2, ((FIFO, 10), (FIFO, 40), (FIFO, 50)), PQ), (n_3, ((FIFO, 10), (FIFO, 40), (FIFO, 50)), PQ), (n_4, ((FIFO, 10), (FIFO, 40), (FIFO, 50)), PQ)\}$$

$EM$  =(Paket, per Hop / Aggregat / deterministisch / Loss / Delay, n Klassen, komplex, Router, Edge, proaktiv, ja, gut, Netzwerk)

**QoS Spektrum (QS):**

$$QS = \{VQSVIDEO, VQSCBR\}$$

$$VQSVIDEO = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

$$VQSCBR = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Video Traces	CBR Flows	TCP Flows
1	1	1, 7
5	5	1, 7

**Tabelle 10.9:** Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente

Simulationsergebnisse:

## Delay, Jitter und Loss:

Die Untersuchungen des QoS-sensitiven Verkehrsmixes unterstreichen die mit CBR und Videoverkehr im Einzelnen gewonnenen Ergebnisse. Sie werden durch das vom Videoverkehr erzeugte, höhere Verkehrsaufkommen im Vergleich zum CBR Verkehr dominiert (siehe C.35, C.36, C.37, C.38, C.39 und C.40).

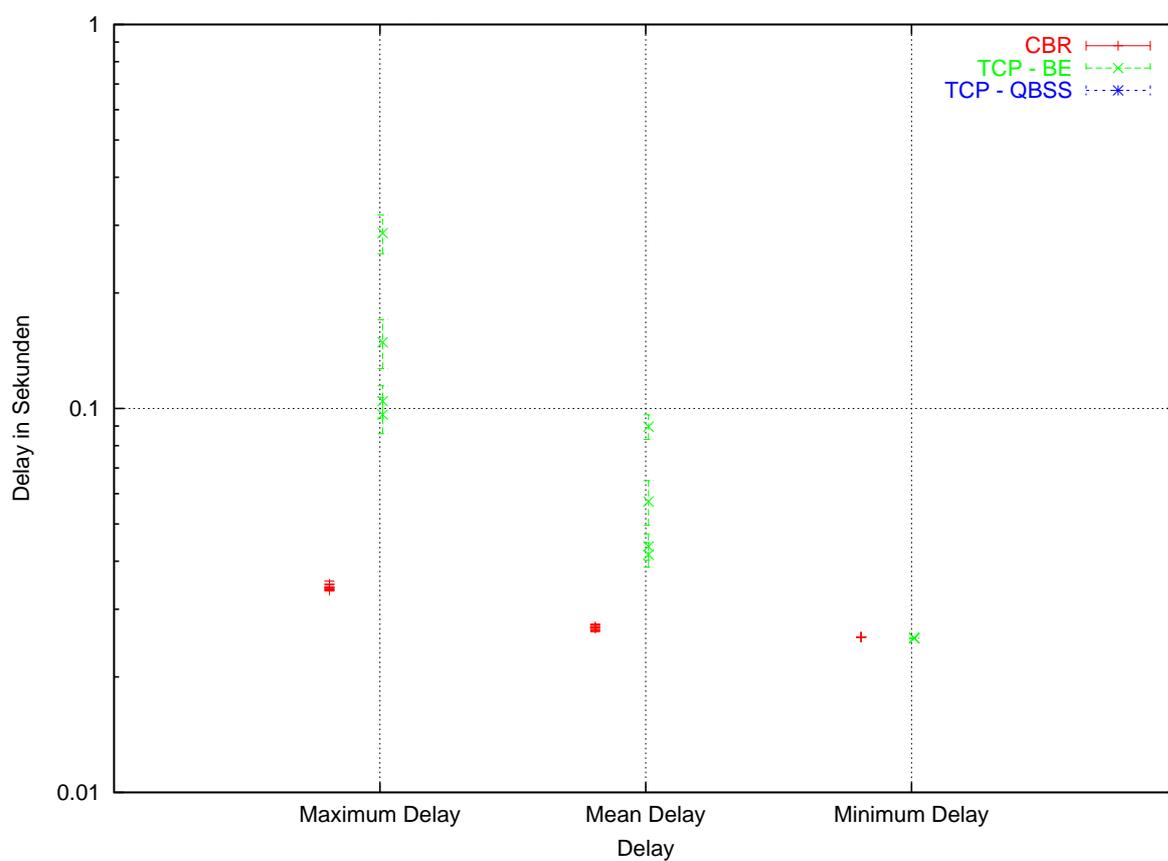
Zusammenfassung der Olympic Simulationsergebnisse:

Durch die für diese Simulationen gewählten Einstellungen überschreitet der Delay bei mittleren Auslastungen von ca. 22 % - 97 % kaum die 100 ms Grenze. Allerdings muß bei DiffServ ebenfalls wie bei IntServ, um die beabsichtigten Dienstgüten realisieren zu können, eine Admission Control ab einem bestimmten Verkehrsaufkommen des QoS-sensitiven Verkehrs eingesetzt werden. Die Grenze hierfür wird im Anschluß für CBR und Videoverkehr für die gegebenen ETS Einstellungen untersucht. Wird keine Admission Control verwendet, steigt der Loss, je nach Verkehrsaufkommen, auf sehr extreme Werte (bei einer mittleren Auslastung von ca. 97 % auf bis zu 50 %). Weiterhin ist zu beachten, daß der maximale Jitter nicht zu groß wird. Dies kann durch eine Verminderung des maximalen Delays durch eine Warteschlangenverkürzung erreicht werden, die allerdings ebenfalls wiederum ein früheres Einsetzen einer Admission Control mit sich bringt. Der TCP Verkehr der BE Klasse erfährt akzeptablen Delay und Jitter. Die QBSS Klasse wird extremem Delay und Jitter unterworfen, allerdings sollen die Anwendungen, die diese Klasse gewählt haben auf diese Faktoren nicht sensitiv reagieren. Ein Problem kann allerdings für beide Klassen der hohe Lossanteil darstellen, der ebenfalls bei extremen Auslastungen auftreten kann. Hier reichte die eigens für die QBSS Klasse bereits im Voraus größer gewählte Warteschlange

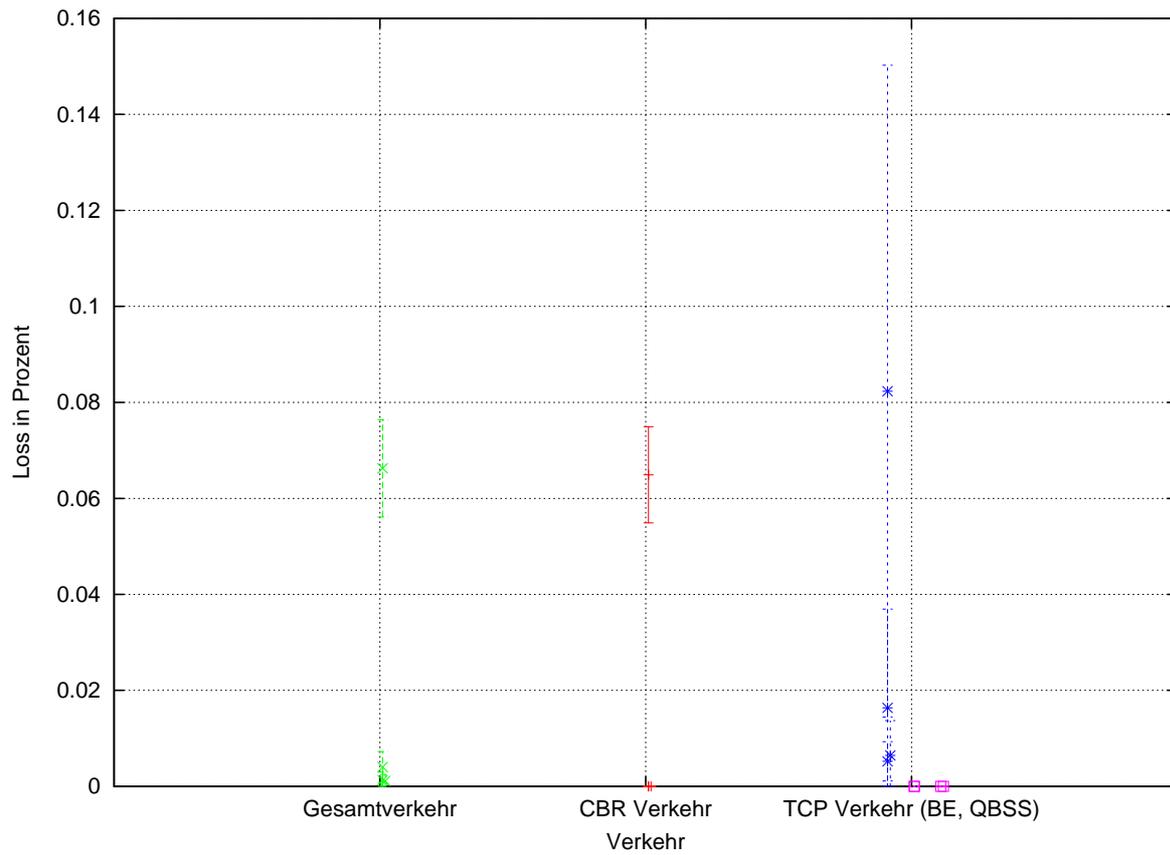
nicht aus und der hohe Loss könnte durch eine weitere Erweiterung der Warteschlangenlängen auf Kosten einer Delay- und Jittererhöhung oder ebenfalls durch das Einsetzen einer Admission Control verhindert werden.

- **AC Grenze für CBR Verkehr mit TCP Hintergrundverkehr:**

Die Abbildungen 10.23 und 10.24 zeigen, daß für diese Verkehrszusammensetzung von 50 CBR Flows und 1 TCP Flow die Grenze, an der ein Überlauf der Warteschlange des hochprioriten Verkehrs stattfindet bei einer mittleren Gesamtauslastung des Bottleneck Links von ca. 65 %, davon ca. 50 % Auslastung hervorgerufen durch CBR Verkehr, noch nicht erreicht wird. Leider stieß das Simulationssystem bzgl. Speicherkapazität an seine Grenzen und es konnte die Flow Anzahl nicht noch mehr gesteigert werden.



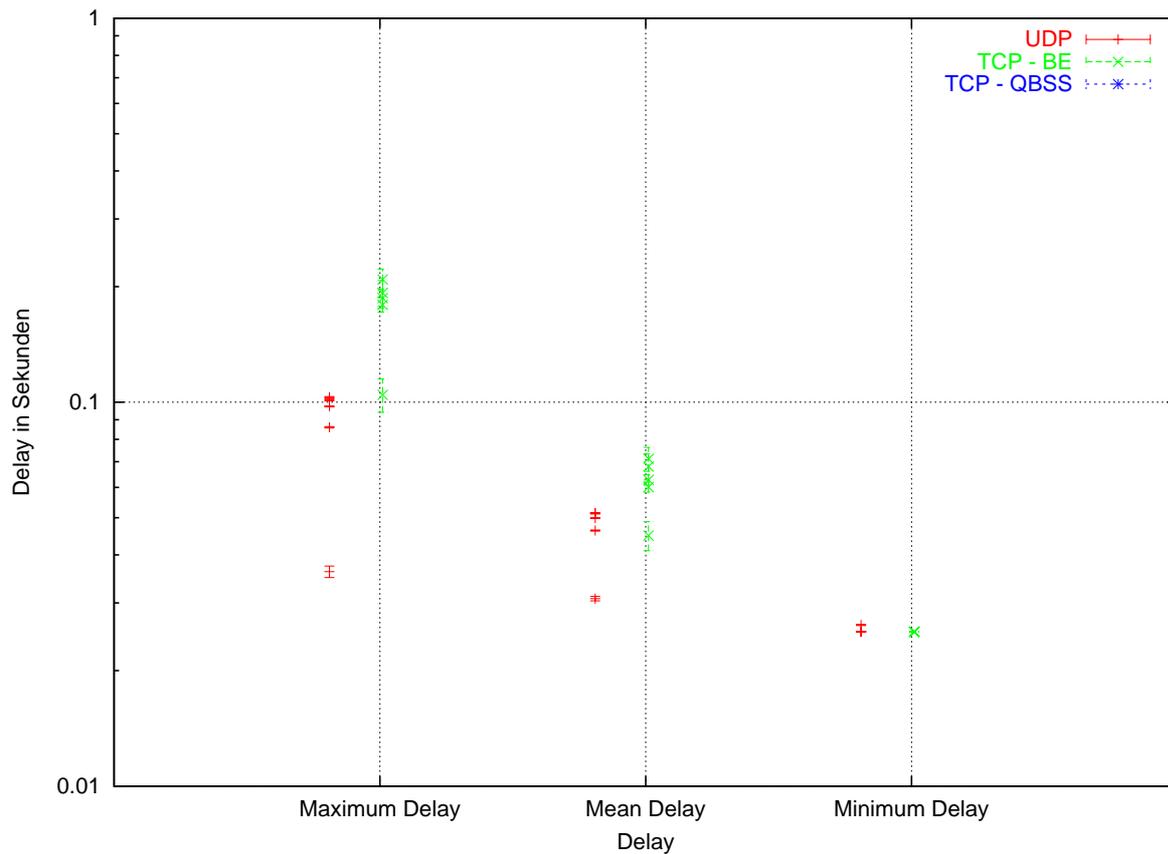
**Abbildung 10.23:** DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von CBR Flows und einem TCP Flow



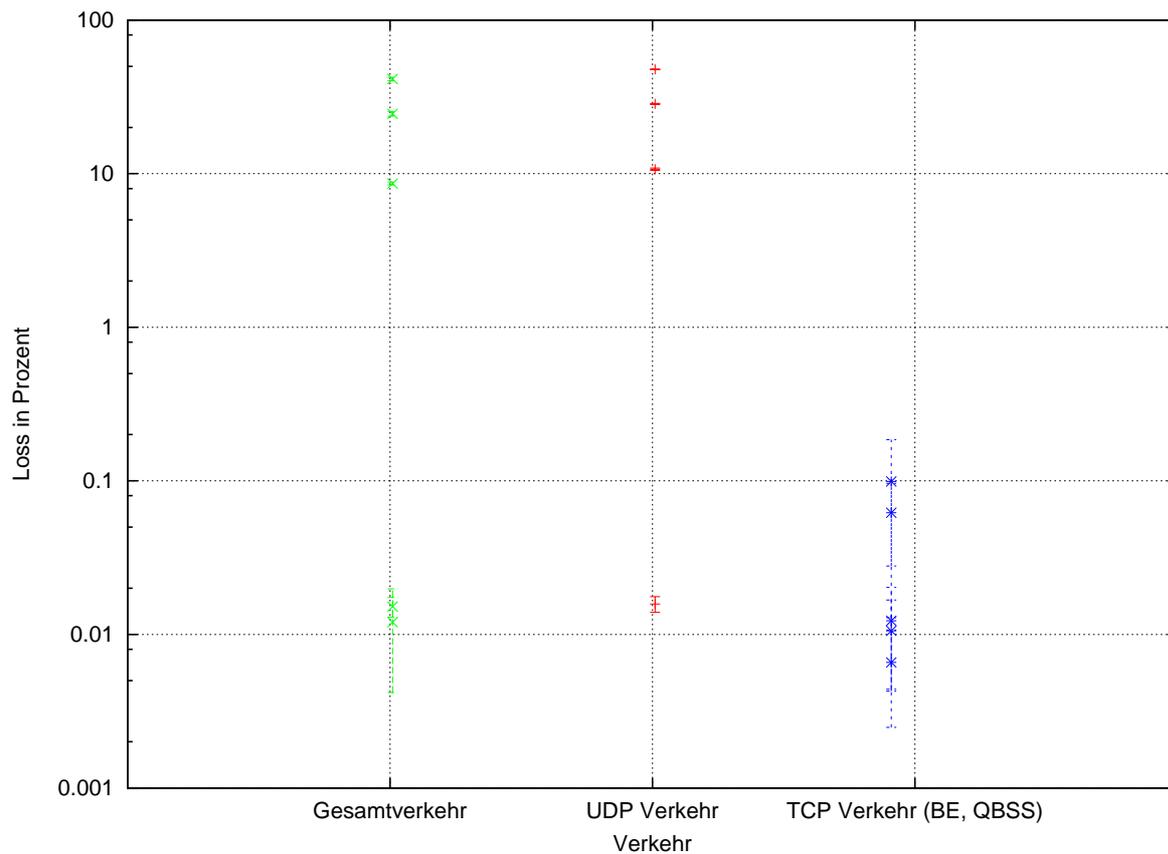
**Abbildung 10.24:** DiffServ - Olympic Ansatz: Loss bei variierender Anzahl von CBR Flows und einem TCP Flow

- **AC Grenze für UDP Verkehr mit TCP Hintergrundverkehr:**

Bei UDP Verkehr ist die Grenze, an der ein Überlauf der Warteschlange des hochpriorien Verkehrs stattfindet und bei der eine AC unumgänglich ist, bei einer mittleren Gesamtauslastung des Bottleneck Links von ca. 50 %, davon ca. 34 % hervorgerufen durch Videoverkehr, bereits überschritten. Dies entspricht einem Verkehrsmix von 5 Video Traces und 1 TCP Flow. Der maximale Ende-zu-Ende Delay beträgt ca. 100 ms, der Jitter ca. 75 ms und der Loss an UDP Paketen steigt auf ca. 30 % an.



**Abbildung 10.25:** DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von Video Traces und einem TCP Flow



**Abbildung 10.26:** DiffServ - Olympic Ansatz: Loss bei variierender Anzahl von Video Traces und einem TCP Flow

#### 10.2.4 Simulationen mit DiffServ (Default Ansatz)

Bei dem hier untersuchten Default Ansatz werden für die einzelnen Warteschlangen folgende Werte eingesetzt: EF = 10, AF1 = 13, AF2 = 13, AF3 = 13. Hier wird die BE Warteschlangenlänge von 50 auf die vier Klassen aufgeteilt, wobei, wie in Kapitel 9 beschrieben, für die Klassen AF2 und AF3 RED Warteschlangen verwendet werden. Die Gewichte für den WRR Algorithmus sind wie folgt verteilt: EF = 900, AF1 = 3, AF2 = 3, AF3 = 4. Diese Werte basieren auf Ergebnissen zahlreicher, im NS2 durchgeführter, Simulations-Testläufe, die eine geeignete Wahl der Gewichte aller Klassen hinsichtlich ihrer Anforderungen und QoS Sensitivität zum Ziel hatte.

##### • CBR-Verkehr mit TCP Hintergrundverkehr:

###### Lastspektrum ( $LS$ ):

$$LS = \{VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$

**QoS Technologien (QT):**

$QT = \{EQTDiffServD\}$

$EQTDiffServD = \{NI, EM\}$

$NI = \{(n_1, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13)), \text{WRR}), (n_2, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}), (n_3, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}), (n_4, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR})\}$

$EM = (\text{Paket, per Hop / Aggregat / deterministisch / Loss / Delay, n Klassen, komplex, Router, Edge, proaktiv, ja, gut, Netzwerk})$

**QoS Spektrum (QS):**

$QS = \{VQS_{CBR}\}$

$VQS_{CBR} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$

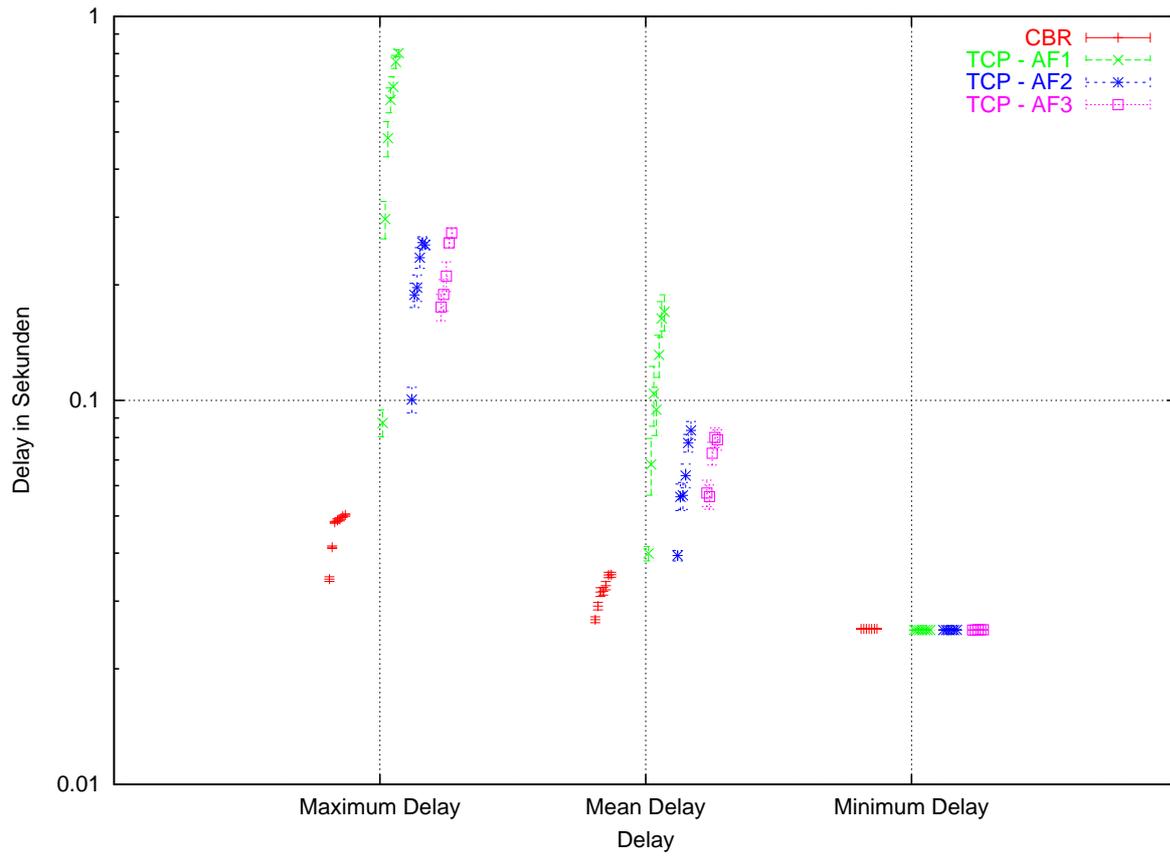
CBR Flows	TCP Flows
1	1, 2, ..., 7
10	1, 2, ..., 7

**Tabelle 10.10:** Variation der Anzahl von CBR Flows und TCP Flows innerhalb der Simulationsexperimente

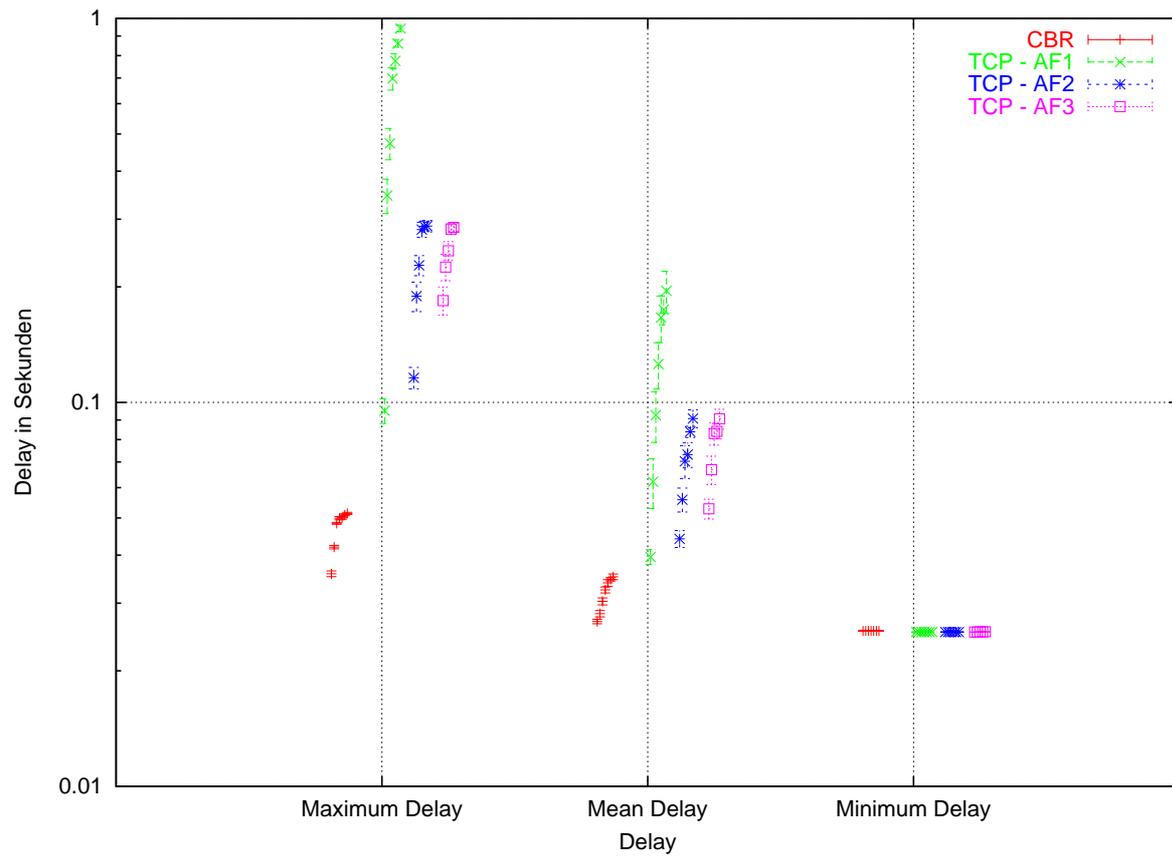
Simulationsergebnisse:

Delay, Jitter und Loss:

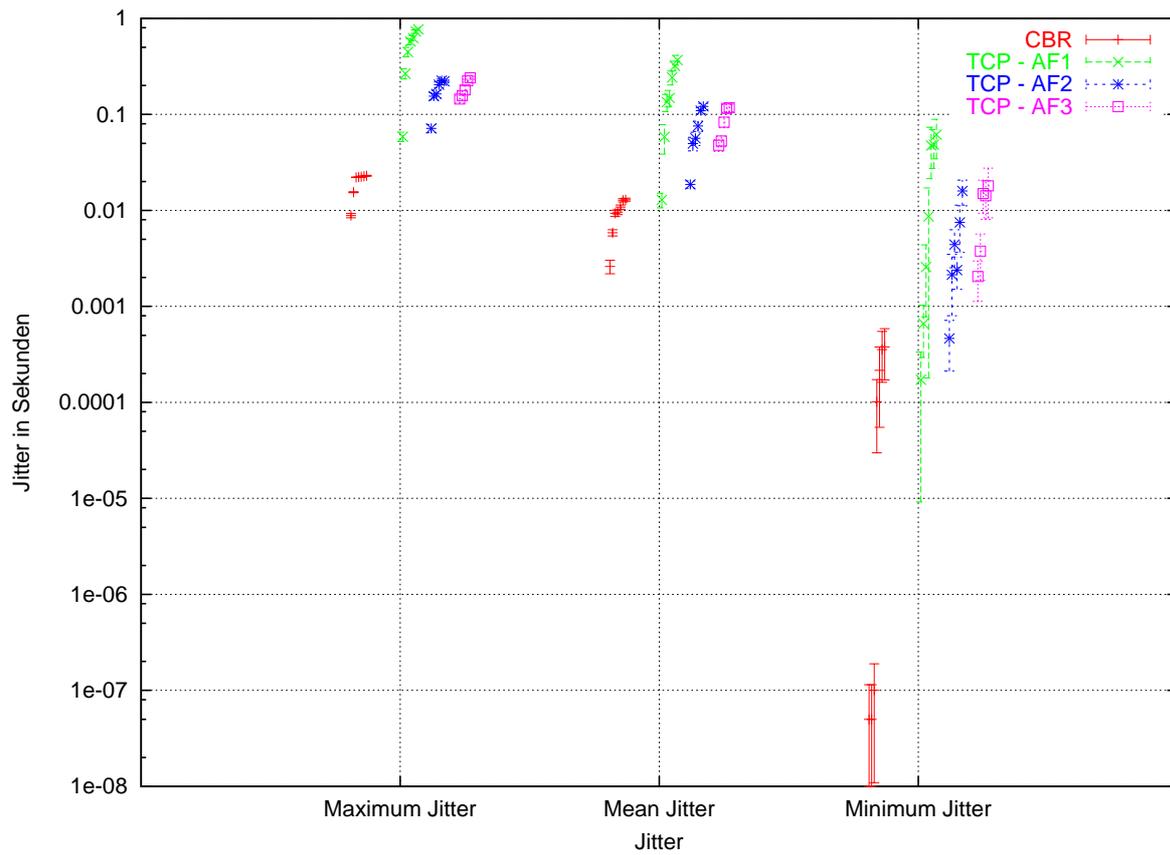
Für die relativ niedrigen Verkehrsanteile des CBR Verkehrs (1 % - 10 % der mittleren Auslastung) liefert der Default Ansatz sehr gute Delaywerte bis zu maximal ca. 50 ms (siehe Abbildungen 10.27 und 10.28). Auch der Delay für alle drei AF Klassen liegt unter 1 s. Daher liegt auch der Jitter für diese Auslastungen in akzeptablen Bereichen (siehe Abbildungen 10.29 und 10.30). Loss tritt allerdings im Gegenteil zum Olympic Ansatz auf Grund des verwendeten WRR im Gegensatz zur strikten Priorität bis zu einem Wert von ca. 1 % für die EF Klasse bei einer mittleren Auslastung von 91 % auf (siehe Abbildungen 10.31 und 10.32). Der Loss der AF Klassen ist noch um einiges höher und müßte durch längere Warteschlangen und/oder durch eine Admission Control reduziert werden.



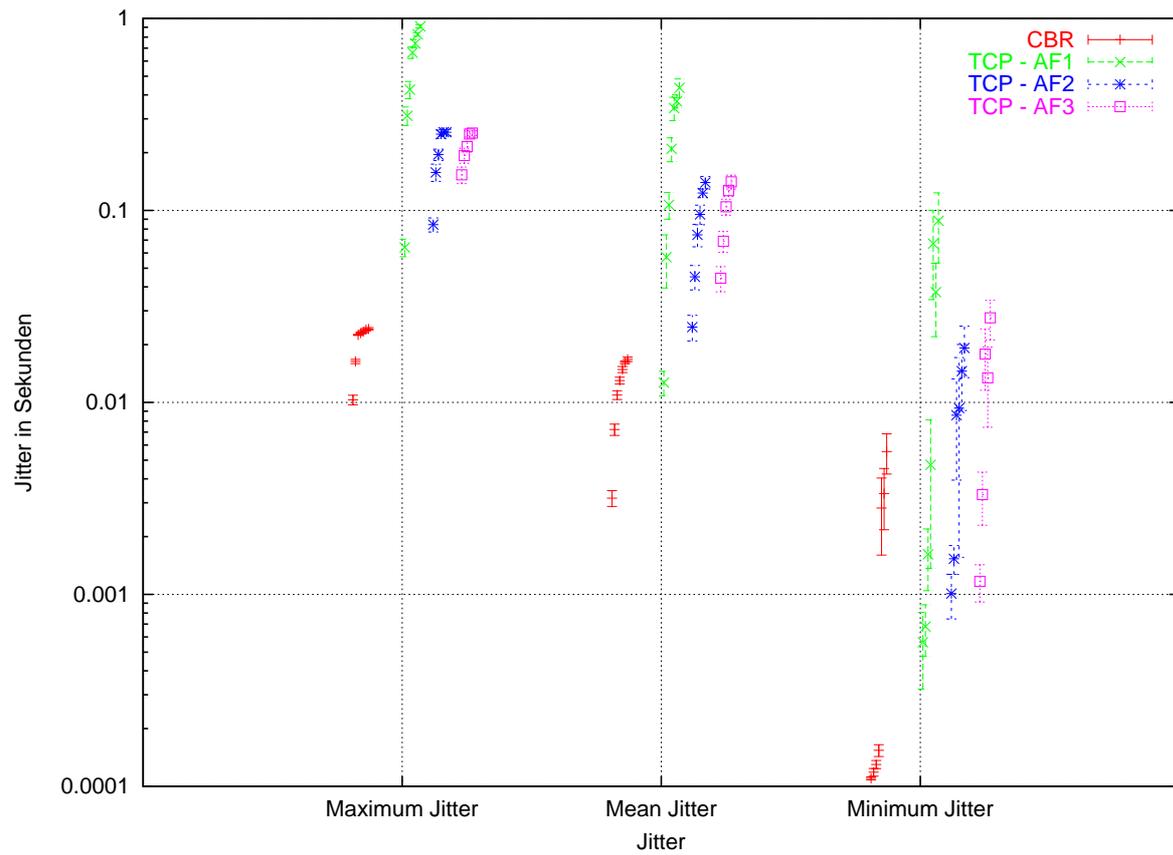
**Abbildung 10.27:** DiffServ - Default Ansatz: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)



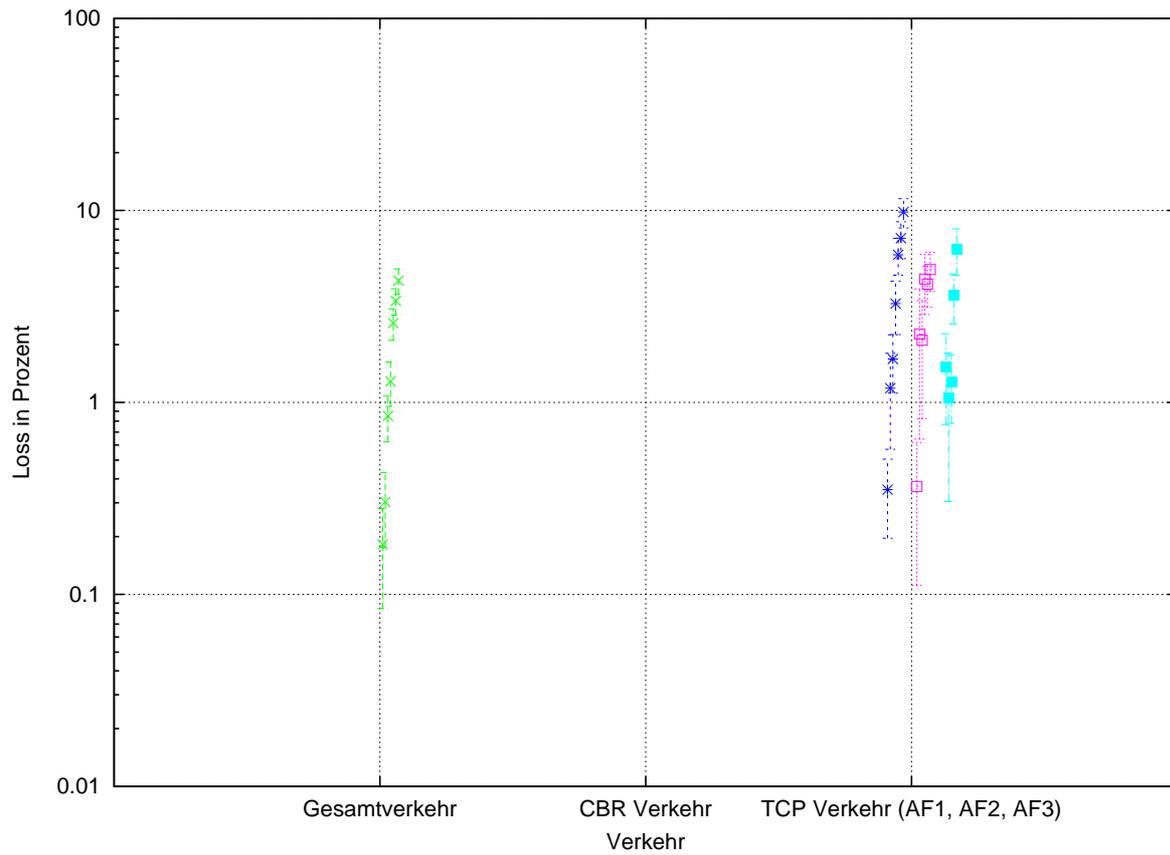
**Abbildung 10.28:** DiffServ - Default Ansatz: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)



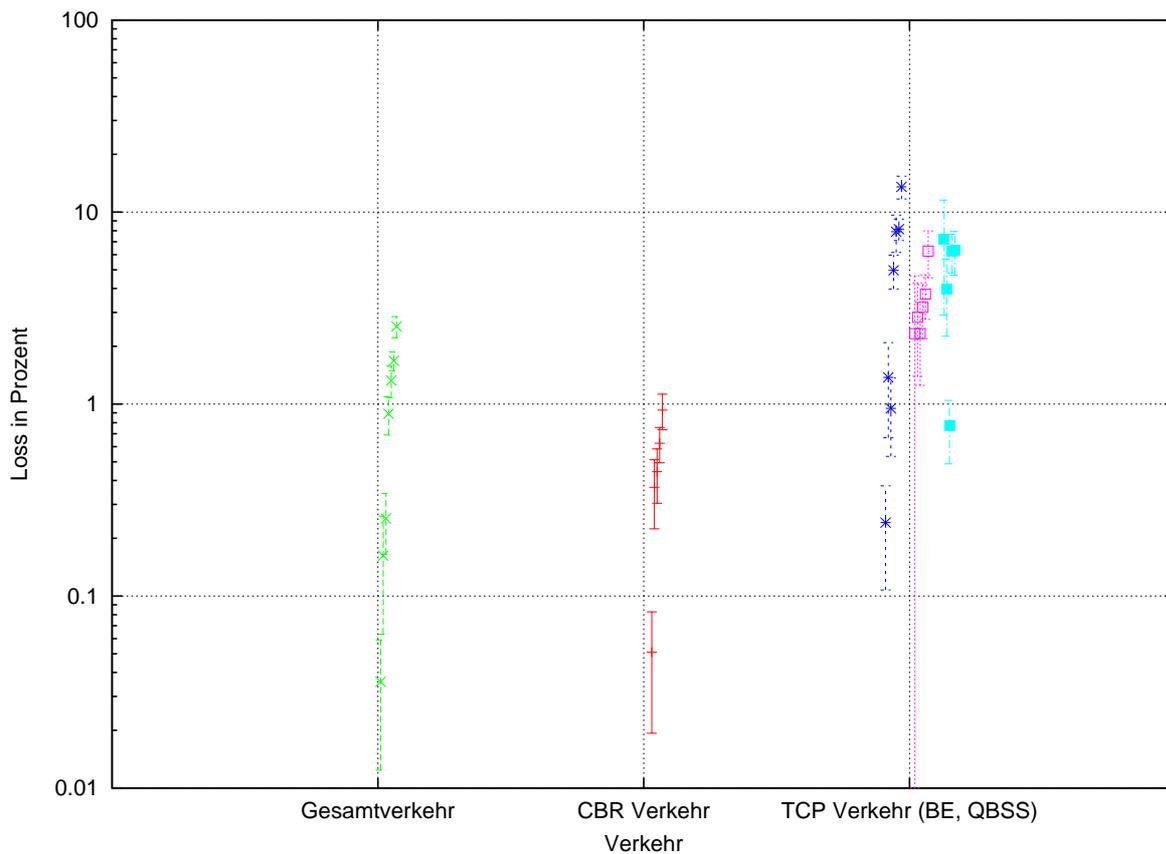
**Abbildung 10.29:** DiffServ - Default Ansatz: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.30:** DiffServ - Default Ansatz: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.31:** DiffServ - Default Ansatz: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7)



**Abbildung 10.32:** DiffServ - Default Ansatz: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7)

- **Videoverkehr mit TCP Hintergrundverkehr:**

**Lastspektrum ( $LS$ ):**

$$LS = \{VLS_{VIDEO}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien ( $QT$ ):**

$$QT = \{EQT_{DiffServD}\}$$

$$EQT_{DiffServD} = \{NI, EM\}$$

$$NI = \{(n_1, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}, (n_2, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}, (n_3, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}, (n_4, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}\}$$

$$EM = (\text{Paket, per Hop / Aggregat / deterministisch / Loss / Delay, n Klassen, komplex, Router, Edge, proaktiv, ja, gut, Netzwerk})$$

**QoS Spektrum (QS):**

$$QS = \{VQS_{VIDEO}\}$$

$$VQS_{VIDEO} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Video Traces	TCP Flows
1	1, 7
10	1, 7

**Tabelle 10.11:** Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente

Simulationsergebnisse:

## Delay, Jitter und Loss:

Der Delay des Videoverkehrs steigt bei mittleren Auslastungen von 36 % bis 96 % kaum über die 100 ms Grenze. Der Delay der einzelnen AF Klassen bleibt auch unter 600 ms in einem akzeptablen Bereich (siehe Abbildungen C.41 und C.42). Der Jitter steigt allerdings beim Video Verkehr auf Grund der Delayschwankung ab einer mittleren Auslastung von 42 % auf über 50 ms bis zu einem Maximum von ca. 80 ms bei einer mittleren Auslastung von 96 % an (siehe Abbildungen C.43 und C.44). Ebenso steigt der Loss für Videoverkehr bei den untersuchten Höchstauslastungen durch überlaufende und sich kaum mehr leerende Warteschlangen auf Werte von bis zu 50 % an. Der TCP Loss erreicht auch zu hohe Werte von bis zu 10 % (siehe Abbildungen C.45 und C.46).

- Videoverkehr, CBR Verkehr mit TCP Hintergrundverkehr:

**Lastspektrum (LS):**

$$LS = \{VLS_{VIDEO}, VLS_{CBR}, VLS_{TCP}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_5)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{CBR} = (SD_{CBR}, TT_{CBR})$$

$$SD_{CBR} = \{(n_0, n_5)\}$$

$$TT_{CBR} = \{\text{CBR Generator}\}$$

$$VLS_{TCP} = \{SD_{TCP}, TT_{TCP}\}$$

$$SD_{TCP} = \{(n_0, n_5)\}$$

$$TT_{TCP} = \{\text{Hintergrundverkehr, TCP Generator}\}$$

**QoS Technologien (QT):**

$$QT = \{EQT_{DiffServD}\}$$

$$EQT_{DiffServD} = \{NI, EM\}$$

$$NI = \{(n_1, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}), (n_2, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}), (n_3, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR}), (n_4, ((\text{FIFO}, 10), (\text{RED}, 13), (\text{RED}, 13), (\text{FIFO}, 13))), \text{WRR})\}$$

EM = (Paket, per Hop / Aggregat / deterministisch / Loss / Delay, n Klassen, komplex, Router, Edge, proaktiv, ja, gut, Netzwerk)

**QoS Spektrum (QS):**

$$QS = \{VQS_{VIDEO}, VQS_{CBR}\}$$

$$VQS_{VIDEO} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

$$VQS_{CBR} = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

Video Traces	CBR Flows	TCP Flows
1	1	1, 7
5	5	1, 7

**Tabelle 10.12:** Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente

#### Simulationsergebnisse:

##### Delay, Jitter und Loss:

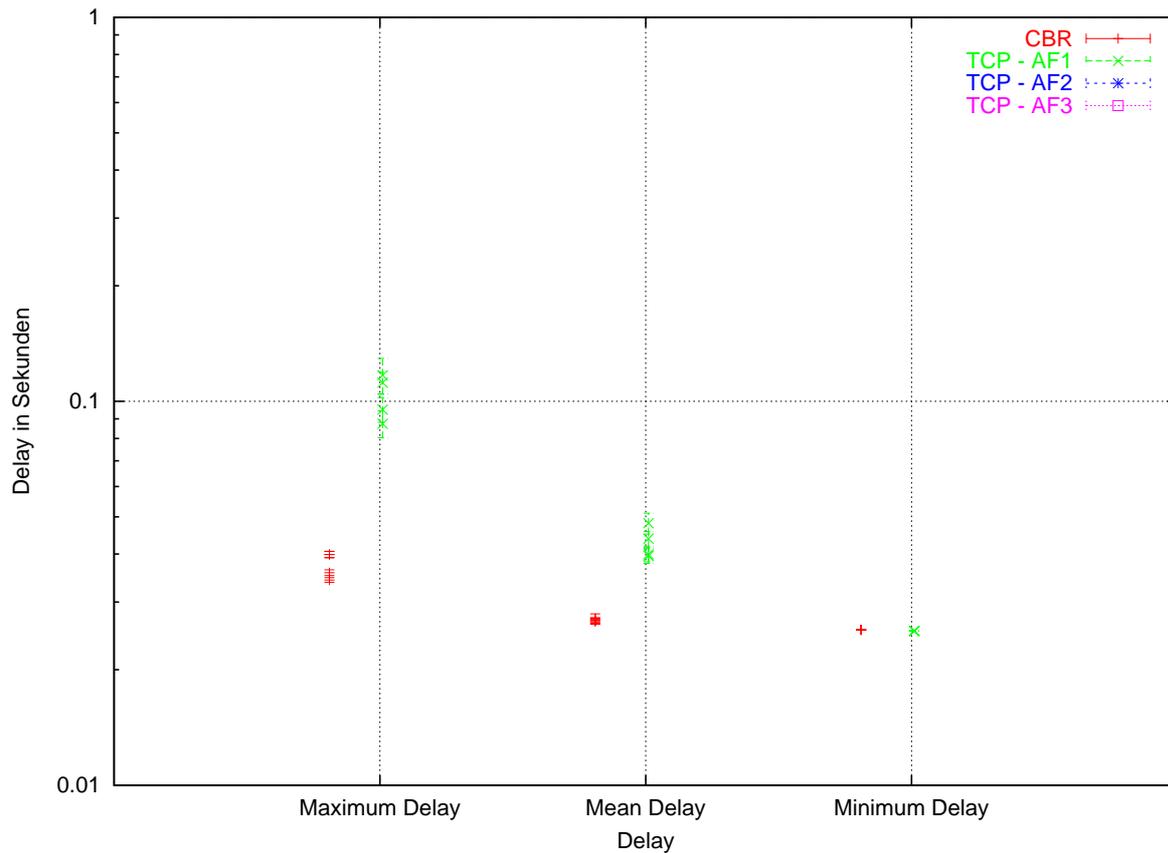
Die Untersuchungen des QoS-sensitiven Verkehrsmixes unterstreichen die mit CBR und Videoverkehr im Einzelnen gewonnen Ergebnisse. Sie werden durch das vom Videoverkehr erzeugte, höhere Verkehrsaufkommen im Vergleich zum CBR Verkehr dominiert (siehe Abbildungen C.47, C.48, C.49, C.50, C.51 und C.52).

##### Zusammenfassung der Default Simulationsergebnisse:

Der Delay des QoS-sensitiven Verkehrs überschreitet durch die gewählten Einstellungen für mittlere Auslastungen von 22 % bis 96 % kaum die 100 ms Grenze. Auch der TCP Delay für alle drei AF Klassen bleibt stets unterhalb von 1 s. Probleme tauchen allerdings, wie auch beim Olympic Ansatz, für den sensitiven Verkehr beim Jitter auf. Dieser müßte durch eine Reduktion des maximalen Delays herabgesetzt werden, was durch eine Warteschlangenreduktion erreicht werden kann. Der Loss steigt auf Grund der überlaufenden und sich kaum mehr leerenden Warteschlangen bei den untersuchten Höchstauslastungen auf bis zu 50 % beim QoS-sensitiven Verkehr an. Dies muß durch eine Admission Control verhindert werden. Die Grenze ab der eine AC verwendet werden sollte wird im Anschluß untersucht. Auch der Loss der drei AF Klassen steigt bis auf 10 % an und muß ebenfalls durch Admission Control oder durch Warteschlangenverlängerungen je nach Anwendungsanforderungen reduziert werden.

- **AC Grenze für CBR Verkehr mit TCP Hintergrundverkehr:**

Die Abbildungen zeigen, daß für diese Verkehrszusammensetzung von ca. 60% mittlerer Gesamtauslastung des Bottleneck Links, davon ca. 47 % hervorgerufen durch den CBR Verkehr, die Warteschlangen bereits übergelaufen sind, der maximale Ende-zu-Ende Delay an sein Maximum von ca. 40 ms gelangt, der Jitter auf ca. 15 ms und der Loss auf ca. 12 % ansteigt. Das entspricht einer Verkehrszusammensetzung aus 50 CBR und 1 TCP Flow. Dies ist der Punkt an dem für dieses Extrapolationsszenario eine AC bereits unumgänglich ist.



**Abbildung 10.33:** DiffServ - Default Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von CBR Flows und einem TCP Flow

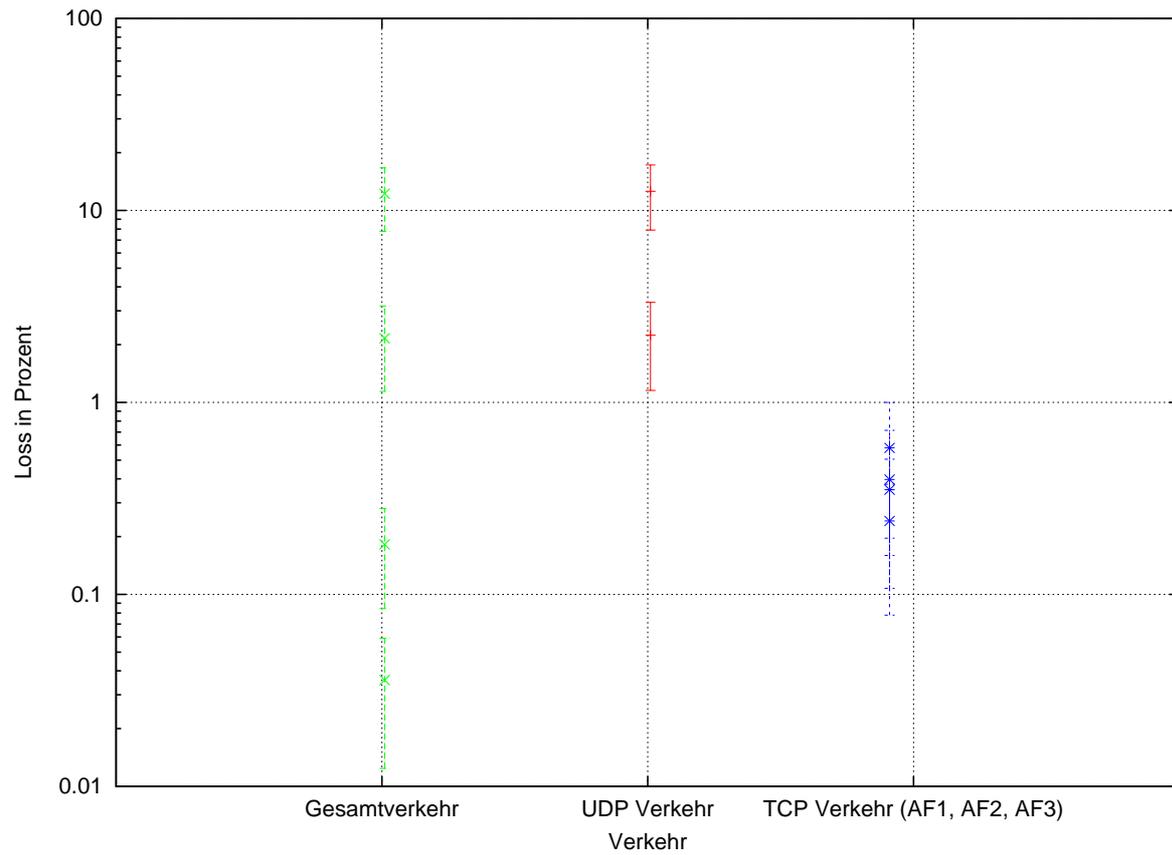
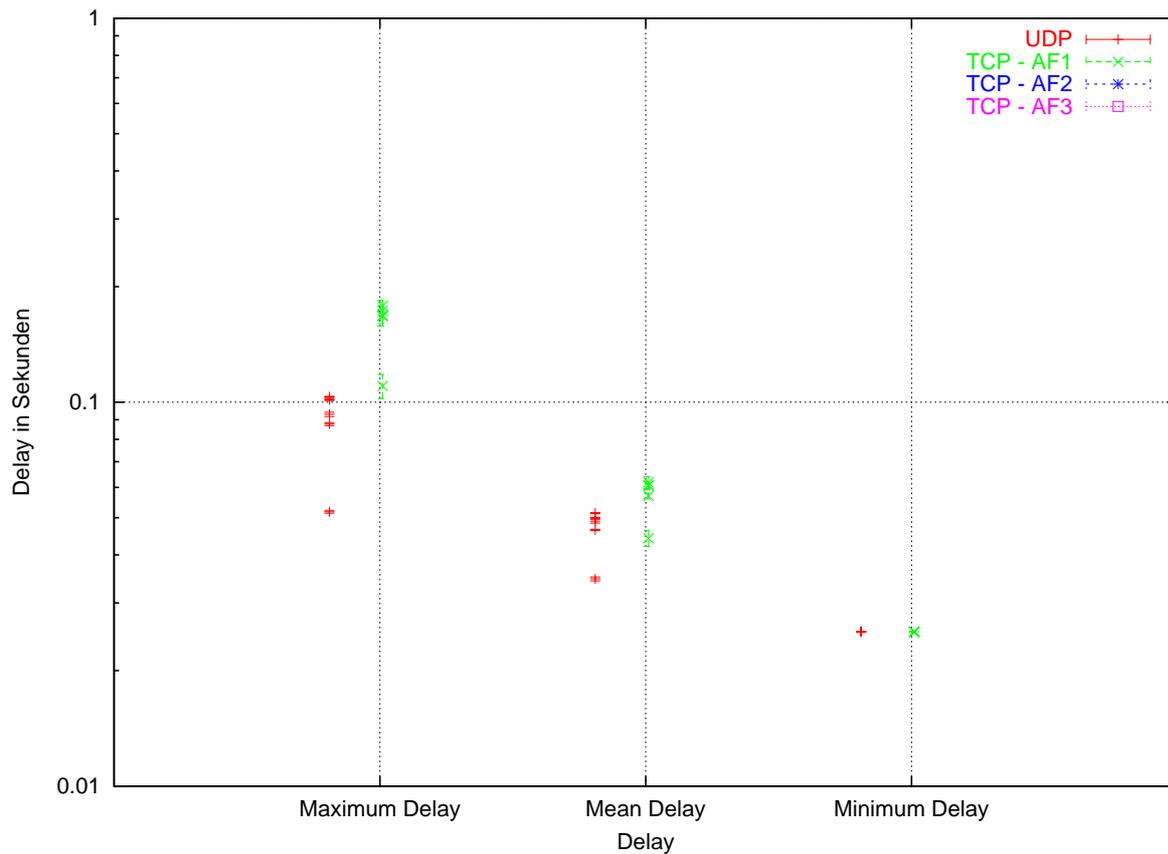


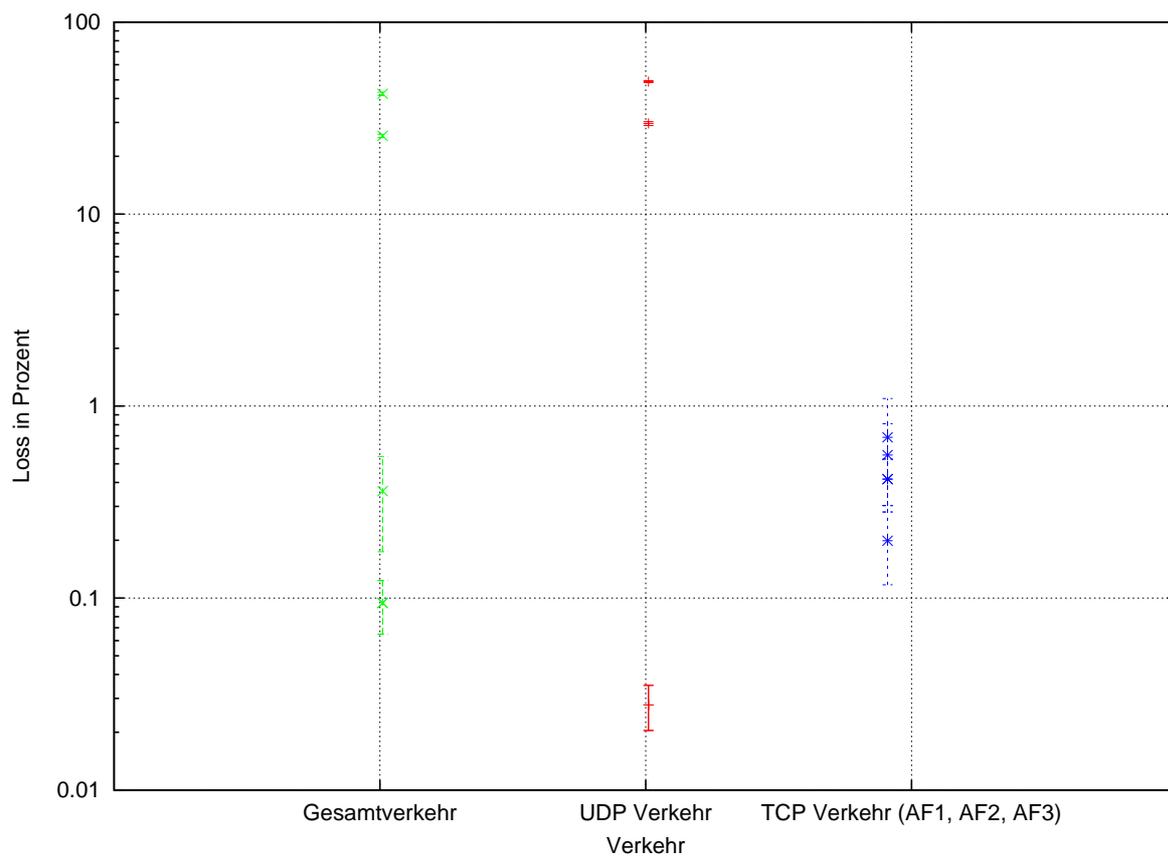
Abbildung 10.34: DiffServ - Default Ansatz: Loss bei variierender Anzahl von CBR Flows und einem TCP Flow

- **AC Grenze für UDP Verkehr mit TCP Hintergrundverkehr:**

Bei UDP Verkehr ist die Grenze, an der ein Überlauf der Warteschlange des hochpriorigen Verkehrs stattfindet und bei der eine AC unumgänglich ist, bei einer mittleren Gesamtauslastung des Bottleneck Links von ca. 50 %, davon ca. 34 % hervorgerufen durch Videoverkehr, bereits überschritten. Dies entspricht einem Verkehrsmix von 5 Video Traces und 1 TCP Flow. Der maximale Ende-zu-Ende Delay beträgt ca. 100 ms, der Jitter ca. 75 ms und der Loss an UDP Paketen steigt auf ca. 30 % an.



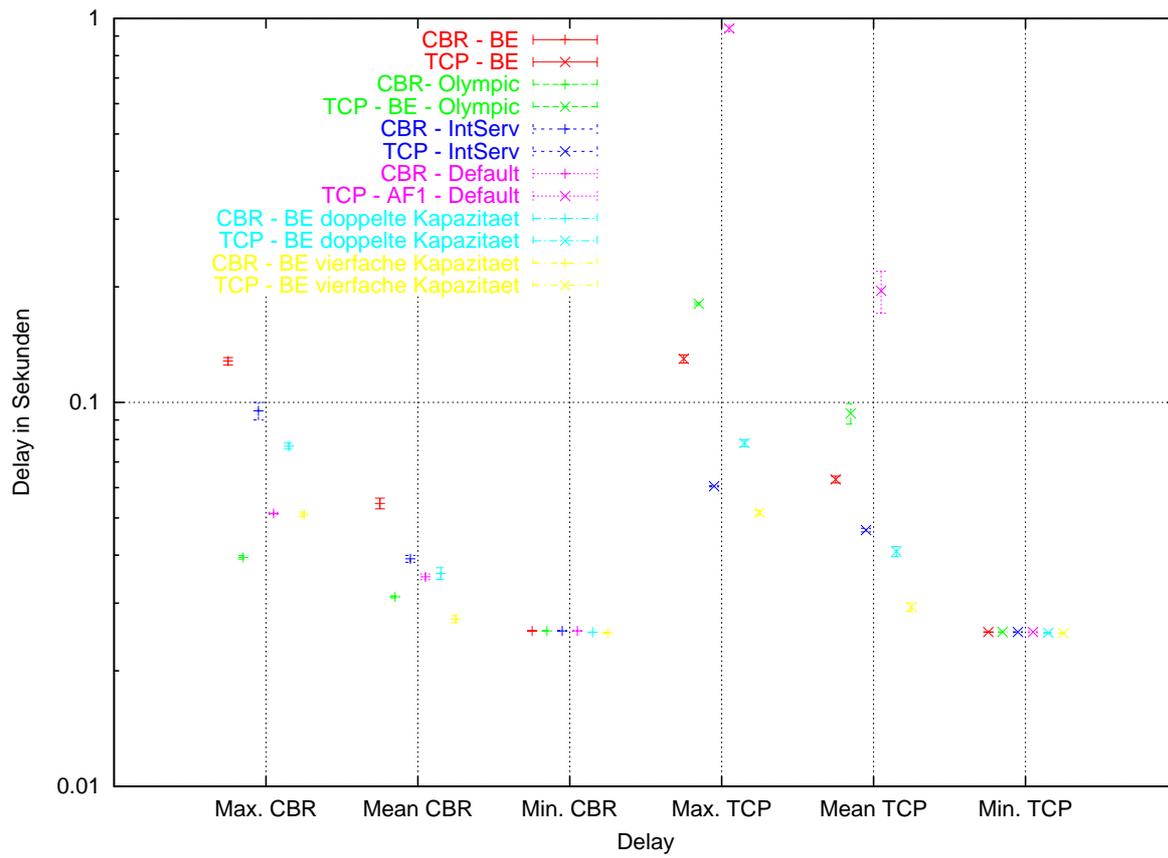
**Abbildung 10.35:** DiffServ - Default Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von Video Traces und einem TCP Flow



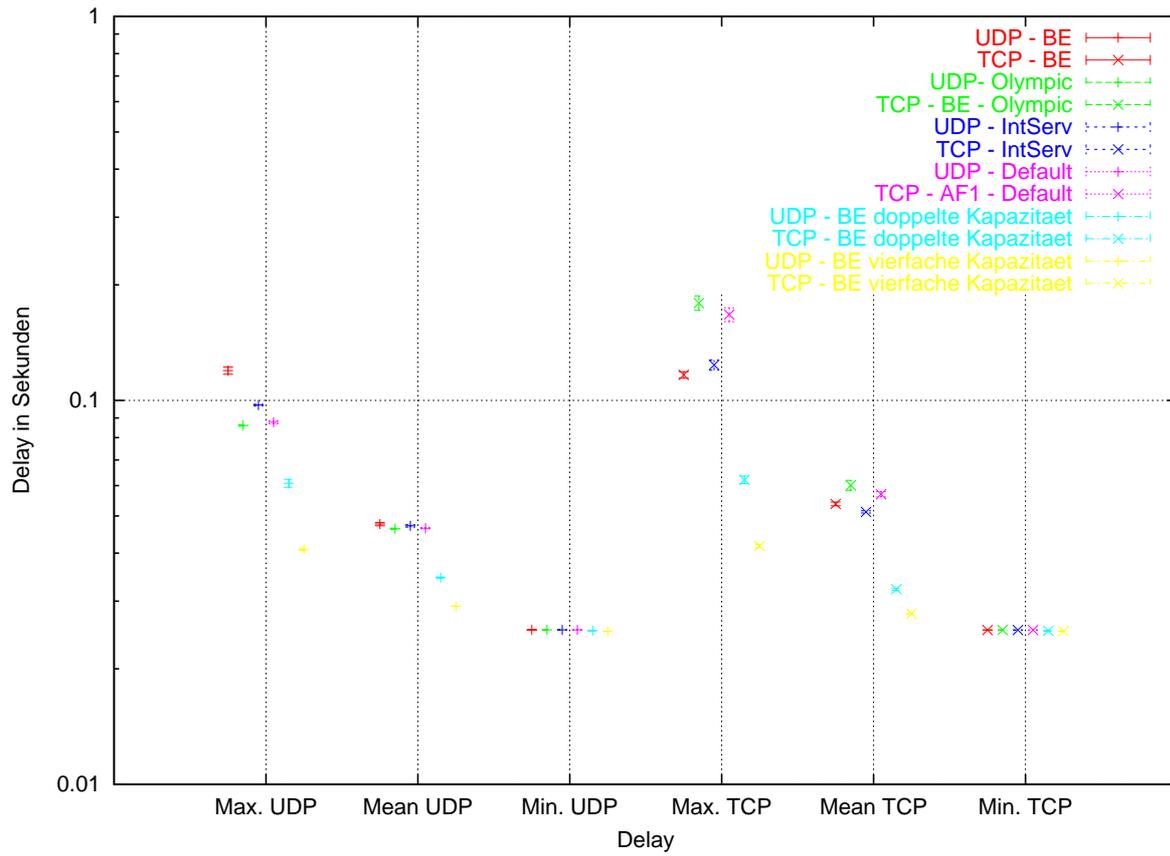
**Abbildung 10.36:** DiffServ - Default Ansatz: Loss bei variierender Anzahl von Video Traces und einem TCP Flow

### 10.2.5 Overprovisioningexperimente

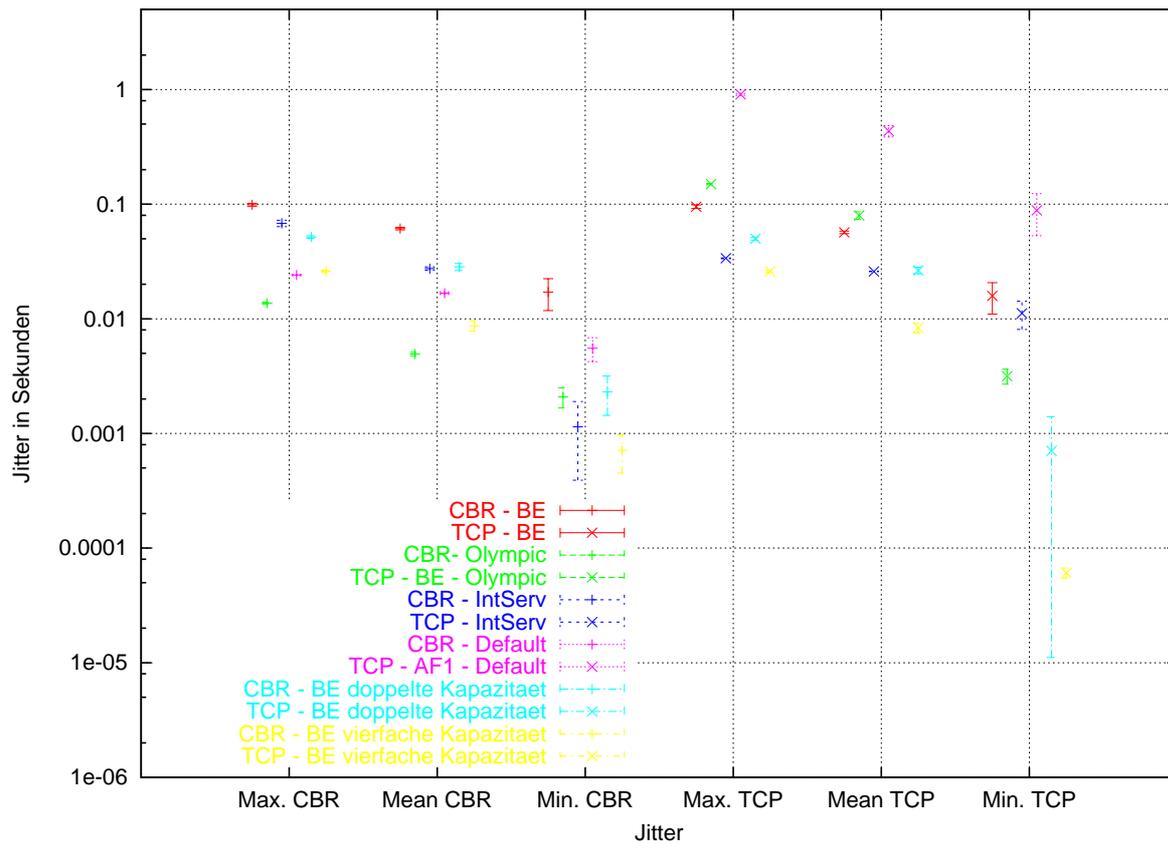
Es werden für eine Bandbreitenverdopplung und -vervierfachung aller Links des ETS die Simulationsergebnisse mit den zuvor mit den Ausgangsbandbreiten erzielten Ergebnissen von BE, IntServ und von den beiden DiffServ Ansätzen verglichen. Die Ergebnisse werden an den Verkehrszusammensetzungen 10 CBR Flows, 7 TCP Flows (entspricht einer mittleren Auslastung von ca. 91 %) und 3 Video Traces und 1 TCP Flow (entspricht einer mittleren Auslastung von ca. 42 %) beispielhaft veranschaulicht. Für den Vergleich des TCP Verkehrs wird für den Olympic Ansatz die BE Klasse und für den Default Ansatz die AF1 Klasse herangezogen. Diese Ergebnisse sind zwar nicht direkt mit denen der unaufgeteilten TCP Anwendungen unter BE zu vergleichen, können aber zumindest zur Beurteilung mit herangezogen werden. Die anderen TCP Klassen der beiden DiffServ Ansätze werden zu Gunsten der Übersichtlichkeit in den Grafiken nicht mit dargestellt.



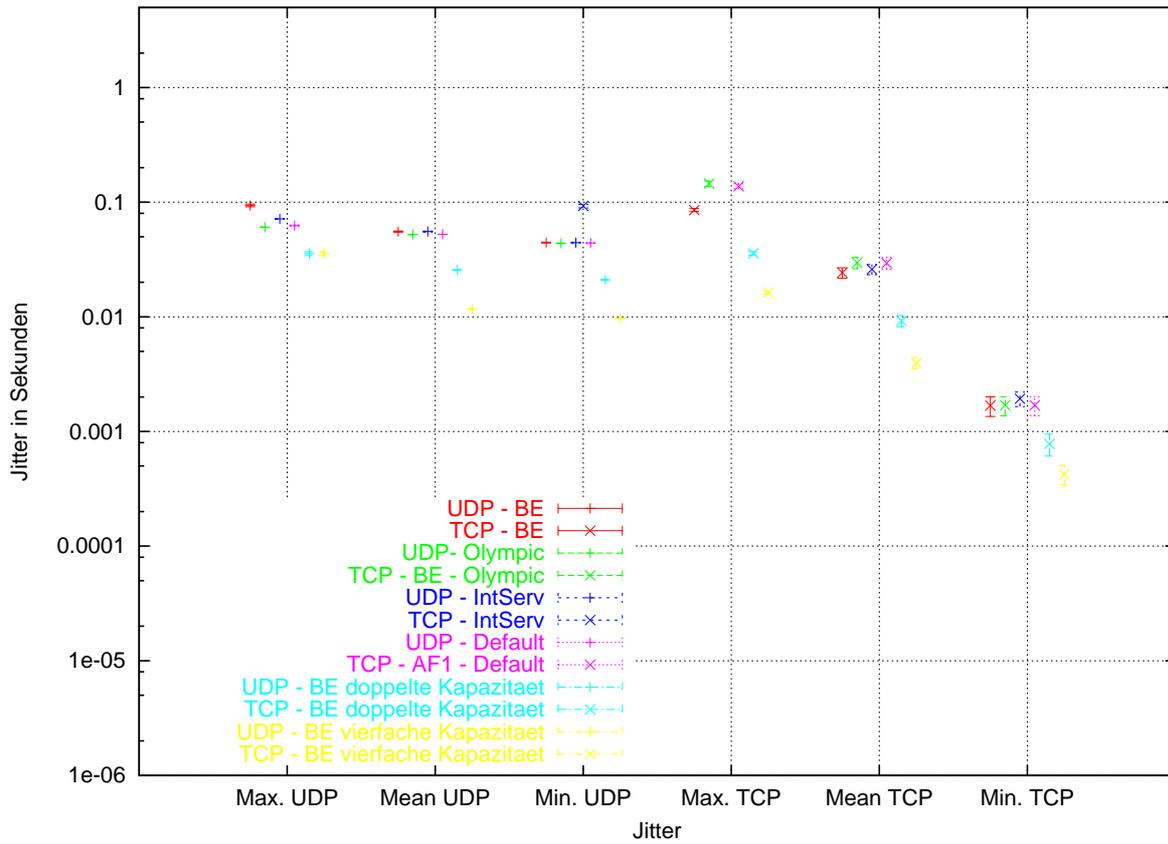
**Abbildung 10.37:** Overprovisioning: Vergleich des Ende-zu-Ende Delays bei 10 CBR und 7 TCP Flows mit und ohne Overprovisioning



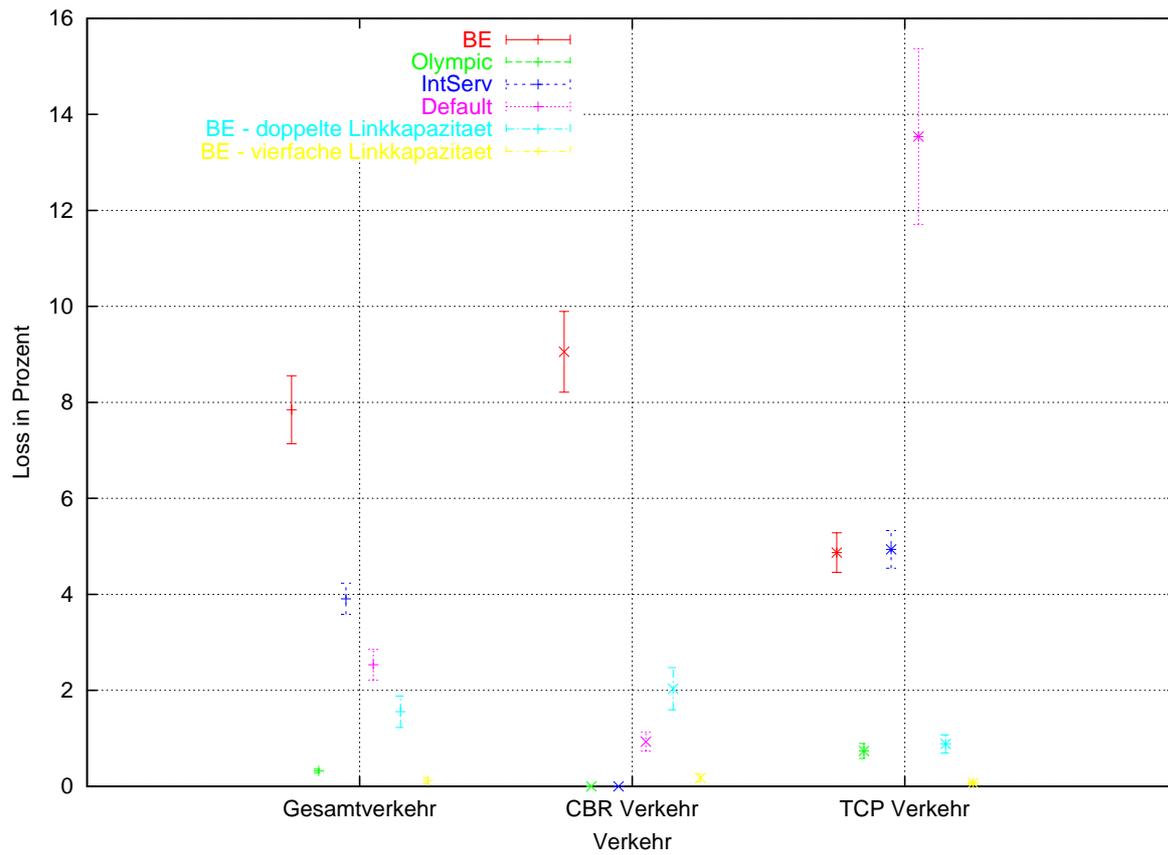
**Abbildung 10.38:** Overprovisioning: Vergleich des Ende-zu-Ende Delays bei 3 Video Traces und 1 TCP Flow mit und ohne Overprovisioning



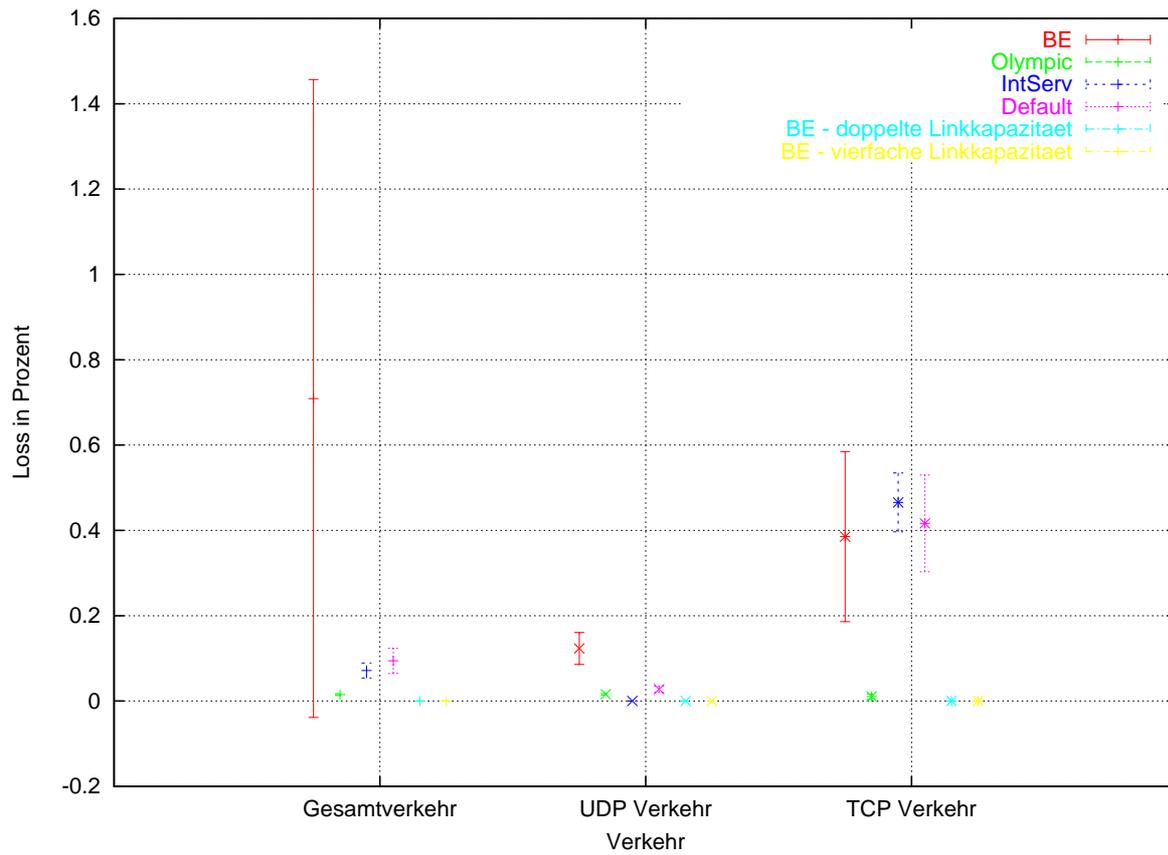
**Abbildung 10.39:** Overprovisioning: Vergleich des Jitters bei 10 CBR und 7 TCP Flows mit und ohne Overprovisioning



**Abbildung 10.40:** Overprovisioning: Vergleich des Jitters bei 3 Video Traces und 1 TCP Flow mit und ohne Overprovisioning



**Abbildung 10.41:** Overprovisioning: Vergleich des Loss' bei 10 CBR und 7 TCP Flows mit und ohne Overprovisioning



**Abbildung 10.42:** Overprovisioning: Vergleich des Loss' bei 3 Video Traces und 1 TCP Flow mit und ohne Overprovisioning

### Simulationsergebnisse:

Die Abbildungen 10.37 - 10.42 zeigen, daß bereits eine Verdoppelung der Linkkapazität der Links des ETS alle Delay und Jitterwerte unabhängig von der Verkehrszusammensetzung für den QoS-sensitiven Verkehr unter die 100 ms für den maximalen Ende-zu-Ende Delay und unter die 50 ms Akzeptanzgrenze für den maximalen Jitter verschiebt. Lediglich bei einer CBR, TCP Verkehrszusammensetzung ist der Loss erst ab einer Vervierfachung der Kapazitäten in einem akzeptablen Bereich unter 0.1 %. Gleichzeitig verbessert sich allerdings im Gegensatz zu IntServ und den beiden DiffServ Ansätzen durch die Gleichbehandlung aller Pakete ebenfalls die Dienstgüte der TCP Übertragung. Allgemein sieht man, daß durch Overprovisioning die Dienstgüten der anderen QoS Technologien erreicht werden können. Der Aufwand bzw. der Faktor der Netzerweiterung und die Erlangung der angestrebten Dienstgüten hängen allerdings sehr stark von den Netzwerkeigenschaften ab.

### **10.2.6 QoS Fallstudie**

Diese Fallstudie faßt die Ergebnisse der zuvor am ETS durchgeführten Simulationen nochmals zusammen. Hierbei wird für Delay und Jitter das Ergebnis jeweils im Verhältnis zu den von der ITU angegebenen Grenzen für diese QoS Metriken angegeben (vergleiche Kapitel 3). Loss wird hingegen als absoluter Wert angegeben. Hierbei bedeutet “>“, daß der dem Rechenzeichen nachfolgende Wert im Rahmen aller durchgeführten Simulationsläufe mindestens einmal überschritten wurde; “<“ hingegen bedeutet, daß der dem Rechenzeichen nachfolgende Wert niemals überschritten wurde.

BE			
Sensitive Parameteränderung	Delay	Jitter	Loss
CBR, TCP			
Anzahl CBR Flows 1 Anzahl TCP Flows steigt	> 100 ms	> 50 ms	<7 %
Anzahl CBR Flows 10 Anzahl TCP Flows steigt	> 100 ms	> 50 ms	<10 %
Video, TCP			
Anzahl Video Traces 1 Anzahl TCP Flows steigt	> 100 ms	> 50 ms	<10 %
Anzahl Video Traces 10 Anzahl TCP Flows steigt	> 100 ms	> 50 ms	<57 %
CBR, Video, TCP			
Anzahl CBR Flows 1 Anzahl Video Traces 1 Anzahl TCP Flows steigt	> 100 ms	> 50 ms	<5 % (CBR) / <12 % (UDP)
Anzahl CBR Flows 5 Anzahl Video Traces 5 Anzahl TCP Flows steigt	> 100 ms	> 50 ms	<41 % (CBR) / <47 % (UDP)
IntServ			
Sensitive Parameter Variation	Delay	Jitter	Loss
CBR, TCP			
Anzahl CBR Flows 1 Anzahl TCP Flows steigt	< 100 ms	> 50 ms	0 %
Anzahl CBR Flows 10 Anzahl TCP Flows steigt	< 100 ms	> 50 ms	0 %
Video, TCP			
Anzahl Video Traces 1 Anzahl TCP Flows steigt	< 100 ms	> 50 ms	0 %
Anzahl Video Traces 10 Anzahl TCP Flows steigt	< 100 ms	> 50 ms	0 %
CBR, Video, TCP			
Anzahl CBR Flows 1 Anzahl Video Traces 1 Anzahl TCP Flows steigt	< 100 ms	> 50 ms	0 % (CBR) / 0 % (UDP)
Anzahl CBR Flows 5 Anzahl Video Traces 5 Anzahl TCP Flows steigt	< 100 ms	> 50 ms	0 % (CBR) / 0 % (UDP)

Tabelle 10.13: ETS Fallstudie für BE und IntServ

DiffServ (Olympic)			
Sensitive Parameter Variation	Delay	Jitter	Loss
CBR, TCP			
Anzahl CBR Flows 1 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 %
Anzahl CBR Flows 10 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 %
Video, TCP			
Anzahl Video Traces 1 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 %
Anzahl Video Traces 10 Anzahl TCP Flows steigt	≤ 100 ms	> 50 ms	<50 %
CBR, Video, TCP			
Anzahl CBR Flows 1 Anzahl Video Traces 1 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 % (CBR) / 0 % (UDP)
Anzahl CBR Flows 5 Anzahl Video Traces 5 Anzahl TCP Flows steigt	≤ 100 ms	> 50 ms	<67 % (CBR) / <26 % (UDP)
DiffServ (Default)			
Sensitive Parameter Variation	Delay	Jitter	Loss
CBR, TCP			
Anzahl CBR Flows 1 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 %
Anzahl CBR Flows 10 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 %
Video, TCP			
Anzahl Video Traces 1 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 %
Anzahl Video Traces 10 Anzahl TCP Flows steigt	> 100 ms	> 50 ms	<74 %
CBR, Video, TCP			
Anzahl CBR Flows 1 Anzahl Video Traces 1 Anzahl TCP Flows steigt	< 100 ms	< 50 ms	0 % (CBR) / 0 % (UDP)
Anzahl CBR Flows 5 Anzahl Video Traces 5 Anzahl TCP Flows steigt	≤ 100 ms	> 50 ms	<67 % (CBR) / <60 % (UDP)

Tabelle 10.14: ETS Fallstudie für DiffServ (Olympic und Default)

### 10.2.7 Validierung

Die Validierung des Simulators sowie einzelner Simulationsergebnisse aus dem Parameterraum des ETS wurde im Rahmen des LetsQoS Projektes in Darmstadt durchgeführt. Hierbei wurde das Testbett entsprechend den Simulatorvorgaben mit Hilfe des Szenariogenerators konfiguriert und mit exakt demselben Verkehr gespeist. Die Validierung wurde auf Grund zeitlicher Verzögerungen in Darmstadt beim Testbettaufbau und der QoS Technologieimplementierung nur punktuell im Rahmen der Overprovisioninguntersuchungen (beinhalten Untersuchungen von BE, IntServ und DiffServ (Olympic, Default)) der TU Darmstadt durchgeführt und nicht alle ETS Untersuchungen im Testbett nachgebildet. Jedoch kann durch diese Untersuchungen davon ausgegangen werden, daß der Simulator für den ETS Simulationsraum und auch die anderen innerhalb dieser Arbeit zu untersuchenden Extrapolationsszenarien exakt arbeitet (siehe [SCHMITT ET AL. 04-1]).

### 10.2.8 Bewertung des simulativen Extrapolationsverfahrens

Eine Extrapolation auf die zu untersuchenden Zielszenariofälle ausgehend vom ETS ist möglich. Dies wurde durch die Sensitivitätsanalyse für alle Zielszenarien, bis auf die mit unterschiedlichen Topologien, demonstriert. Die Extrapolation auf topologievarierte Zielszenarien ist durch Implementation der abweichenden Topologien in den Simulator kein Problem. Es existieren allerdings Beschränkungen für den Umfang (Knotenanzahl, Linkanzahl), die Linkkapazitäten bzw. Verkehrsvolumen der Zielszenarien, die durch die Speicherkapazität des Simulationsrechners und die Dauer der Generierung der Simulationsergebnisse beschränkt sind. Die Exaktheit der Ergebnisse und somit der Implementation der QoS Technologien und die Realisierung der Verkehrsgenerierung unter Verwendung des Szenariogenerators wurde, wie zuvor beschrieben, durch punktuelle Validierung im Testbett belegt. Der Einsatz dieses Extrapolationsverfahrens ist für einen Netzbetreiber zwar eine relativ aufwendige Sache, doch liefert die Simulation nach erfolgter Kalibrierung und Validierung für alle Extrapolationsfälle aussagekräftige Ergebnisse.

## 10.3 Best/Worst Case Analyse

Durch die Best/Worst Case Analyse werden wiederum die 7 relevanten Extrapolationsfälle ausgehend vom Ausgangsszenario (ETS) untersucht. Bei der Best/Worst Case Analyse werden die, in den Simulationen des ETS gemessenen und bei unterschiedlichen Auslastungen auftretenden, mittleren und maximalen Ende-zu-Ende Delaywerte für die drei unterschiedlichen Verkehrszusammensetzungen für alle QoS Technologien (ausgenommen IntServ) mit den berechneten Best und Worst Case Ende-zu-Ende Delaywerten verglichen, die auf den Formeln von Kapitel 9 basieren. Auf diese Weise werden wie beim simulativen Extrapolationsverfahren alle 7 Extrapolationsfälle untersucht. Zusätzlich werden zur Überprüfung des Worst Case Ende-zu-Ende Delays Simulationen mit sehr hoher mittlerer (annähernd 100 %) Auslastung auf dem Bottleneck Link des ETS für den ersten Verkehrsmix aus CBR und TCP Verkehr durchgeführt und mit den berechneten Worst Case Werten verglichen. Ein entsprechender Vergleich wird für den berechneten Worst Case Jitter durchgeführt. Daraufhin folgt eine Bewertung der Best/Worst Case Berechnungen. Da bei IntServ der maximale Ende-zu-Ende Delay vorgegeben wird, wird IntServ bei diesem Extrapolationsverfahren nicht berücksichtigt (siehe Kapitel 9).

### 10.3.1 Best Effort

Für das ETS ergeben sich basierend auf den Formeln 9.1 und 9.2 folgende Formeln für den maximalen und minimalen Ende-zu-Ende Delay und den maximalen Jitter.

$$\begin{aligned}
D_{max} &= \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{(Q_{max,i} - 1) * MTU_i}{\gamma_i} + \frac{MTU_i}{\gamma_i} + \frac{M}{\gamma_i} \right) \\
&= \left( 0.005 \text{ s} + \frac{49 * MTU_0}{20 \text{ Mbit/s}} + \frac{MTU_0}{20 \text{ Mbit/s}} + \frac{M}{20 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{49 * MTU_1}{5 \text{ Mbit/s}} + \frac{MTU_1}{5 \text{ Mbit/s}} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{49 * MTU_2}{5 \text{ Mbit/s}} + \frac{MTU_2}{5 \text{ Mbit/s}} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{49 * MTU_3}{5 \text{ Mbit/s}} + \frac{MTU_3}{5 \text{ Mbit/s}} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{49 * MTU_4}{20 \text{ Mbit/s}} + \frac{MTU_4}{20 \text{ Mbit/s}} + \frac{M}{20 \text{ Mbit/s}} \right)
\end{aligned} \tag{10.1}$$

$$\begin{aligned}
D_{min} &= \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{M}{\gamma_i} \right) \\
&= \left( 0.005 \text{ s} + \frac{M}{20 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{M}{20 \text{ Mbit/s}} \right)
\end{aligned} \tag{10.2}$$

Auf Grund des betrachteten Zeitintervalls bei der Jitterberechnung entsteht keine Minderung der maximalen Wartezeiten in den Warteschlangen, da die Warteschlangen innerhalb des 3 Sekunden Intervalls vollständig geleert oder gefüllt werden können. Die benötigte maximale Entleerungsdauer der kompletten Warteschlange am leistungsschwächsten Link der Kapazität 5 Mbit/s beträgt 0.12 Sekunden. Dies unterschreitet die gewählte Intervalldauer deutlich. Somit ergibt sich folgende Formel für den maximalen Jitter:

$$\begin{aligned}
J_{max} &= \sum_{i=0}^{n-1} \left( \frac{(Q_{max,i} - 1) * MTU_i}{\gamma_i} + \frac{MTU_i}{\gamma_i} + \frac{\Delta M_i}{\gamma_i} \right) \\
&= \left( \frac{49 * MTU_0}{20 \text{ Mbit/s}} + \frac{MTU_0}{20 \text{ Mbit/s}} + \frac{\Delta M_0}{20 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{49 * MTU_1}{5 \text{ Mbit/s}} + \frac{MTU_1}{5 \text{ Mbit/s}} + \frac{\Delta M_1}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{49 * MTU_2}{5 \text{ Mbit/s}} + \frac{MTU_2}{5 \text{ Mbit/s}} + \frac{\Delta M_2}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{49 * MTU_3}{5 \text{ Mbit/s}} + \frac{MTU_3}{5 \text{ Mbit/s}} + \frac{\Delta M_3}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{49 * MTU_4}{20 \text{ Mbit/s}} + \frac{MTU_4}{20 \text{ Mbit/s}} + \frac{\Delta M_4}{20 \text{ Mbit/s}} \right)
\end{aligned} \tag{10.3}$$

Abgeleitet von den unterschiedlichen Verkehrsarten ergeben sich folgende Variablenwerte (siehe Tabelle 10.15), die zu entsprechenden Ergebnissen führen (siehe Tabellen 10.16, 10.17, 10.18, 10.19 und 10.20, 10.21).

Variablen	CBR, TCP Verkehrsmix	Video, TCP Verkehrsmix	CBR, Video und TCP Verkehrsmix
$MTU_0$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_1$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_2$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_3$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_4$	1500 Bytes	1500 Bytes	1500 Bytes
$M$	72 Bytes	782 Bytes	782 Bytes
$\Delta M_0$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_1$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_2$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_3$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_4$	0 Bytes	1344 Bytes	1344 Bytes

**Tabelle 10.15:** Verkehrsspezifische Variablenwerte

Für CBR mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.15), folgende Ergebnisse (siehe Tabellen 10.16, 10.17).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0732	CBR
maximaler Delay bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.1404	CBR
mittlerer Delay bei 1 CBR, 1 TCP Flow	0.0302	CBR
maximaler Delay bei 1 CBR, 1 TCP Flow	0.0862	CBR
mittlerer Delay bei 1 CBR, 7 TCP Flows	0.0768	CBR
maximaler Delay bei 1 CBR, 7 TCP Flows	0.1478	CBR
mittlerer Delay bei 10 CBR, 1 TCP Flow	0.0280	CBR
maximaler Delay bei 10 CBR, 1 TCP Flow	0.0805	CBR
mittlerer Delay bei 10 CBR, 7 TCP Flows	0.0545	CBR
maximaler Delay bei 10 CBR, 7 TCP Flows	0.1281	CBR
berechneter Best Case Delaywert	0.0254	CBR
berechneter Worst Case Delaywert	0.4454	CBR

**Tabelle 10.16:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0546	CBR
maximaler Jitter bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0997	CBR
mittlerer Jitter bei 1 CBR, 1 TCP Flow	0.0073	CBR
maximaler Jitter bei 1 CBR, 1 TCP Flow	0.0586	CBR
mittlerer Jitter bei 1 CBR, 7 TCP Flows	0.0584	CBR
maximaler Jitter bei 1 CBR, 7 TCP Flows	0.1218	CBR
mittlerer Jitter bei 10 CBR, 1 TCP Flow	0.0108	CBR
maximaler Jitter bei 10 CBR, 1 TCP Flow	0.0549	CBR
mittlerer Jitter bei 10 CBR, 7 TCP Flows	0.0613	CBR
maximaler Jitter bei 10 CBR, 7 TCP Flows	0.0988	CBR
berechneter Worst Case Jitterwert	0.42	CBR

**Tabelle 10.17:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR und TCP Verkehr

Für Video Traces mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.15), folgende Ergebnisse (siehe Tabellen 10.18, 10.19).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 1 Video Trace, 1 TCP Flow	0.0404	Video Trace
maximaler Delay bei 1 Video Trace, 1 TCP Flow	0.1103	Video Trace
mittlerer Delay bei 1 Video Trace, 7 TCP Flows	0.0845	Video Trace
maximaler Delay bei 1 Video Trace, 7 TCP Flows	0.1507	Video Trace
mittlerer Delay bei 10 Video Traces, 1 TCP Flow	0.0559	Video Trace
maximaler Delay bei 10 Video Traces, 1 TCP Flow	0.1170	Video Trace
mittlerer Delay bei 10 Video Traces, 7 TCP Flows	0.0706	Video Trace
maximaler Delay bei 10 Video Traces, 7 TCP Flows	0.1513	Video Trace
berechneter Best Case Delaywert	0.0294	Video Trace
berechneter Worst Case Delaywert	0.4494	Video Trace

**Tabelle 10.18:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus Video und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 1 Video Trace, 1 TCP Flow	0.0265	Video Trace
maximaler Jitter bei 1 Video Trace, 1 TCP Flow	0.0811	Video Trace
mittlerer Jitter bei 1 Video Trace, 7 TCP Flows	0.0747	Video Trace
maximaler Jitter bei 1 Video Trace, 7 TCP Flows	0.1216	Video Trace
mittlerer Jitter bei 10 Video Traces, 1 TCP Flow	0.0769	Video Trace
maximaler Jitter bei 10 Video Traces, 1 TCP Flow	0.0915	Video Trace
mittlerer Jitter bei 10 Video Traces, 7 TCP Flows	0.0883	Video Trace
maximaler Jitter bei 10 Video Traces, 7 TCP Flows	0.1206	Video Trace
berechneter Worst Case Jitterwert	0.4275	Video Trace

**Tabelle 10.19:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus Video und TCP Verkehr

Für CBR und Video Traces mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.15), folgende Ergebnisse (siehe Tabellen 10.20, 10.21).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0365	Video Trace
maximaler Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0998	Video Trace
mittlerer Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0821	Video Trace
maximaler Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.1504	Video Trace
mittlerer Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0557	Video Trace
maximaler Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.1076	Video Trace
mittlerer Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0741	Video Trace
maximaler Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.1418	Video Trace
berechneter Best Case Delaywert	0.0294	Video Trace
berechneter Worst Case Delaywert	0.4494	Video Trace

mittlerer Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0308	CBR
maximaler Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0954	CBR
mittlerer Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0760	CBR
maximaler Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.1447	CBR
mittlerer Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0420	CBR
maximaler Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.1049	CBR
mittlerer Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0685	CBR
maximaler Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.1341	CBR
berechneter Best Case Delaywert	0.0254	CBR
berechneter Worst Case Delaywert	0.4454	CBR

**Tabelle 10.20:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR, Video und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0236	Video Trace
maximaler Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0696	Video Trace
mittlerer Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0776	Video Trace
maximaler Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.1216	Video Trace
mittlerer Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0708	Video Trace
maximaler Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0823	Video Trace
mittlerer Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0768	Video Trace
maximaler Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.1108	Video Trace
berechneter Worst Case Jitterwert	0.4275	Video Trace

mittlerer Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0177	CBR
maximaler Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0681	CBR
mittlerer Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0557	CBR
maximaler Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.1191	CBR
mittlerer Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0680	CBR
maximaler Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0795	CBR
mittlerer Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0662	CBR
maximaler Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.1075	CBR
berechneter Worst Case Jitterwert	0.42	CBR

**Tabelle 10.21:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR, Video und TCP Verkehr

### 10.3.2 Olympic Ansatz

Für das ETS ergeben sich basierend auf den Formeln 9.9 und 9.10 folgende Formeln für den maximalen und minimalen Ende-zu-Ende Delay und den maximalen Jitter.

$$\begin{aligned}
 D_{max} &= \left( D_{prop,0} + \frac{(Q_{max,0} - 1) * MTU_0}{\gamma_0} + \frac{MTU_0}{\gamma_0} + \frac{M}{\gamma_0} \right) + \\
 &\left( D_{prop,1} + \frac{(Q_{max,1} - 1) * MTU_{EF,1}}{\gamma_1} + \frac{MTU_1}{\gamma_1} + \frac{M}{\gamma_1} \right) + \\
 &\left( D_{prop,2} + \frac{(Q_{max,2} - 1) * MTU_{EF,2}}{\gamma_2} + \frac{MTU_2}{\gamma_2} + \frac{M}{\gamma_2} \right) + \\
 &\left( D_{prop,3} + \frac{(Q_{max,3} - 1) * MTU_{EF,3}}{\gamma_3} + \frac{MTU_3}{\gamma_3} + \frac{M}{\gamma_3} \right) + \\
 &\left( D_{prop,4} + \frac{(Q_{max,4} - 1) * MTU_4}{\gamma_4} + \frac{MTU_4}{\gamma_4} + \frac{M}{\gamma_4} \right) \\
 &= \left( 0.005 \text{ s} + \frac{49 * MTU_0}{20 \text{ Mbit/s}} + \frac{MTU_0}{20 \text{ Mbit/s}} + \frac{M}{20 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{9 * MTU_{EF,1}}{5 \text{ Mbit/s}} + \frac{MTU_1}{5 \text{ Mbit/s}} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{9 * MTU_{EF,2}}{5 \text{ Mbit/s}} + \frac{MTU_2}{5 \text{ Mbit/s}} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{9 * MTU_{EF,3}}{5 \text{ Mbit/s}} + \frac{MTU_3}{5 \text{ Mbit/s}} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{49 * MTU_4}{20 \text{ Mbit/s}} + \frac{MTU_4}{20 \text{ Mbit/s}} + \frac{M}{20 \text{ Mbit/s}} \right)
 \end{aligned} \tag{10.4}$$

$$\begin{aligned}
 D_{min} &= \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{M}{\gamma_i} \right) \\
 &= \left( 0.005 \text{ s} + \frac{M}{20 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\left( 0.005 \text{ s} + \frac{M}{20 \text{ Mbit/s}} \right)
 \end{aligned} \tag{10.5}$$

$$\begin{aligned}
J_{max} &= \left( \frac{(Q_{max,0} - 1) * MTU_0}{\gamma_0} + \frac{MTU_0}{\gamma_0} + \frac{\Delta M_0}{\gamma_0} \right) + \\
&\left( \frac{(Q_{max,1} - 1) * MTU_{EF,1}}{\gamma_1} + \frac{MTU_1}{\gamma_1} + \frac{\Delta M_1}{\gamma_1} \right) + \\
&\left( \frac{(Q_{max,2} - 1) * MTU_{EF,2}}{\gamma_2} + \frac{MTU_2}{\gamma_2} + \frac{\Delta M_2}{\gamma_2} \right) + \\
&\left( \frac{(Q_{max,3} - 1) * MTU_{EF,3}}{\gamma_3} + \frac{MTU_3}{\gamma_3} + \frac{\Delta M_3}{\gamma_3} \right) + \\
&\left( \frac{(Q_{max,4} - 1) * MTU_4}{\gamma_4} + \frac{MTU_4}{\gamma_4} + \frac{\Delta M_4}{\gamma_4} \right) \\
&= \left( \frac{49 * MTU_0}{20 \text{ Mbit/s}} + \frac{MTU_0}{20 \text{ Mbit/s}} + \frac{\Delta M_0}{20 \text{ Mbit/s}} \right) + \\
&\left( \frac{9 * MTU_{EF,1}}{5 \text{ Mbit/s}} + \frac{MTU_1}{5 \text{ Mbit/s}} + \frac{\Delta M_1}{5 \text{ Mbit/s}} \right) + \\
&\left( \frac{9 * MTU_{EF,2}}{5 \text{ Mbit/s}} + \frac{MTU_2}{5 \text{ Mbit/s}} + \frac{\Delta M_2}{5 \text{ Mbit/s}} \right) + \\
&\left( \frac{9 * MTU_{EF,3}}{5 \text{ Mbit/s}} + \frac{MTU_3}{5 \text{ Mbit/s}} + \frac{\Delta M_3}{5 \text{ Mbit/s}} \right) + \\
&\left( \frac{49 * MTU_4}{20 \text{ Mbit/s}} + \frac{MTU_4}{20 \text{ Mbit/s}} + \frac{\Delta M_4}{20 \text{ Mbit/s}} \right)
\end{aligned} \tag{10.6}$$

Abgeleitet von den unterschiedlichen Verkehrsarten ergeben sich folgende Variablenwerte (siehe Tabelle 10.22), die zu entsprechenden Ergebnissen führen (siehe Tabellen 10.23, 10.24, 10.25, 10.26 und 10.27, 10.28).

Variablen	CBR, TCP Verkehrsmix	Video, TCP Verkehrsmix	CBR, Video und TCP Verkehrsmix
$MTU_0$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_1$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_2$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_3$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_4$	1500 Bytes	1500 Bytes	1500 Bytes
$M$	72 Bytes	782 Bytes	782 Bytes
$\Delta M_0$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_1$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_2$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_3$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_4$	0 Bytes	1344 Bytes	1344 Bytes
$MTU_{EF,1}$	72 Bytes	1375 Bytes	1375 Bytes
$MTU_{EF,2}$	72 Bytes	1375 Bytes	1375 Bytes
$MTU_{EF,3}$	72 Bytes	1375 Bytes	1375 Bytes

**Tabelle 10.22:** Verkehrsspezifische Variablenwerte

Für CBR Verkehr mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.22), folgende Ergebnisse (siehe Tabellen 10.23, 10.24).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 99.7 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0317	CBR
maximaler Delay bei 99.7 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0528	CBR
mittlerer Delay bei 1 CBR, 1 TCP Flow	0.0269	CBR
maximaler Delay bei 1 CBR, 1 TCP Flow	0.0336	CBR
mittlerer Delay bei 1 CBR, 7 TCP Flows	0.0312	CBR
maximaler Delay bei 1 CBR, 7 TCP Flows	0.0387	CBR
mittlerer Delay bei 10 CBR, 1 TCP Flow	0.0269	CBR
maximaler Delay bei 10 CBR, 1 TCP Flow	0.0341	CBR
mittlerer Delay bei 10 CBR, 7 TCP Flows	0.0312	CBR
maximaler Delay bei 10 CBR, 7 TCP Flows	0.0395	CBR
berechneter Best Case Delaywert	0.0254	CBR
berechneter Worst Case Delaywert	0.0957	CBR

**Tabelle 10.23:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 99.7 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0046	CBR
maximaler Jitter bei 99.7 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0274	CBR
mittlerer Jitter bei 1 CBR, 1 TCP Flow	0.0024	CBR
maximaler Jitter bei 1 CBR, 1 TCP Flow	0.0081	CBR
mittlerer Jitter bei 1 CBR, 7 TCP Flows	0.0035	CBR
maximaler Jitter bei 1 CBR, 7 TCP Flows	0.0124	CBR
mittlerer Jitter bei 10 CBR, 1 TCP Flow	0.0029	CBR
maximaler Jitter bei 10 CBR, 1 TCP Flow	0.0085	CBR
mittlerer Jitter bei 10 CBR, 7 TCP Flows	0.0049	CBR
maximaler Jitter bei 10 CBR, 7 TCP Flows	0.0137	CBR
berechneter Worst Case Jitterwert	0.0703	CBR

**Tabelle 10.24:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR und TCP Verkehr

Für Video Traces mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.22), folgende Ergebnisse (siehe Tabellen 10.25, 10.26).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 1 Video Trace, 1 TCP Flow	0.0308	Video Trace
maximaler Delay bei 1 Video Trace, 1 TCP Flow	0.0362	Video Trace
mittlerer Delay bei 1 Video Trace, 7 TCP Flows	0.0362	Video Trace
maximaler Delay bei 1 Video Trace, 7 TCP Flows	0.0494	Video Trace
mittlerer Delay bei 10 Video Traces, 1 TCP Flow	0.0499	Video Trace
maximaler Delay bei 10 Video Traces, 1 TCP Flow	0.1030	Video Trace
mittlerer Delay bei 10 Video Traces, 7 TCP Flows	0.0516	Video Trace
maximaler Delay bei 10 Video Traces, 7 TCP Flows	0.1045	Video Trace
berechneter Best Case Delaywert	0.0294	Video Trace
berechneter Worst Case Delaywert	0.1559	Video Trace

**Tabelle 10.25:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus Video und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 1 Video Trace, 1 TCP Flow	0.0058	Video Trace
maximaler Jitter bei 1 Video Trace, 1 TCP Flow	0.0098	Video Trace
mittlerer Jitter bei 1 Video Trace, 7 TCP Flows	0.0172	Video Trace
maximaler Jitter bei 1 Video Trace, 7 TCP Flows	0.0231	Video Trace
mittlerer Jitter bei 10 Video Traces, 1 TCP Flow	0.0679	Video Trace
maximaler Jitter bei 10 Video Traces, 1 TCP Flow	0.0777	Video Trace
mittlerer Jitter bei 10 Video Traces, 7 TCP Flows	0.0645	Video Trace
maximaler Jitter bei 10 Video Traces, 7 TCP Flows	0.0754	Video Trace
berechneter Worst Case Jitterwert	0.1341	Video Trace

**Tabelle 10.26:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus Video und TCP Verkehr

Für CBR und Video Traces mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.22), folgende Ergebnisse (siehe Tabellen 10.27, 10.28).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0339	Video Trace
maximaler Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0490	Video Trace
mittlerer Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0364	Video Trace
maximaler Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0497	Video Trace
mittlerer Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0527	Video Trace
maximaler Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0993	Video Trace
mittlerer Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0537	Video Trace
maximaler Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.1002	Video Trace
berechneter Best Case Delaywert	0.0294	Video Trace
berechneter Worst Case Delaywert	0.1560	Video Trace

mittlerer Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0275	CBR
maximaler Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0470	CBR
mittlerer Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0316	CBR
maximaler Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0489	CBR
mittlerer Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0313	CBR
maximaler Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0982	CBR
mittlerer Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0336	CBR
maximaler Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0983	CBR
berechneter Best Case Delaywert	0.0254	CBR
berechneter Worst Case Delaywert	0.1040	CBR

**Tabelle 10.27:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR, Video und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0188	Video Trace
maximaler Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0237	Video Trace
mittlerer Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0172	Video Trace
maximaler Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0232	Video Trace
mittlerer Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0657	Video Trace
maximaler Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0741	Video Trace
mittlerer Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0620	Video Trace
maximaler Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0721	Video Trace
berechneter Worst Case Jitterwert	0.1341	Video Trace

mittlerer Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0133	CBR
maximaler Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0215	CBR
mittlerer Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0120	CBR
maximaler Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0219	CBR
mittlerer Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0631	CBR
maximaler Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0728	CBR
mittlerer Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0585	CBR
maximaler Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0706	CBR
berechneter Worst Case Jitterwert	0.0786	CBR

**Tabelle 10.28:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR, Video und TCP Verkehr

### 10.3.3 Default Ansatz

Für das ETS ergeben sich basierend auf den Formeln 9.9 und 9.10 folgende Formeln für den maximalen und minimalen Ende-zu-Ende Delay und den maximalen Jitter.

$$\begin{aligned}
 D_{min} &= \sum_{i=0}^{n-1} \left( D_{prop,i} + \frac{M}{\gamma_i} \right) \\
 &= \left( 0.005 \text{ s} + \frac{M}{20 \text{ Mbit/s}} \right) + \\
 &\quad \left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\quad \left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\quad \left( 0.005 \text{ s} + \frac{M}{5 \text{ Mbit/s}} \right) + \\
 &\quad \left( 0.005 \text{ s} + \frac{M}{20 \text{ Mbit/s}} \right)
 \end{aligned} \tag{10.7}$$

$$\begin{aligned}
 D_{max} &= \left( D_{prop,0} + \frac{(Q_{max,0} - 1) * MTU_0}{\gamma_0} + \frac{MTU_0}{\gamma_0} + \frac{M}{\gamma_0} \right) + \\
 &\quad \left( D_{prop,1} + \frac{(Q_{max,1} - 1) * MTU_{EF,1}}{\gamma_1} \right) + \\
 &\quad \left( \frac{(MTU_{AF1,1} * AF1_1) + (MTU_{AF2,1} * AF2_1) + (MTU_{AF3,1} * AF3_1)}{\gamma_1} \right) + \\
 &\quad \left( \frac{M}{\gamma_1} \right) + \\
 &\quad \left( D_{prop,2} + \frac{(Q_{max,2} - 1) * MTU_{EF,2}}{\gamma_2} \right) + \\
 &\quad \left( \frac{(MTU_{AF1,2} * AF1_2) + (MTU_{AF2,2} * AF2_2) + (MTU_{AF3,2} * AF3_2)}{\gamma_2} \right) + \\
 &\quad \left( \frac{M}{\gamma_2} \right) + \\
 &\quad \left( D_{prop,3} + \frac{(Q_{max,3} - 1) * MTU_{EF,3}}{\gamma_3} \right) + \\
 &\quad \left( \frac{(MTU_{AF1,3} * AF1_3) + (MTU_{AF2,3} * AF2_3) + (MTU_{AF3,3} * AF3_3)}{\gamma_3} \right) + \\
 &\quad \left( \frac{M}{\gamma_3} \right) + \\
 &\quad \left( D_{prop,4} + \frac{(Q_{max,4} - 1) * MTU_4}{\gamma_4} + \frac{MTU_4}{\gamma_4} + \frac{M}{\gamma_4} \right)
 \end{aligned} \tag{10.8}$$

$$\begin{aligned}
&= \left( 0.005 \text{ s} + \frac{49 * MTU_0}{20 \text{ Mbit/s}} + \frac{MTU_0}{20 \text{ Mbit/s}} + \frac{M}{20 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{9 * MTU_{EF,1}}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{(MTU_{AF1,i} * 3) + (MTU_{AF2,i} * 3) + (MTU_{AF3,i} * 4)}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{9 * MTU_{EF,2}}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{(MTU_{AF1,i} * 3) + (MTU_{AF2,i} * 3) + (MTU_{AF3,i} * 4)}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{9 * MTU_{EF,3}}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{(MTU_{AF1,i} * 3) + (MTU_{AF2,i} * 3) + (MTU_{AF3,i} * 4)}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{M}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( 0.005 \text{ s} + \frac{49 * MTU_0}{20 \text{ Mbit/s}} + \frac{MTU_0}{20 \text{ Mbit/s}} + \frac{M}{20 \text{ Mbit/s}} \right) \\
\\
J_{max} &= \left( \frac{(Q_{max,0} - 1) * MTU_0}{\gamma_0} + \frac{MTU_0}{\gamma_0} + \frac{\Delta M_0}{\gamma_0} \right) + \\
&\quad \left( \frac{(Q_{max,1} - 1) * MTU_{EF,1}}{\gamma_1} \right) + \\
&\quad \left( \frac{(MTU_{AF1,1} * AF1_1) + (MTU_{AF2,1} * AF2_1) + (MTU_{AF3,1} * AF3_1)}{\gamma_1} \right) + \\
&\quad \left( \frac{\Delta M_1}{\gamma_1} \right) + \\
&\quad \left( \frac{(Q_{max,2} - 1) * MTU_{EF,2}}{\gamma_2} \right) + \\
&\quad \left( \frac{(MTU_{AF1,2} * AF1_2) + (MTU_{AF2,2} * AF2_2) + (MTU_{AF3,2} * AF3_2)}{\gamma_2} \right) + \\
&\quad \left( \frac{\Delta M_2}{\gamma_2} \right) + \\
&\quad \left( \frac{(Q_{max,3} - 1) * MTU_{EF,3}}{\gamma_3} \right) + \\
&\quad \left( \frac{(MTU_{AF1,3} * AF1_3) + (MTU_{AF2,3} * AF2_3) + (MTU_{AF3,3} * AF3_3)}{\gamma_3} \right) + \\
&\quad \left( \frac{\Delta M_3}{\gamma_3} \right) + \\
&\quad \left( \frac{(Q_{max,4} - 1) * MTU_4}{\gamma_4} + \frac{MTU_4}{\gamma_4} + \frac{\Delta M_4}{\gamma_4} \right)
\end{aligned}$$

$$\begin{aligned}
&= \left( \frac{49 * MTU_0}{20 \text{ Mbit/s}} + \frac{MTU_0}{20 \text{ Mbit/s} \cdot \text{Mbit/s}} + \frac{\Delta M_0}{20 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{9 * MTU_{EF,1}}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{(MTU_{AF1,1} * 3) + (MTU_{AF2,1} * 3) + (MTU_{AF3,1} * 4)}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{\Delta M_1}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{9 * MTU_{EF,2}}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{(MTU_{AF1,2} * 3) + (MTU_{AF2,2} * 3) + (MTU_{AF3,2} * 4)}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{\Delta M_2}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{9 * MTU_{EF,3}}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{(MTU_{AF1,3} * 3) + (MTU_{AF2,3} * 3) + (MTU_{AF3,3} * 4)}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{\Delta M_3}{5 \text{ Mbit/s}} \right) + \\
&\quad \left( \frac{49 * MTU_4}{20 \text{ Mbit/s}} + \frac{MTU_4}{20 \text{ Mbit/s}} + \frac{\Delta M_4}{20 \text{ Mbit/s}} \right)
\end{aligned}$$

Abgeleitet von den unterschiedlichen Verkehrsarten ergeben sich folgende Variablenwerte (siehe Tabelle 10.29), die zu entsprechenden Ergebnissen führen (siehe Tabellen 10.30, 10.31, 10.32, 10.33 und 10.34, 10.35).

Variablen	CBR, TCP Verkehrsmix	Video, TCP Verkehrsmix	CBR, Video und TCP Verkehrsmix
$MTU_0$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_1$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_2$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_3$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_4$	1500 Bytes	1500 Bytes	1500 Bytes
$M$	72 Bytes	782 Bytes	782 Bytes
$\Delta M_0$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_1$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_2$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_3$	0 Bytes	1344 Bytes	1344 Bytes
$\Delta M_4$	0 Bytes	1344 Bytes	1344 Bytes
$MTU_{EF,1}$	72 Bytes	1375 Bytes	1375 Bytes
$MTU_{EF,2}$	72 Bytes	1375 Bytes	1375 Bytes
$MTU_{EF,3}$	72 Bytes	1375 Bytes	1375 Bytes
$MTU_{AF1,1}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF1,2}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF1,3}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF2,1}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF2,2}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF2,3}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF3,1}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF3,2}$	1500 Bytes	1500 Bytes	1500 Bytes
$MTU_{AF3,3}$	1500 Bytes	1500 Bytes	1500 Bytes

**Tabelle 10.29:** Verkehrsspezifische Variablenwerte

Für CBR mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.29), folgende Ergebnisse (siehe Tabellen 10.30, 10.31).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0381	CBR
maximaler Delay bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0624	CBR
mittlerer Delay bei 1 CBR, 1 TCP Flow	0.0268	CBR
maximaler Delay bei 1 CBR, 1 TCP Flow	0.0343	CBR
mittlerer Delay bei 1 CBR, 7 TCP Flows	0.0352	CBR
maximaler Delay bei 1 CBR, 7 TCP Flows	0.0503	CBR
mittlerer Delay bei 10 CBR, 1 TCP Flow	0.0269	CBR
maximaler Delay bei 10 CBR, 1 TCP Flow	0.0358	CBR
mittlerer Delay bei 10 CBR, 7 TCP Flows	0.0351	CBR
maximaler Delay bei 10 CBR, 7 TCP Flows	0.0514	CBR
berechneter Best Case Delaywert	0.0254	CBR
berechneter Worst Case Delaywert	0.1605	CBR

**Tabelle 10.30:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0192	CBR
maximaler Jitter bei 99.6 % mittlerer Auslastung bei 10 CBR, 20 TCP Flows	0.0370	CBR
mittlerer Jitter bei 1 CBR, 1 TCP Flow	0.0026	CBR
maximaler Jitter bei 1 CBR, 1 TCP Flow	0.0088	CBR
mittlerer Jitter bei 1 CBR, 7 TCP Flows	0.0129	CBR
maximaler Jitter bei 1 CBR, 7 TCP Flows	0.0229	CBR
mittlerer Jitter bei 10 CBR, 1 TCP Flow	0.0032	CBR
maximaler Jitter bei 10 CBR, 1 TCP Flow	0.0103	CBR
mittlerer Jitter bei 10 CBR, 7 TCP Flows	0.0168	CBR
maximaler Jitter bei 10 CBR, 7 TCP Flows	0.0241	CBR
berechneter Worst Case Jitterwert	0.1351	CBR

**Tabelle 10.31:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR und TCP Verkehr

Für Video Traces mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.29), folgende Ergebnisse (siehe Tabellen 10.32, 10.33).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 1 Video Trace, 1 TCP Flow	0.0347	Video Trace
maximaler Delay bei 1 Video Trace, 1 TCP Flow	0.0518	Video Trace
mittlerer Delay bei 1 Video Trace, 7 TCP Flows	0.0422	Video Trace
maximaler Delay bei 1 Video Trace, 7 TCP Flows	0.0690	Video Trace
mittlerer Delay bei 10 Video Traces, 1 TCP Flow	0.0500	Video Trace
maximaler Delay bei 10 Video Traces, 1 TCP Flow	0.1032	Video Trace
mittlerer Delay bei 10 Video Traces, 7 TCP Flows	0.0546	Video Trace
maximaler Delay bei 10 Video Traces, 7 TCP Flows	0.1097	Video Trace
berechneter Best Case Delaywert	0.0294	Video Trace
berechneter Worst Case Delaywert	0.2208	Video Trace

**Tabelle 10.32:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus Video und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 1 Video Trace, 1 TCP Flow	0.0197	Video Trace
maximaler Jitter bei 1 Video Trace, 1 TCP Flow	0.0263	Video Trace
mittlerer Jitter bei 1 Video Trace, 7 TCP Flows	0.0294	Video Trace
maximaler Jitter bei 1 Video Trace, 7 TCP Flows	0.0403	Video Trace
mittlerer Jitter bei 10 Video Traces, 1 TCP Flow	0.0678	Video Trace
maximaler Jitter bei 10 Video Traces, 1 TCP Flow	0.0780	Video Trace
mittlerer Jitter bei 10 Video Traces, 7 TCP Flows	0.0661	Video Trace
maximaler Jitter bei 10 Video Traces, 7 TCP Flows	0.0803	Video Trace
berechneter Worst Case Jitterwert	0.1989	Video Trace

**Tabelle 10.33:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus Video und TCP Verkehr

Für CBR und Video Traces mit TCP Hintergrundverkehr ergeben sich durch Einsetzen der aus den Verkehrsarten abgeleiteten Variablenwerten (siehe Tabelle 10.29), folgende Ergebnisse (siehe Tabellen 10.34, 10.35).

Delayart	Delaywert [s]	Verkehrstyp
mittlerer Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0342	Video Trace
maximaler Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0516	Video Trace
mittlerer Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0422	Video Trace
maximaler Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0693	Video Trace
mittlerer Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0524	Video Trace
maximaler Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0992	Video Trace
mittlerer Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0535	Video Trace
maximaler Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.1000	Video Trace
berechneter Best Case Delaywert	0.0294	Video Trace
berechneter Worst Case Delaywert	0.2208	Video Trace

mittlerer Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0279	CBR
maximaler Delay bei 1 Video, 1 CBR, 1 TCP Flow	0.0494	CBR
mittlerer Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0375	CBR
maximaler Delay bei 1 Video, 1 CBR, 7 TCP Flows	0.0673	CBR
mittlerer Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0316	CBR
maximaler Delay bei 5 Video, 5 CBR, 1 TCP Flow	0.0976	CBR
mittlerer Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0388	CBR
maximaler Delay bei 5 Video, 5 CBR, 7 TCP Flows	0.0933	CBR
berechneter Best Case Delaywert	0.0254	CBR
berechneter Worst Case Delaywert	0.1688	CBR

**Tabelle 10.34:** Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR, Video und TCP Verkehr

Jitterart	Jitterwert [s]	Verkehrstyp
mittlerer Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0194	Video Trace
maximaler Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0262	Video Trace
mittlerer Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0295	Video Trace
maximaler Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0409	Video Trace
mittlerer Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0652	Video Trace
maximaler Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0739	Video Trace
mittlerer Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0584	Video Trace
maximaler Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0715	Video Trace
berechneter Worst Case Jitterwert	0.1989	Video Trace

mittlerer Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0138	CBR
maximaler Jitter bei 1 Video, 1 CBR, 1 TCP Flow	0.0239	CBR
mittlerer Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0222	CBR
maximaler Jitter bei 1 Video, 1 CBR, 7 TCP Flows	0.0393	CBR
mittlerer Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0616	CBR
maximaler Jitter bei 5 Video, 5 CBR, 1 TCP Flow	0.0722	CBR
mittlerer Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0487	CBR
maximaler Jitter bei 5 Video, 5 CBR, 7 TCP Flows	0.0671	CBR
berechneter Worst Case Jitterwert	0.1434	CBR

**Tabelle 10.35:** Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR, Video und TCP Verkehr

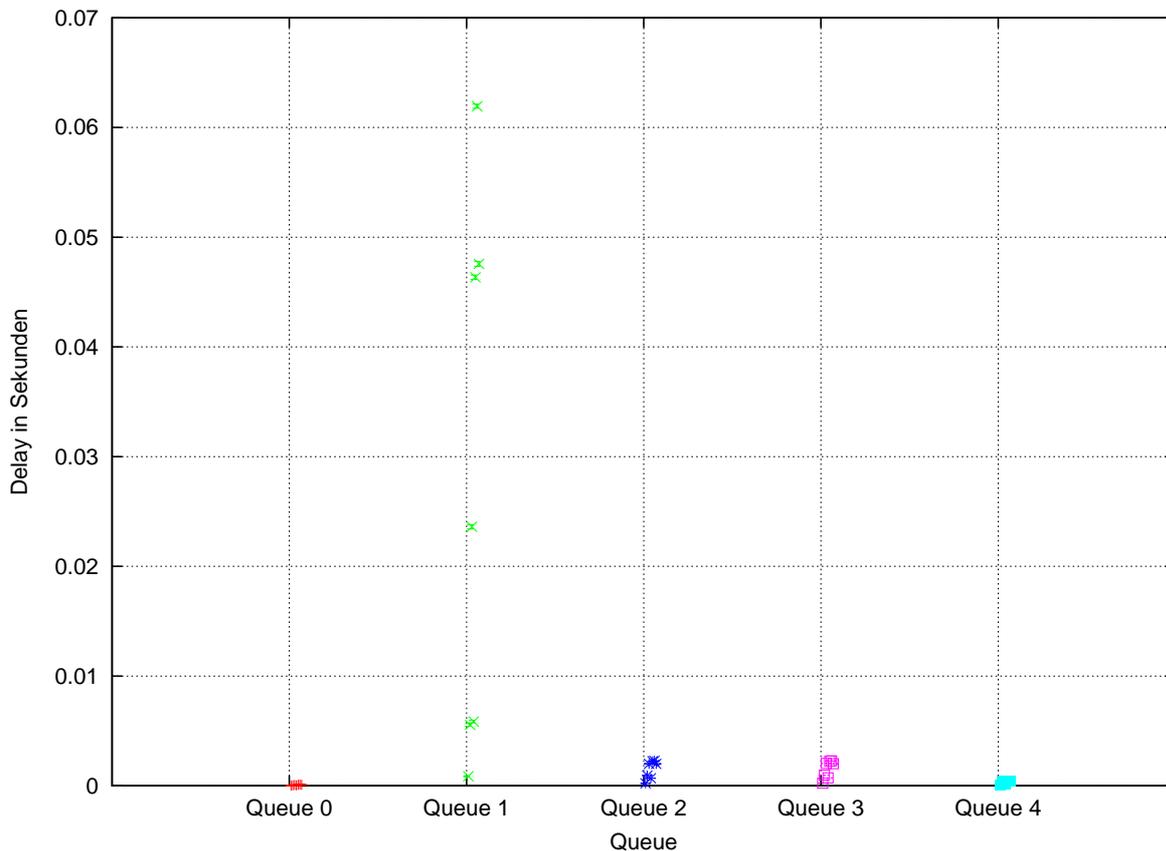
### 10.3.4 Bewertung der Best/Worst Case Extrapolation

Die Best/Worst Case Analyse ermöglicht Extrapolationen auf die sieben zu untersuchenden Zielszenarien (siehe Tabelle 1.1 und 8.1) für die Dienstgütemetriken Delay und Jitter. Die Best/Worst Case Analyse zeigt, daß eine Worst Case Berechnung für allgemeine, grobe Abschätzungen des Delays, wie auch des Jitters, geeignet sein kann. Dies gilt besonders für QoS Technologien, die kleine Warteschlangenlängen verwenden, Verkehr mit kleinen Paketgrößen und für hoch ausgelastete Netze.

Die Abweichung der Worst Case Berechnung von den Simulationsergebnissen steigt bei längeren Warteschlangen an, da die Berechnung von einer maximalen Füllung aller Warteschlangen auf einem Pfad vom Sender zum Empfänger ausgeht. Diese Situation kann aber in den meisten Netzen, wie auch im ETS oder G-WiN, nicht auftreten. Dort existiert meistens ein (im ETS so gewählt) oder mehrere Engpässe an denen die Warteschlangen maximal gefüllt sind. Die restlichen Warteschlangen sind

meist nur wenig ausgelastet bzw. gefüllt. Dies zeigen die Abbildungen 10.43 und 10.44 anhand der Wartezeiten in den einzelnen Warteschlangen des ETS am Beispiel des CBR und TCP Verkehrsmixes unter der Verwendung von BE. Diese Wirkung kann sich auch noch durch große Pakete der QoS-sensitiven Anwendung verstärken, da die Warteschlangenlänge oft nur durch die Anzahl der Pakete, aber nicht durch die Anzahl von Bits oder Bytes begrenzt ist. Außerdem sieht natürlich eine Worst Case Berechnung im Verhältnis zu Simulationsergebnissen in niedrig ausgelasteten Netzen immer schlecht aus. Daher sollte ein Vergleich oder eine Beurteilung immer im Bezug auf eine Maximalauslastung des Netzes durchgeführt werden.

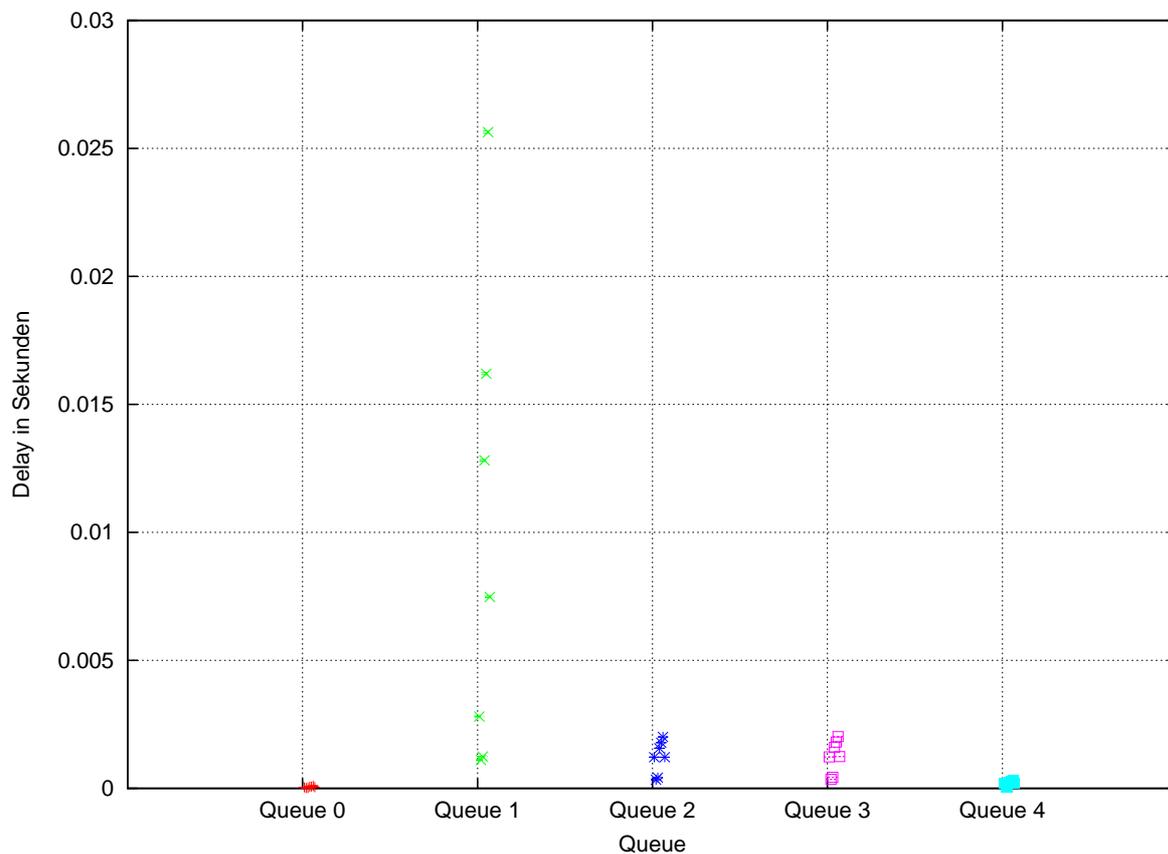
Wie die Ergebnisse zeigen, liefern die entwickelten Formeln bei den zuvor beschriebenen Voraussetzungen aussagekräftige Resultate für einen Netzbetreiber. Durch die Berechnung z.B.: des maximalen Ende-zu-Ende Delays für CBR mit TCP Hintergrundverkehr unter der Verwendung des Olympic Ansatzes (siehe Tabelle 10.23) kann belegt werden, daß ein für VoIP Verkehr kritischer Ende-zu-Ende Delay von 100 ms niemals überschritten werden kann und somit weitere Untersuchungen durch Simulationen und das stochastische Verfahren hinfällig sind. Durch die Berechnungen kann ausgelotet werden, wann, bei welcher Verkehrszusammensetzung, der Ende-zu-Ende Delay überschritten werden könnte.



**Abbildung 10.43:** ETS unter Einsatz von BE: Wartezeit in den Warteschlangen bei einem CBR Flow und variierender Anzahl von TCP Flows (1, 2, ..., 7)

Die Best Case Berechnung gibt den minimalen Ende-zu-Ende Delay für den untersuchten Pfad eines Netzes an. Daran kann der Netzbetreiber die Netzdimensionierung überprüfen, z.B.: überschreitet der minimale Ende-zu-Ende Delay bereits die maximal von den zu bedienenden Anwendungen akzeptierten

Delaywerte. Außerdem kann anhand der Best Case Berechnung ein Vergleich einer momentan in einem Netz vorliegenden Delaysituation mit den theoretisch maximal in diesem Netz erzielbaren Delaywerten durchgeführt werden.



**Abbildung 10.44:** ETS unter Einsatz von BE: Wartezeit in der Warteschlange bei 10 CBR Flows und variierender Anzahl von TCP Flows (1, 2, ..., 7)

## 10.4 Stochastisches Verfahren

Das stochastische Verfahren, das der Gruppe der Berechnungen angehört, wird auf das ETS angewendet. Um die 7 relevanten Extrapolationsfälle zu untersuchen, wird zunächst auf ein Zielszenario mit unterschiedlicher QoS Technologie extrapoliert. Danach wird das Lastspektrum und die Netzkonfiguration des Zielszenarios variiert.

### 10.4.1 Variation der QoS Technologie

Die Extrapolation auf eine andere QoS Technologie beschränkt sich auf ein Ausgangsszenario, das den BE Dienst verwendet und von dem aus auf ein Zielszenario mit einer DiffServ QoS Technologie extrapoliert wird. Eine Extrapolation von einem einen DiffServ Ansatz verwendenden Ausgangsszenario auf ein den BE verwendendes Zielszenario ist möglich, jedoch nicht realistisch. Ein Netzbetreiber wird die

DiffServ Implementation nicht zu Gunsten von BE nicht nutzen. Die Extrapolation von einem implementierten auf einen anderen DiffServ Ansatz ist zwar bei einer Änderung der Anforderungen an ein Netz (neue Anwendungen) denkbar und mit Hilfe des stochastischen Verfahrens durchführbar, ist aber hier von geringem Interesse.

$$\begin{aligned}
E[D_Z]_{olympic} &= \sum_{i=0}^{n-1} E[D_Z]_i = \sum_{i=0}^{n-1} (E[W_Z]_i + E[B_Z]_i + D_{prop,Z,i}) \\
&= \sum_{i=0}^{n-1} \left( \frac{\rho_Z}{1-\rho_Z} \cdot (\rho_{A,ef} \cdot \frac{1-\rho_{A,tcp}}{\rho_{A,tcp}} \cdot (E[D_{A,tcp}]_{be} - D_{prop,A} - E[B_{A,tcp}]) + (\rho_{A,be} \right. \\
&\quad \left. + \rho_{A,qbss}) \cdot \frac{1-\rho_{A,ef}}{\rho_{A,ef}} \cdot (E[D_{A,ef}]_{be} - D_{prop,A} - E[B_{A,ef}]))_i + E[B_Z]_i + D_{prop,Z,i} \right) \\
&\text{mit } n = \text{Anzahl der Netzwerkelemente (hier } n = 5)
\end{aligned}$$

Durch Einsetzen der durch Simulation gewonnenen Ausgangsszenario-parameterwerte und der sich durch die Netzkonfiguration des Zielszenarios ergebenden Zielszenarioparameterwerte entsteht folgendes Ergebnis:

$$\begin{aligned}
&= \frac{0.0537}{1-0.0537} \cdot (0.0026 \cdot \frac{1-0.0511}{0.0511} \cdot 0.000634 \text{ s} + 0.0511 \cdot \frac{1-0.0026}{0.0026} \\
&\quad \cdot 0.000037 \text{ s}) + 0.0000288 \text{ s} + 0.005 \text{ s} \\
&\quad + \frac{0.2148}{1-0.2148} \cdot (0.0104 \cdot \frac{1-0.2014}{0.2014} \cdot 0.00795 \text{ s} + 0.2014 \cdot \frac{1-0.0104}{0.0104} \\
&\quad \cdot 0.000972 \text{ s}) + 0.0001152 \text{ s} + 0.005 \text{ s} \\
&\quad + \frac{0.2148}{1-0.2148} \cdot (0.0104 \cdot \frac{1-0.2014}{0.2014} \cdot 0.00239 \text{ s} + 0.2014 \cdot \frac{1-0.0104}{0.0104} \\
&\quad \cdot 0.000340 \text{ s}) + 0.0001152 \text{ s} + 0.005 \text{ s} \\
&\quad + \frac{0.2148}{1-0.2148} \cdot (0.0104 \cdot \frac{1-0.2014}{0.2014} \cdot 0.00239 \text{ s} + 0.2014 \cdot \frac{1-0.0104}{0.0104} \\
&\quad \cdot 0.000350 \text{ s}) + 0.0001152 \text{ s} + 0.005 \text{ s} \\
&\quad + \frac{0.0537}{1-0.0537} \cdot (0.0026 \cdot \frac{1-0.0511}{0.0511} \cdot 0.000598 \text{ s} + 0.0511 \cdot \frac{1-0.0026}{0.0026} \\
&\quad \cdot 0.000079 \text{ s}) + 0.0000288 \text{ s} + 0.005 \text{ s} \\
&= 0.0344 \text{ s}
\end{aligned} \tag{10.10}$$

Für die Verkehrszusammensetzungen aus 1 CBR Flow, 1 TCP Flow und 1 Video Trace, 1 TCP Flow werden die Extrapolationen durchgeführt. Die Variation der generierten Dateigrößen des TCP Webverkehrs wird für alle Simulationsläufe einmal gering (g) (Variation von 22 bis 3022 Paketen pro Datei) und einmal hoch (h) (Variation von 21 bis 1470880 Paketen pro Datei) gehalten, wobei der Maximalwert der zweiten Variation ein normalerweise niemals erreichbares Extrema darstellt. Dies dient dazu, den Einfluß der Verkehrscharakteristik auf die Gültigkeit des stochastischen Verfahrens zu zeigen, da die Restbedienzeitformeln der DiffServ Ansätze auf M/G/1 Formeln basieren. Für eine Extrapolation auf ein Zielszenario, das den Olympic Ansatz verwendet bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr wird die Berechnung der Ergebnisse als Beispiel demonstriert (siehe Rechnung 10.11).

Die Tabellen 10.36 bis 10.39 zeigen die Ergebnisse der untersuchten Extrapolationen. Hierbei werden die durch Simulation erzielten Extrapolationsergebnisse mit den berechneten verglichen und die prozentuale Abweichung der berechneten von den simulierten Ergebnissen angegeben.

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow (g)	berechnet	0.0257
	simuliert	0.0256
	prozentuale Abweichung	0.39
1 CBR Flow, 1 TCP Flow (h)	berechnet	0.0344
	simuliert	0.0269
	prozentuale Abweichung	30.11

**Tabelle 10.36:** Extrapolation von BE auf Olympic bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow (g)	berechnet	0.0357
	simuliert	0.0333
	prozentuale Abweichung	7.21
1 Video Trace, 1 TCP Flow (h)	berechnet	0.0554
	simuliert	0.0343
	prozentuale Abweichung	61.52

**Tabelle 10.37:** Extrapolation von BE auf Olympic bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow (g)	berechnet	0.0379
	simuliert	0.0257
	prozentuale Abweichung	47.47
1 CBR Flow, 1 TCP Flow (h)	berechnet	0.0420
	simuliert	0.0272
	prozentuale Abweichung	54.41

**Tabelle 10.38:** Extrapolation von BE auf Default bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow (g)	berechnet	0.5947
	simuliert	0.0333
	prozentuale Abweichung	1685.89
1 Video Trace, 1 TCP Flow (h)	berechnet	0.5644
	simuliert	0.0347
	prozentuale Abweichung	1527.51

**Tabelle 10.39:** Extrapolation von BE auf Default bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

An diesen Ergebnissen sieht man sehr gut, daß dieser stochastische Ansatz bei einem extremen Ankunftsprozess, der sich stark von den M/G/1 Voraussetzungen unterscheidet, keine befriedigenden Ergebnisse mehr liefert. Für typischen TCP Webverkehr werden allerdings bei der Extrapolation auf ein den Olympic Ansatz verwendendes Zielszenario akzeptable Werte erzielt. Wenn CBR Verkehr als priorisierter Verkehr verwendet wird, bleiben die Abweichungen von den Simulationsergebnissen unter 1 %, für priorisierten Videoverkehr bleiben sie unter 8 % (siehe Tabelle 10.36 und 10.37). Für die Extrapolation auf ein den Default Ansatz verwendendes Zielszenario liegen sogar die Abweichungen bei verwendetem CBR Verkehr und niedriger Variabilität der TCP Dateigrößen bei ca. 50 %, die Abweichungen bei hoher Variabilität sind noch weit extremer (siehe Tabelle 10.36). Bei verwendetem Videoverkehr weichen alle Ergebnisse extrem ab (siehe Tabelle 10.39). Dies liegt an der Restbedienzeitformel des Default Ansatzes, die ebenfalls wie die Restbedienzeitformel des Olympic Ansatzes auf den M/G/1 Formeln basiert, aber noch stärkere Abhängigkeiten vom Ankunftsprozess demonstriert.

### 10.4.2 Variation des Lastspektrums

Als nächstes wird die Veränderung des Lastspektrums, im Speziellen der Verkehrsintensität getestet. Zunächst wird die Verkehrsintensität um den Faktor  $m$  verändert. Es wird wiederum von demselben Ausgangsszenario mit einer Verkehrszusammensetzung aus 1 CBR Flow, 1 TCP Flow und 1 Video Trace, 1 TCP Flow ausgegangen und auf ein Zielszenario mit 10 CBR, 1 TCP Flow und 10 Video Traces, 1 TCP Flow extrapoliert. Entsprechende Untersuchungen werden für die beiden DiffServ Ansätze durchgeführt. Für die CBR und TCP Verkehrszusammensetzung unter Verwendung von BE wird die Berechnung der Ergebnisse für die Erhöhung der Anzahl von CBR Flows von 1 auf 10 als Beispiel demonstriert (siehe Rechnung 10.12). Da sich der Gesamtverkehr bei den Untersuchungen aus zwei Verkehrsarten zusammensetzt und nur die Verkehrsintensität einer Verkehrsart erhöht, muß der Faktor  $m$  wie in Kapitel 9 beschrieben dies berücksichtigen.

$$\begin{aligned}
 E[D_Z]_{be} &= \sum_{i=0}^{n-1} E[D_Z]_i = \sum_{i=0}^{n-1} (E[W_Z]_i + E[B_Z]_i + D_{prop,Z,i}) \\
 &= \sum_{i=0}^{n-1} \left( \frac{\rho_Z}{1 - \rho_Z} \cdot E[B_Z] \cdot \frac{C_{A,Z}^2 + C_{B,Z}^2}{2} + E[B_Z] + D_{prop,Z} \right) \\
 &= \sum_{i=0}^{n-1} \left( \frac{1 - \rho_A}{1 - (m \cdot \rho_A)} \cdot m \cdot \frac{\rho_A}{1 - \rho_A} \cdot E[B_A] \cdot \frac{C_{A,A}^2 + C_{B,A}^2}{2} + E[B_A] + D_{prop,Z} \right) \\
 &= \sum_{i=0}^{n-1} \left( \frac{1 - \rho_A}{1 - (m \cdot \rho_A)} \cdot m \cdot (E[D_A]_{be} - E[B_A] - D_{prop,A}) + E[B_A] + D_{prop,Z} \right)
 \end{aligned}$$

mit  $n =$  Anzahl der Netzwerkelemente (hier  $n = 5$ )

Durch Einsetzen der durch Simulation gewonnenen Ausgangsszenario-parameterwerte und der sich durch die Topologie des Zielszenarios ergebenden Zielszenarioparameterwerte entsteht folgendes Ergebnis:

$$\begin{aligned}
 &= \frac{0.0537}{1 - 0.0771} \cdot 0.7582 \cdot 0.000037 \text{ s} + 0.0000288 \text{ s} + 0.005 \text{ s} \\
 &\quad \frac{0.2148}{1 - 0.3054} \cdot 0.7582 \cdot 0.000972 \text{ s} + 0.0001152 \text{ s} + 0.005 \text{ s} \\
 &\quad \frac{0.2148}{1 - 0.3054} \cdot 0.7582 \cdot 0.000340 \text{ s} + 0.0001152 \text{ s} + 0.005 \text{ s} \\
 &\quad \frac{0.2148}{1 - 0.3054} \cdot 0.7582 \cdot 0.000350 \text{ s} + 0.0001152 \text{ s} + 0.005 \text{ s} \\
 &\quad \frac{0.0537}{1 - 0.0771} \cdot 0.7582 \cdot 0.000079 \text{ s} + 0.0000288 \text{ s} + 0.005 \text{ s} \\
 &= 0.0269 \text{ s}
 \end{aligned}$$

Die Tabellen 10.40 bis 10.45 zeigen die Ergebnisse der untersuchten Extrapolationen. Hierbei werden die durch Simulation erzielten Extrapolationsergebnisse mit den berechneten verglichen und die prozentuale Abweichung der berechneten von den simulierten Ergebnissen angegeben.

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
10 CBR Flows, 1 TCP Flow	berechnet	0.0269
	simuliert	0.0281
	prozentuale Abweichung	4.27
1 CBR Flow, 7 TCP Flows	berechnet	0.0526
	simuliert	0.0768
	prozentuale Abweichung	31.51

**Tabelle 10.40:** Extrapolation von BE auf BE bei einer veränderten Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
10 Video Traces, 1 TCP Flow	berechnet	0.0643
	simuliert	0.0559
	prozentuale Abweichung	15.03
1 Video Trace, 7 TCP Flows	berechnet	0.1547
	simuliert	0.0845
	prozentuale Abweichung	83.08

**Tabelle 10.41:** Extrapolation von BE auf BE bei einer veränderten Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
10 CBR Flows, 1 TCP Flow	berechnet	0.0263
	simuliert	0.0269
	prozentuale Abweichung	2.23
1 CBR Flow, 7 TCP Flows	berechnet	0.0431
	simuliert	0.0312
	prozentuale Abweichung	38.08

**Tabelle 10.42:** Extrapolation von Olympic auf Olympic bei einer veränderten Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
10 Video Traces, 1 TCP Flow	berechnet	0.0473
	simuliert	0.0499
	prozentuale Abweichung	5.21
1 Video Trace, 7 TCP Flows	berechnet	0.1113
	simuliert	0.0362
	prozentuale Abweichung	207.46

**Tabelle 10.43:** Extrapolation von Olympic auf Olympic bei einer veränderten Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
10 CBR Flows, 1 TCP Flow	berechnet	0.0265
	simuliert	0.0269
	prozentuale Abweichung	1.47
1 CBR Flow, 7 TCP Flows	berechnet	0.0464
	simuliert	0.0352
	prozentuale Abweichung	31.77

**Tabelle 10.44:** Extrapolation von Default auf Default bei einer veränderten Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
10 Video Traces, 1 TCP Flow	berechnet	0.0522
	simuliert	0.0500
	prozentuale Abweichung	4.4
1 Video Trace, 7 TCP Flows	berechnet	0.1307
	simuliert	0.0422
	prozentuale Abweichung	209.72

**Tabelle 10.45:** Extrapolation von Default auf Default bei einer veränderten Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Die Untersuchungen der Extrapolation auf ein Zielszenario mit veränderter Verkehrsintensität zeigen, daß eine Extrapolation auf ein Zielszenario mit einem höheren CBR Verkehrsanteil gute Ergebnisse mit Abweichung von unter 5 % von den Simulationsergebnissen liefert. Auch bei Erhöhung der Intensität des Videoverkehrs im Zielszenario werden Abweichungen unter 15 % erzielt, wobei diese unter der Verwendung von BE weit über den anderen Abweichungen der DiffServ Ansätze liegt. Die geringeren Abweichungen entstehen durch die bereits im Ausgangsszenario durchgeführte Verkehrsregelung durch die verwendeten DiffServ Ansätze. Bei einer Intensivierung des TCP Verkehrs (Erhöhung der Anzahl der TCP Flows) treten bei einer Vermischung mit CBR Verkehr Abweichung zwischen 30 und 40 % auf; für eine Verkehrszusammensetzung aus TCP und Video steigen diese allerdings über 200 %. Dies liegt wiederum an der Schwankung der Paketgrößen und der Ankunftsrate der Videopakete und der Variation der Dateigrößen des TCP Webverkehrs.

Eine Verminderung des Verkehrs könnte auch berechnet werden (entspricht dem Verfahren der Vergrößerung), wird aber hier nicht betrachtet, da diese Extrapolation sehr unrealistisch ist. Ein Netzbetreiber wird sich kaum dafür interessieren, wie hoch sein Ende-zu-Ende Delay bei Reduktion seines Verkehrsaufkommens ausfallen wird, außer er hat rückläufige Benutzerzahlen bzw. ein reduziertes Verkehrsaufkommen und will gleichzeitig seine Kapazitäten reduzieren. Er interessiert sich dafür, wieviel er noch transportieren kann, wenn die Dienstgüte verbessert werden muß.

### 10.4.3 Variation der Netzkonfiguration

Zum Abschluß wird die Veränderung der Netzwerkkonfiguration, im Speziellen der Linkkapazität getestet. Es wird die Linkkapazität aller Links des ETS einmal verdoppelt und einmal halbiert. Das Ausgangsszenario bleibt identisch. Die beiden Verkehrsmischungen 1 CBR Flow, 1 TCP Flow und 1 Video Trace, 1 TCP Flow werden untersucht. Die Extrapolation wird für BE, Olympic und Default Ansatz durchgeführt. Für den CBR und TCP Verkehrszusammensetzung unter Verwendung von BE wird die Berechnung der Ergebnisse für eine Verdoppelung der Linkkapazität als Beispiel demonstriert.

$$\begin{aligned}
 E[D_Z]_{be} &= E[W_Z] + E[B_Z] + D_{prop,Z} \\
 &= \frac{\rho_Z}{1 - \rho_Z} \cdot E[B_Z] \cdot \frac{C_{A,Z}^2 + C_{B,Z}^2}{2} + E[B_Z] + D_{prop,Z} \\
 &= \frac{\frac{\rho_A}{k - \rho_A}}{\frac{\rho_A}{1 - \rho_A}} \cdot \frac{1}{k} \cdot \frac{\rho_A}{1 - \rho_A} \cdot E[B_A] \cdot \frac{C_{A,A}^2 + C_{B,A}^2}{2} + \frac{1}{k} \cdot E[B_A] + D_{prop,Z} \\
 &= \frac{\frac{\rho_A}{k - \rho_A}}{\frac{\rho_A}{1 - \rho_A}} \cdot \frac{1}{k} \cdot (E[D_A]_{be} - E[B_A] - D_{prop,A}) + \frac{1}{k} \cdot E[B_A] + D_{prop,Z}
 \end{aligned}$$

mit  $n$  = Anzahl der Netzwerkelemente (hier  $n = 5$ )

Durch Einsetzen der durch Simulation gewonnenen Ausgangsszenario-parameterwerte und der sich durch die Topologie des Zielszenarios ergebenden Zielszenarioparameterwerte entsteht für  $m$  (ergibt sich aus der Erhöhung der Anzahl an CBR Flows) folgendes Ergebnis:

$$\begin{aligned}
&= \frac{\frac{0.0537}{2-0.0537}}{\frac{0.0537}{1-0.0537}} \cdot \frac{1}{2} \cdot 0.000037 \text{ s} + \frac{1}{2} \cdot 0.000029 \text{ s} + 0.0005 \text{ s} \\
&= \frac{\frac{0.2148}{2-0.2148}}{\frac{0.2148}{1-0.2148}} \cdot \frac{1}{2} \cdot 0.000972 \text{ s} + \frac{1}{2} \cdot 0.000115 \text{ s} + 0.0005 \text{ s} \\
&= \frac{\frac{0.2148}{2-0.2148}}{\frac{0.2148}{1-0.2148}} \cdot \frac{1}{2} \cdot 0.000340 \text{ s} + \frac{1}{2} \cdot 0.000115 \text{ s} + 0.0005 \text{ s} \\
&= \frac{\frac{0.2148}{2-0.2148}}{\frac{0.2148}{1-0.2148}} \cdot \frac{1}{2} \cdot 0.000350 \text{ s} + \frac{1}{2} \cdot 0.000115 \text{ s} + 0.0005 \text{ s} \\
&= \frac{\frac{0.0537}{2-0.0537}}{\frac{0.0537}{1-0.0537}} \cdot \frac{1}{2} \cdot 0.000079 \text{ s} + \frac{1}{2} \cdot 0.000029 \text{ s} + 0.0005 \text{ s} \\
&= 0.0256 \text{ s}
\end{aligned}$$

Insgesamt kommt man zu folgenden Ergebnissen:

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow	berechnet	0.0256
	simuliert	0.0259
	prozentuale Abweichung	1.16

**Tabelle 10.46:** Extrapolation von BE auf BE mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow	berechnet	0.0355
	simuliert	0.0467
	prozentuale Abweichung	23.98

**Tabelle 10.47:** Extrapolation von BE auf BE mit halbiertem Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow	berechnet	0.0302
	simuliert	0.0295
	prozentuale Abweichung	2.37

**Tabelle 10.48:** Extrapolation von BE auf BE mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow	berechnet	0.1683
	simuliert	0.0772
	prozentuale Abweichung	118.01

**Tabelle 10.49:** Extrapolation von BE auf BE mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow	berechnet	0.0254
	simuliert	0.0257
	prozentuale Abweichung	1.17

**Tabelle 10.50:** Extrapolation von Olympic auf Olympic mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow	berechnet	0.0315
	simuliert	0.0298
	prozentuale Abweichung	5.70

**Tabelle 10.51:** Extrapolation von Olympic auf Olympic mit halbiertem Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow	berechnet	0.0290
	simuliert	0.0290
	prozentuale Abweichung	0.00

**Tabelle 10.52:** Extrapolation von Olympic auf Olympic mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow	berechnet	0.1179
	simuliert	0.0462
	prozentuale Abweichung	155.19

**Tabelle 10.53:** Extrapolation von Olympic auf Olympic mit halbierten Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow	berechnet	0.0255
	simuliert	0.0258
	prozentuale Abweichung	1.16

**Tabelle 10.54:** Extrapolation von Default auf Default mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 CBR Flow, 1 TCP Flow	berechnet	0.0325
	simuliert	0.0312
	prozentuale Abweichung	4.17

**Tabelle 10.55:** Extrapolation von Default auf Default mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow	berechnet	0.0275
	simuliert	0.0291
	prozentuale Abweichung	5.50

**Tabelle 10.56:** Extrapolation von Default auf Default mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

Die Extrapolationen auf ein Zielszenario mit doppelter Linkkapazität liefern stets sehr geringe Abweichungen von den Simulationsergebnissen unter 5 % für alle QoS Technologien und Verkehrszusammensetzungen. Wird die Linkkapazität allerdings halbiert können nur mehr für eine Verkehrszusammensetzung aus CBR und TCP Verkehr akzeptable Werte erzielt werden. Dabei liefern ebenfalls wieder die Extrapolationen auf Zielszenarien mit eingesetzten DiffServ Ansätzen von Ausgangsszenarien, die bereits diese Ansätze einsetzen, auf Grund der Verkehrsregelung im Ausgangsszenario bessere Ergebnisse unter 6 % Abweichung, als die Extrapolationen von BE verwendenden Szenarien. Für Verkehrszusammensetzungen aus Video und TCP Verkehr liefert lediglich der Olympic Ansatz bei Halbierung der Linkkapazität akzeptable Abweichungen von ca. 15 %.

Verkehrszusammensetzung	Erzeugung	mittlerer E2E Delay [s]
1 Video Trace, 1 TCP Flow	berechnet	0.0405
	simuliert	0.0479
	prozentuale Abweichung	15.45

**Tabelle 10.57:** Extrapolation von Default auf Default mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten

#### 10.4.4 Bewertung des stochastischen Verfahrens

Für eine Extrapolation vom angegebenen Ausgangsszenario auf ein Zielszenario mit differierender QoS Technologie (Olympic oder Default Ansatz) sind die Ergebnisse für einen stark variierenden Ankunftsprozess für alle QoS Technologien unzureichend. Für Kombinationen von CBR oder Videoverkehr mit typischem TCP Webverkehr (s.o. beschränkte Dateigrößen) werden für die Extrapolation auf den Olympic Ansatz akzeptable Ergebnisse (5% - 15% Abweichung) erzielt. Die Extrapolation auf den Default Ansatz liefert auch für diese Verkehrszusammensetzungen unzureichende Ergebnisse.

Eine Extrapolation auf ein Zielszenario mit einer höheren Verkehrsintensität des CBR oder des Videoverkehrs liefert gute Ergebnisse (1% - 15% Abweichung). Eine Variation der Intensität des TCP Webverkehrs führt zu unzureichenden Ergebnissen.

Eine Extrapolation auf ein Zielszenario mit erhöhten Linkkapazitäten führt durchwegs für alle QoS Technologien zu guten Ergebnissen (- 5% Abweichung). Eine Verminderung liefert lediglich für den Olympic Ansatz für alle untersuchten Verkehrszusammensetzungen akzeptable Werte (- 15% Abweichung). Für BE und den Default Ansatz bleiben die Werte nur für CBR mit TCP Hintergrundverkehr in diesen Grenzen.

Es sind natürlich auch Kombinationen einer QoS Technologievariation und einer Änderung des Lastspektrums denkbar. Zur stochastischen Ermittlung der Ende-zu-Ende Delaywerte dieser Extrapolationsszenarien werden die zuvor vorgestellten Formeln je nach Veränderung des Extrapolationszielszenarios kombiniert. Auf Grund der zahlreichen Kombinationsmöglichkeiten wird allerdings auf die detaillierte Untersuchung dieser Kombinationen verzichtet. Desweiteren ist eine Extrapolation auf ein Zielszenario mit einer anderen Topologie nicht möglich, da die Inputdaten hinzugefügter Netzwerkelemente aus dem Ausgangsszenario fehlen. Für den Netzbetreiber ist das stochastische Verfahren je nach Verkehrscharakteristik bergrenzt nutzbar.

## 10.5 Allgemeine Extrapolationsdurchführung

Basierend auf den zuvor aus den ETS Extrapolationen mit Hilfe der drei entwickelten Verfahren erzielten Ergebnissen wird eine allgemeine Vorgehensweise zur Durchführung einer Extrapolation gegeben und diese grafisch in Ablaufdiagrammen dargestellt:

Zu Beginn muß ein Ausgangs- und Zielszenario, basierend auf der Extrapolationsszenariodefinition von Kapitel 8, spezifiziert werden. Durch eine Analyse der zu untersuchenden Netzauslastung des Zielszenarios (niedrig ausgelastete Teilnetze z.B. Kernnetz benötigen keine QoS Technologie) und der im Ausgangsszenario vorhandenen Topologie (beinhaltet hier auch Routereigenschaften, wie DiffServ- oder IntServ Fähigkeit) kann eine Auswahl der zu untersuchenden Extrapolationsfälle getroffen werden oder auch eine Einschränkung der geeigneten QoS Technologien für das gewählte Netzwerk angegeben werden und somit die Anzahl der Extrapolationen verringert und eine Vielzahl von Ergebnisvergleichen eingespart werden (siehe Ablaufdiagramm 10.45).

Soll dann eine Extrapolation vom Ausgangs- auf das Zielszenario durchgeführt werden, werden die Voraussetzungen für eine Extrapolation geprüft (siehe Ablaufdiagramm 10.45 und 10.46). Dabei wird zunächst getestet, ob Informationen oder Vorstellungen über die genaue Verkehrszusammensetzung (in dieser Arbeit lediglich CBR/TCP und Video/TCP) und -intensität im Zielszenario vorhanden sind. Ist dies nicht der Fall, kann keine Extrapolation durchgeführt werden. Danach wird ein Test zur Feststellung der Existenz von Meßdaten (Queueing Delays, Auslastungen und Bedienzeiten) des Ausgangsszenarios vollzogen. Sind keine Meßdaten vorhanden, kann lediglich die Simulation unter der Voraussetzung als Extrapolationsverfahren eingesetzt werden, daß der Simulator für ein Ausgangsszenario kalibriert und dessen Ergebnisse validiert wurden; die auf den Meßdaten basierenden analytischen Verfahren können nicht verwendet werden. Weiter wird geprüft, ob die Netzausdehnung des Zielszenarios extrem ist, dies bedeutet, man testet ob die Kapazitäten des Simulationsrechners und die akzeptable Dauer der Ergebniserzeugung überschritten werden. Ist dies der Fall, kann keine Simulation zur Extrapolation verwendet werden. Wenn sowohl keine Meßdaten vorhanden sind als auch die Netzausdehnung zu groß ist, kann wiederum keine Extrapolation mit den drei entwickelten Verfahren ausgeführt werden (siehe Ablaufdiagramm 10.46).

Als nächster Schritt werden alle Extrapolationsfälle und die mit Hilfe der drei Extrapolationsverfahren erzielbaren Ergebnisgüten, die auf den ETS Ergebnissen basieren, für jeden Fall für die untersuchten Verkehrszusammensetzungen und etwaige Durchführungsempfehlungen dargestellt (siehe Ablaufdiagramm 10.47, 10.48 und 10.49). Unter Durchführungsempfehlungen versteht man hier eine Angabe der Reihenfolge der Anwendung der unterschiedlichen Extrapolationsverfahren, z.B. kann eine vorgezogene Worst Case Analyse weitere Untersuchungen überflüssig machen. Für jeden Extrapolationsfall, ausgenommen der zuvor behandelten Sonderfälle und die Extrapolation auf ein IntServ verwendendes Zielszenario, können die drei Extrapolationsverfahren eingesetzt werden. Für IntServ wurde ein stochastisches Verfahren aus geringem technischen Interesse nicht verfolgt, da für IntServ der Worst Case Ende-zu-Ende Delay vorgegeben wird. Einige weitere Extrapolationsvarianten wurden auf Grund ihrer fehlenden Relevanz für reale Extrapolationsanwendungen im ETS nicht untersucht oder wegen ihrer Trivialität vernachlässigt. Die Simulationsergebnisse werden auf Grund der durchgeführten Validierung des Simulators als exakt vorausgesetzt. Für die Best/Worst Case Analyse und das stochastische Verfahren wird jeweils die aus den ETS Simulationsergebnissen verallgemeinerte Abweichungsqualität von den Simulationsergebnissen angegeben. Hierbei entspricht eine Abweichung von den Simulationsergebnissen im ETS von 0 - 5 % einer sehr guten Abweichungsqualität (in den Ablaufdiagrammen abgekürzt mit sg), von 5 - 20 % einer guten Abweichungsqualität (abgekürzt mit g) und von über 20 % einer schlechten Abweichungsqualität (abgekürzt mit s).

Es werden je nach Extrapolationszielszenario für einen oder mehrere Freiheitsgrade (QoS Technologie, Lastspektrum und Netzkonfiguration) zunächst alle möglichen Extrapolationsverfahren, deren Ergebnisgüte und etwaige Durchführungsempfehlungen für das zu untersuchende Extrapolationszielszenario

ausgewählt (siehe Ablaufdiagramm 10.47, 10.48 und 10.49). Die Verfahren und Güten werden dann jeweils einzeln für jeden Freiheitsgrad mit den angestrebten Ergebnisgüten verglichen und somit gefiltert. Die resultierenden Verfahren und Güten werden entweder für die weitere Extrapolationsdurchführung weitergereicht oder nochmals durch eine durch Kombination von Freiheitsgraden auftretende Kombination von Extrapolationen gefiltert und dann wiederum weitergereicht. Da sich diese Ergebnisse je nach Extrapolationsszenario unterscheiden, sind die Ergebnisfelder in Ablaufdiagramm 10.50 freigelassen und lediglich die Erzeugung der Ergebnisse dargestellt.

Das oder die für die Extrapolation gefundene Verfahren und dessen Durchführungsempfehlung werden dann ausgeführt und liefern die zur Beurteilung des Netzwerkes hinsichtlich einer Übertragung von QoS-sensitivem Verkehr notwendigen QoS Parameter.

Die grafische Extrapolationsdurchführungsbeschreibung beschränkt sich bei Extrapolationen auf Ziel-szenarien mit variierenden QoS Technologien aus Gründen der Übersichtlichkeit auf Ausgangsszenarien, die BE verwenden.

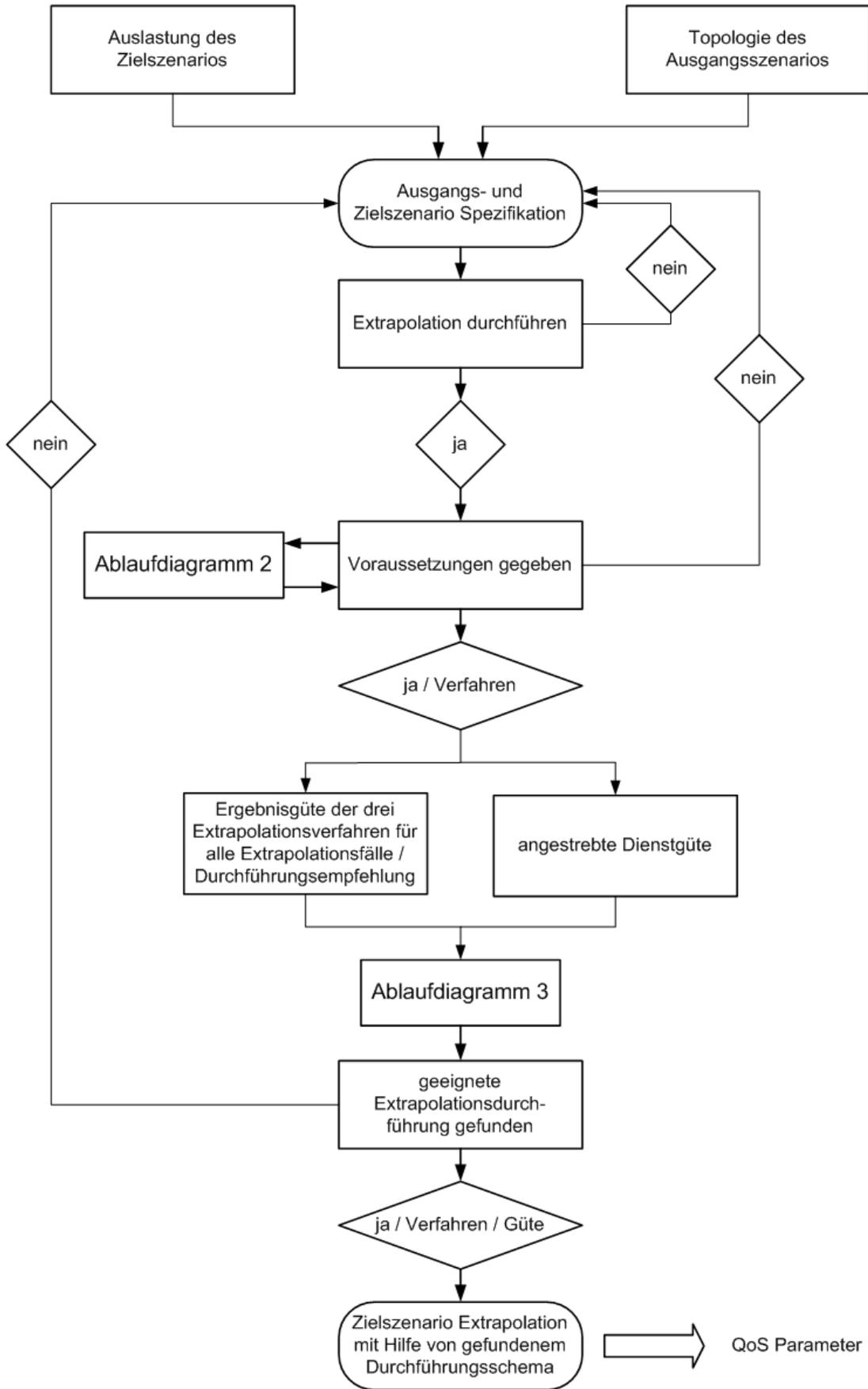


Abbildung 10.45: Ablaufdiagramm 1

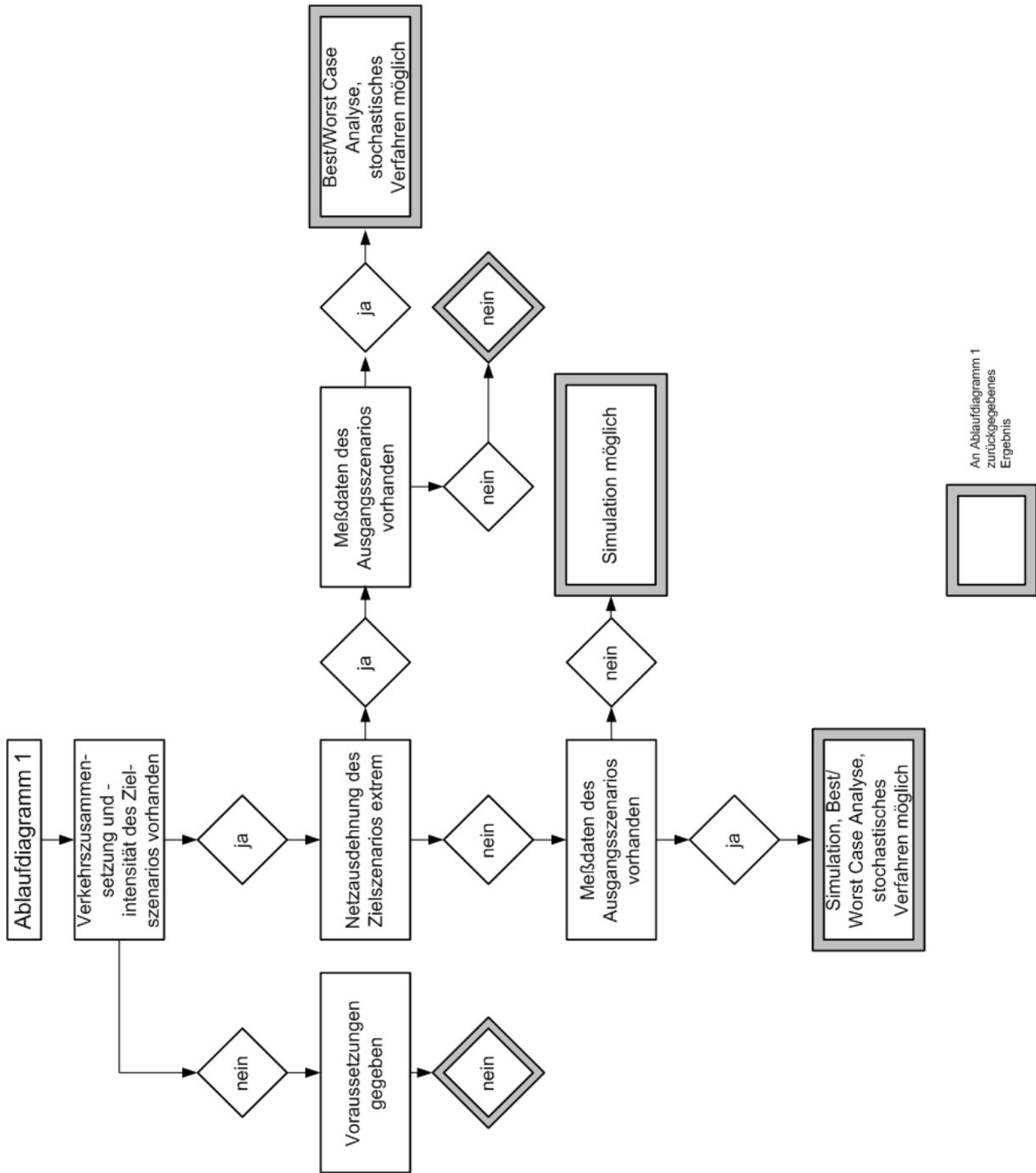


Abbildung 10.46: Ablaufdiagramm 2



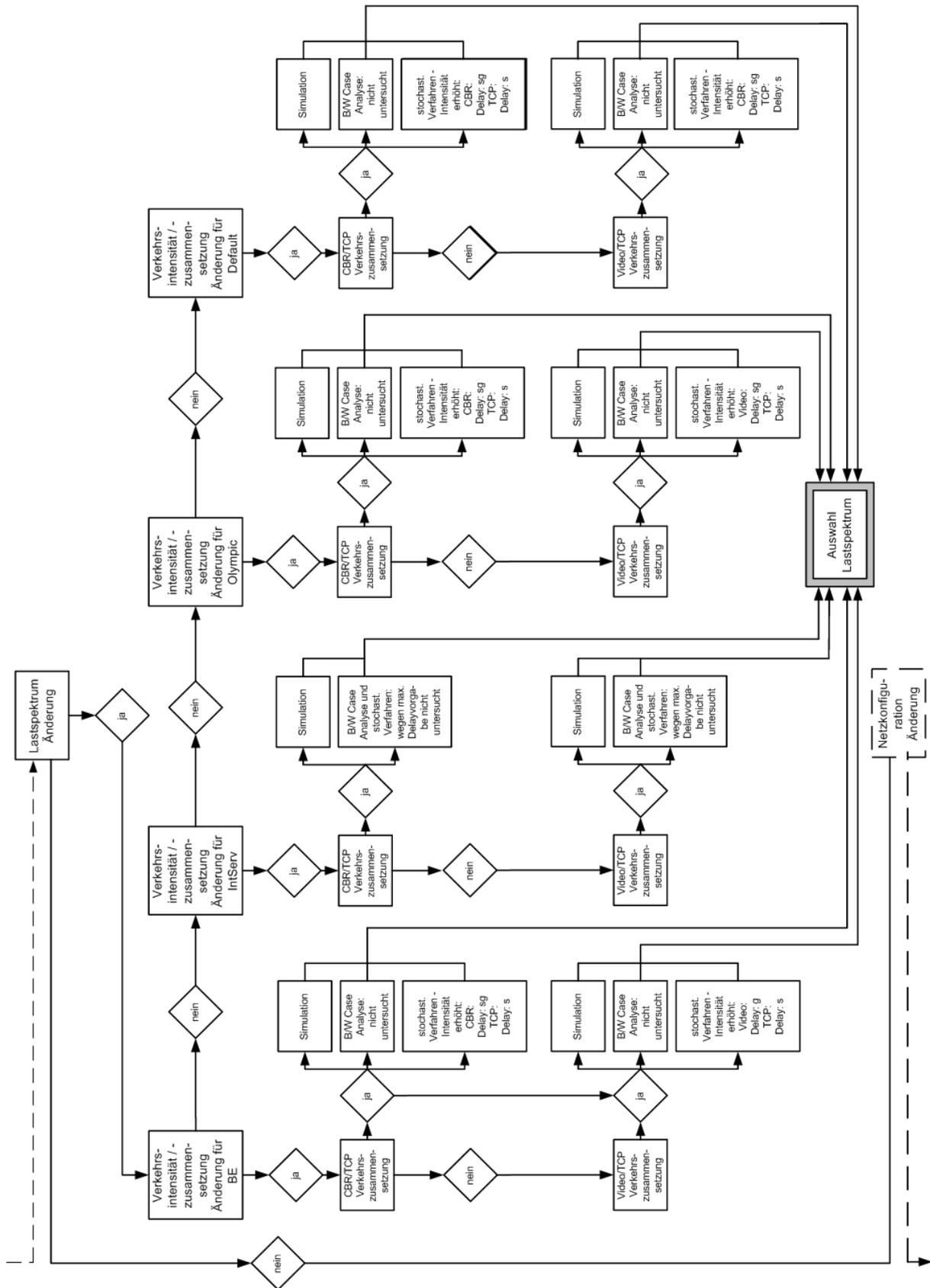


Abbildung 10.48: Ablaufdiagramm 4

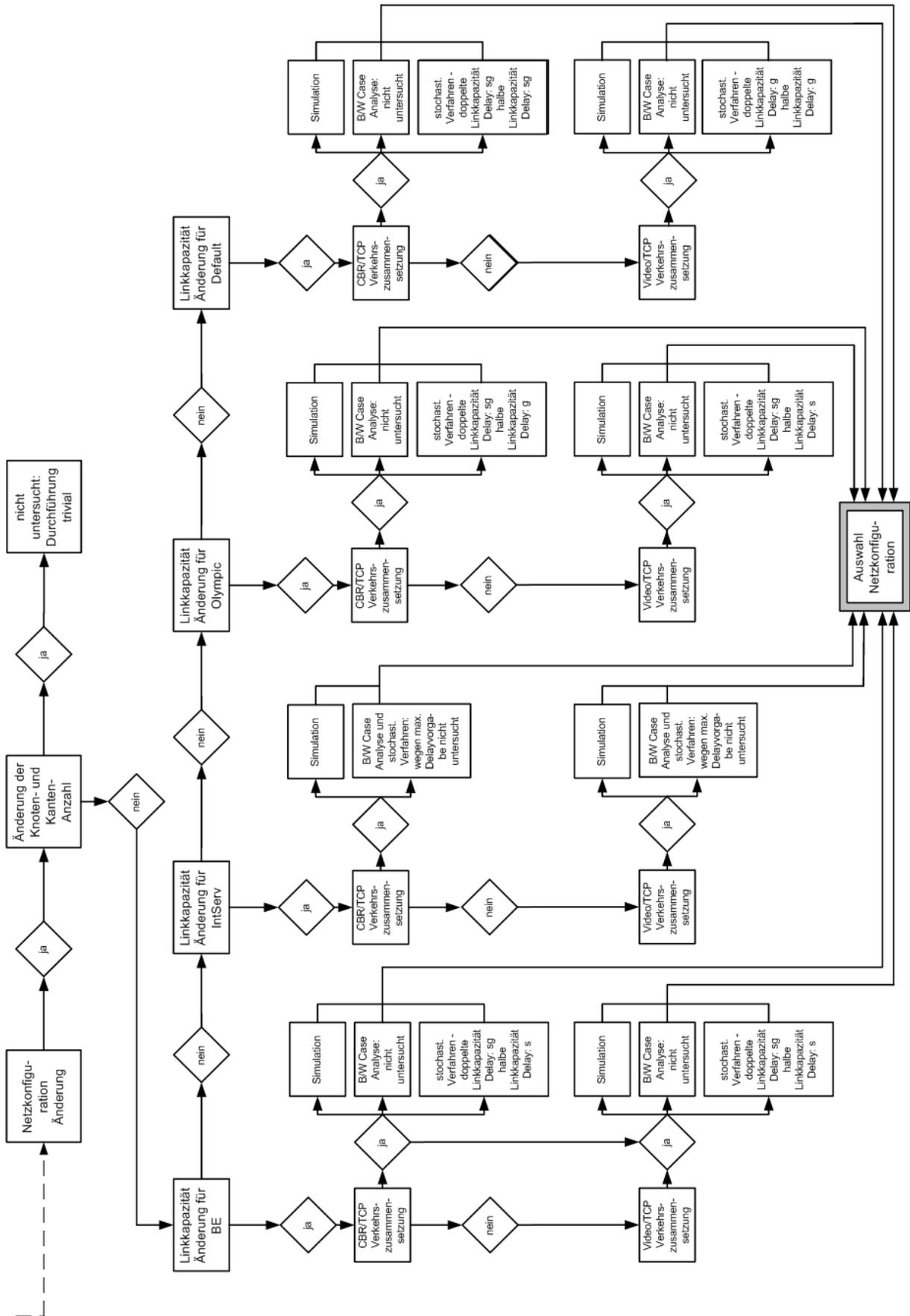


Abbildung 10.49: Ablaufdiagramm 5

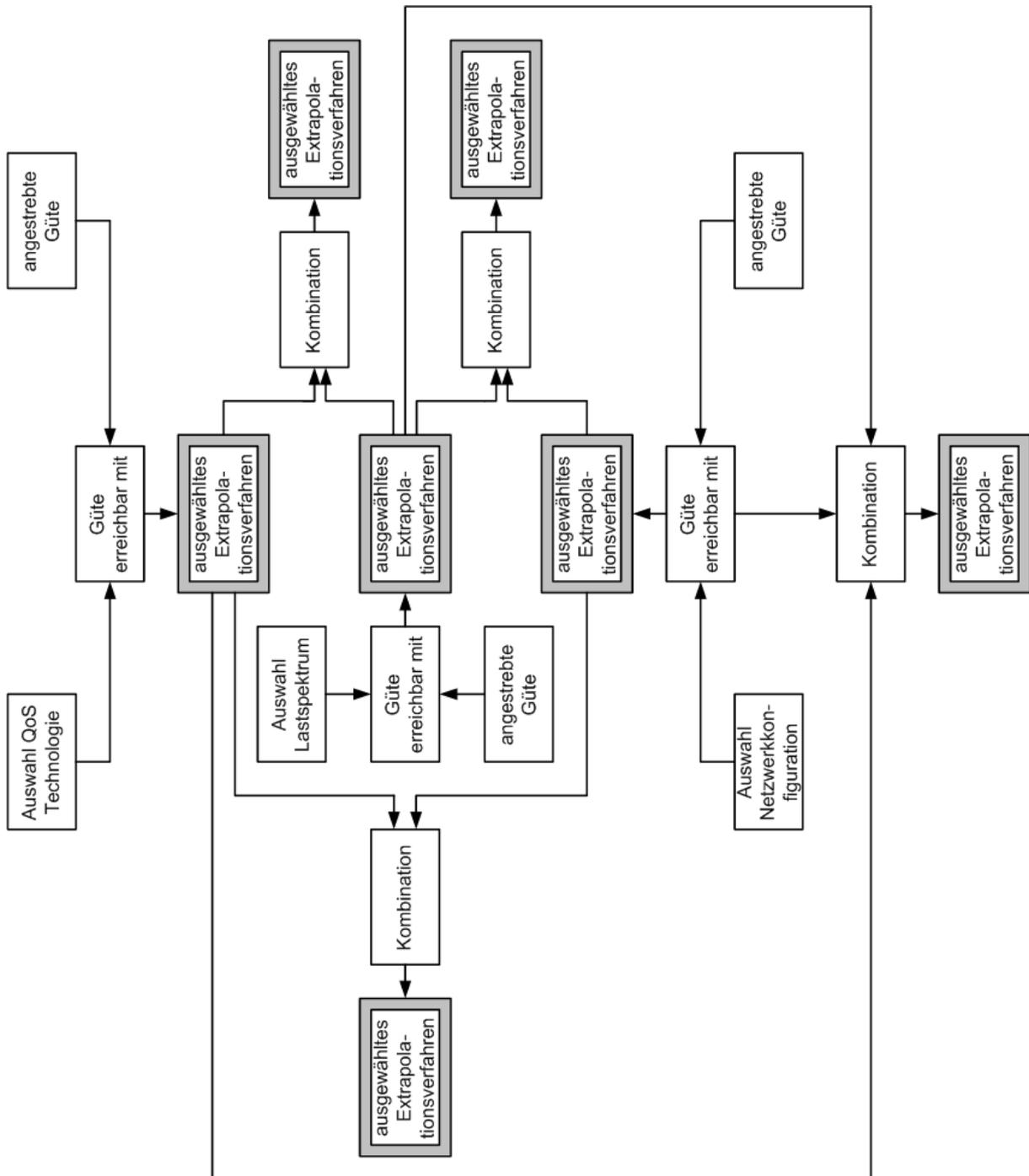


Abbildung 10.50: Ablaufdiagramm 6

## Kapitel 11

# G-WiN Extrapolations-Szenario

Da eine wesentliche Aufgabe dieser Arbeit die Extrapolation der zuvor am Testszenario erzielten Ergebnisse auf das G-WiN und die Extrapolation von G-WiN Ausgangsverhältnissen auf geänderte G-WiN Zielverhältnisse ist, wird die in Kapitel 9 beschriebene Vorgehensweise bei einer Extrapolation auf das G-WiN angewendet. Zunächst wird das Ausgangs- und die Zielszenarien auf Grund der im Anschluß durchgeführten Untersuchungen der Topologie und der Lastverhältnisse des G-WiN spezifiziert. Danach werden die speziellen Voraussetzungen des G-WiN, die sich aus diesen Untersuchungen mit ergeben haben, überprüft und eine Vorauswahl adäquater Extrapolationsverfahren getroffen. Zur Prüfung dieser auf der Durchführungsanleitung basierenden Vorauswahl werden nicht nur diese empfohlenen, sondern alle drei Verfahren für die gewählten Extrapolationen verwendet. Die Ergebnisse der drei Extrapolationsverfahren werden dargestellt und mit ihrer Hilfe eine Bewertung der verwendeten Ansätze zur Erlangung der geforderten Zielszenario-Dienstgütern durchgeführt.

### 11.1 G-WiN Topologie und Auslastung

#### 11.1.1 Beschreibung

“Das Deutsche Forschungsnetz (DFN) ist das von der Wissenschaft selbst verwaltete Hochleistungsnetz für Lehre und Forschung in Deutschland. Es verbindet Hochschulen und Forschungseinrichtungen miteinander und unterstützt die Entwicklung und Erprobung neuer Anwendungen innerhalb der Internet2-Community in Deutschland. Das nationale Backbone des DFN ist das Gigabit-Wissenschaftsnetz G-WiN. Über den europäischen Backbone GÉANT ist das G-WiN mit dem weltweiten Verbund der Forschungs- und Wissenschaftsnetze verbunden. Verträge und Peering-Vereinbarungen integrieren das G-WiN in das globale Internet“ [DFN 01].

#### 11.1.2 Die G-WiN Kernnetztopologie

Momentan ist das G-WiN bzgl. der Anzahl der Verbindungen und des Durchsatzes eines der größten Wissenschaftsnetze der Welt, mit einer für November 2003 geschätzten Anzahl von ca. 2 Millionen Benutzern bei ca. 600 Anschlußpunkten und einem exportierten Datenvolumen von ca. 1,2 PBytes/Monat (Stand: 31.11.03). Das G-WiN besteht aus einem Kernnetz, das 27 Router beinhaltet (Stand 31.11.2003, siehe Abbildung 11.1, [ADLER 02]). Dieses wird erweitert durch 71 Zugangsrouten (*access router*), die über Zugangsleitungen mit Bandbreiten zwischen 128 Kbit/s und 622 Mbit/s verfügen. Dabei werden Cisco Router vom Typ 7507 für Zugangsleitungen bis zu 155 Mbit/s und Cisco Router vom Typ 12008 für Zugangsleitungen ab 155 Mbit/s verwendet.

Zusätzlich bestehen Verbindungen zu anderen Telekommunikationsnetzen:

- Eine 2.5 Gbit/s Verbindung zum europäischen GÉANT Netzwerk

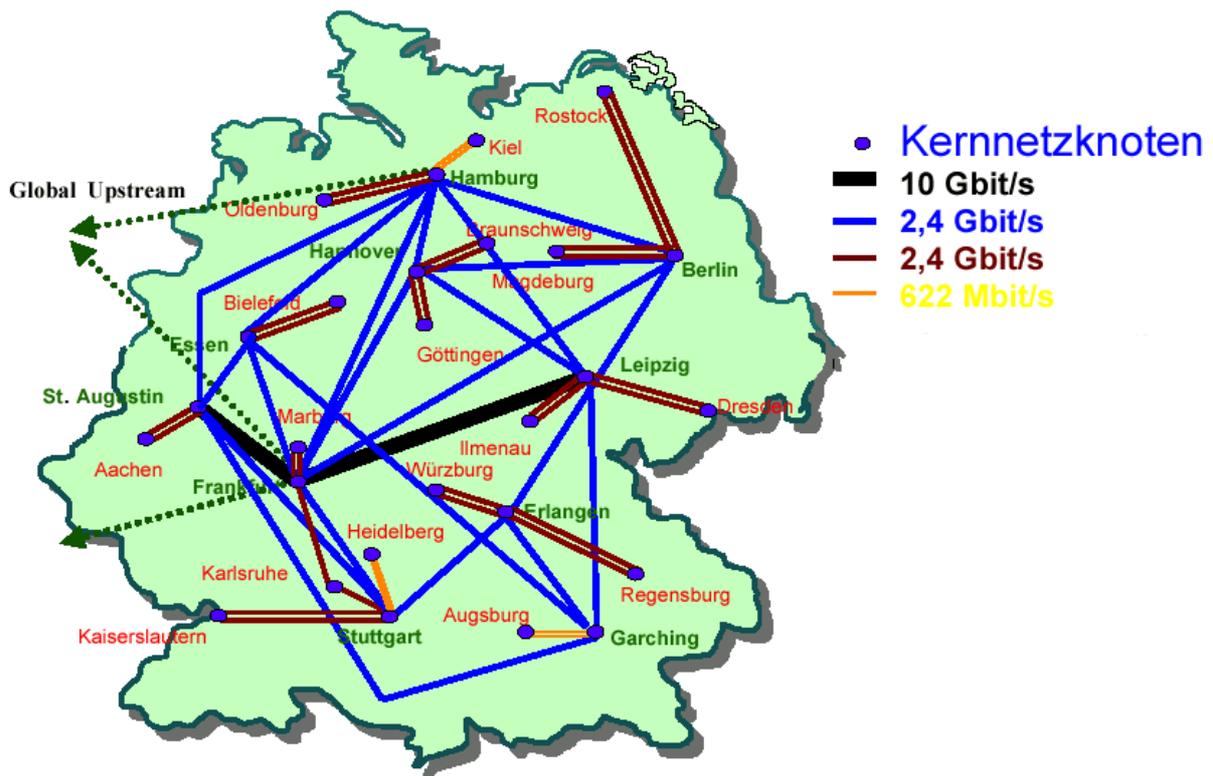


Abbildung 11.1: G-WiN Kernnetztopologie (Stand: 31.11.2003)

- Vom GÉANT Netzwerk drei 2.5 Gbit/s Verbindungen nach Nordamerika
- Zwei 2.5 Gbit/s Verbindungen zu kommerziellen Netzwerken (Provider: Global Crossing von Frankfurt aus / TELIA von Hamburg aus)

Das Kernnetzwerk besteht, wie bereits zuvor erwähnt, aus 27 Routern. Diese werden nach ihrer Positionierung im Kernnetz in 10 sogenannte Level-1 und 17 Level-2 Router unterteilt. Die Level-1 Router sind vom Typ Cisco 12016, die Level-2 Router vom Typ Cisco 12008 (mehr Informationen über die Leistungsparameter dieser Router kann [CISCO SYSTEMS 12008, CISCO SYSTEMS 12016] entnommen werden). Eine Tabelle mit einer Auflistung der einzelnen Router Standorte, Level und Typbezeichnungen ist im Anhang D zu finden.

Die typische Übertragungstechnik für extrem leistungsfähige Netzwerke ist IP über SDH/WDM. SDH (Synchronous Digital Hierarchy) dient zur Aufspaltung eines Netzes aus optischen Kanälen in individuelle Ende-zu-Ende Kanäle fester Bandbreite. Zusätzlich wird WDM (Wavelength Division Multiplexing) zum Betrieb mehrerer optischer Kanäle (Wellenlängen) auf einem Lichtleiter eingesetzt. Durch diese Techniken können optische Kanäle von 622 Mbit/s bis zu 10.0 Gbit/s im Kernnetz zur Verfügung gestellt werden (Stand 31.11.2003).

Insgesamt enthält das Kernnetz 56 interne Verbindungen, die sich, charakterisiert durch den Typ der Knoten, in folgende Verbindungsarten aufspalten (siehe Abbildung 11.1):

Verbindungen zwischen Level-1 Routern:

- 20 x 2.4 Gbit/s
- 2 x 10 Gbit/s

Verbindung zwischen Level-1 und Level-2 Routern:

- 27 x 2.4 Gbit/s
- 6 x 622 Mbit/s

Jeder der 17 Level-2 Router ist zur Ausfallsicherheit (Protection) genau über zwei Leitungen mit einem Level-1 Router verbunden, was zu einer Gesamtanzahl von 34 Level-1 zu Level-2 Router-Verbindungen führt. Level 1 Router sind über mehrere benachbarte Router erreichbar und daher mit großer Sicherheit immer über wenigstens eine Leitung miteinander verbunden.

### 11.1.3 Lastsituation des G-WiN

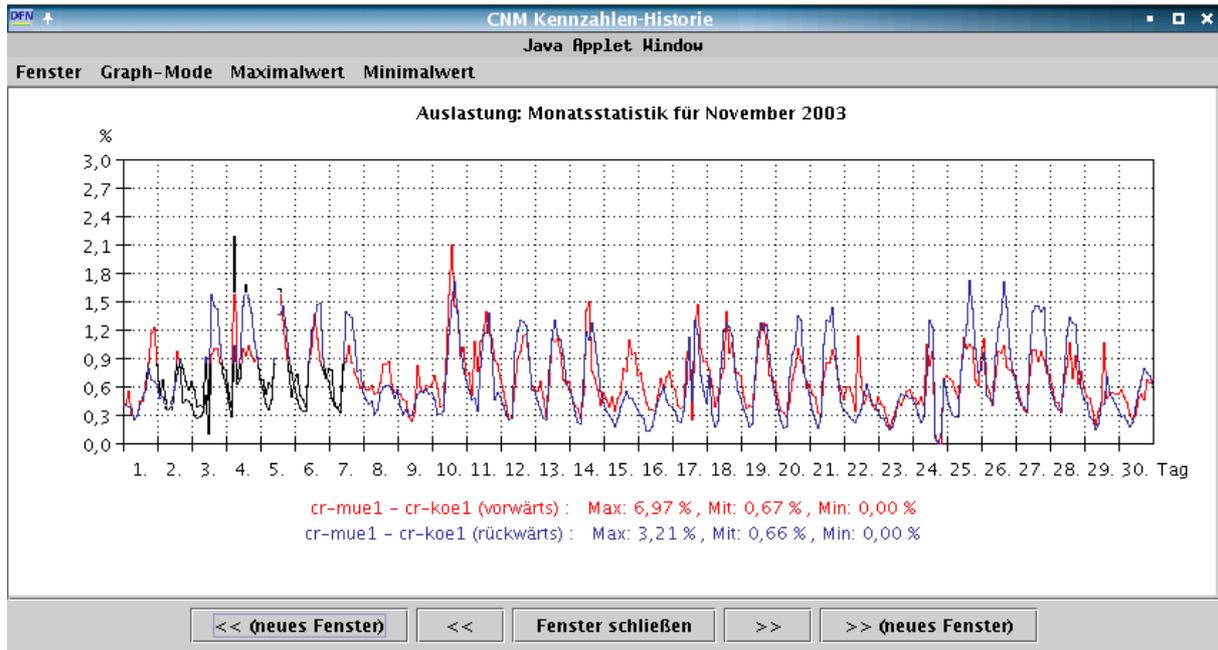
Im Folgenden wird die Lastsituation im G-WiN betrachtet, um basierend auf dieser QoS Technologieauswahlen für den Einsatz im G-WiN zu treffen. Es wird hierbei zwischen dem Kernnetz und den Zugangsleitungen differenziert.

#### 11.1.3.1 Kernnetzauslastung

Um einen Eindruck von der Lastsituation im G-WiN zu bekommen, wird die durchschnittliche und maximale Auslastung der einzelnen Kernnetzverbindungen über einen Monat betrachtet. Dabei werden die Auslastungen dem CNM (Customer Network Management) System des DFN entnommen. Diese werden in Intervallen von 15 Minuten gemessen. Messungen in kürzeren Abständen sind auf Grund der dadurch für die Router zusätzlich entstehenden Belastungen nicht möglich.

Wie den Tabellen D.2, D.3 und D.4 im Anhang D zu entnehmen ist, überschreitet die durchschnittliche Auslastung einer jeden einzelnen Verbindung nicht die 10 % Marke, wobei meist noch wesentlich geringere Auslastungen auftreten. Die höchsten Auslastungen liegen, wie nicht anders zu erwarten war, auf Leitungen vor, die auf dem Pfad zum Level 1 Knoten Frankfurt liegen, da an diesem die Verbindung zum GÉANT Netz und eine Überseeanbindung vorhanden sind. Diese Verhältnisse werden zusätzlich von Delaymessungen für jeden Hop im Kernnetz belegt, in deren Rahmen der Delay den Wert von 10 ms nicht überschritten hat (Messungen des DFN). Es wurden zwar bei zwei, drei einzelnen Verbindungen maximale Auslastungen von 40 % erreicht, diese wurden jedoch jeweils nur an einem einzelnen Tag gemessen und können daher auf Konferenzen oder ähnliche Veranstaltungen oder Anlässe zurückgeführt werden und stellen somit einen Ausnahmestand dar. Die meisten Maximalwerte liegen wiederum weit unterhalb dieser Werte. Die Abbildung 11.2 zeigt einen Beispielplot einer Kernnetzverbindung mit den Kurven für die Auslastung in 15-Minutenintervallen in beiden Verbindungsrichtungen.

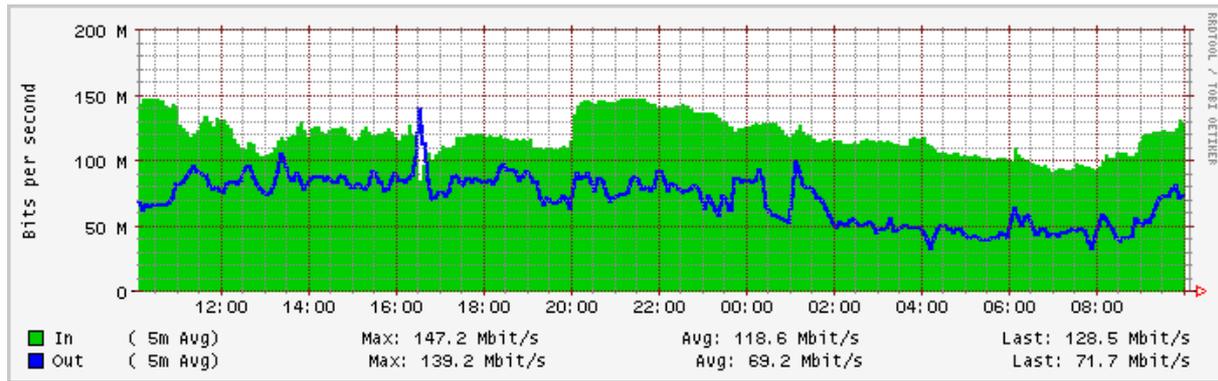
Die ebenfalls zu beachtenden Kernnetz-Routerauslastungen, die im Extrembereich durch zeitliche Verzögerung der Paketverarbeitung erheblichen Einfluß auf die Güte eines Dienstes haben können, spielen momentan im G-WiN laut DFN keine Rolle. Etwaige entstehende Engpässe könnten durch ein Aufrüsten der Router auf neue Linecardtypen, natürlich in Verbindung mit Zusatzkosten, behoben werden.



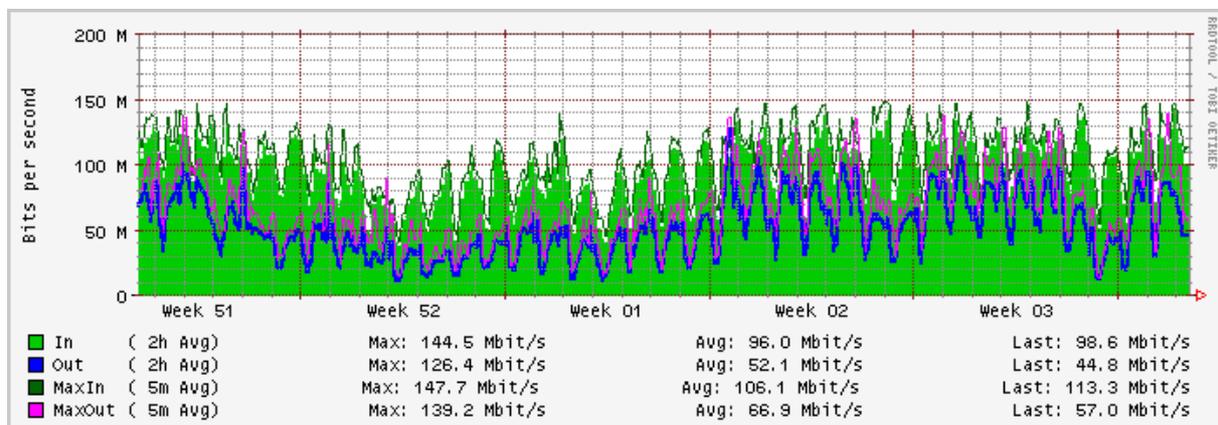
**Abbildung 11.2:** Beispiel einer Kernnetzverbindung: G-WiN Kernnetzleitungsauslastung über einen Monat im 15 min - Mittel (Stand: 31.11.2003)

### 11.1.3.2 Auslastung der Zugangsleitungen

Die Auslastung der Zugangsleitungen wird lediglich an einer beispielhaften, anonymisierten, hochausgelasteten Zugangsleitung aufgezeigt, da Detailinformationen über individuelle Zugangsleitungen dem vertraglichen Vertrauensschutz unterliegen und andererseits für diese Arbeit lediglich die Existenz von hochausgelasteten Zugangsleitungen sowie deren Auslastungsniveau von Interesse ist. Die Abbildungen 11.3 und 11.4 zeigen die Auslastungen einer Zugangsleitung zu einem Kernnetzknotten des G-WiN in beiden Richtungen über einen Tag (Abbildung 11.3) und über eine Woche (Abbildung 11.4). Hierbei wurde die durchschnittliche und die maximale Auslastung in 5 Minutenintervallen gemessen. Bei der Monatsauslastung wurde zusätzlich auch in 2 Stundenintervallen gemessen. Die Messungen zeigen, daß auf Zugangsleitungen durchaus mittlere Auslastungen von mehr als 90 % auftreten können.



**Abbildung 11.3:** Durchsatz einer 155 Mb/s - G-WiN Zugangsleitung über einen Tag im 5 min - Mittel (Stand: 31.11.2003)



**Abbildung 11.4:** Durchsatz derselben 155 Mb/s - G-WiN Zugangsleitung wie in Abb. 11.3 über einen Monat im 15 min - und 2 h - Mittel (Stand: 31.11.2003)

#### 11.1.4 Verkehrszusammensetzung

Über die Verkehrszusammensetzung konnten keine aussagekräftigen Daten gesammelt werden. Die vom LRZ (Leibniz Rechenzentrum) zur Verfügung gestellten Daten (prozentuale Portbelegungen des Gesamtverkehrs) waren auf Grund der Granularität unzureichend für die erforderlichen, verkehrsspezifischen Implementierungen in die entwickelten Extrapolationsverfahren.

## 11.2 Ausgangsszenario- und Zielszenarienspezifikation für das G-WiN

Auf Grund der zuvor durchgeführten Untersuchungen lassen sich sowohl das Ausgangsszenario, als auch die Zielszenarien spezifizieren:

Ziel ist es, die Ende-zu-Ende Delay-, Jitter- und Losswerte der Pakete einer Videokonferenz bei der

Übertragung im G-WiN am effizientesten für die momentanen Topologie- und Lastverhältnisse zu optimieren. Aus der zuvor beschriebenen Lastsituation ergibt sich als favorisierter Technologieansatz eine Kombination aus DiffServ auf den Zugangsleitungen und Overprovisioning im Kernnetz. Für Overprovisioning im Kernnetz spricht die niedrige Auslastung der Leitungen, die durch die Erweiterungs- bzw. Leitungsaufstockungsverträge mit der Telekom auch bei steigendem Verkehrsaufkommen zumindest noch bis Mai 2004 gesichert war. Die Kernnetzrouter werden durch die Overprovisioning Technologie nicht zusätzlich belastet.

Für den Einsatz von DiffServ auf den Zugangsleitungen spricht, daß erstens die Router schon DiffServ-fähig sind und die Zugangsleitungen für Realzeitanwendungen auf Grund der möglichen Extrembelastung eine QoS Technologie benötigen, um extremen Delay, Jitter und Loss zu vermeiden, und daß zweitens DiffServ nicht Ende-zu-Ende verwendet werden muß. Außerdem belastet DiffServ die Edge-Router des Kernnetzes nur minimal mit Zusatzarbeit (lediglich wie Core-Router einer DiffServ Domäne), da die Kernnetz Edge-Router die Pakete des ausgehenden Verkehrs auf Grund des DSCPs bzw. der Zuordnung des DSCPs zu einer PHB weiterleiten. Die Kernnetz Core-Router tunneln nur die Pakete bzw. leiten sie ohne Rücksichtnahme auf den DSCP unter Verwendung des BE Dienstes weiter. Bei niedrigem Verkehrsaufkommen von QoS-sensitiven Anwendungen kommt man, wie die Untersuchungen in Kapitel 10 zeigen, auch ohne Admission Control oder einen Bandwidth Broker aus. Bei höherem Verkehrsaufkommen sind diese Verfahren bei burstartigen Anwendungen (z.B. Video) unumgänglich. Probleme können hierbei jedoch, z.B. bei der Realisierung eines Bandwidth Brokers, bei der Aushandlung der SLAs der einzelnen Zugangsleitungen auftreten, die entweder durch globale Vorschriften oder durch Signalisierung geregelt werden müssen, um eine reibungslose Verwendung eines DiffServ Ansatzes zu ermöglichen. Die Probleme bestehen in der Aushandlung allgemein, sowie in der Problematik extrem unterschiedlicher Zugangskapazitäten.

Der momentan vielversprechendste Ansatz ist ein einfacher DiffServ Ansatz mit anfangs zwei bis höchstens drei Klassen (z.B. Olympic Ansatz), deren PHBs für alle QoS Technologieteilnehmer auf den Zugangsleitungen bindend sind.

IntServ ist wegen der enormen Ausdehnung des G-WiN und auch des Implementierungs- und Signalisierungsaufwandes auch in Verbindung mit der Ende-zu-Ende Verbindungsrealisierung nicht zu empfehlen. Der Einsatz von Overprovisioning auch auf den Zugangsleitungen kann für einzelne Links eine kurzfristige Lösung sein. Jedoch wäre bei einem Netz dieser Ausdehnung und bei mehreren zu erweiternden Zugangsleitungen eine einmal eingeführte QoS Technologie sinnvoller, um ein ständiges Nachrüsten zu vermeiden.

Da die Untersuchung des kompletten G-WiN Kernnetzes mit den gesamten Zugangsleitungen im Simulator auf Grund der extremen Leitungskapazitäten, des zu erzeugenden Verkehrsaufkommens und der speicherintensiven Simulationsauswertung nicht möglich ist, wird zur Untersuchung des G-WiN ein Beispielpfad durch das G-WiN mit maximaler Hopanzahl des Kernnetzpfades und jeweils einer Zugangsleitung an beiden Endpunkten des Kernnetzpfades verwendet (siehe Abbildung 11.5).

Es wird folgendes Ausgangsszenario, das der zuvor beschriebenen Lastsituation im G-WiN entspricht, für die weiteren Untersuchungen verwendet. Die Verkehrszusammensetzung ist nach dem Untersuchungsinteresse (hohe Auslastung auf den Zugangsleitungen, niedrige Auslastung im Kernnetz, Übertragung von QoS-sensitiven Daten) ausgewählt worden und entspricht, auf Grund der fehlenden Informationen über reale Verkehrszusammensetzungen im G-WiN, nicht einer beobachteten, sondern einer theoretisch möglichen, extremen Verkehrszusammensetzung.

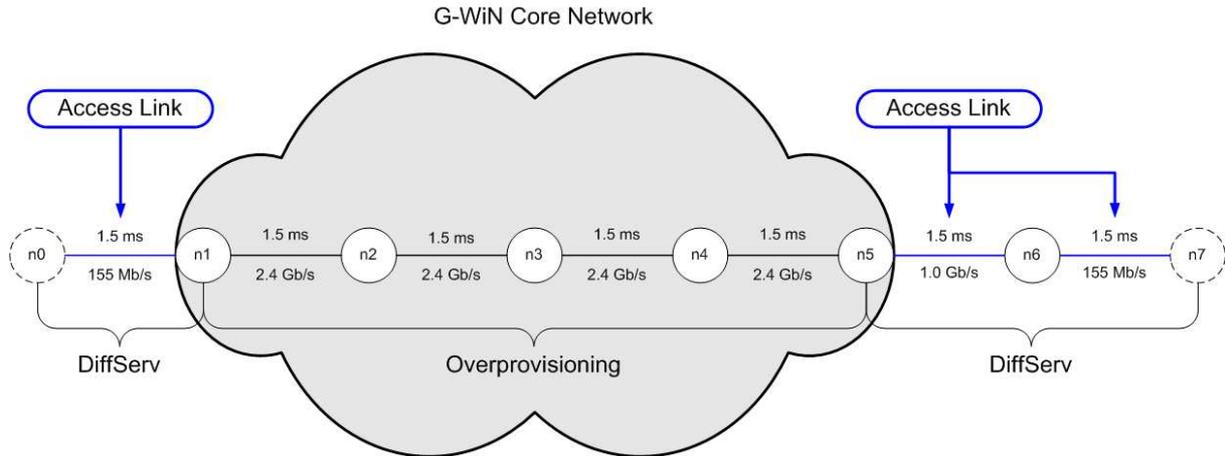


Abbildung 11.5: G-WiN Beispielpfad

Netzwerkconfiguration bestehend aus:

- der Topologie  $T = \{ID, N, E\}$ , mit  $ID = ETS$ ,  $N = \{n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7\}$  Knoten und  $E = \{e_{0,1}, e_{1,2}, e_{2,3}, e_{3,4}, e_{4,5}, e_{5,6}, e_{6,7}\}$  Kanten
- der Menge von Bandbreiten  $B = \{B_{0,1}, B_{1,2}, B_{2,3}, B_{3,4}, B_{4,5}, B_{5,6}, B_{6,7}\}$ , mit  $B_{0,1} = 155 \text{ Mb/s}$ ,  $B_{1,2} = B_{2,3} = B_{3,4} = B_{4,5} = 2.4 \text{ Gb/s}$ ,  $B_{5,6} = 1.0 \text{ Gb/s}$  und  $B_{6,7} = 34.0 \text{ Mb/s}$
- der Menge von Leitungsverzögerungen  $LV = \{LV_{0,1}, LV_{1,2}, LV_{2,3}, LV_{3,4}, LV_{4,5}, LV_{5,6}, LV_{6,7}\}$ , mit  $LV_{0,1} = LV_{1,2} = LV_{2,3} = LV_{3,4} = LV_{4,5} = LV_{5,6} = LV_{6,7} = 1.5 \text{ ms}$  (mittlerer Propagationsdelay im G-WiN)
- der Routinginformation  $R = \{\text{IP Routing}\}$

Es wird für das **Lastspektrum** ( $LS$ ) eine Mischung aus Video und WWW Verkehr gewählt:

$$LS = \{VLS_{VIDEO}, VLS_{WWW}\}$$

$$VLS_{VIDEO} = (SD_{VIDEO}, TT_{VIDEO})$$

$$SD_{VIDEO} = \{(n_0, n_7)\}$$

$$TT_{VIDEO} = \{\text{H.323 Videokonferenz, Verkehr aus Trace Files generiert}\}$$

$$VLS_{WWW} = (SD_{WWW}, TT_{WWW})$$

$$SD_{WWW} = \{(n_0, n_7)\}$$

$$TT_{WWW} = \{\text{Hintergrundverkehr, WWW Generator}\}$$

Auf Grund der höchsten Relevanz für das DFN und der schwierigsten Erlangung von akzeptabler Dienstgüte, wird beispielhaft eine Extrapolation ausgehend von den bestehenden G-WiN Verhältnissen (Topologie, Lastsituation, Best Effort, etc.) mit einer verwendeten Verkehrszusammensetzung aus Videoverkehr und TCP Hintergrundverkehr durchgeführt. Folgende Verkehrszusammensetzung wird untersucht:

**Verkehrszusammensetzung:**

20 Video Traces

49 TCP Flows von n0 nach n7

Von diesem Ausgangsszenario wird auf ein Zielszenario extrapoliert, das den favorisierten Kombiansatz aus Overprovisioning im Kernnetz und dem Olympic Ansatz auf den Zugangsleitungen verwendet. Die Ergebnisse werden mit denen von Extrapolationsszenarien verglichen, die einen warteschlangenoptimierten BE Ansatz und einem Overprovisioningansatz auch auf den Zugangsleitungen (Bottleneck Link von Knoten n6 nach n7) verwenden.

### 11.3 Prüfung der G-WiN Voraussetzungen

Angaben über die Verkehrszusammensetzung des G-WiN sind nicht vorhanden. Allerdings kann aber auf Grund des Interesses an der Untersuchung von Videokonferenzen und der Lastverhältnisse des G-WiN eine Verkehrszusammensetzung, wie zuvor beschrieben, für die Zielszenarien angegeben werden. Die Linkkapazitäten des G-WiN sind extrem groß, daher kann nicht das gesamte G-WiN simuliert werden. Es wird auf eine repräsentative Pfaduntersuchung übergegangen. Meßdaten des Ausgangsszenarios werden durch Simulationen gewonnen.

Es existieren keine Einschränkungen bezüglich der Nutzung eines Extrapolationsverfahrens. Alle drei Extrapolationsverfahren können zur Extrapolation herangezogen werden.

### 11.4 Extrapolationsdurchführung

Für jedes Zielszenario werden die möglichen Extrapolationsverfahren ermittelt und ihre aus den am ETS durchgeführten Untersuchungsergebnissen verallgemeinerten Ergebnisgüten prognostiziert. Laut den Angaben in der entwickelten Durchführungsanleitung für eine Extrapolation kann man folgende Prognosen für die Verwendbarkeit der einzelnen Extrapolationsverfahren geben:

Die Best/Worst Case Analyse dürfte auf Grund der Verkehrszusammensetzung und -intensität und der Verwendung von BE (auch im Kombiansatz) mit großen Warteschlangen für die meisten Zielszenarien unzureichende Ergebnisse liefern. Nur bei sehr hohen Übertragungskapazitäten und somit kleinem maximalem Ende-zu-Ende Delay ( $< 100$  ms) und möglicherweise gleichzeitig kleinem maximalem Jitter ( $< 50$  ms) können die Ergebnisse relevant sein. Simulationen können für alle Zielszenarien durchgeführt werden und liefern aussagekräftige Ergebnisse. Das stochastische Verfahren dürfte keine akzeptablen Ergebnisse auf Grund der auf M/G/1 Formeln basierenden Restbedienzeitformel und den bei diesen Untersuchungen nicht eingehaltenen Voraussetzungen für die verwendeten Wartezeitformeln (siehe Kapitel 9) liefern.

Zur Unterstützung dieser Vorhersagen für die Tauglichkeit der Extrapolationsverfahren werden alle drei Verfahren für die spezifizierten Extrapolationszielszenarien durchgeführt:

#### 11.4.1 Best/Worst Case Analyse

Als erstes Extrapolationsverfahren wird die Best/Worst Case Analyse auf folgende Zielszenarien angewendet, da hierdurch möglicherweise bereits Ergebnisse erzielt werden können, die den Einsatz der beiden anderen Verfahren für das G-WiN unnötig machen würde (was aber auf Grund der Vorhersage für das G-WiN nicht zu erwarten ist).

Zielszenarien:

- Ausgangsszenario unter Verwendung der angegebenen Verkehrszusammensetzung
- Ausgangsszenario unter Verwendung eines Kombiansatzes und der angegebenen Verkehrszusammensetzung:

Als **QoS Technologien** wird für die Zugangsleitungen der DiffServ Olympic Ansatz und im Kernnetz BE verwendet. Die Warteschlangenlängen werden an die extremen Leitungskapazitätsunterschiede von Kernnetz zu Zugangsnetz angepaßt und somit für die drei Klassen auf folgende Werte erhöht: EF = 100, BE = 1000, QBSS = 2000.

$$QT = \{EQTDiffServO\}$$

$$EQTDiffServO = \{NI, EM\}$$

$$NI = \{(n_0, (FIFO, (100, 1000, 2000)), PQ), (n_1, (FIFO, (100, 1000, 2000)), PQ), (n_5, (FIFO, (100, 1000, 2000)), PQ), (n_6, (FIFO, (100, 1000, 2000)), PQ), (n_7, (FIFO, (100, 1000, 2000)), PQ)\}$$

EM =(Paket, per Hop / Aggregat / deterministisch / Loss / Delay, n Klassen, komplex, Router, Edge, proaktiv, ja, gut, Netzwerk)

#### QoS Spektrum (QS):

$$QS = \{VQSVIDEO\}$$

$$VQSVIDEO = (0.1 \%, 100 \text{ ms}, 50 \text{ ms})$$

- Ausgangsszenario unter Verwendung der angegebenen Verkehrszusammensetzung mit vergrößerten Warteschlangenlängen:  
Die Warteschlangenlängen aller Warteschlangen des Pfades werden um den Faktor 16 auf 800 erhöht (in Tabelle 11.1 mit 16fach abgekürzt).
- Ausgangsszenario unter Verwendung der angegebenen Verkehrszusammensetzung mit reinem Overprovisioning:  
Es wird die Kapazität des Bottleneck Links verdoppelt und vervierfacht (in Tabelle 11.1 mit x2 und x4 abgekürzt).

Delay Typ	Delay [s]	Verkehrstyp
mittlerer Delay bei 20 Video Traces, 49 TCP Flows - Kombiansatz	0.0207	Video Trace
maximum Delay bei 20 Video Traces, 49 TCP Flows - Kombiansatz	0.0585	Video Trace
berechneter Best Case Delay - Kombiansatz	0.0106	Video Trace
berechneter Worst Case Delay - Kombiansatz	0.2429	Video Trace
mittlerer Delay bei 20 Video Traces, 49 TCP Flows - BE x2	0.0137	Video Trace
maximum Delay bei 20 Video Traces, 49 TCP Flows - BE x2	0.0208	Video Trace
berechneter Best Case Delay - BE x2	0.0106	Video Trace
berechneter Worst Case Delay - BE x2	0.1046	Video Trace
mittlerer Delay bei 20 Video Traces, 49 TCP Flows - BE x4	0.0116	Video Trace
maximum Delay bei 20 Video Traces, 49 TCP Flows - BE x4	0.0162	Video Trace
berechneter Best Case Delay - BE x4	0.0106	Video Trace
berechneter Worst Case Delay - BE x4	0.0758	Video Trace
mittlerer Delay bei 20 Video Traces, 49 TCP Flows - BE 16fach	0.1257	Video Trace
maximum Delay bei 20 Video Traces, 49 TCP Flows - BE 16fach	0.2883	Video Trace
berechneter Best Case Delay - BE 16fach	0.0106	Video Trace
berechneter Worst Case Delay - BE 16fach	0.3352	Video Trace
mittlerer Delay bei 20 Video Traces, 49 TCP Flows - BE x4, 16fach	0.0143	Video Trace
maximum Delay bei 20 Video Traces, 49 TCP Flows - BE x4, 16fach	0.0521	Video Trace
berechneter Best Case Delay - BE x4, 16fach	0.0106	Video Trace
berechneter Worst Case Delay - BE x4, 16fach	0.1409	Video Trace

**Tabelle 11.1:** G-WiN: Vergleich von berechneten und simulierten Ende-zu-Ende Delaywerten

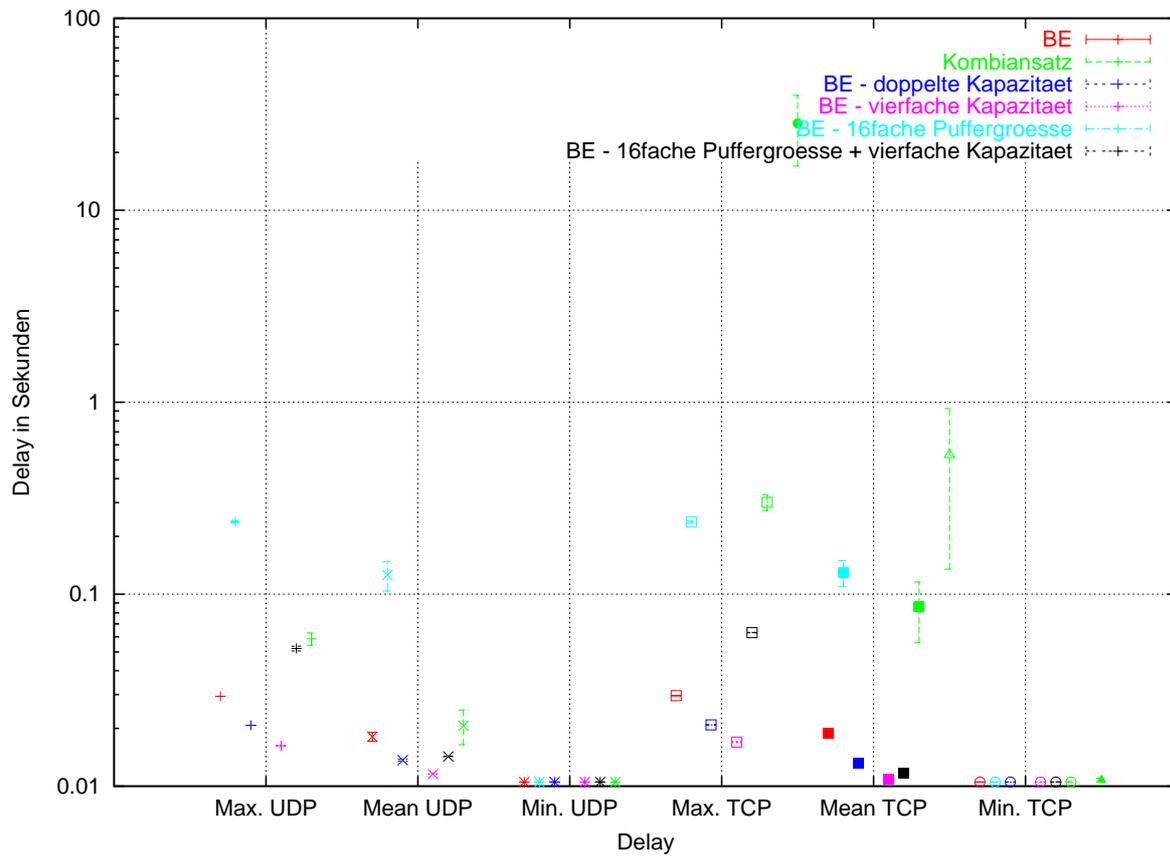
Jitter Typ	Jitter [s]	Verkehrstyp
mittlerer Jitter at 20 Video Traces, 49 TCP Flows - Kombiansatz	0.0300	Video Trace
maximum Jitter at 20 Video Traces, 49 TCP Flows - Kombiansatz	0.0480	Video Trace
berechneter Worst Case Jitterwert - Kombiansatz	0.2327	Video Trace
mittlerer Jitter at 20 Video Traces, 49 TCP Flows - BE x2	0.0089	Video Trace
maximum Jitter at 20 Video Traces, 49 TCP Flows - BE x2	0.0103	Video Trace
berechneter Worst Case Jitter - BE x2	0.0942	Video Trace
mittlerer Jitter at 20 Video Traces, 49 TCP Flows - BE x4	0.0038	Video Trace
maximum Jitter at 20 Video Traces, 49 TCP Flows - BE x4	0.0057	Video Trace
berechneter Worst Case Jitter - BE x4	0.0653	Video Trace
mittlerer Jitter at 20 Video Traces, 49 TCP Flows - BE 16fach	0.0963	Video Trace
maximum Jitter at 20 Video Traces, 49 TCP Flows - BE 16fach	0.2253	Video Trace
berechneter Worst Case Jitter - BE 16fach	0.3248	Video Trace
mittlerer Jitter at 20 Video Traces, 49 TCP Flows - BE x4, 16fach	0.0125	Video Trace
maximum Jitter at 20 Video Traces, 49 TCP Flows - BE x4, 16fach	0.0416	Video Trace
berechneter Worst Case Jitter - BE x4, 16fach	0.1305	Video Trace

**Tabelle 11.2:** G-WiN: Vergleich von berechneten und simulierten Ende-zu-Ende Jitterwerten

Die Worst Case Analyse liefert hier allgemein Ergebnisse, die sehr weit von dem bei diesen Auslastungsverhältnissen im Simulator gemessenen maximalen Ende-zu-Ende Delay abweichen (siehe Tabelle 11.1). Identisches gilt für den Jitter. Diese Ergebnisse werden durch den Widerspruch der durch sehr niedrige Auslastung im Kernnetz des G-WiN gering gefüllten Warteschlangen zu den als maximal gefüllt angesehenen Warteschlangen in der Worst Case Berechnung verursacht. Dies hat zur Folge, daß bei den untersuchten Zielszenarien lediglich unter Verwendung von BE und vierfacher Linkkapazität ein aussagekräftiges Resultat für den maximalen Ende-zu-Ende Delay erzielt werden kann. In diesem Fall kann der maximale Ende-zu-Ende Delay die kritische Grenze von 100 ms nicht überschreiten. Somit ist für dieses Zielszenario die Extrapolation mit Hilfe des analytischen Verfahrens sehr gut geeignet und ausreichend. Allerdings kann der maximale Jitter dennoch die Grenze von 50 ms überschreiten und macht damit weitere Untersuchungen durch die anderen Extrapolationsverfahren erforderlich. Die Extrapolation auf die anderen Zielszenarien ermöglicht lediglich die Aussage, daß sowohl der maximale Ende-zu-Ende Delay, als auch der maximale Jitter die angestrebten Grenzen überschreiten können. Die durch Simulation gewonnenen Mittelwerte nähern sich aber auf Grund des geringfügig ausgelasteten Kernnetzes den Best Case Werten an.

### 11.4.2 Simulationen

Es werden Simulationsexperimente für die zur Best/Worst Case Analyse identischen Zielszenarien durchgeführt:



**Abbildung 11.6:** G-WiN: Vergleich von Ende-zu-Ende Delay unter Verwendung von BE, Overprovisioning und Kombiansatz

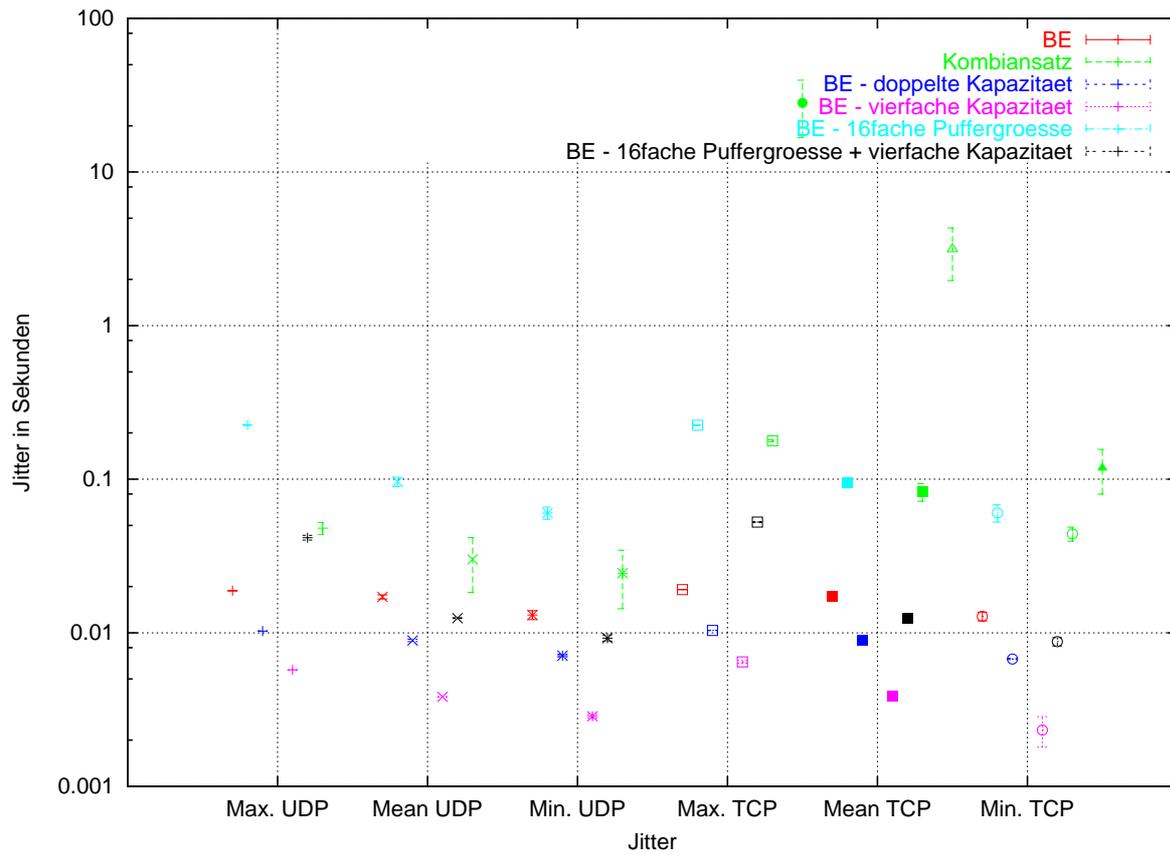


Abbildung 11.7: G-WiN: Vergleich von Jitter unter Verwendung von BE, Overprovisioning und Kombiansatz

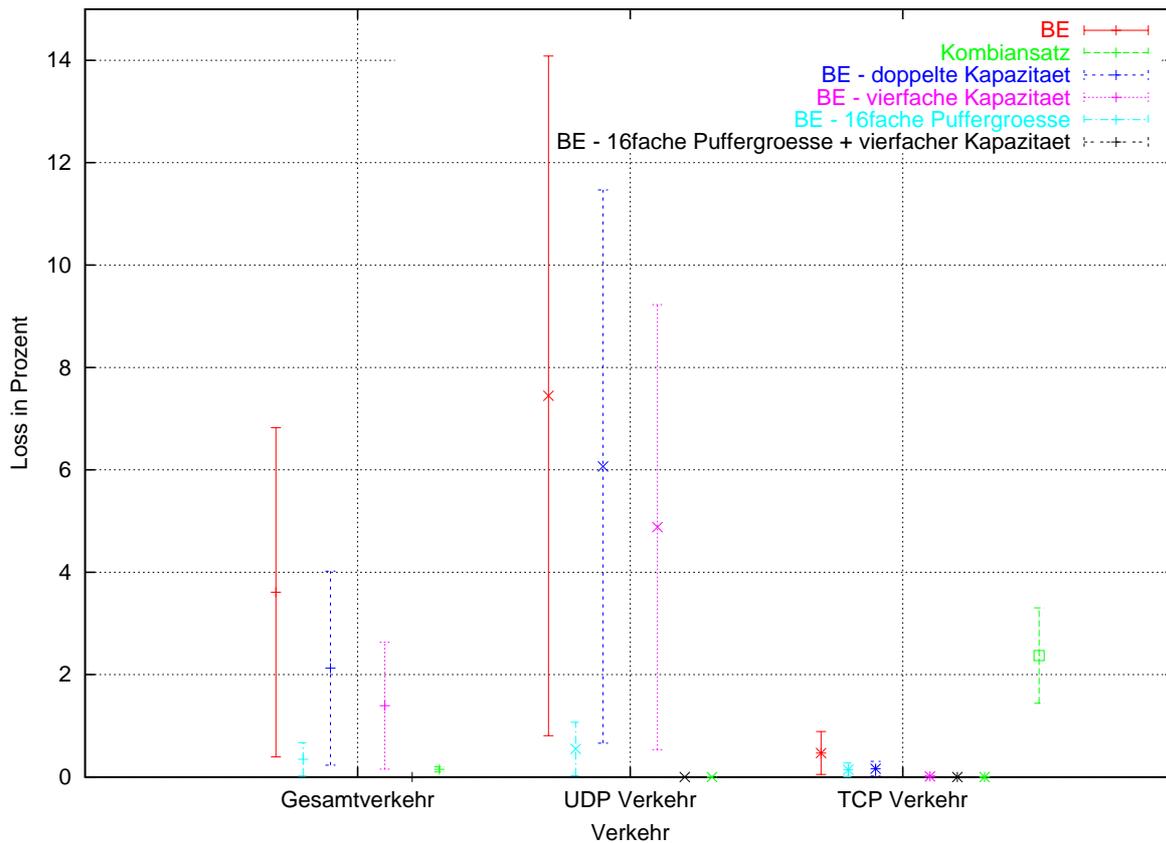


Abbildung 11.8: G-WiN: Vergleich von Loss unter Verwendung von BE, Overprovisioning und Kombiansatz

### Simulationsergebnisse:

Der Vergleich des Kombiansatzes mit dem momentan implementierten BE Dienst zeigt, daß der Kombiansatz zwar für den maximalen Ende-zu-Ende Delay und Jitter des Videoverkehrs höhere Werte annimmt, diese aber unter 100 ms für den Delay und 50 ms für Jitter bleiben. Daß bei dem Kombiansatz ein höherer Delay und Jitter für den Videoverkehr auftritt, liegt daran, daß die Warteschlangenlängen (Puffergrößen) für den Kombiansatz erhöht wurden. Dies führt dazu, daß kein Loss auftritt, der Delay aber ansteigt. Beim implementierten BE Dienst ergibt sich dagegen ein Loss von ca. 7.5 %. Für TCP Verkehr steigt beim Kombiansatz besonders für die QBSS Klasse der Delay und Jitter sehr stark an. Die Anwendungen, die diese Klassen gewählt haben, reagieren allerdings nicht sensitiv auf diese QoS Metriken. Loss tritt für die BE Klasse keiner auf, für die QBSS Klasse beträgt er ca. 2 %. Dieser kann durch eine noch extremere Warteschlangenvergrößerung auf Kosten eines noch höheren Delays und Jitters vermieden werden.

Durch eine Erhöhung der Warteschlangenlängen kann ebenfalls beim implementierten reinen BE Dienst Loss vermieden werden. Dies wird allerdings auf Kosten eines stark ansteigenden Ende-zu-Ende Delays und Jitters erreicht.

Eine Verdoppelung und Vervierfachung der Linkkapazität alleine erreicht nur eine minimale Reduktion des Loss'.

Kombiniert man allerdings beide Maßnahmen miteinander erzielt man für alle QoS Metriken für alle Verkehrsarten hervorragende Werte.

### 11.4.3 Stochastisches Verfahren

Das dritte Extrapolationsverfahren ist das stochastische Verfahren. Es wird mit Hilfe der in Kapitel 9 entwickelten Formeln vom Ausgangs- auf das Zielszenario extrapoliert. Das Ausgangsszenario ist das zuvor spezifizierte, die Zielszenarien sind wiederum zum einen das G-WiN unter Verwendung des Kombiansatzes und zum anderen ein BE Ansatz mit 16facher Pufferlänge und ein reiner Overprovisioning Ansatz mit doppelter und vierfacher Bandbreite des Bottleneck Links ( $n6 - n7$ ). Für die Extrapolation auf den Kombiansatz ergibt sich folgende Formel:

$$\begin{aligned}
E[D_Z]_{kombi} &= \sum_{i=0}^{n-1} E[D_Z]_i = \sum_{i=0}^{n-1} (E[W_Z]_i + E[B_Z]_i + D_{prop,Z,i}) \\
&= \frac{\rho_{Z,0}}{1 - \rho_{Z,0}} \cdot (\rho_{A,ef,0} \cdot \frac{1 - \rho_{A,tcp,0}}{\rho_{A,tcp,0}} \cdot (E[D_{A,tcp}]_{be,0} - D_{prop,A,0} - E[B_{A,tcp}]_0) + (\rho_{A,be,0} \\
&\quad + \rho_{A,qbss,0}) \cdot \frac{1 - \rho_{A,ef,0}}{\rho_{A,ef,0}} \cdot (E[D_{A,ef}]_{be,0} - D_{prop,A,0} - E[B_{A,ef}]_0)) + E[B_Z]_0 + D_{prop,Z,0} \\
&\quad + \sum_{i=1}^4 ((E[D_A]_{be,i} - D_{prop,A,i} - E[B_A]_i) + E[B_Z]_i + D_{prop,Z,i}) \\
&\quad + \sum_{i=5}^6 \left( \frac{\rho_{Z,i}}{1 - \rho_{Z,i}} \cdot (\rho_{A,ef,i} \cdot \frac{1 - \rho_{A,tcp,i}}{\rho_{A,tcp,i}} \cdot (E[D_{A,tcp}]_{be,i} - D_{prop,A,i} - E[B_{A,tcp}]_i) + (\rho_{A,be,i} \right. \\
&\quad \left. + \rho_{A,qbss,i}) \cdot \frac{1 - \rho_{A,ef,i}}{\rho_{A,ef,i}} \cdot (E[D_{A,ef}]_{be,i} - D_{prop,A,i} - E[B_{A,ef}]_i)) + E[B_Z]_i + D_{prop,Z,i} \right)
\end{aligned}$$

mit  $n =$  Anzahl der Netzwerkelemente (hier  $n = 7$ )

Durch Einsetzen der durch Simulation gewonnenen Ausgangsszenario-parameterwerte und der sich durch die Netzkonfiguration des Zielszenarios ergebenden Zielszenarioparameterwerte entsteht folgendes Ergebnis:

$$\begin{aligned}
&= \frac{0.2195}{1 - 0.2195} \cdot (0.0266 \cdot \frac{1 - 0.1929}{0.1929} \cdot 0.000124 \text{ s} + 0.1929 \cdot \frac{1 - 0.0266}{0.0266} \\
&\quad \cdot 0.000750 \text{ s}) + 0.000040 \text{ s} + 0.0015 \text{ s} \\
&\quad + 0.000002 \text{ s} + 0.000003 \text{ s} + 0.0015 \text{ s} \\
&\quad + 0.000002 \text{ s} + 0.000003 \text{ s} + 0.0015 \text{ s} \\
&\quad + 0.000002 \text{ s} + 0.000003 \text{ s} + 0.0015 \text{ s} \\
&\quad + \frac{0.0340}{1 - 0.0340} \cdot (0.0041 \cdot \frac{1 - 0.0299}{0.0299} \cdot 0.000012 \text{ s} + 0.0299 \cdot \frac{1 - 0.0041}{0.0041} \\
&\quad \cdot 0.000005 \text{ s}) + 0.000006 \text{ s} + 0.0015 \text{ s} \\
&\quad + \frac{0.9068}{1 - 0.9068} \cdot (0.0600 \cdot \frac{1 - 0.8468}{0.8468} \cdot 0.007993 + 0.8468 \text{ s} \cdot \frac{1 - 0.0600}{0.0600} \\
&\quad \cdot 0.06320 \text{ s}) + 0.000184 \text{ s} + 0.0015 \text{ s} \\
&= 0.8289 \text{ s}
\end{aligned}$$

Das erzielte Ergebnis wird mit den Simulationsergebnissen verglichen und die prozentuale Abweichung von diesen angegeben:

QoS Ansatz	Erzeugung	mittlerer E2E Delay [s]
Kombiansatz	berechnet	0.8289
	simuliert	0.0207
	prozentuale Abweichung	3904

**Tabelle 11.3:** Extrapolation von BE auf Kombiansatz bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechnetem und durch Simulation erzeugtem Ende-zu-Ende Delay

Die Formel zur Extrapolation auf ein Zielszenario mit doppelter Bottleneck Kapazität sieht folgendermaßen aus:

$$\begin{aligned}
E[D_Z]_{be} &= \sum_{i=0}^{n-1} (E[W_Z]_i + E[B_Z]_i + D_{prop,Z,i}) \\
&= \sum_{i=0}^{n-1} \left( \frac{\rho_{Z,i}}{1-\rho_{Z,i}} \cdot E[B_Z]_i \cdot \frac{C_{A,Z,i}^2 + C_{B,Z,i}^2}{2} + E[B_Z]_i + D_{prop,Z,i} \right) \\
&= \sum_{i=0}^{n-1} \left( \frac{\frac{\rho_{A,i}}{k-\rho_{A,i}}}{\frac{\rho_{A,i}}{1-\rho_{A,i}}} \cdot \frac{1}{k} \cdot \frac{\rho_{A,i}}{1-\rho_{A,i}} \cdot E[B_A]_i \cdot \frac{C_{A,A,i}^2 + C_{B,A,i}^2}{2} + \frac{1}{k} \cdot E[B_A]_i + D_{prop,Z,i} \right) \\
&= \sum_{i=0}^{n-1} \left( \frac{\frac{\rho_{A,i}}{k-\rho_{A,i}}}{\frac{\rho_{A,i}}{1-\rho_{A,i}}} \cdot \frac{1}{k} \cdot (E[D_A]_{be,i} - E[B_A]_i - D_{prop,A,i}) + \frac{1}{k} \cdot E[B_A]_i + D_{prop,Z,i} \right)
\end{aligned}$$

mit n = Anzahl der Netzwerkelemente (hier n = 7)

Durch Einsetzen der durch Simulation gewonnenen Ausgangsszenario-parameterwerte und der sich durch die Netzkonfiguration des Zielszenarios ergebenden Zielszenarioparameterwerte entsteht folgendes Ergebnis:

$$\begin{aligned}
&= \frac{\frac{0.2195}{2-0.2195}}{\frac{0.2195}{1-0.2195}} \cdot \frac{1}{2} \cdot 0.000750s + \frac{1}{2} \cdot 0.000040s + 0.0015s \\
&= \frac{\frac{0.0142}{2-0.0142}}{\frac{0.0142}{1-0.0142}} \cdot \frac{1}{2} \cdot 0.000002s + \frac{1}{2} \cdot 0.000003s + 0.0015s \\
&= \frac{\frac{0.0142}{2-0.0142}}{\frac{0.0142}{1-0.0142}} \cdot \frac{1}{2} \cdot 0.000002s + \frac{1}{2} \cdot 0.000003s + 0.0015s \\
&= \frac{\frac{0.0142}{2-0.0142}}{\frac{0.0142}{1-0.0142}} \cdot \frac{1}{2} \cdot 0.000002s + \frac{1}{2} \cdot 0.000003s + 0.0015s \\
&= \frac{\frac{0.0142}{2-0.0142}}{\frac{0.0142}{1-0.0142}} \cdot \frac{1}{2} \cdot 0.000002s + \frac{1}{2} \cdot 0.000003s + 0.0015s
\end{aligned}$$

$$\begin{aligned}
&= \frac{\frac{0.0340}{2-0.0340}}{\frac{0.0340}{1-0.0340}} \cdot \frac{1}{2} \cdot 0.000005s + \frac{1}{2} \cdot 0.000006s + 0.0015s \\
&= \frac{\frac{0.9068}{2-0.9068}}{\frac{0.9068}{1-0.9068}} \cdot \frac{1}{2} \cdot 0.006320s + \frac{1}{2} \cdot 0.000184s + 0.0015s \\
&= 0.0117
\end{aligned}$$

Das erzielte Ergebnis dieser Extrapolation, wie auch das auf ein Zielszenario mit vierfacher Bottleneck Kapazität wird mit den Simulationsergebnissen verglichen und die prozentuale Abweichung von diesen angegeben.

QoS Ansatz	Erzeugung	mittlerer E2E Delay [s]
verdoppelte Bottleneck Kapazität	berechnet	0.0117
	simuliert	0.0137
	prozentuale Abweichung	14.60
vervierfachte Bottleneck Kapazität	berechnet	0.0114
	simuliert	0.0116
	prozentuale Abweichung	1.72

**Tabelle 11.4:** Extrapolation von BE auf BE mit Overprovisioning bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechnetem und durch Simulation erzeugtem Ende-zu-Ende Delay

Die Ergebnisse des stochastischen Verfahrens für das G-WiN unterstützen die Ergebnisse aus dem ETS. Eine Extrapolation auf den Kombiansatz ist auf Grund der ausgewählten Verkehrszusammensetzung und auf Grund der auf M/G/1 Formeln basierenden Restbedienzeitformel des Kombiansatzes nicht exakt möglich. Hingegen können Extrapolationen auf ein Zielszenario mit erhöhter Bottleneck Kapazität relativ exakt durchgeführt werden. Die Genauigkeit der Extrapolation steigt dabei mit zunehmender Erhöhung der Bottleneck Kapazität. Untersuchungen für Zielszenarien mit 16facher Puffergröße können mit diesem Verfahren nicht durchgeführt werden.

#### 11.4.4 Zusammenfassung und Empfehlung

Für das G-WiN ist, wie zuvor begründet und bereits in der Vorhersage für die Tauglichkeit der Extrapolationsverfahren mit Hilfe der Durchführungsanleitung angegeben, lediglich das simulative Extrapolationsverfahren anwendbar. Der empfehlenswerte QoS Technologieansatz ist der Kombiansatz aus DiffServ auf den Zugangsleitungen und Overprovisioning im Kernnetz. Eine Gewährleistung der Dienstgüte ist allerdings auch durch einen kompletten Overprovisioning Ansatz in Kombination mit einer Warteschlangenlängenoptimierung möglich. Dies kann für einzelne hoch ausgelastete Leitungen der optimale Ansatz sein. Doch für ein Netz mit einer Vielzahl solcher Leitungen ist der Kombiansatz zu bevorzugen, da hiermit auch ein ständiges Nachrüsten unterbunden werden kann.

## Kapitel 12

# Zusammenfassung und Ergebnisse

In dieser Arbeit wurde ein neues Verfahren zur Vorhersage von Dienstgüteaussagen in Zielnetzen basierend auf Messungen in ähnlichen Ausgangsnetzen entwickelt, das als Extrapolation bezeichnet wird. Es wurden drei Freiheitsgrade Netzkonfiguration, Lastspektrum und QoS Technologie definiert, aus denen sich 8 Extrapolationsfälle ergeben durch die sich das Zielnetz vom Ausgangsnetz unterscheiden kann. Die Extrapolation auf die unterschiedlichen Zielnetze wird durch drei Verfahren realisiert: Simulation, Best/Worst Case Analyse und ein stochastisches Verfahren. Die Simulationen wurden mit Hilfe des NS2 Simulators durchgeführt und durch ein innerhalb des LETS QoS Projektes aufgebautes Testbett validiert. Die beiden anderen Verfahren wurden rechnerisch durchgeführt. Die drei Extrapolationsverfahren wurden im Rahmen dieser Arbeit entwickelt und die Verwendbarkeit und Genauigkeit anhand eines Extrapolations-Testszenarios untersucht. Dabei wurden zusätzlich die Verwendbarkeit und Eigenschaften dreier ausgewählter QoS Technologien (BE (Overprovisioning), IntServ und DiffServ (Olympic und Default)) getestet. Die Untersuchungen zeigten erneut, wie bereits in anderen Projekten, daß QoS Technologien unbedingt erforderlich sind, um QoS-sensitiven Verkehr entsprechend seinen Anforderungen trotz hoher Auslastung zu übertragen.

Der reine Best Effort Dienst liefert für sämtliche QoS Metriken ab einer gewissen unteren Auslastung (von ca. 30 % für alle untersuchten Verkehrszusammensetzungen) des betrachteten Übertragungspfades (Teilpfades) unzureichende Werte. Dies kann entweder durch Overprovisioning, DiffServ oder IntServ verbessert werden.

Overprovisioning ist für den Einsatz auf einzelnen hoch ausgelasteten Links innerhalb eines niedrig ausgelasteten Netzwerks (sollte je nach Verkehrszusammensetzung innerhalb des gesamten Netzwerks zu Auslastungen unter 10 % führen) oder für eine kurzfristige Lösung prädestiniert. Um ein ständiges Aufrüsten zu vermeiden, sollte allerdings (in den in dieser Arbeit untersuchten Netzen) eine der beiden anderen QoS Technologien eingesetzt werden. Dies hängt allerdings stets von Netzwerk Charakteristiken, wie Topologie, Auslastung und Verkehrszusammensetzung und Kostenfaktoren für die QoS-Technologie-Implementierung oder die Netzkapazitätserweiterung ab und kann nicht pauschalisiert werden.

Für IntServ spricht die absolute Garantie für den maximalen Ende-zu-Ende Delay und Loss, die durch Best Case Ende-zu-Ende Delay Berechnungen ermöglichte Worst Case Jitter Berechnung und die höher bzw. flexibler einstellbare Akzeptanz des hochprioren Verkehrs (Reservierungen) durch die Admission Control. Der hohe Implementierungs-, Signalisierungs- und somit zusätzliche Aufwand für die Router, sowie die Ende-zu-Ende Implementierungsanforderung sind Argumente gegen die Verwendung von IntServ.

DiffServ hingegen erreicht ebenfalls in allen Bereichen sehr gute Werte für die QoS Metriken. Die Anzahl an hochprioren Verkehrsflows ist allerdings auf Grund der maximalen, akzeptablen, den Delay begrenzenden Warteschlangenlänge durch die Admission Control begrenzt. Dafür kann DiffServ durch unterschiedlichste definierbare Ansätze auf die in dem betrachteten Netz auftretenden Verkehrstypen und -anforderungen angepaßt werden und muß nicht Ende-zu-Ende implementiert sein.

Die drei entwickelten Extrapolationsverfahren verhielten sich innerhalb des Extrapolations-Testszenarios wie folgt:

Durch Simulationen konnte die Extrapolation für alle untersuchten Fälle durchgeführt werden. Es ist allerdings der Ausdehnung des untersuchbaren Netzes durch Rechnerleistung und Speicherkapazität des Simulationsrechners eine Grenze gesetzt, die vor allem durch die rechen- und speicherintensive Ergebnisgenerierung hervorgerufen wird. Die Best und Worst Case Analyse kann nur begrenzt aussagekräftige Ergebnisse liefern. Generell überschätzt die Worst Case Analyse die simulierten Werte für maximale Delays, da alle Warteschlangen als maximal gefüllt in die Berechnung mit eingehen und dies in den meisten Netzen nicht der Fall ist. Sie kann allerdings bei bestimmten Netz- und Verkehrscharakteristiken bereits dazu dienen nachzuweisen, daß der maximale Delay eine geforderte maximale Grenze nicht überschreitet und somit weitere aufwendigere Verfahren überflüssig machen. Daher sollte dieses Extrapolationsverfahren immer als erstes für einen eventuellen Einsatz geprüft werden. Die Best Case Analyse kann mit Hilfe des bei IntServ vorgegebenen maximalen Delays eine Abschätzung des maximalen Jitters ermöglichen und bei einem Vergleich mit dem gemessenen Delay von Paketen über den berechneten Pfad eine Aussage über die Auslastung dieses Teilabschnitts eines Netzes ermöglichen. Die Güte des stochastischen Verfahrens hängt sehr stark von der Verkehrscharakteristik ab, obwohl durch Validierung an einem Single Queue Simulator eine akzeptable Wartezeitformel gefunden werden konnte, da sich der Variationskoeffizient des Ankunftsprozesses im Zielszenario gegenüber dem des Ausgangsszenarios bei bestimmten Extrapolationen ändern kann und die unterschiedlichen DiffServ Restbedienzeitformeln auf M/G/1 Formeln basieren. Außerdem ist für den Einsatz der ausgewählten Wartezeitformeln ein Ein-Stationen-Wartesystem und eine endlose Warteschlange vorausgesetzt.

Durch eine Verallgemeinerung der erzielten Ergebnisse innerhalb des Extrapolations-Testszenarios konnte eine allgemeine Vorgehensweise für Extrapolationen zur Verfügung gestellt werden. Die Anwendung dieser Vorgehensweise auf das G-WiN und die Analyse des G-WiN ergaben eine Empfehlung eines zu verwendenden Kombinationsansatzes aus Overprovisioning im Kernnetz und einem einfachen DiffServ Ansatz (wie Olympic) auf den Zugangsleitungen. Von den drei entwickelten Verfahren lieferte lediglich die Simulation gute Ergebnisse. Die Best/Worst Case Analyse scheitert an der Lastsituation im G-WiN und das stochastische Verfahren an der untersuchten Verkehrszusammensetzung.

## Kapitel 13

# Offene Probleme und Weiterentwicklungsmöglichkeiten

Rückblickend auf diese Arbeit und das LetsQoS Projekt läßt sich sagen, daß einige allgemeine Probleme auftraten. Zunächst ist die Komplexität einer Implementation eigener Konzepte bzw. Ideen in NS2, bedingt durch den Aufbau des Simulators, nicht zu unterschätzen. Ein späterer Startpunkt des Projektes hätte die Entscheidung für einen anderen, strukturell übersichtlicheren Simulator auf Grund der fortschreitenden Entwicklung von NS2 Konkurrenzsimulatoren wahrscheinlich gemacht.

Außerdem wurde beim LetsQoS Projektpartner in Darmstadt die Komplexität eines Testbettauflbaus und der entsprechenden Konfigurationen zur Untersuchung der QoS Technologien unterschätzt, so daß die Validierung der Extrapolation nicht im vollen Umfang durchgeführt werden konnte.

Desweiteren erwies sich die Gewinnung der G-WiN Daten als sehr schwieriges Problem.

Im Speziellen liegen die größten Schwierigkeiten bei der Durchführung einer Extrapolation an der Unkenntnis des Variationskoeffizienten des Ankunftsprozesses und der Nichtunabhängigkeit von Ankunfts- und Bedienprozess (in sich und untereinander) auf einem Übertragungspfad und der somit mangelnden Verfügbarkeit von geeigneten GI/G/1 Formeln zur Wartezeitbestimmung bei extremen Verkehrstypen und dem Einsatz von QoS Technologien. Hier sollte die weitere Entwicklung verfolgt werden.

Die QoS Metriken wurden nur im Hinblick auf hochprioren UDP Verkehr gemessen. Sollte eine hochpriore Anwendung TCP verwenden, müßten die Retransmissionen in den Untersuchungen berücksichtigt werden.

Zusätzlich war die Untersuchung des G-WiN ein Ressourcen (Speicherkapazität, Rechenleistung) Problem, das in späteren Zeiten auf Grund der technischen Entwicklung der Speicherkapazitäten und Prozessorgeschwindigkeiten bestimmt zufriedenstellender gelöst werden kann.

Die erzielten G-WiN Ergebnisse sollten, wenn möglich, in Realität geprüft und mit anderen vergleichbaren Ergebnissen eines anderen Netzes ähnlicher Charakteristik (GÉANT Netz) verglichen werden. Der Einsatz von QoS Technologien ist bei weiter steigendem Datenvolumen ein guter Ansatz um QoS-sensitiven Verkehr entsprechend seinen Anforderungen zu übertragen. Allerdings sollte auch die Entwicklung der optischen Netze hin zu rein optischen Netzen (All Optical Networks, AON) und weiterführend zu automatisch geschalteten optischen Netzen (Automatically Switched Optical Networks, ASON) beobachtet werden, da die Entwicklung letzterer den Einsatz einer IP-basierten QoS Technologie eines Netzes in Zukunft überflüssig machen könnte.

## Anhang A

# Definitionen und Bezeichnungen

### A.1 Definitionen

Die wichtigsten Definitionen sind im Anschluß zusammengefaßt. Sie basieren auf [BLAKE ET AL. 98, WROCLAWSKI 97, BRADEN ET AL. 94, JESSEN 01]

#### **Anwendung**

Unter Anwendung versteht man hier ein Anwendungsprogramm. Ein Anwendungsprogramm ist ein Programm, das dazu entwickelt wurde, eine bestimmte Funktion direkt einem Anwender zur Verfügung zu stellen.

#### **Auftrag**

Ein Auftrag ist eine Verpflichtung einer Funktionseinheit zu einer Handlung.

#### **Auslastung**

Die Auslastung ist das Verhältnis von Durchsatz zu Grenzdurchsatz. Bei exklusiven Funktionseinheiten ist die Auslastung zugleich der Zeitanteil, in dem die Funktionseinheit belegt ist.

#### **Bandbreite (entspricht hier Kapazität)**

In der Nachrichtentechnik ist die Bandbreite des Signals die Breite der Frequenzspanne, die ein Signal auf einem gegebenen Kanal besetzt. Entsprechend wird ein Medium durch seine nutzbare Bandbreite charakterisiert. Im Bezug auf digitale Netze bedeutet sie die Anzahl von Daten, die maximal über eine Kommunikationsverbindung in einer bestimmten Zeitspanne übertragen werden kann, geteilt durch die Länge der Zeitspanne (gemessen in Zahl der Dateneinheiten/Zeiteinheit). Die Bandbreite einer digitalen Kommunikationsverbindung bildet das Maximum für den Durchsatz derselben, daher spricht man auch vom Grenzdurchsatz.

#### **Bedienzeit**

Die Bedienzeit ist die Zeit, in der ein Auftrag in einer Funktionseinheit bearbeitet wird, bei Bestand eins.

#### **Bestand (Füllung)**

Der Bestand ist die Zahl der Aufträge, die aktuell ihre Verweilzeit in der betrachteten Funktionseinheit zubringen.

**Classifier**

Ein Classifier ist eine Funktionseinheit, die Pakete basierend auf Paketheaderinformationen nach definierten Regeln auswählt.

**Delay**

Unter dem Ende-zu-Ende Delay versteht man die Zeit, die Pakete vom vollständigen Einreihen in die Ausgangswarteschlange des Senders bis zum vollständigen Einreihen in die Eingangswarteschlange des Empfängers benötigen. Der Router-Delay ist die Zeitspanne zwischen dem Paketein- und -abgang innerhalb eines einzelnen Routers, bestehend aus Berechnungs- und Pufferzeiten. Delay ist im allgemeinen Fall ein Synonym zu Verweilzeit (beides gemessen in Zeiteinheiten).

**Dienst**

Ein Dienst (Service) ist das Angebot Aufträge auszuführen. Hierbei liefert jeder Dienst eine gewisse Dienstgüte (siehe Dienstgüte Definition).

**Dienstgüte**

Die Güte eines Dienstes wird hier durch die Angaben der Dienstgütemetriken Delay, Jitter und Loss für die mit Hilfe dieses Dienstes übertragenen Pakete charakterisiert.

**Domäne**

Allgemein versteht man unter einer Domäne einen Steuerungs-, Überwachungs- oder Wissensbereich.

**Dropper**

Ein Dropper ist eine Funktionseinheit, die Pakete wegwirft.

**Drop Precedence**

Unter Drop Precedence versteht man eine Angabe einer Priorität, die eine Aussage über die Reihenfolge des Wegwerfens von Paketen macht.

**Durchsatz**

Der Durchsatz ist die Anzahl der Aufträge (Dateneinheiten, Pakete, etc.), die von einer Funktionseinheit in einer vorgegebenen Zeitspanne bearbeitet bzw. übertragen wird, geteilt durch die Zeitspanne (gemessen in Aufträgen bzw. Dateneinheiten/Zeiteinheit).

**Edge-Router**

Ein Edge-Router ist ein Router über den Verkehr in eine Domäne einströmen kann, also ein Zugangspunkt zu einer Domäne.

**Funktionseinheit**

Nach Aufgabe oder Wirkung abgrenzbares Gebilde.

**Grenzdurchsatz**

Der Grenzdurchsatz stellt den maximalen Durchsatz (entspricht Kapazität) einer Funktionseinheit (z.B.: Link) dar.

**Jitter**

Allgemein versteht man unter Jitter die Schwankung des Delays in einer Paketfolge. In dieser Arbeit ist Jitter die maximale Ende-zu-Ende Delayvariation (Differenz des minimalen und maximalen Ende-zu-Ende Delays) in einem 3 Sekunden Intervall (gemessen in Zeiteinheiten). Die Intervalldauer wurde im Hinblick auf die durchschnittliche Dauer eines Sprachbursts gewählt.

**Kanal**

Unter einem Kanal versteht man den einem Strom zugewiesenen Teil eines Übertragungsmediums.

**Kapazität**

Siehe Bandbreite.

**Klasse**

Eine Klasse ist eine Aggregation von Strömen, die dieselbe Behandlung während ihrer Übertragung erfahren sollen (siehe Kapitel 2).

**Knoten**

Ein Knoten ist in dieser Arbeit ein Host oder ein Router.

**Loss**

Loss gibt die prozentuale Zahl der verlorengegangenen Pakete relativ zur Anzahl der empfangenen Pakete an. Verlust von Paketen entsteht durch Pufferüberläufe auf dem Paketübertragungsweg vom Sender zum Empfänger, durch das Wegwerfen von Paketen, durch eine übermäßige Verspätung oder Verfälschung, die die Verwendbarkeit der Pakete für den Empfänger unmöglich machen oder durch fehladressierte Pakete. Pakete die am Domäneneingang auf Grund von Policing weggeworfen werden, fließen nicht mit in die Lossberechnung ein.

**Marker**

Ein Marker ist eine Funktionseinheit, die Pakete markiert.

**Meter**

Ein Meter ist eine Funktionseinheit, die Messungen durchführt.

**QoS Technologie (QT)**

Unter einer QoS Technologie versteht man eine Technologie, die verwendet wird, um eine bestimmte Güte eines Dienstes zu realisieren. In dieser Arbeit werden Overprovisioning, Differentiated Services und Integrated Services (siehe Kapitel 2) als QoS Technologien eingesetzt.

**Per-Hop Behaviour**

Beschreibung des Forwarding Behaviour eines Netzwerkknotens.

**Policy**

Eine Policy ist eine Menge von formellen Aussagen, die definieren, wie die Netzwerkressourcen unter den Kunden (z.B.: Strömen) aufgeteilt werden.

**Rate**

Eine Rate ist eine Anzahl von Ereignissen pro Zeiteinheit, z.B. Auftragsfertigstellung (= Durchsatz).

**Sensitivität**

Unter Sensitivität versteht man hier die Stärke des Einflusses, den ein Wert einer Dienstgütemetrik auf eine Auftrags erledigung ausübt.

**Shaper**

Ein Shaper ist eine Funktionseinheit, die einen Strom formt.

**Strom**

Unter einem Strom versteht man eine identifizierbare Sequenz zusammengehörender Pakete einer einzelnen Benutzeraktivität, deren Pakete dieselbe Dienstgüte erfordern.

**Verweilzeit**

Die Verweilzeit eines Auftrags in der ihn ausführenden Funktionseinheit ist die Zeitdauer von Auftragsübergabe/ -übernahme bis zum Ende der durch den Auftrag verlangten Handlung.

**Wartezeit (Warteschlangenverzögerung)**

Die Wartezeit ist die Zeit, die ein Auftrag in einer Warteschlange auf Grund von konkurrierenden Aufträgen warten muß, ohne bedient zu werden.

## A.2 Variablenbezeichnungen

### Extrapolations-Szenario Parameter

Abschnitt 8

<i>ES</i>	Extrapolations-Szenario
<i>ETS</i>	Extrapolations-Testszenario
<i>NK</i>	Netzwerkkonfiguration
<i>T</i>	Topologie
<i>B</i>	Menge von Bandbreiten
<i>LV</i>	Menge von Leitungsverzögerungen
<i>R</i>	Routinginformationen
<i>ID</i>	Identifizierer der Topologie
<i>N</i>	Menge von Knoten
<i>n</i>	Knoten
<i>E</i>	Menge von Kanten
<i>e</i>	Kante
<i>LS</i>	Lastspektrum
<i>VLS<sub>l</sub></i>	verkehrsspezifisches Lastspektrum des Verkehrstyps <i>l</i>
<i>Q</i>	Verkehrsquelle
<i>S</i>	Verkehrssenke
<i>SD<sub>l</sub></i>	Menge von Verkehrsquellenknoten und -senkenknoten des Verkehrstyps <i>l</i>
<i>TT<sub>l</sub></i>	Beschreibung des Verkehrstyps <i>l</i>
<i>QT</i>	QoS Technologien
<i>EQT<sub>t</sub></i>	einzelne QoS Technologie <i>t</i>
<i>KI</i>	Knoteninformationen
<i>BM</i>	Bewertungsmetriken
<i>WM</i>	Warteschlangenmanagement
<i>ST</i>	Schedulertyp
<i>QS</i>	QoS Spektrum
<i>VQS<sub>l</sub></i>	verkehrsspezifisches QoS Spektrum vom Verkehrstyp <i>l</i>

### Ende-zu-Ende Delay Parameter

Abschnitt 9

<i>r</i>	Token Generierungsrate
<i>b</i>	Token Bucket Größe
<i>p</i>	Peak Rate
<i>M</i>	maximale Paketgröße
<i>C<sub>i</sub></i>	ratenabhängiger Fehlerterm des Netzwerkelements <i>i</i>
<i>E<sub>i</sub></i>	ratenunabhängiger Fehlerterm des Netzwerkelements <i>i</i>
<i>R</i>	reservierte Rate
<i>MTU</i>	Maximum Transmission Unit
<i>MTU<sub>EF</sub></i>	Maximum Transmission Unit der EF Klasse
<i>MTU<sub>AF1,i</sub></i>	Maximum Transmission Unit der AF1 Klasse
<i>MTU<sub>AF2,i</sub></i>	Maximum Transmission Unit der AF2 Klasse
<i>MTU<sub>AF3,i</sub></i>	Maximum Transmission Unit der AF3 Klasse
<i>AF1</i>	Anteile (Gewicht) der AF1 Klasse
<i>AF2</i>	Anteile (Gewicht) der AF2 Klasse
<i>AF3</i>	Anteile (Gewicht) der AF3 Klasse
<i>D<sub>prop</sub></i>	Propagationdelay

$B_r$	Restbedienzeit
$\gamma$	Link Kapazität

### Parameter der Wartezeitformeln

$E(W)$	mittlere Wartezeit
$\rho$	Auslastung
$E(B)$	mittlere Bedienzeit
$C_A^2$	Variationskoeffizient des Ankunftsprozesses
$C_B^2$	Variationskoeffizient des Bedienprozesses
$E[D_Z]_{be}$	Delay im Zielszenario unter Verwendung von BE
$E[W_Z]$	mittlere Wartezeit im Zielszenario
$E[B_Z]$	mittlere Bedienzeit im Zielszenario
$D_{prop,Z}$	Propagationdelay im Zielszenario
$\rho_Z$	Auslastung des betrachteten Links im Zielszenario
$C_{B,Z}^2$	Variationskoeffizient des Bedienprozesses im Zielszenario
$C_{A,Z}^2$	Variationskoeffizient des Ankunftsprozesses im Zielszenario
$E[RBZZ]_{be}$	mittlere Restbedienzeit im Zielszenario unter der Verwendung von BE
$E[B_A]$	mittlere Bedienzeit im Ausgangsszenario
$C_{B,A}^2$	Variationskoeffizient der Bedienzeit im Ausgangsszenario
$C_{A,A}^2$	Variationskoeffizient der Zwischenankunftszeit im Ausgangsszenario
$\rho_A$	Auslastung des betrachteten Links im Ausgangsszenario
$E[D_A]_{be}$	Delay im Ausgangsszenario unter der Verwendung von BE
$D_{prop,A}$	Propagationdelay im Ausgangsszenario
$E[B_A]$	mittlere Bedienzeit im Ausgangsszenario
$E[RBZZ]_{olympic}$	mittlere Restbedienzeit im Zielszenario unter der Verwendung des Olympic Ansatzes
$\rho_{A,ef}$	Auslastung des betrachteten Links im Ausgangsszenario verursacht durch den Verkehr der EF Klasse
$\rho_{A,tcp}$	Auslastung des betrachteten Links im Ausgangsszenario verursacht durch den TCP Verkehr der Klassen BE und QBSS
$E[D_{A,tcp}]_{be}$	Delay des TCP Verkehrs der Klassen BE und QBSS im Ausgangsszenario unter der Verwendung von BE
$E[B_{A,tcp}]$	mittlere Bedienzeit des TCP Verkehrs der Klassen BE und QBSS im Ausgangsszenario unter der Verwendung von BE
$\rho_{A,be}$	Auslastung des betrachteten Links im Ausgangsszenario verursacht durch den Verkehr der Klasse BE
$\rho_{A,qbss}$	Auslastung des betrachteten Links im Ausgangsszenario verursacht durch den Verkehr der Klasse QBSS
$E[D_{A,ef}]_{be}$	Delay des Verkehrs der EF Klasse im Ausgangsszenario unter der Verwendung von BE
$E[B_{A,ef}]$	mittlere Bedienzeit des Verkehrs der EF Klasse im Ausgangsszenario
$E[RBZZ]_{default}$	mittlere Restbedienzeit im Zielszenario unter der Verwendung des Default Ansatzes
$w_{ef}$	Gewicht der EF Klasse im WRR Scheduling Algorithmus
$\rho_{A,af1}$	Auslastung des betrachteten Links im Ausgangsszenario verursacht durch den Verkehr der AF1 Klasse

---

$w_{af1}$	Gewicht der AF1 Klasse im WRR Scheduling Algorithmus
$\rho_{A,af2}$	Auslastung des betrachteten Links im Ausgangsszenario verursacht durch den Verkehr der AF2 Klasse
$w_{af2}$	Gewicht der AF2 Klasse im WRR Scheduling Algorithmus
$\rho_{A,af3}$	Auslastung des betrachteten Links im Ausgangsszenario verursacht durch den Verkehr der AF3 Klasse
$w_{af3}$	Gewicht der AF3 Klasse im WRR Scheduling Algorithmus

## Anhang B

# Singlequeue Simulator Ergebnisse

### B.1 Ergebnisse der Heavy Traffic Formel

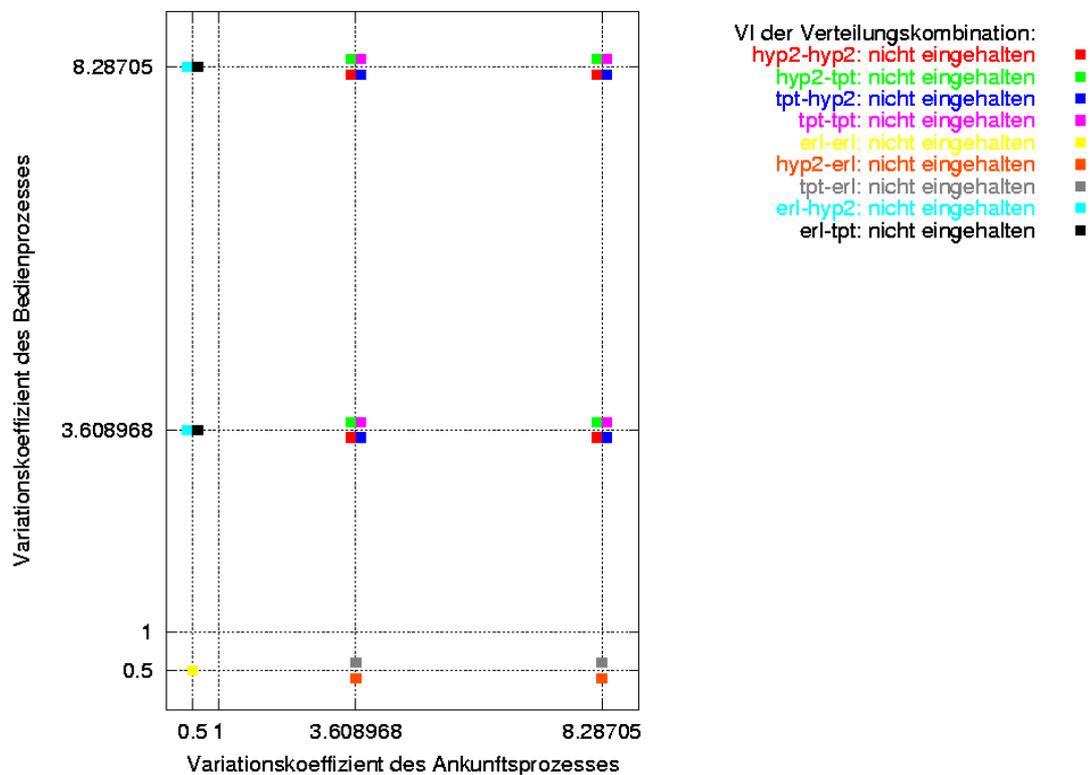


Abbildung B.1: Auslastung 20 %: Heavy Traffic Formel

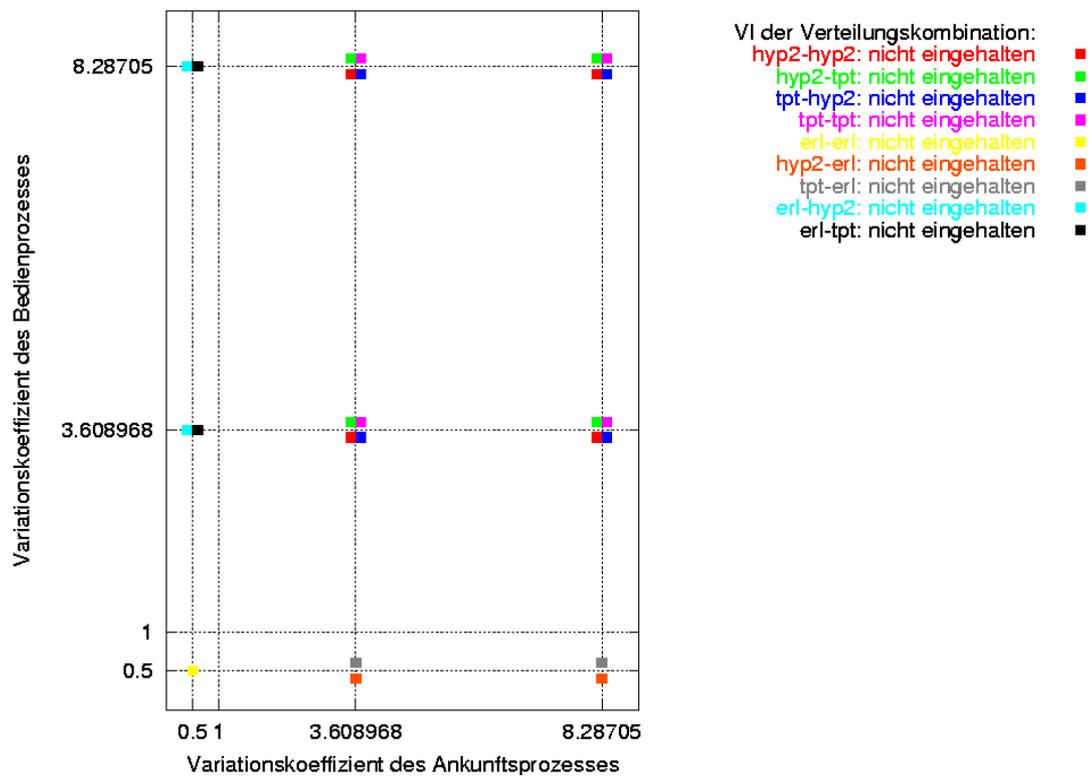


Abbildung B.2: Auslastung 50 %: Heavy Traffic Formel

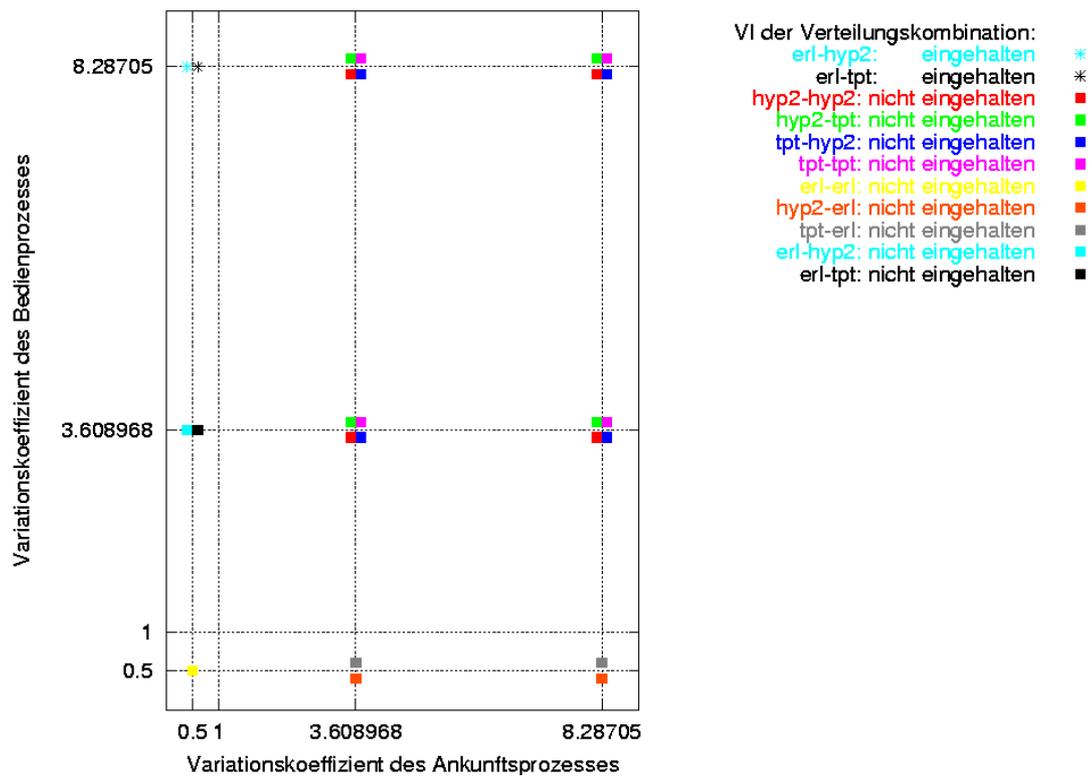


Abbildung B.3: Auslastung 80 %: Heavy Traffic Formel

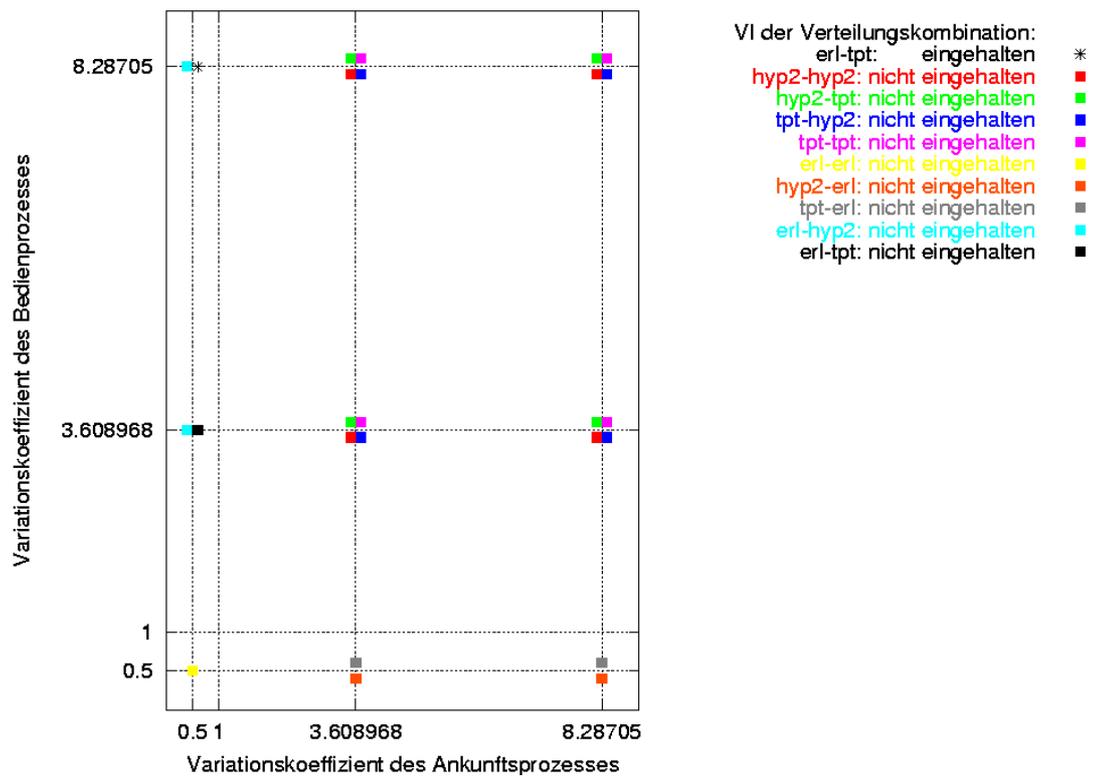


Abbildung B.4: Auslastung 90 %: Heavy Traffic Formel

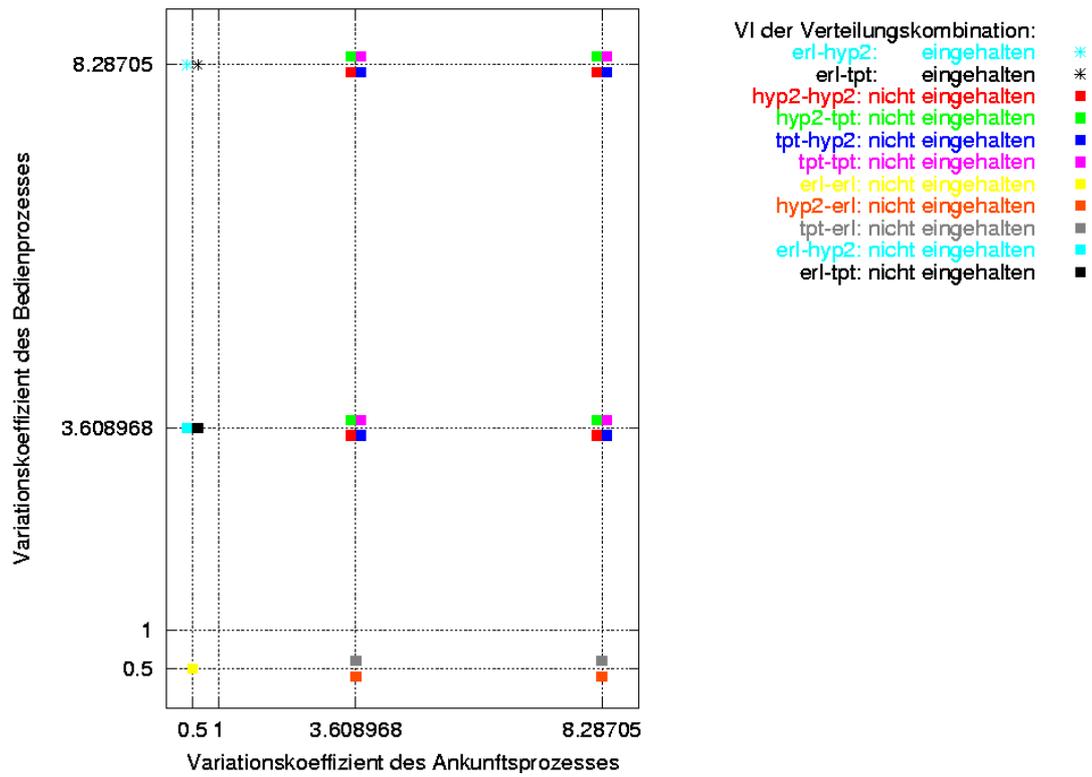


Abbildung B.5: Auslastung 95 %: Heavy Traffic Formel

## B.2 Ergebnisse zusätzlicher Variationen der Variationskoeffizienten

Verteilung	Parameter	Werte
hyp2-hyp2	$C_A^2 / C_B^2$	9/9, 100/100, 225/225, 289/289, 324/324, 400/400, 100/9, 9/100, 225/9, 9/225, 289/9, 9/289, 225/100, 100/225, 289/100, 100/289, 289/225, 225/289
hyp2-tpt	$C_A^2 / C_B^2 / T$	9/8.28705/10, 9/19.3588/13
tpt-hyp2	$T / C_A^2 / C_B^2$	10/8.28705/9, 13/19.3588/9
tpt-tpt	$T / C_A^2 = C_B^2$	10/8.28705/8.28705, 13/19.3588/19.3588
erl-hyp2	$C_A^2 / C_B^2$	0.1/9, 0.5/9, 0.707107/9, 0.1/289, 0.5/289, 0.707107/289,
erl-tpt	$C_A^2 / C_B^2 / T$	0.1/8.28705/10, 0.5/8.28705/10, 0.707107/8.28705/10, 0.1/19.3588/13, 0.5/19.3588/13, 0.707107/19.3588/13

**Tabelle B.1:** Verteilungen und Variation der Variationskoeffizienten innerhalb der GI/G/1 Untersuchungen

	Auslastung	0.2	0.5	0.8	0.9	0.95
Verteilung Ankunfts- / Bedi- enprozess	nexp / hyp2					
Variations- koeffizienten	$C_A^2 = 1 / C_B^2 = 9$					
	Simulationen	0.124695	0.49989	2.00615	4.49745	9.45975
	M/G/1 Formel bzw. Krämer Formel	0.125	0.5	2.0	4.5	9.5
	Prozentuale Ab- weichung	0.24	0.02	0.31	0.06	0.42
Variations- koeffizienten	$C_A^2 = 1 / C_B^2 = 400$					
	Simulationen	4.9773	19.9565	81.4575	179.02	383.63
	M/G/1 Formel bzw. Krämer Formel	5.0125	20.05	80.2	180.45	380.95
	Prozentuale Ab- weichung	0.69	0.47	1.57	0.79	0.7

**Tabelle B.2:** nexp-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Verteilung Ankunfts- / Bedi- enprozess	nexp / tpt					
Tail / Variations- koeffizienten	$T = 10 / C_A^2 = 1 / C_B^2 = 8.28705$					
	Simulationen	0.116095	0.463365	1.85495	4.1671	8.83675
	M/G/1 Formel bzw. Krämer Formel	0.116088	0.464353	1.85741	4.179173	8.822698
	Prozentuale Ab- weichung	0.01	0.21	0.13	0.29	0.16
Tail / Variations- koeffizienten	$T = 13 / C_A^2 = 1 / C_B^2 = 19.3588$					
	Simulationen	0.254475	1.01775	4.0922	9.1759	19.2695
	M/G/1 Formel bzw. Krämer Formel	0.254485	1.01794	4.07176	9.16146	19.34086
	Prozentuale Ab- weichung	0.0	0.02	0.5	0.16	0.37

**Tabelle B.3:** nexp-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Verteilung Ankunfts- / Bedi- enprozess	hyp2 / hyp2					
Variations- koeffizienten	$C_A^2 = 9 / C_B^2 = 9$					
	Simulationen	0.225965	0.91104	3.64305	8.1819	16.969
	Krämer Formel	0.195171	0.823453	3.474249	7.957272	16.948674
	Prozentuale Ab- weichung	13.63	9.61	4.63	2.75	0.12
	Heavy Traffic For- mel	2.925	2.25	4.6125	9.05	18.023684
	Prozentuale Ab- weichung	1194.45	146.97	26.61	10.61	6.22
	Whitt Ansatz	0.225	0.9	3.6	8.1	17.1
	Prozentuale Ab- weichung	0.43	1.21	1.18	1.0	0.77
Variations- koeffizienten	$C_A^2 = 100 / C_B^2 = 100$					
	Simulationen	2.5188	10.0135	40.7405	92.7555	194.775
	Krämer Formel	2.133771	9.057427	38.446953	88.235526	188.12828
	Prozentuale Ab- weichung	15.29	9.55	5.63	4.87	3.41
	Heavy Traffic For- mel	32.5	25	51.25	100.55556	200.26316
	Prozentuale Ab- weichung	1190.3	149.66	25.8	8.41	2.82
	Whitt Ansatz	2.5	10	40	90	190
	Prozentuale Ab- weichung	0.75	0.13	1.82	2.97	2.45
Variations- koeffizienten	$C_A^2 = 225 / C_B^2 = 225$					
	Simulationen	5.59785	22.284	87.5465	193.63	403.73
	Krämer Formel	4.796719	20.367892	86.486424	198.50788	423.26512
	Prozentuale Ab- weichung	13.4	8.6	1.21	2.52	4.84
	Heavy Traffic For- mel	73.125	56.25	115.3125	226.25	450.59211
	Prozentuale Ab- weichung	1206.31	152.42	31.72	16.85	11.61
	Whitt Ansatz	5.625	22.5	90	202.5	427.5
	Prozentuale Ab- weichung	0.49	0.97	2.8	4.58	5.89

**Tabelle B.4:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit identischem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 289 / C_B^2 = 289$					
	Simulationen	7.0796	28.1405	108.9805	239.085	454.6
	Krämer Formel	6.160148	26.158851	111.08263	254.96732	543.65517
	Prozentuale Ab- weichung	12.99	7.04	1.93	6.64	19.59
	Heavy Traffic For- mel	93.925	72.25	148.1125	290.60556	578.76053
	Prozentuale Ab- weichung	1226.7	156.75	35.91	21.55	27.31
	Whitt Ansatz	7.225	28.9	115.6	260.1	549.1
	Prozentuale Ab- weichung	2.05	2.7	6.07	8.79	20.79
Variations- koeffizienten	$C_A^2 = 324 / C_B^2 = 324$					
	Simulationen	7.8921	30.3885	118.05	251.69	475.65
	Krämer Formel	6.905774	29.325782	124.53368	285.84358	609.49349
	Prozentuale Ab- weichung	12.5	3.5	5.49	13.57	28.14
	Heavy Traffic For- mel	105.3	81	166.05	325.8	648.85263
	Prozentuale Ab- weichung	1234.25	166.55	40.66	29.44	36.41
	Whitt Ansatz	8.1	32.4	129.6	291.6	615.6
	Prozentuale Ab- weichung	2.63	6.62	9.78	15.86	29.42
Variations- koeffizienten	$C_A^2 = 400 / C_B^2 = 400$					
	Simulationen	9.0835	37.7125	142.31	287.715	559.82
	Krämer Formel	8.524847	36.202546	153.74168	352.88917	752.45668
	Prozentuale Ab- weichung	6.08	4.0	8.03	22.65	34.41
	Heavy Traffic For- mel	130	100	205	402.22222	801.05263
	Prozentuale Ab- weichung	1315.58	165.16	44.05	39.8	43.09
	Whitt Ansatz	10	40	160	360	760
	Prozentuale Ab- weichung	8.89	6.07	12.43	25.12	35.76

**Tabelle B.5:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit identischem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 100 / C_B^2 = 9$					
	Simulationen	0.315685	2.9649	20.972	49.042	105.424
	Krämer Formel	0.761069	3.787268	18.846397	45.606309	99.848849
	Prozentuale Ab- weichung	141.08	27.74	10.14	7.01	5.29
	Heavy Traffic For- mel	31.3625	20.45	33.05	59.605556	113.813116
	Prozentuale Ab- weichung	9834.75	589.74	57.59	21.54	7.96
	Whitt Ansatz	1.3625	5.45	21.8	49.05	103.55
	Prozentuale Ab- weichung	331.6	83.82	3.95	0.02	1.78
Variations- koeffizienten	$C_A^2 = 9 / C_B^2 = 100$					
	Simulationen	1.51265	5.6347	22.4415	50.565	110.329
	Krämer Formel	1.341346	5.396959	21.714885	48.954152	103.44878
	Prozentuale Ab- weichung	11.32	4.22	3.24	3.19	6.24
	Heavy Traffic For- mel	4.0625	6.8	22.8125	50	104.47368
	Prozentuale Ab- weichung	168.57	20.68	1.65	1.12	5.31
	Whitt Ansatz	1.3625	5.45	21.8	49.05	103.55
	Prozentuale Ab- weichung	9.93	3.28	2.86	2.99	6.14

**Tabelle B.6:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 225 / C_B^2 = 9$					
	Simulationen	0.32543	4.5153	44.0025	105.845	223.53
	Krämer Formel	1.472121	7.617666	39.418518	96.639703	212.96247
	Prozentuale Ab- weichung	352.36	68.71	10.42	8.7	4.73
	Heavy Traffic For- mel	70.425	45.45	72.1125	129.05	245.39211
	Prozentuale Ab- weichung	21540.6	906.58	63.88	21.92	9.78
	Whitt Ansatz	2.925	11.7	46.8	105.3	222.3
	Prozentuale Ab- weichung	798.81	159.12	6.36	0.52	0.55
Variations- koeffizienten	$C_A^2 = 9 / C_B^2 = 225$					
	Simulationen	3.0689	11.7175	46.0075	101.4635	200.175
	Krämer Formel	2.904478	11.648628	46.717696	105.20737	222.2022
	Prozentuale Ab- weichung	5.36	0.59	1.54	3.69	11.0
	Heavy Traffic For- mel	5.625	13.05	47.8125	106.25	223.22368
	Prozentuale Ab- weichung	83.29	11.37	3.92	4.72	11.51
	Whitt Ansatz	2.9525	11.7	46.8	105.3	222.3
	Prozentuale Ab- weichung	3.79	0.15	1.72	3.78	11.05

**Tabelle B.7:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 289 / C_B^2 = 9$					
	Simulationen	3.79	14.9055	56.778	127.58	247.755
	Krämer Formel	1.833348	9.566662	49.920154	122.727999	270.83032
	Prozentuale Ab- weichung	51.64	35.82	12.08	3.8	9.31
	Heavy Traffic For- mel	90.425	58.25	92.1125	164.60556	312.76053
	Prozentuale Ab- weichung	2285.38	290.8	62.23	29.02	26.24
	Whitt Ansatz	3.725	14.9	59.6	134.1	283.1
	Prozentuale Ab- weichung	0.45	0.04	4.97	5.11	14.27
Variations- koeffizienten	$C_A^2 = 9 / C_B^2 = 289$					
	Simulationen	3.8117	14.691	56.348	122.08	245.985
	Krämer Formel	3.704593	14.848929	59.518202	134.00795	283.00281
	Prozentuale Ab- weichung	2.81	1.08	5.56	9.77	15.05
	Heavy Traffic For- mel	6.425	16.25	60.6125	135.05	284.02368
	Prozentuale Ab- weichung	68.56	10.61	7.57	10.62	15.46
	Whitt Ansatz	3.725	14.9	59.6	134.1	283.1
	Prozentuale Ab- weichung	2.27	1.42	5.77	9.85	15.09

**Tabelle B.8:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 225 / C_B^2 = 100$					
	Simulationen	2.8698	14.458	65.272	158.995	331.255
	Krämer Formel	3.049808	13.584004	60.503866	141.10122	303.26648
	Prozentuale Ab- weichung	6.27	6.05	4.77	11.25	8.45
	Heavy Traffic For- mel	71.5625	50	90.3125	170	331.84211
	Prozentuale Ab- weichung	2393.64	245.83	38.36	6.92	0.18
	Whitt Ansatz	4.0625	16.25	65	146.25	308.75
	Prozentuale Ab- weichung	41.56	12.39	0.42	8.02	6.79
Variations- koeffizienten	$C_A^2 = 100 / C_B^2 = 225$					
	Simulationen	4.67395	16.8865	63.4255	139.29	283.63
	Krämer Formel	3.753162	15.465209	63.725658	144.80927	307.22546
	Prozentuale Ab- weichung	19.7	8.42	0.47	3.96	8.32
	Heavy Traffic For- mel	34.0625	31.25	76.25	156.80556	319.01316
	Prozentuale Ab- weichung	628.77	85.06	19.75	12.57	12.48
	Whitt Ansatz	4.0625	16.25	65	146.25	308.75
	Prozentuale Ab- weichung	13.08	3.77	2.48	5.0	8.86

**Tabelle B.9:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 289 / C_B^2 = 100$					
	Simulationen	2.96255	16.2695	76.6945	178.17	397.81
	Krämer Formel	3.480427	15.78166	71.560412	167.88378	361.90661
	Prozentuale Ab- weichung	17.48	3.0	6.69	5.77	9.03
	Heavy Traffic For- mel	91.5625	62.8	110.3125	205.55556	399.21053
	Prozentuale Ab- weichung	2990.67	286.0	43.83	15.37	0.35
	Whitt Ansatz	4.8625	19.45	77.8	175.05	369.55
	Prozentuale Ab- weichung	64.13	19.55	1.44	1.75	7.10
Variations- koeffizienten	$C_A^2 = 100 / C_B^2 = 289$					
	Simulationen	5.63945	19.9	75.4595	162.64	311.62
	Krämer Formel	4.565351	18.698368	76.583152	173.67565	368.09644
	Prozentuale Ab- weichung	19.05	6.04	1.49	6.79	18.12
	Heavy Traffic For- mel	34.8625	34.45	89.05	185.60556	379.813116
	Prozentuale Ab- weichung	518.19	73.16	18.01	14.12	21.88
	Whitt Ansatz	4.8625	19.45	77.8	175.05	369.55
	Prozentuale Ab- weichung	13.78	2.26	3.10	8.18	18.59

**Tabelle B.10:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 289 / C_B^2 = 225$					
	Simulationen	5.8342	24.6785	100.1255	213.89	481.285
	Krämer Formel	5.293186	22.768564	97.938652	225.76475	482.42186
	Prozentuale Ab- weichung	9.27	7.74	2.18	5.55	0.24
	Heavy Traffic For- mel	93.125	69.05	135.3125	261.80556	517.96053
	Prozentuale Ab- weichung	1496.19	179.8	35.14	22.4	7.62
	Whitt Ansatz	6.425	25.7	102.8	231.3	488.3
	Prozentuale Ab- weichung	10.13	4.14	2.67	8.14	1.46
Variations- koeffizienten	$C_A^2 = 225 / C_B^2 = 289$					
	Simulationen	6.71515	25.6175	101.51	210.71	411.77
	Krämer Formel	5.643111	23.697993	99.518653	227.57855	484.35587
	Prozentuale Ab- weichung	15.96	7.49	1.96	8.01	17.63
	Heavy Traffic For- mel	73.925	59.45	128.1125	255.05	511.39211
	Prozentuale Ab- weichung	1000.87	132.07	26.21	21.04	24.19
	Whitt Ansatz	6.425	25.7	102.8	231.3	488.3
	Prozentuale Ab- weichung	4.32	0.32	1.27	9.77	18.59

**Tabelle B.11:** hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem  $C_A^2$  und  $C_B^2$

	Auslastung	0.2	0.5	0.8	0.9	0.95
Tail / Variationskoeffizienten	$T = 10 / C_A^2 = 8.28705 / C_B^2 = 9$					
	Simulationen	0.188422	0.74717	3.181075	7.37765	16.1375
	Krämer Formel	0.189436	0.796088	3.345844	7.65222	16.288141
	Prozentuale Abweichung	0.54	6.15	5.18	3.72	0.93
	Heavy Traffic Formel	2.702203	2.10741	4.389703	8.653917	17.273211
	Prozentuale Abweichung	1334.12	182.05	37.99	17.3	7.04
	Whitt Ansatz	0.216088	0.864353	3.45741	7.779173	16.422697
	Prozentuale Abweichung	14.68	15.68	8.69	5.44	1.77
Tail / Variationskoeffizienten	$T = 13 / C_A^2 = 19.3588 / C_B^2 = 9$					
	Simulationen	0.193503	0.779675	3.473917	8.3766	18.843
	Krämer Formel	0.27188	1.201281	5.307776	12.345189	26.49782
	Prozentuale Abweichung	40.5	54.07	52.79	47.38	40.62
	Heavy Traffic Formel	6.162125	4.32176	7.849625	14.804889	28.927684
	Prozentuale Abweichung	3084.51	454.3	125.96	76.74	53.52
	Whitt Ansatz	0.354485	1.41784	5.67176	12.76146	26.94086
	Prozentuale Abweichung	83.19	81.86	63.27	52.35	42.98

Tabelle B.12: tpt-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Tail / Variationskoeffizienten	$T = 10 / C_A^2 = 9 / C_B^2 = 8.28705$					
	Simulationen	0.157175	0.644785	2.606925	5.955725	13.311258
	Krämer Formel	0.185646	0.786095	3.328622	7.632911	16.267578
	Prozentuale Abweichung	18.11	21.92	27.68	28.16	22.21
	Heavy Traffic Formel	2.916088	2.214353	4.46991	8.729173	17.346382
	Prozentuale Abweichung	1755.31	243.43	71.46	46.57	30.31
	Whitt Ansatz	0.216088	0.864353	3.45741	7.779173	16.422697
	Prozentuale Abweichung	37.48	34.05	32.62	30.62	23.37
Tail / Variationskoeffizienten	$T = 13 / C_A^2 = 9 / C_B^2 = 19.3588$					
	Simulationen	0.224133	0.896257	3.622	8.21315	18.916843
	Krämer Formel	0.329185	1.353817	5.567736	12.643891	26.816473
	Prozentuale Abweichung	46.87	51.05	53.72	46.54	41.76
	Heavy Traffic Formel	3.054485	2.76794	6.68426	13.71146	27.864544
	Prozentuale Abweichung	1262.8	208.83	84.55	66.95	47.3
	Whitt Ansatz	0.354485	1.41794	5.67176	12.76146	26.94086
	Prozentuale Abweichung	58.16	58.21	56.59	55.38	42.42

**Tabelle B.13:** hyp2-tp Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Tail / Variationskoeffizienten	$T = 10 / C_A^2 = 8.28705 / C_B^2 = 8.28705$					
	Simulationen	0.168465	0.682855	2.9766	7.012015	15.2415
	Krämer Formel	0.179986	0.758947	3.200254	7.328325	15.607548
	Prozentuale Abweichung	6.84	11.14	7.51	4.51	2.4
	Heavy Traffic Formel	2.693291	2.071763	4.247113	8.333089	16.595908
	Prozentuale Abweichung	1498.72	203.4	42.68	18.84	8.89
	Whitt Ansatz	0.207176	0.828705	3.31482	7.458345	15.745395
	Prozentuale Abweichung	22.98	21.36	11.36	6.37	3.31
Tail / Variationskoeffizienten	$T = 13 / C_A^2 = 19.3588 / C_B^2 = 19.3588$					
	Simulationen	0.342215	1.40945	6.1964	15.3495	33.815
	Krämer Formel	0.415835	1.760729	7.455281	17.095576	36.434552
	Prozentuale Abweichung	21.51	24.92	20.32	11.38	7.75
	Heavy Traffic Formel	6.29161	4.8397	9.921385	19.466349	38.768544
	Prozentuale Abweichung	1738.5	243.38	60.12	26.82	14.65
	Whitt Ansatz	0.48397	1.93588	7.74352	17.42292	36.78172
	Prozentuale Abweichung	41.42	37.35	24.97	13.51	8.77

**Tabelle B.14:** tpt-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 0.1 / C_B^2 = 9$					
	Simulationen	0.088075	0.41883	1.77115	4.01595	8.51275
	Krämer Formel	0.087244	0.424356	1.782477	4.04705	8.576406
	Prozentuale Ab- weichung	1.71	1.32	0.64	0.77	0.75
	Heavy Traffic For- mel	0.130883	0.461765	1.818383	4.08268	8.61192
	Prozentuale Ab- weichung	48.6	10.25	2.67	1.6	1.16
	Whitt Ansatz	0.113235	0.452941	1.811765	4.076471	8.605883
	Prozentuale Ab- weichung	28.57	8.14	2.29	1.51	1.09
Variations- koeffizienten	$C_A^2 = 0.5 / C_B^2 = 9$					
	Simulationen	0.09599	0.43887	1.81775	4.1228	8.7033
	Krämer Formel	0.110702	0.466739	1.891685	4.266675	9.016671
	Prozentuale Ab- weichung	15.37	6.35	4.07	3.49	3.6
	Heavy Traffic For- mel	9.076316	4.327778	1.95625	0.55	0.26875
	Prozentuale Ab- weichung	9355.48	886.12	7.67	86.66	96.91
	Whitt Ansatz	0.11875	0.475	1.9	4.275	9.025
	Prozentuale Ab- weichung	23.71	8.23	4.52	3.69	3.7
Variations- koeffizienten	$C_A^2 = 0.707107 / C_B^2 = 9$					
	Simulationen	0.10556	0.459175	1.87785	4.27665	9.06415
	Krämer Formel	0.118513	0.482504	1.938564	4.36534	9.218893
	Prozentuale Ab- weichung	12.27	5.08	3.23	2.07	1.71
	Heavy Traffic For- mel	0.333471	0.591421	2.020971	4.442837	9.294323
	Prozentuale Ab- weichung	215.91	28.8	7.62	3.89	2.54
	Whitt Ansatz	0.121339	0.485355	1.941421	4.368198	9.2217516
	Prozentuale Ab- weichung	14.95	5.7	3.39	2.14	1.74

**Tabelle B.15:** erl-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Variations- koeffizienten	$C_A^2 = 0.1 / C_B^2 = 289$					
	Simulationen	3.5658	14.3925	57.993	129.475	286.0
	Krämer Formel	3.586851	14.428025	57.793006	130.068	
	Prozentuale Ab- weichung	0.59	0.25	0.34	0.46	3.98
	Heavy Traffic For- mel	3.64375	14.47	57.83125	130.10556	274.65526
	Prozentuale Ab- weichung	2.19	0.54	0.28	0.49	3.97
	Whitt Ansatz	3.61375	14.455	57.82	130.095	274.645
	Prozentuale Ab- weichung	1.34	0.43	0.3	0.48	3.97
Variations- koeffizienten	$C_A^2 = 0.5 / C_B^2 = 289$					
	Simulationen	3.59115	14.481	58.7045	129.36	266.735
	Krämer Formel	3.610426	14.466669	57.891667	130.26667	275.01667
	Prozentuale Ab- weichung	0.54	0.1	1.38	0.7	3.1
	Heavy Traffic For- mel	3.76875	14.55	57.95625	130.32778	275.07632
	Prozentuale Ab- weichung	4.95	0.48	1.27	0.75	3.13
	Whitt Ansatz	3.61875	14.475	57.9	130.275	275.025
	Prozentuale Ab- weichung	0.77	0.04	1.37	0.71	3.11
Variations- koeffizienten	$C_A^2 = 0.707107 / C_B^2 = 289$					
	Simulationen	3.5824	14.571	57.7585	126.965	293.255
	Krämer Formel	3.61848	14.482496	57.938562	130.36534	275.21889
	Prozentuale Ab- weichung	1.01	0.61	0.31	2.68	6.15
	Heavy Traffic For- mel	3.833471	14.591421	58.020971	130.44284	275.29432
	Prozentuale Ab- weichung	7.01	0.14	0.45	2.74	6.13
	Whitt Ansatz	3.621339	14.485355	57.941421	130.3682	275.22175
	Prozentuale Ab- weichung	1.09	0.59	0.32	2.68	6.15

Tabelle B.16: erl-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Tail / Variationskoeffizienten	$T = 10 / C_A^2 = 0.1 / C_B^2 = 8.28705$					
	Simulationen	0.081735	0.3843	1.63185	3.7189	7.954
	Krämer Formel	0.081035	0.393203	1.650626	3.747269	7.940743
	Prozentuale Abweichung	0.86	2.32	1.15	0.76	0.17
	Heavy Traffic Formel	0.134838	0.434353	1.68866	3.784728	7.977961
	Prozentuale Abweichung	64.97	13.02	3.48	1.77	0.3
	Whitt Ansatz	0.104838	0.419353	1.67741	3.774173	7.967698
	Prozentuale Abweichung	28.27	9.12	2.79	1.49	0.17
Tail / Variationskoeffizienten	$T = 10 / C_A^2 = 0.5 / C_B^2 = 8.28705$					
	Simulationen	0.089235	0.403345	1.68085	3.81905	7.9974
	Krämer Formel	0.101813	0.431098	1.749096	3.945848	8.339368
	Prozentuale Abweichung	14.1	6.88	4.06	3.32	4.28
	Heavy Traffic Formel	0.259838	0.514353	1.81366	4.00695	8.399013
	Prozentuale Abweichung	191.18	27.52	7.9	4.92	5.02
	Whitt Ansatz	0.109838	0.439353	1.75741	3.954173	8.347698
	Prozentuale Abweichung	23.09	8.93	4.55	3.54	4.38
Tail / Variationskoeffizienten	$T = 10 / C_A^2 = 0.707107 / C_B^2 = 8.28705$					
	Simulationen	0.0975	0.424105	1.74415	3.9487	8.2798
	Krämer Formel	0.109603	0.446857	1.795974	4.044512	8.54159
	Prozentuale Abweichung	12.41	5.36	2.97	2.43	3.16
	Heavy Traffic Formel	0.324559	0.555774	1.878381	4.12201	8.617021
	Prozentuale Abweichung	232.88	31.05	7.7	4.39	4.07
	Whitt Ansatz	0.112427	0.449708	1.798831	4.047371	8.544449
	Prozentuale Abweichung	15.31	6.04	3.14	2.5	3.2

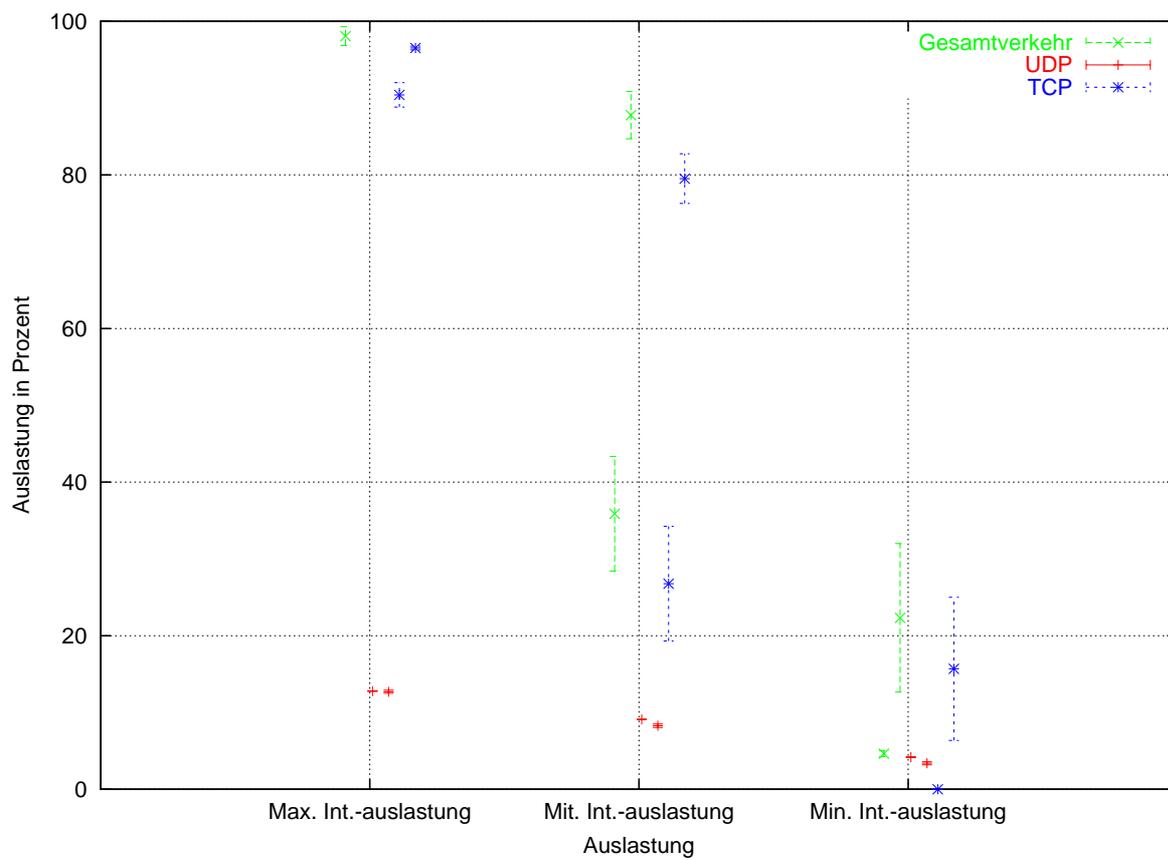
**Tabelle B.17:** erl-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

	Auslastung	0.2	0.5	0.8	0.9	0.95
Tail / Variationskoeffizienten	$T = 13 / C_A^2 = 0.1 / C_B^2 = 19.3588$					
	Simulationen	0.21884	0.9381	3.8261	8.6007	18.6445
	Krämer Formel	0.21768	0.946311	3.864853	8.729502	18.45888
	Prozentuale Abweichung	0.53	0.88	1.01	1.5	1.0
	Heavy Traffic Formel	0.273235	0.98794	3.90301	8.767016	18.496123
	Prozentuale Abweichung	24.86	5.31	2.01	1.93	0.8
	Whitt Ansatz	0.243235	0.97294	3.89176	8.75646	18.48586
	Prozentuale Abweichung	11.68	3.71	1.72	1.81	0.85
Tail / Variationskoeffizienten	$T = 13 / C_A^2 = 0.5 / C_B^2 = 19.3588$					
	Simulationen	0.23636	0.9816	3.9164	8.97	18.9875
	Krämer Formel	0.24004	0.984642	3.963435	8.928131	18.857529
	Prozentuale Abweichung	1.56	0.31	1.2	0.47	0.68
	Heavy Traffic Formel	0.398235	1.06794	4.02801	8.989238	18.917176
	Prozentuale Abweichung	68.49	8.8	2.85	0.21	0.37
	Whitt Ansatz	0.248235	0.99294	3.97176	8.93646	18.86586
	Prozentuale Abweichung	5.02	1.16	1.4	0.37	0.64
Tail / Variationskoeffizienten	$T = 13 / C_A^2 = 0.707107 / C_B^2 = 19.3588$					
	Simulationen	0.229975	0.95185	3.8895	8.92925	18.876
	Krämer Formel	0.247981	1.00044	4.010323	9.026799	19.059752
	Prozentuale Abweichung	7.83	5.13	3.12	1.09	0.97
	Heavy Traffic Formel	0.462956	1.109361	4.092731	9.104297	19.135183
	Prozentuale Abweichung	101.31	16.55	5.23	1.96	1.37
	Whitt Ansatz	0.250824	1.003295	4.013181	9.029658	19.062612
	Prozentuale Abweichung	9.07	5.4	3.18	1.12	0.99

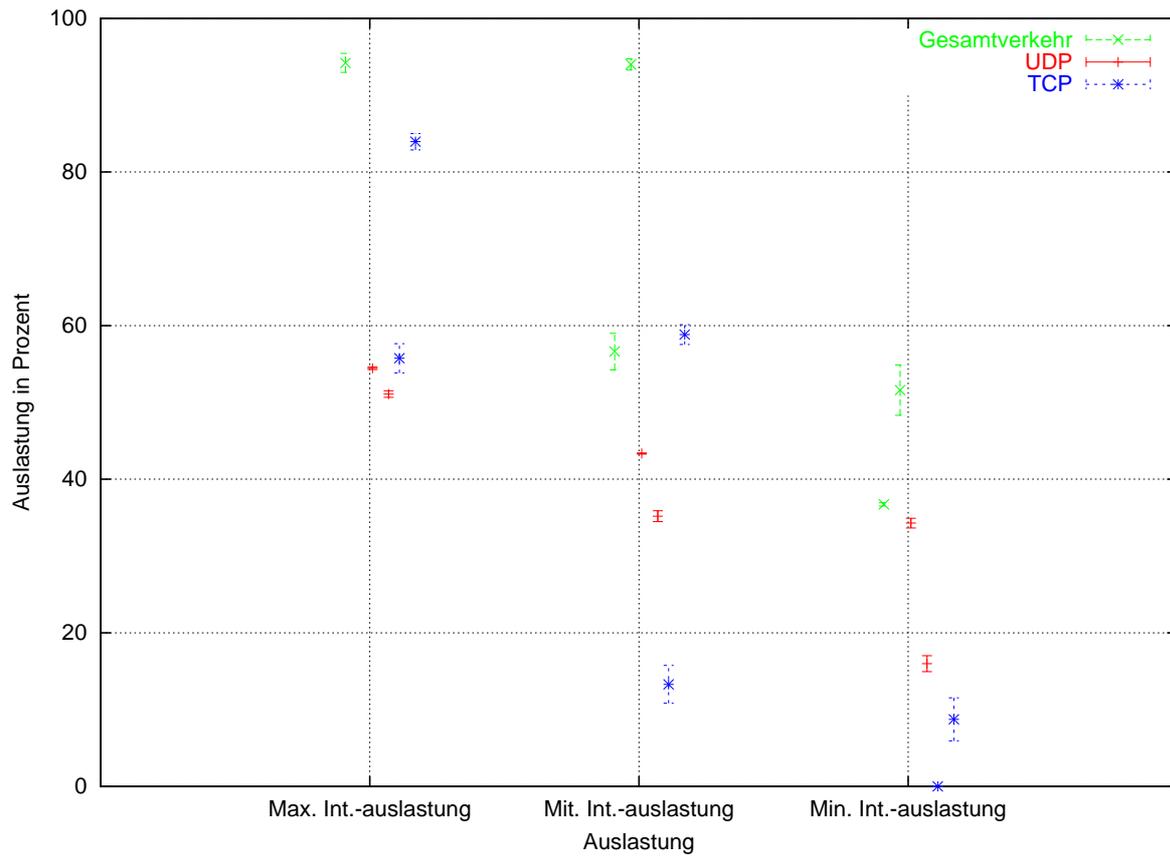
Tabelle B.18: erl-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit

## Anhang C

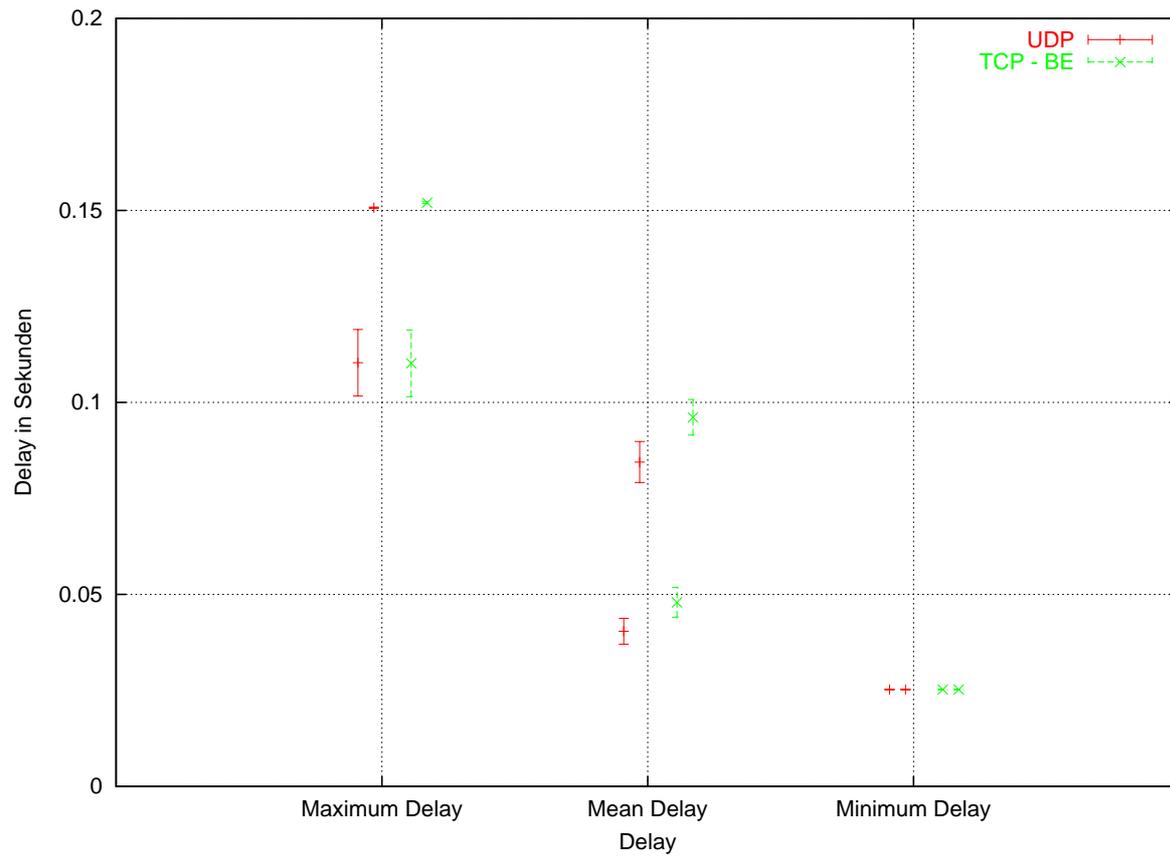
# Simulationsergebnisse des Extrapolations-Testszenarios



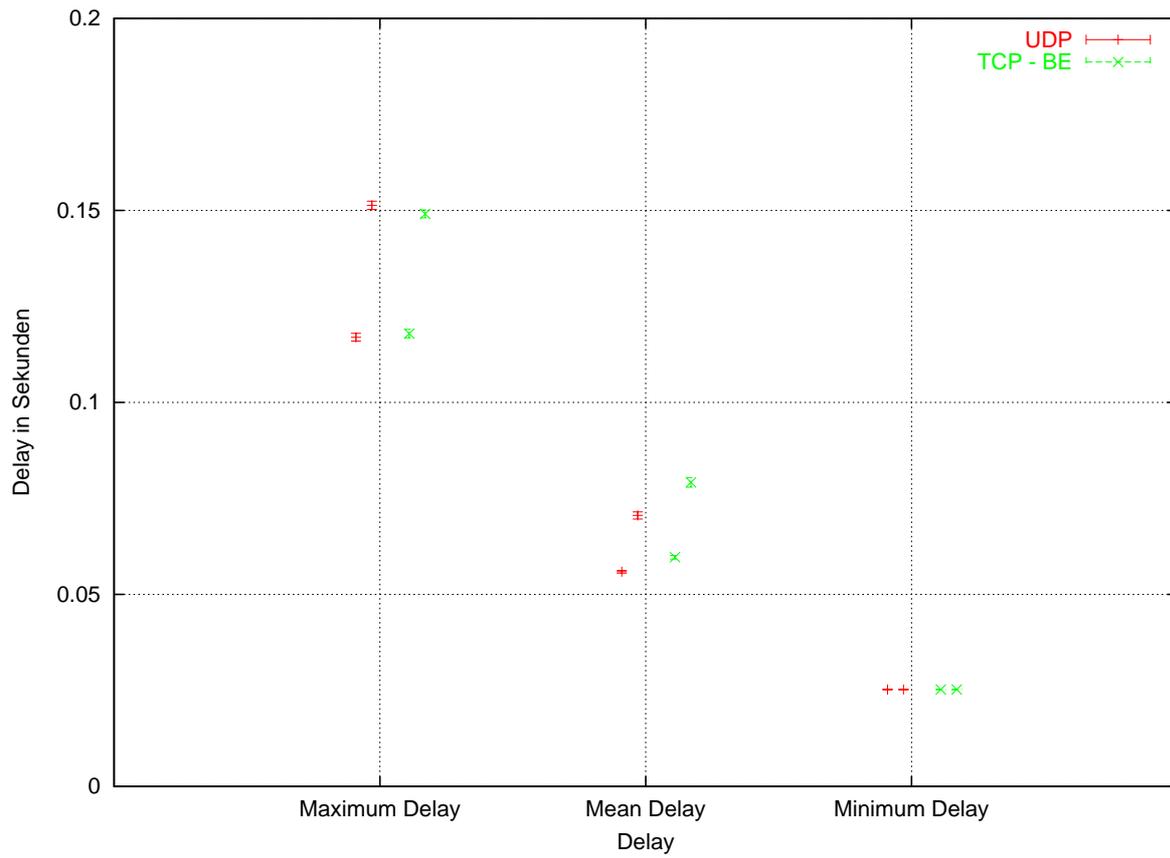
**Abbildung C.1:** BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



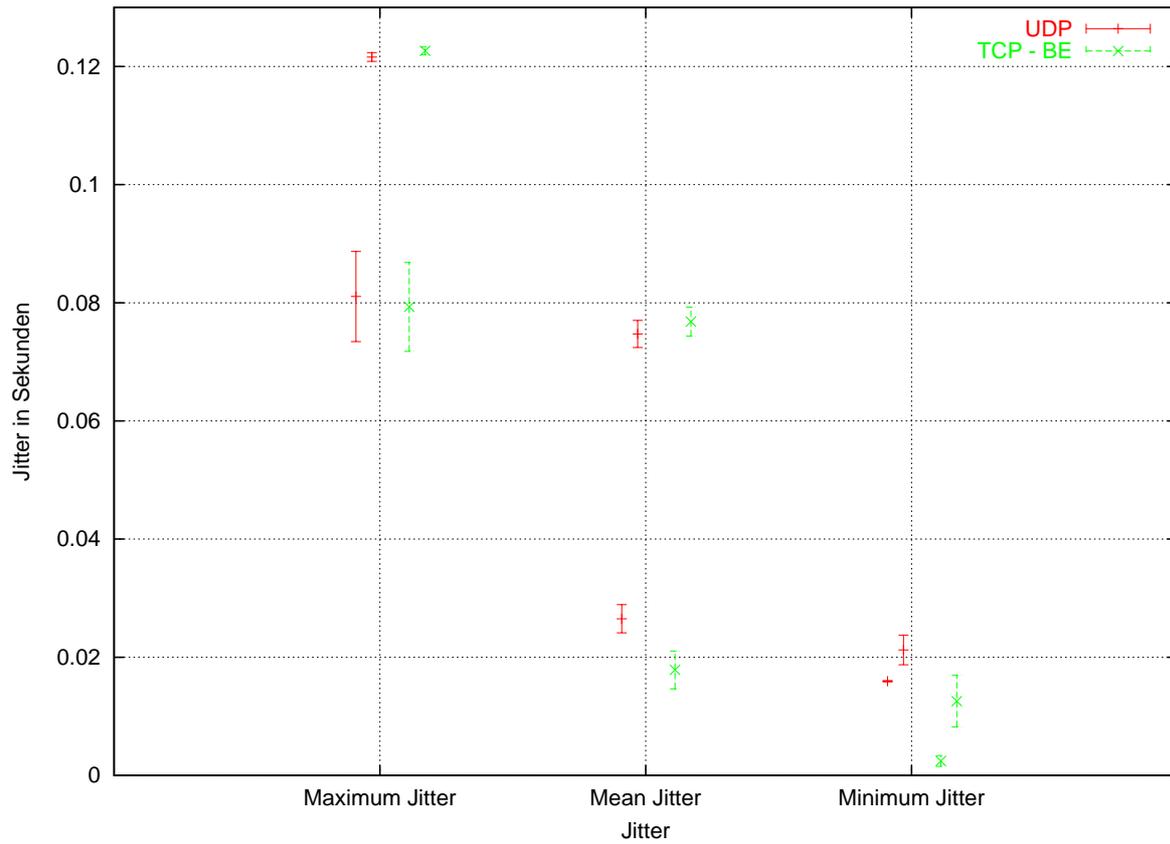
**Abbildung C.2:** BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



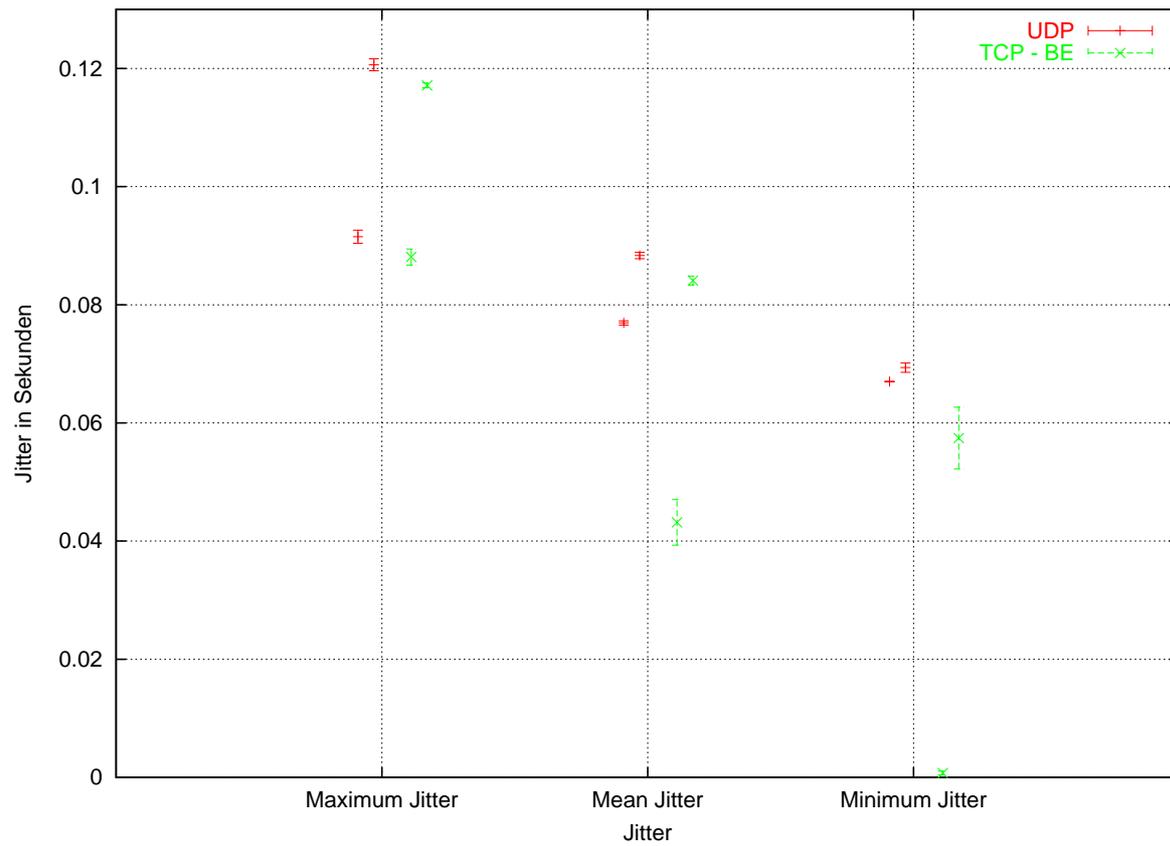
**Abbildung C.3:** BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



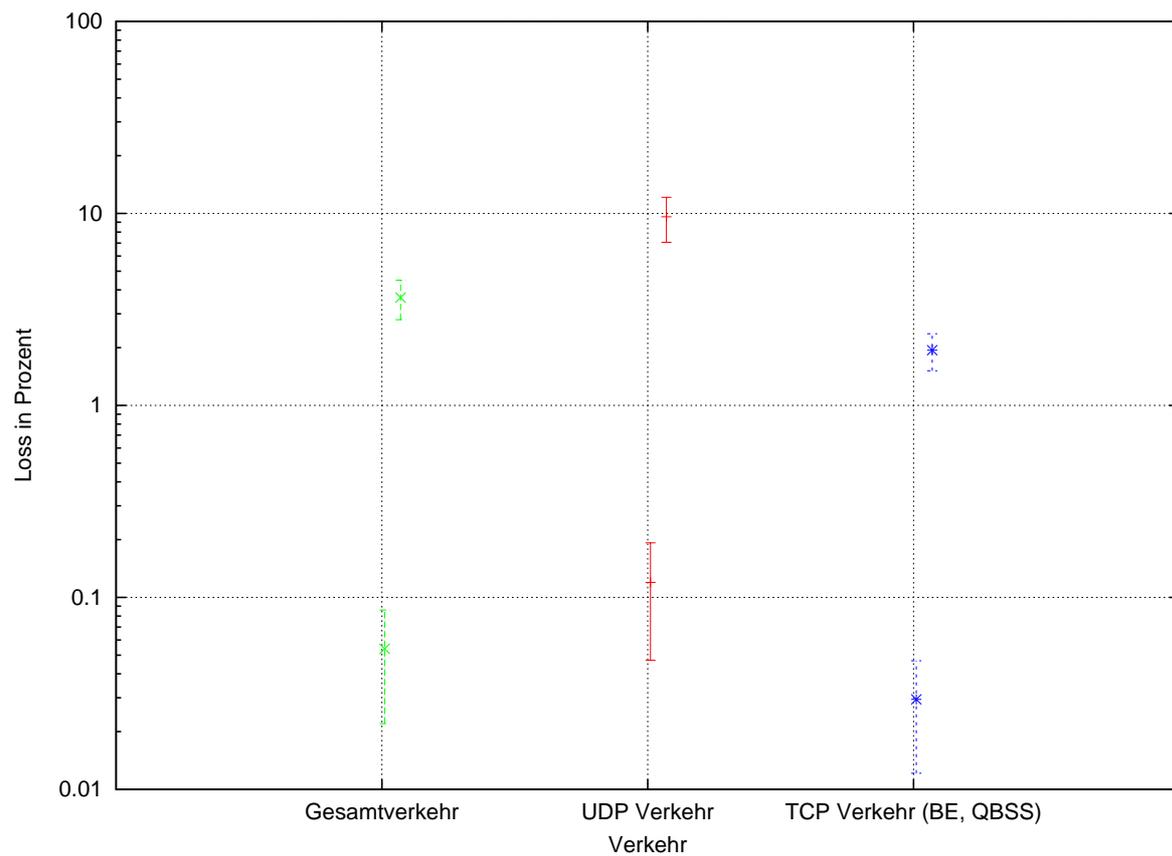
**Abbildung C.4:** BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



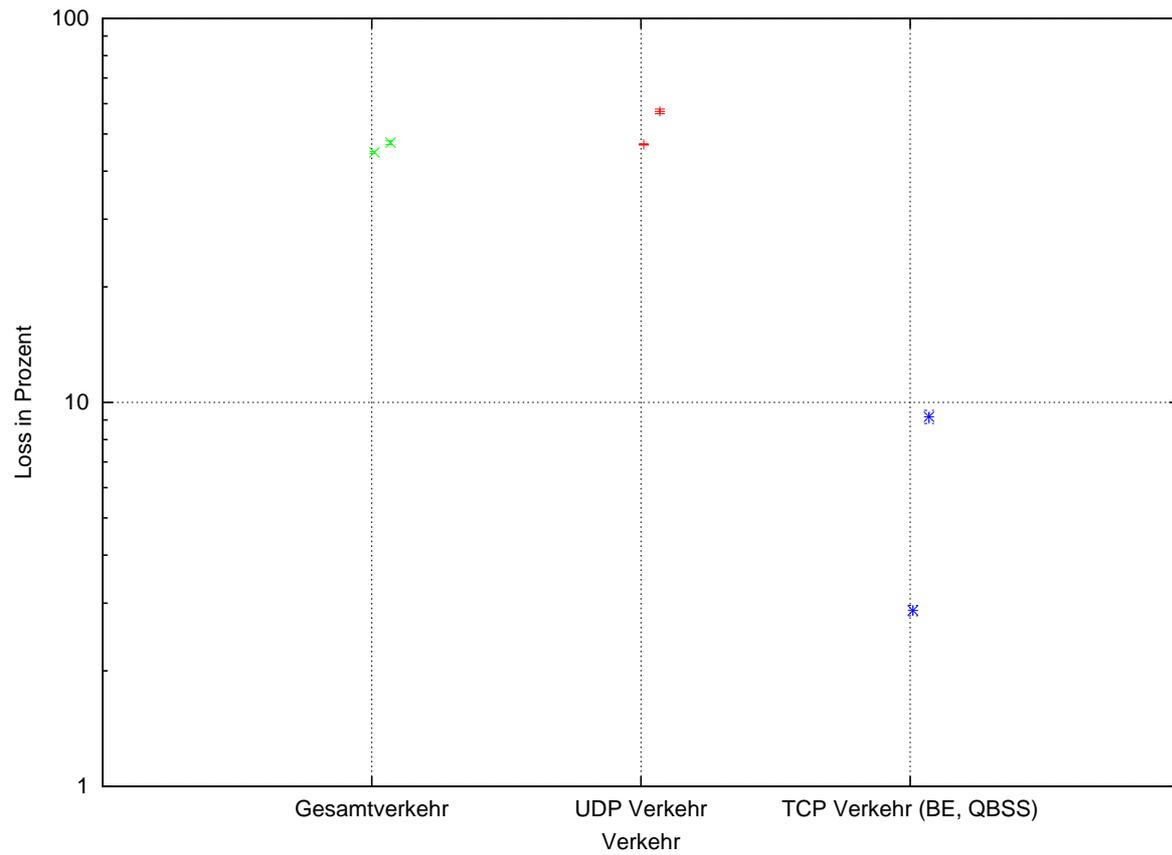
**Abbildung C.5:** BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



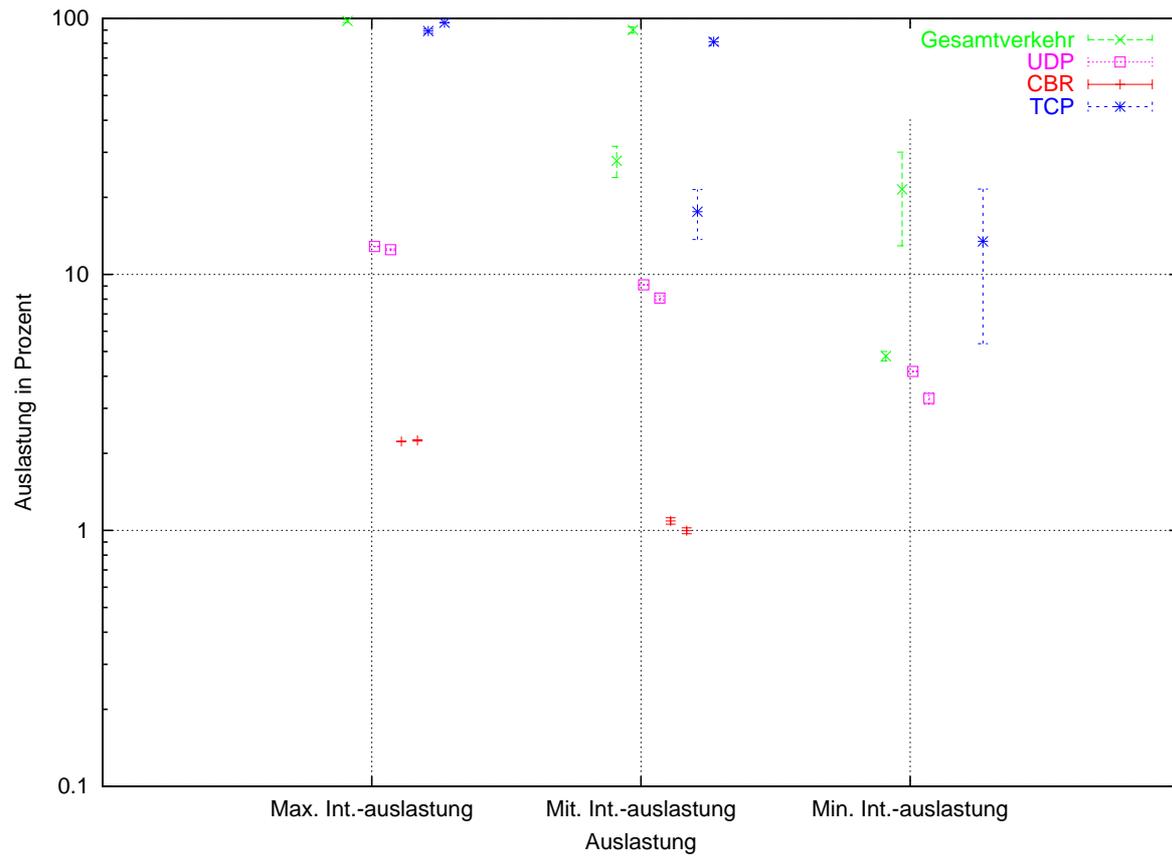
**Abbildung C.6:** BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



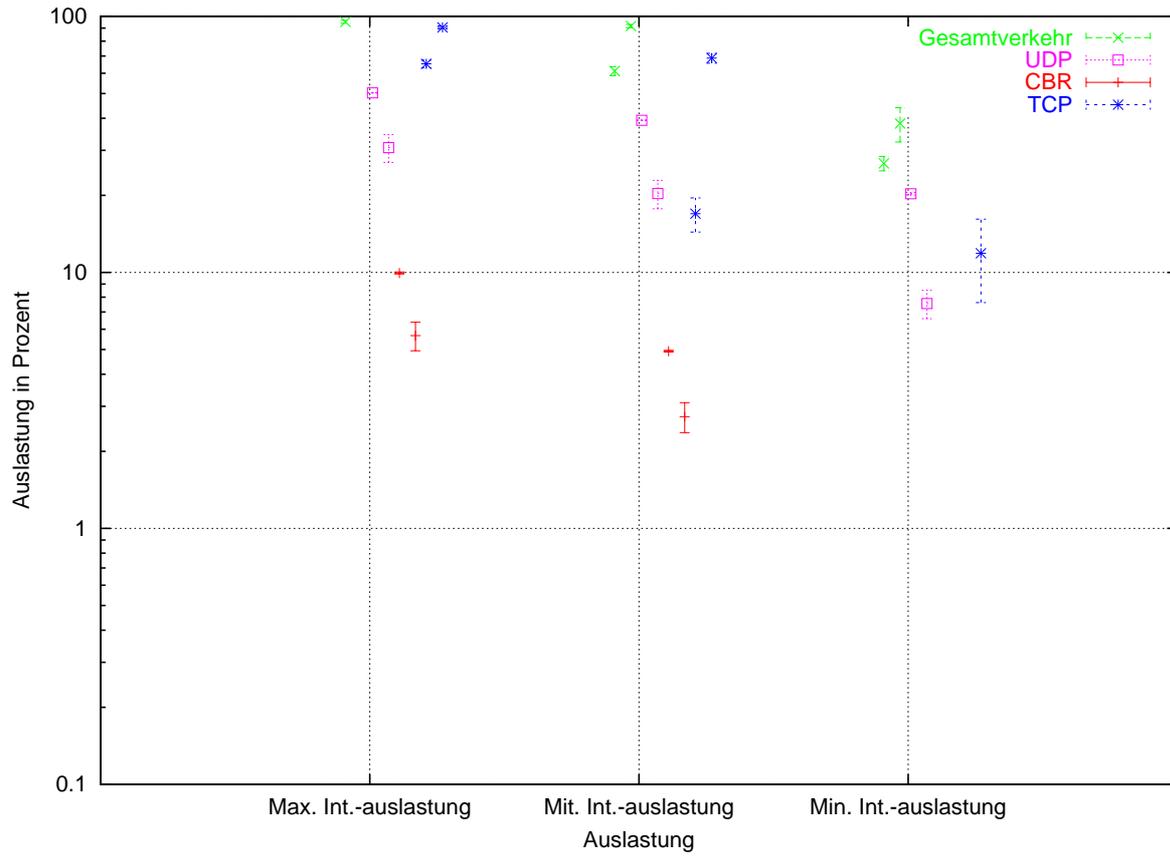
**Abbildung C.7:** BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



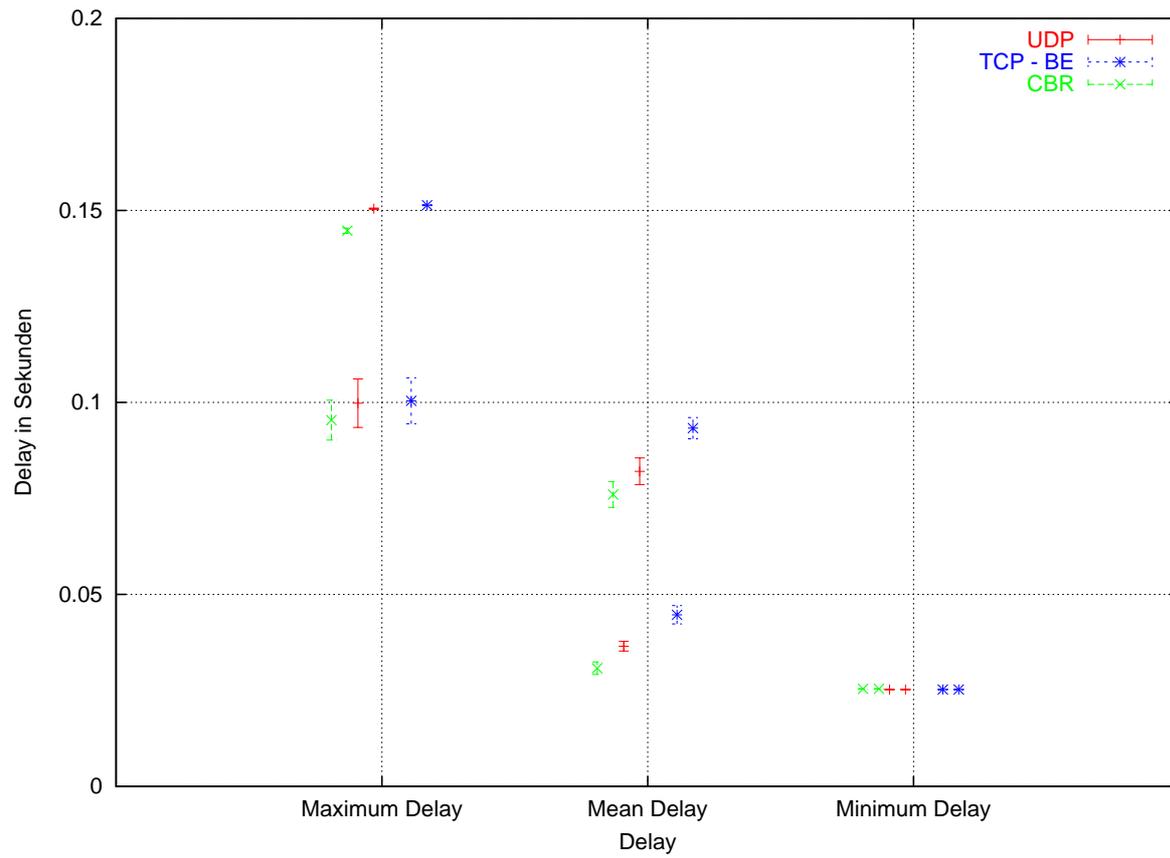
**Abbildung C.8:** BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



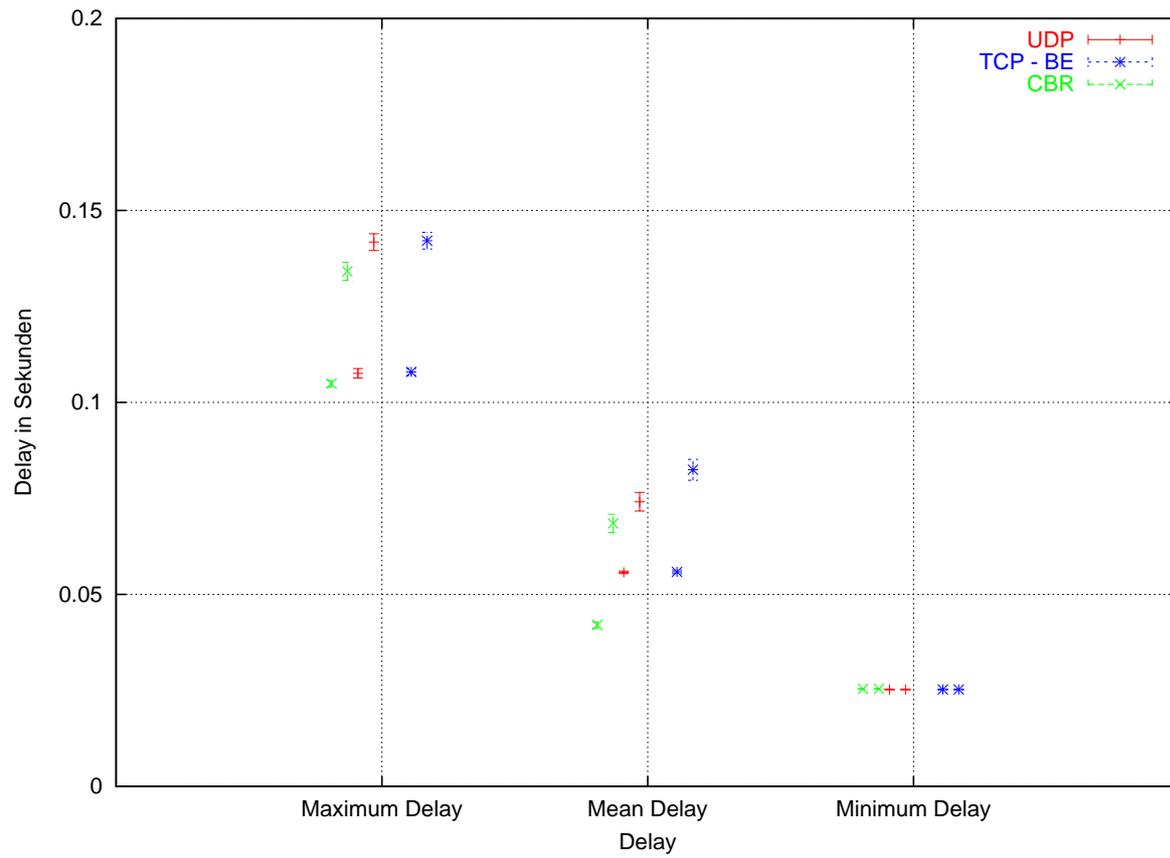
**Abbildung C.9:** BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



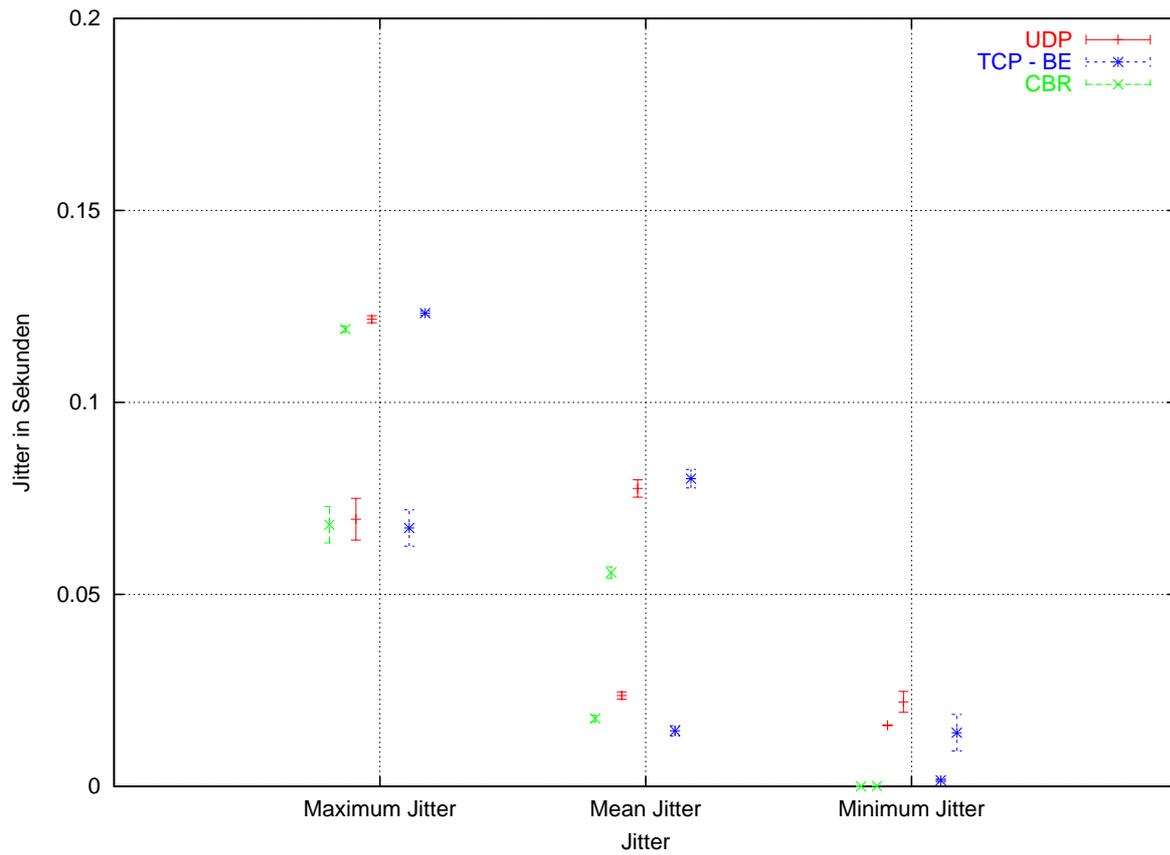
**Abbildung C.10:** BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



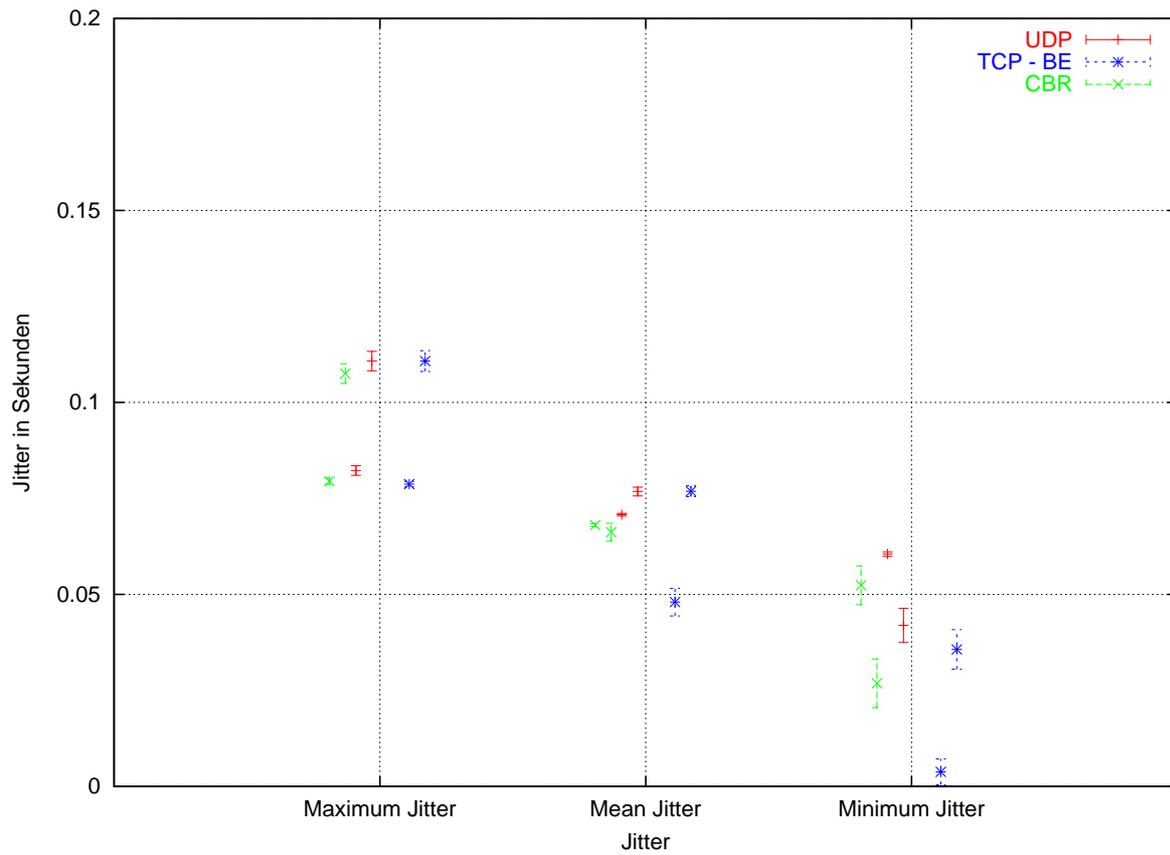
**Abbildung C.11:** BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



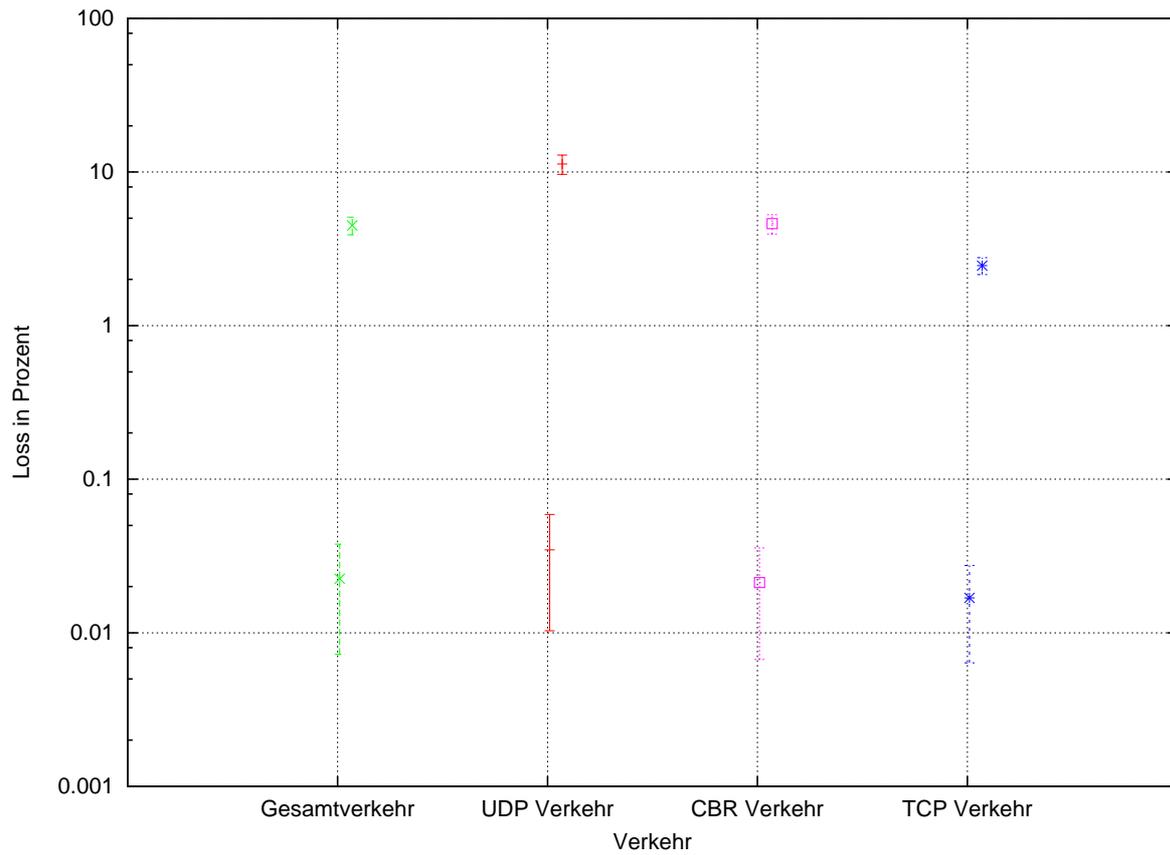
**Abbildung C.12:** BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



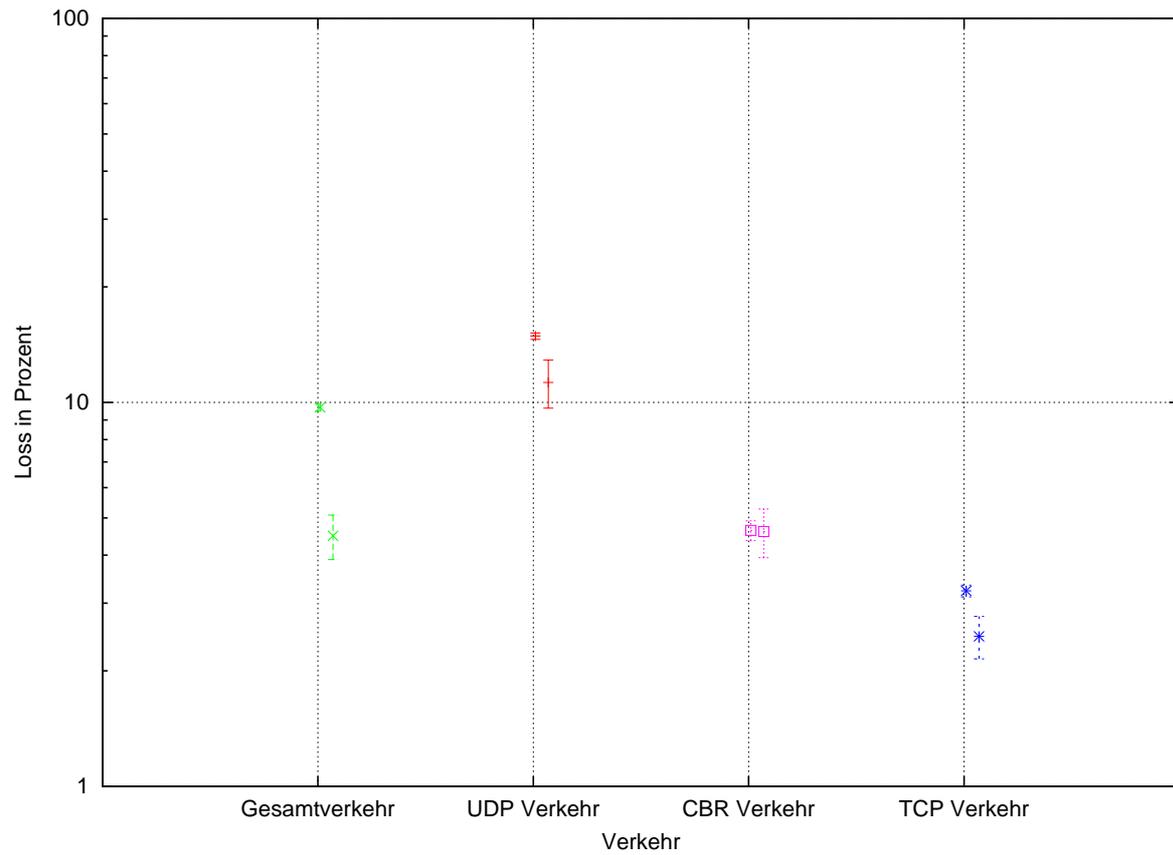
**Abbildung C.13:** BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



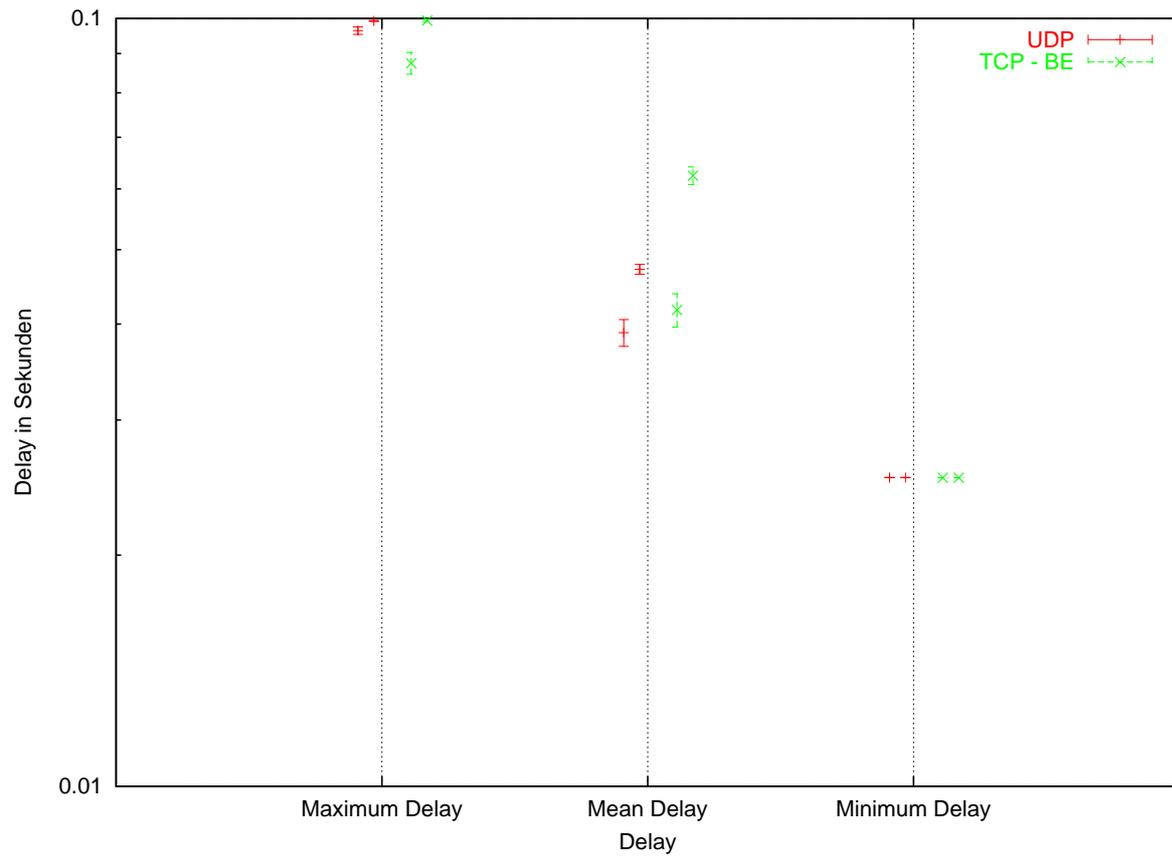
**Abbildung C.14:** BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



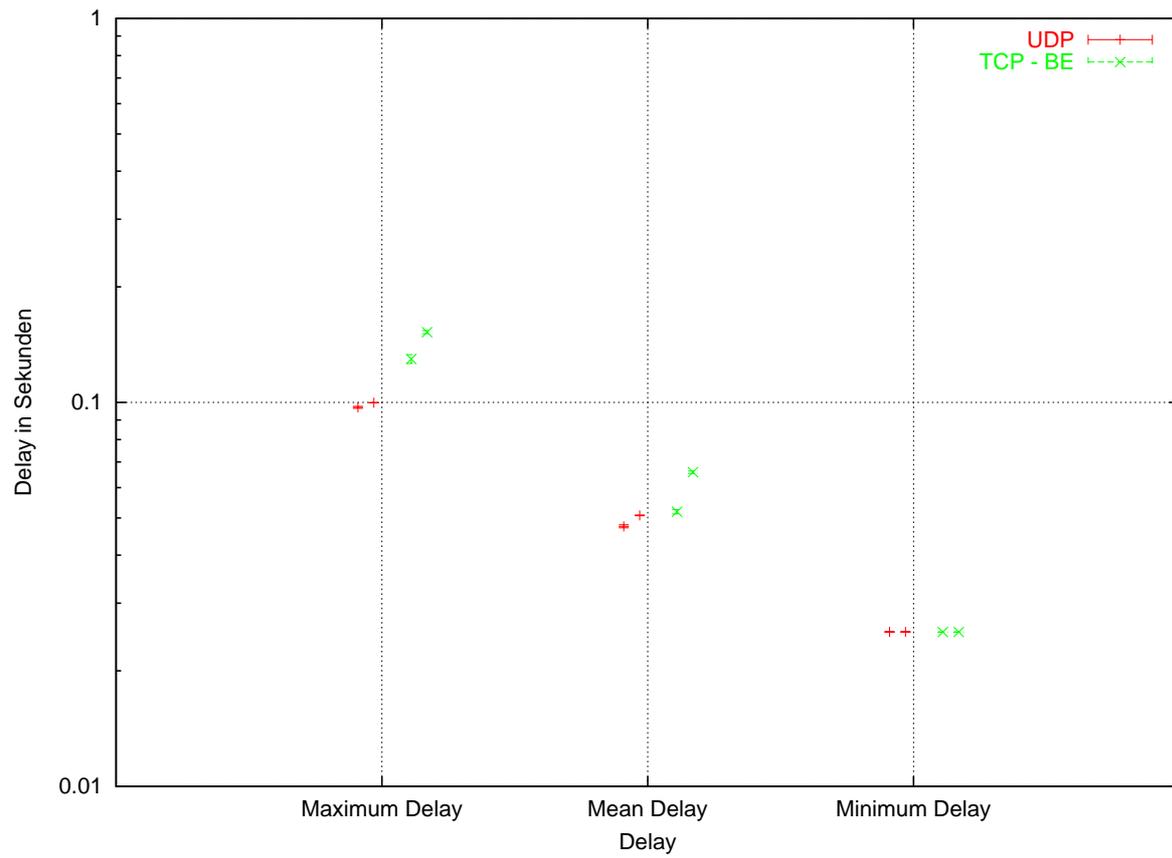
**Abbildung C.15:** BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



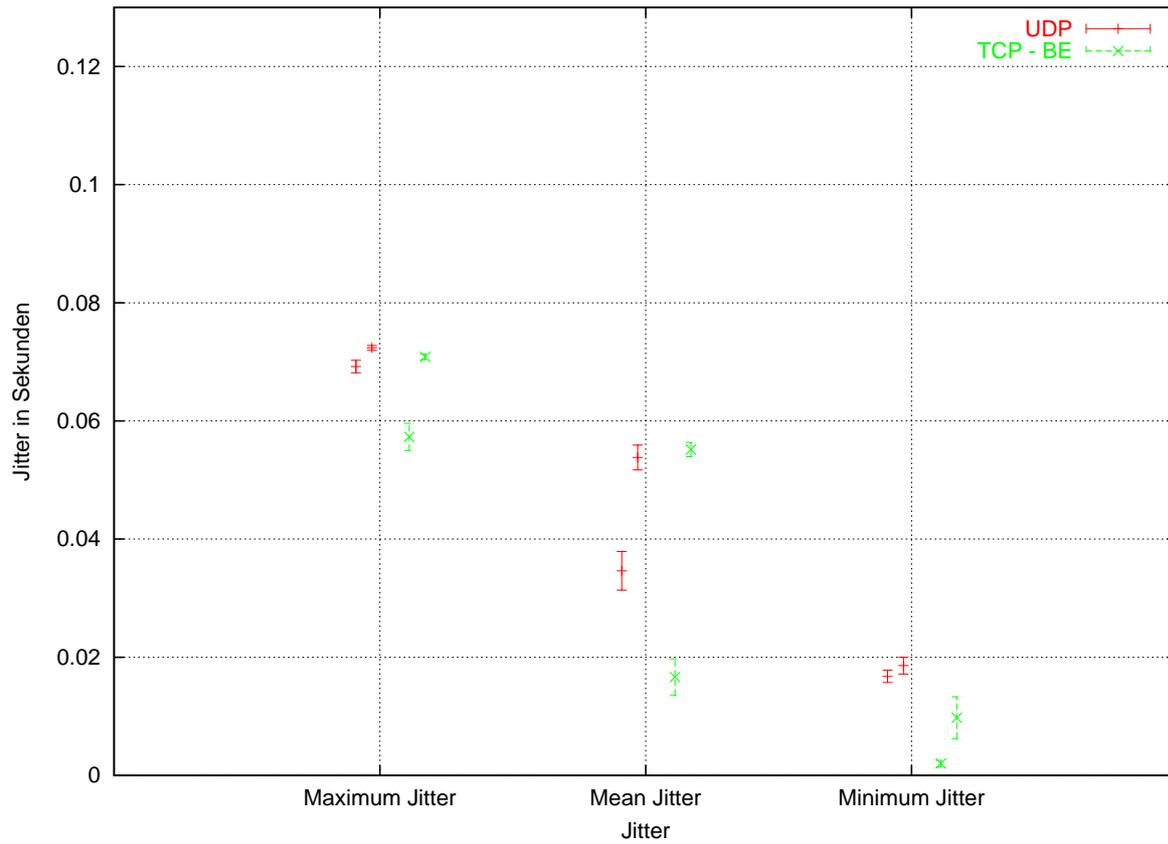
**Abbildung C.16:** BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



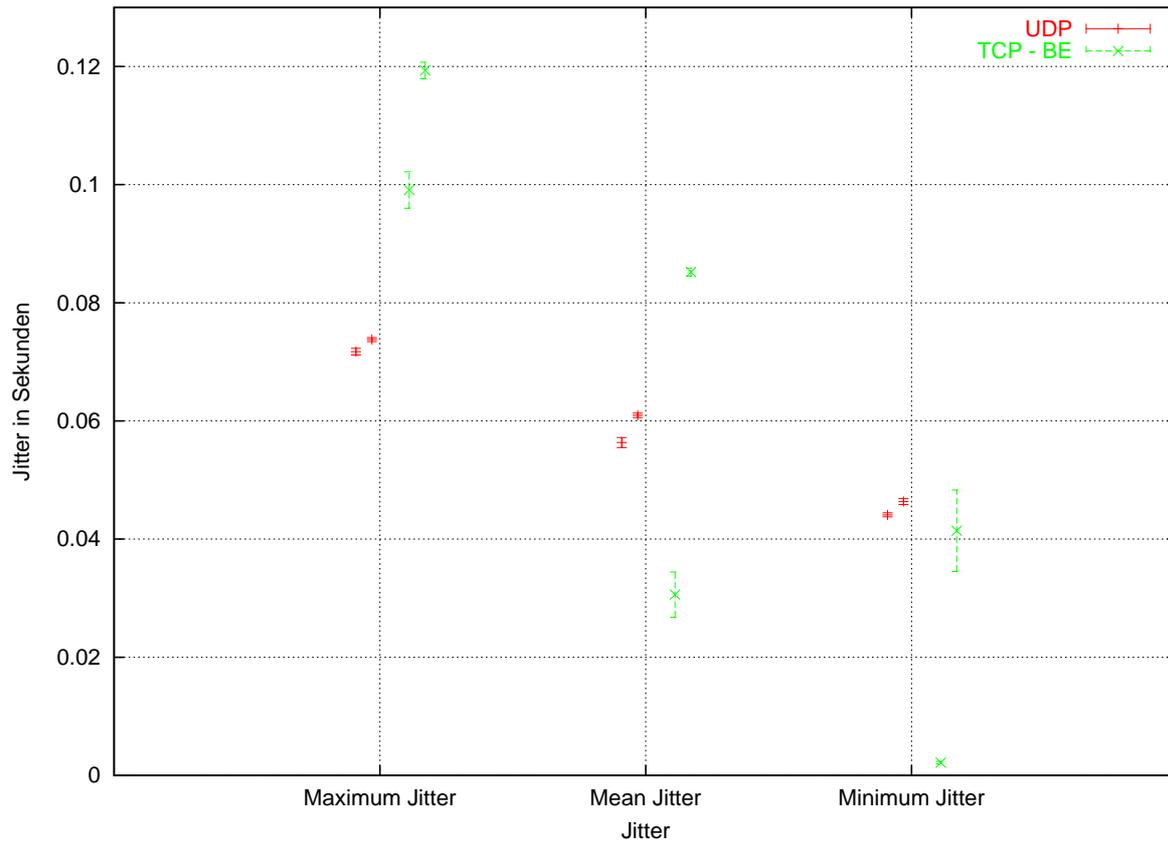
**Abbildung C.17:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



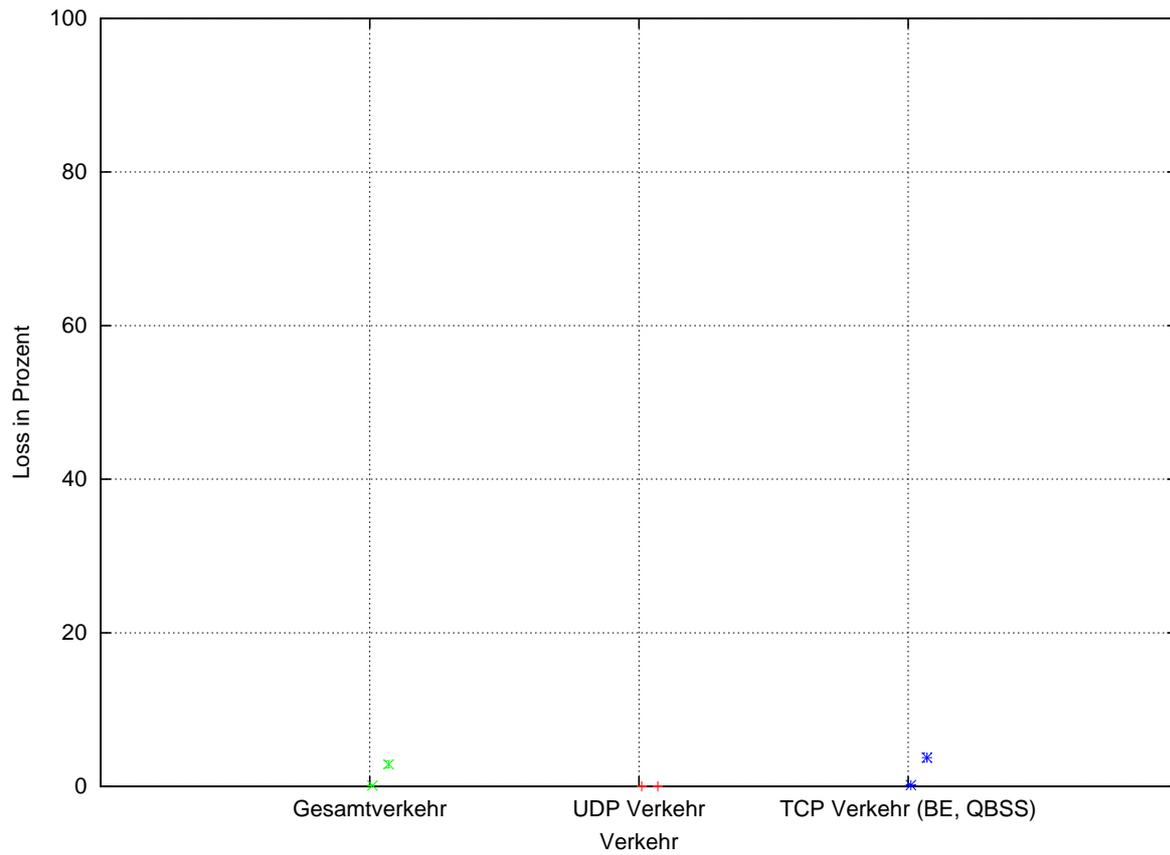
**Abbildung C.18:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



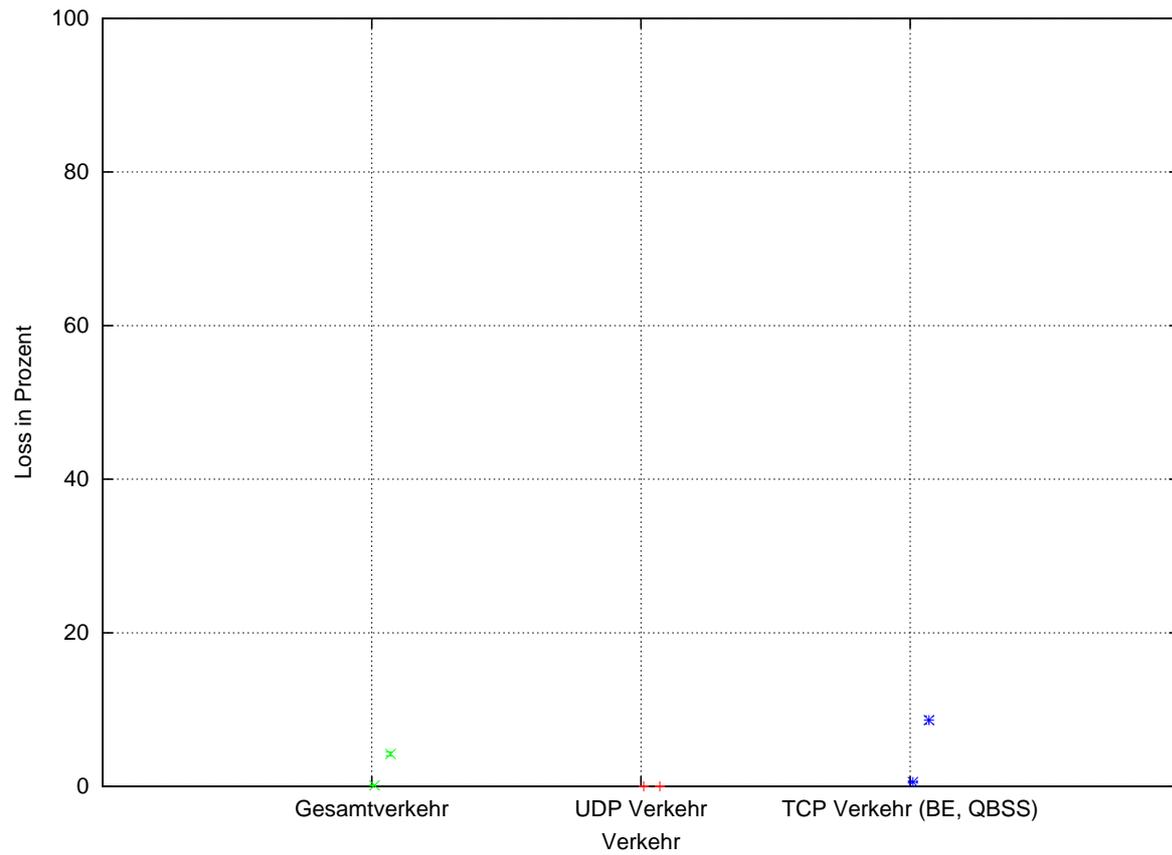
**Abbildung C.19:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



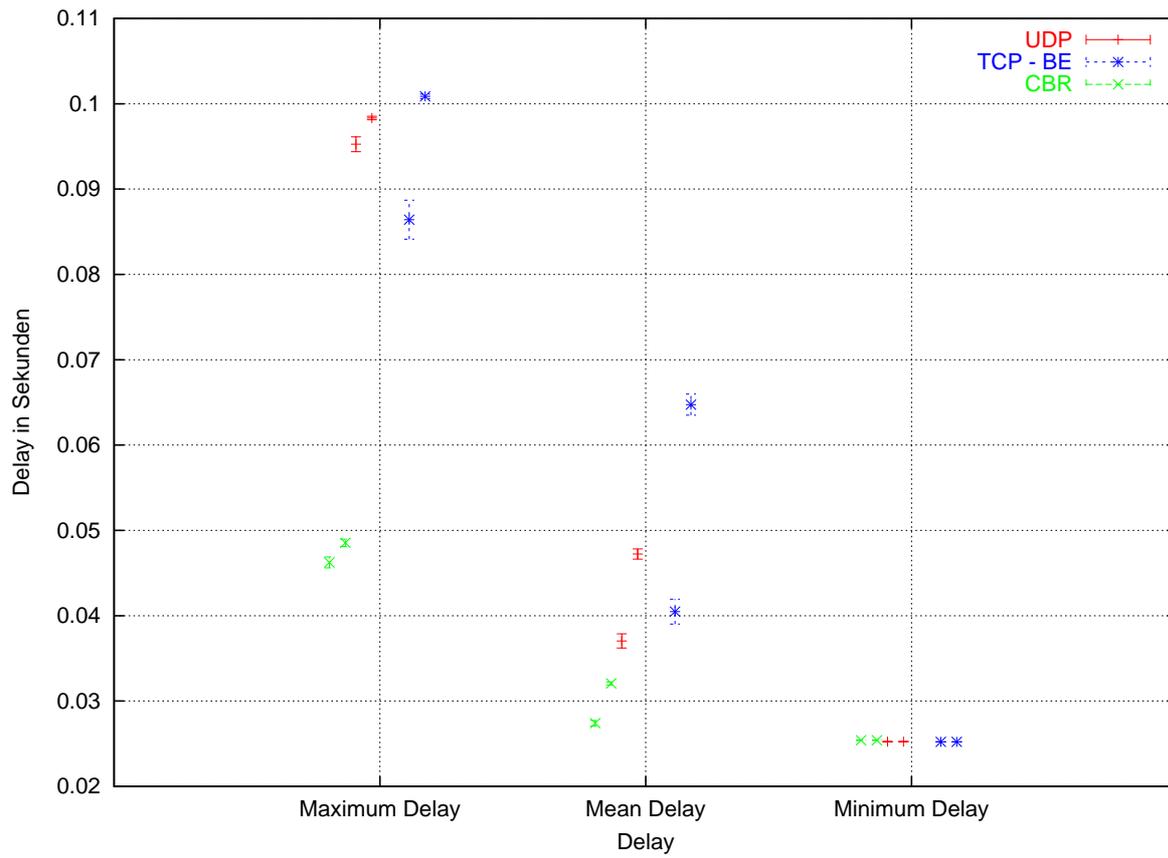
**Abbildung C.20:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



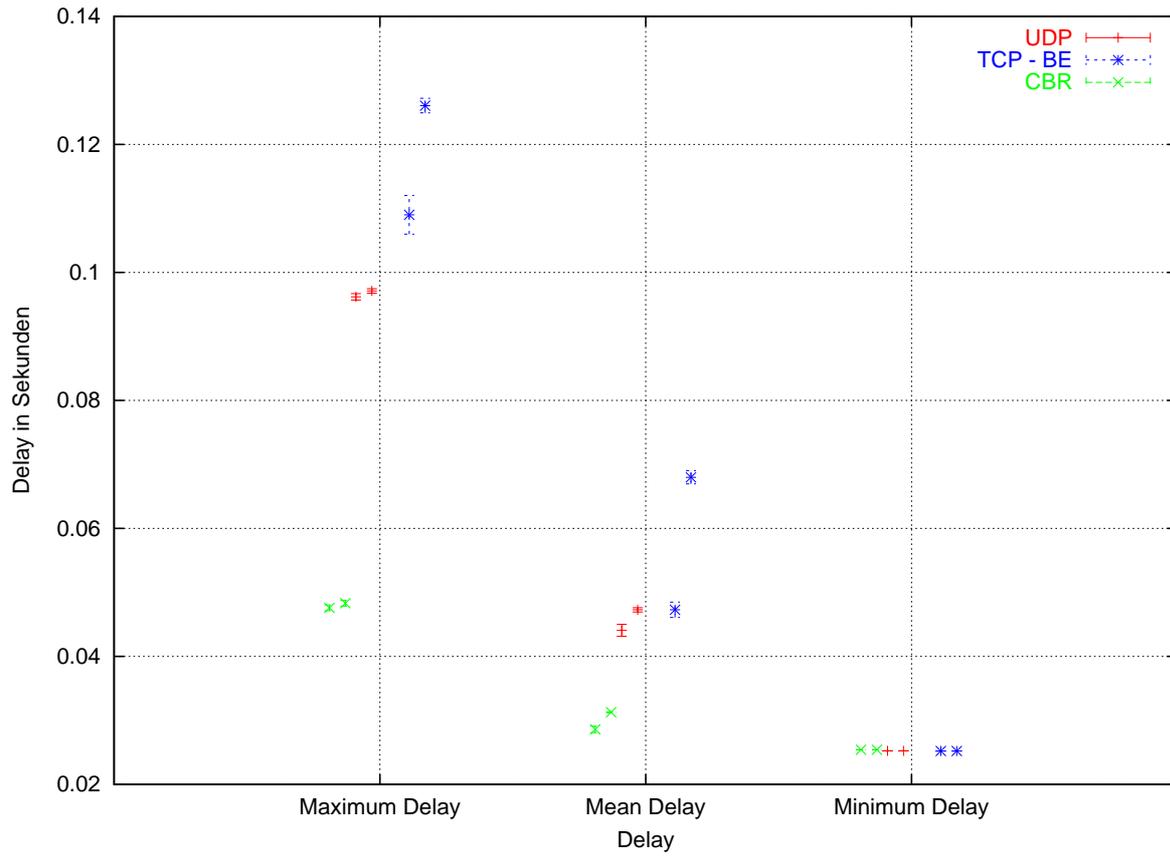
**Abbildung C.21:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



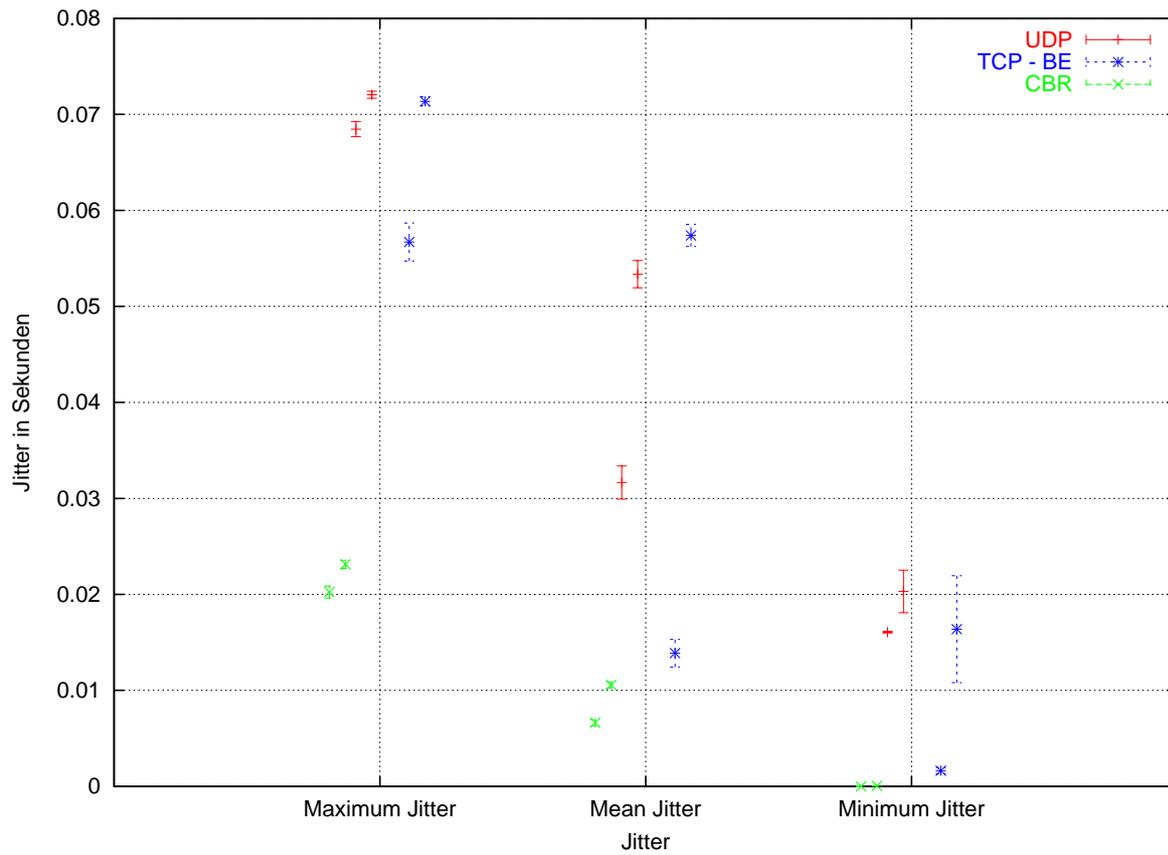
**Abbildung C.22:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



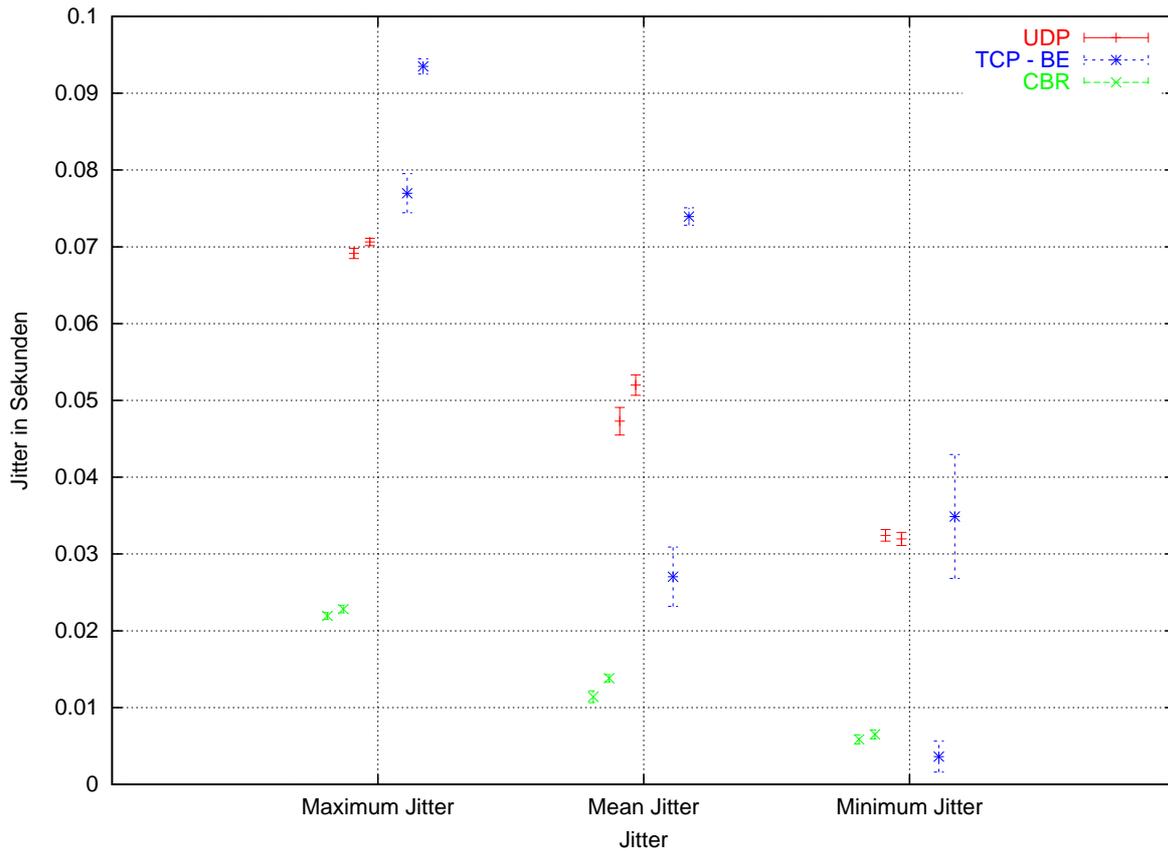
**Abbildung C.23:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



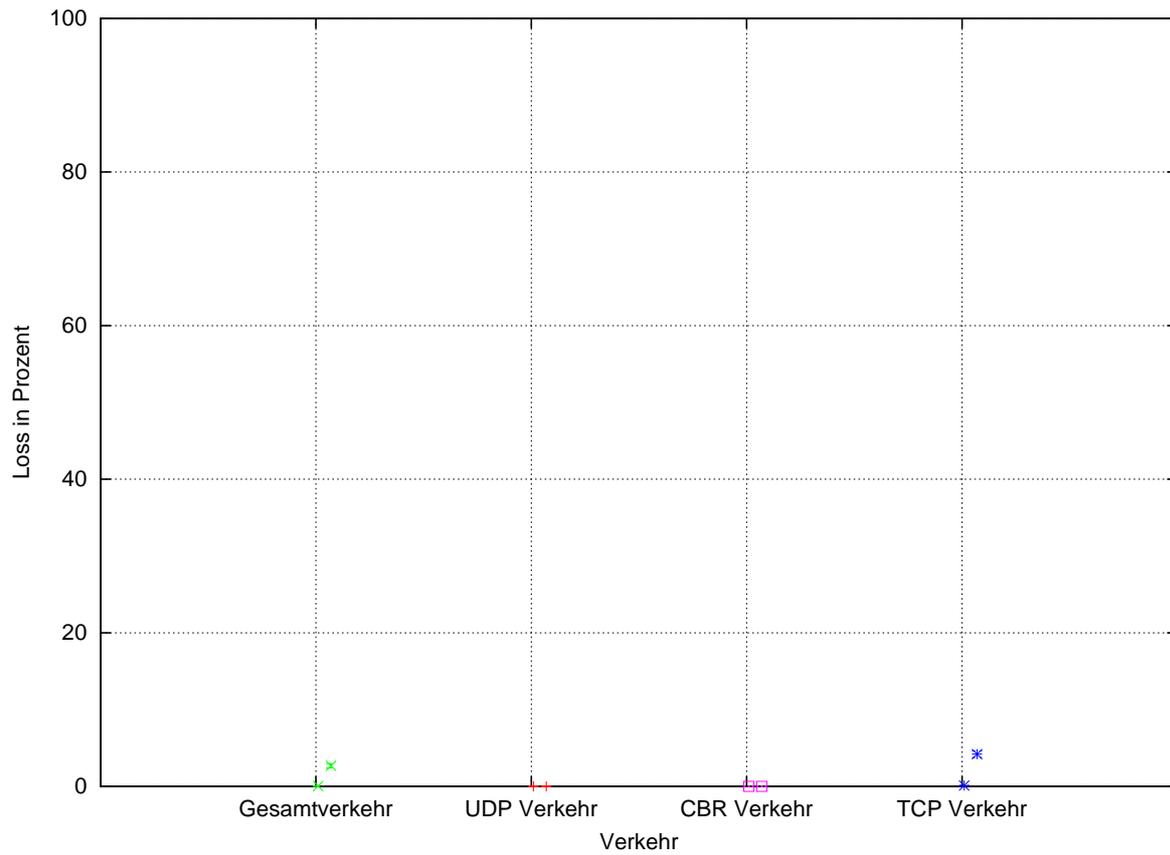
**Abbildung C.24:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



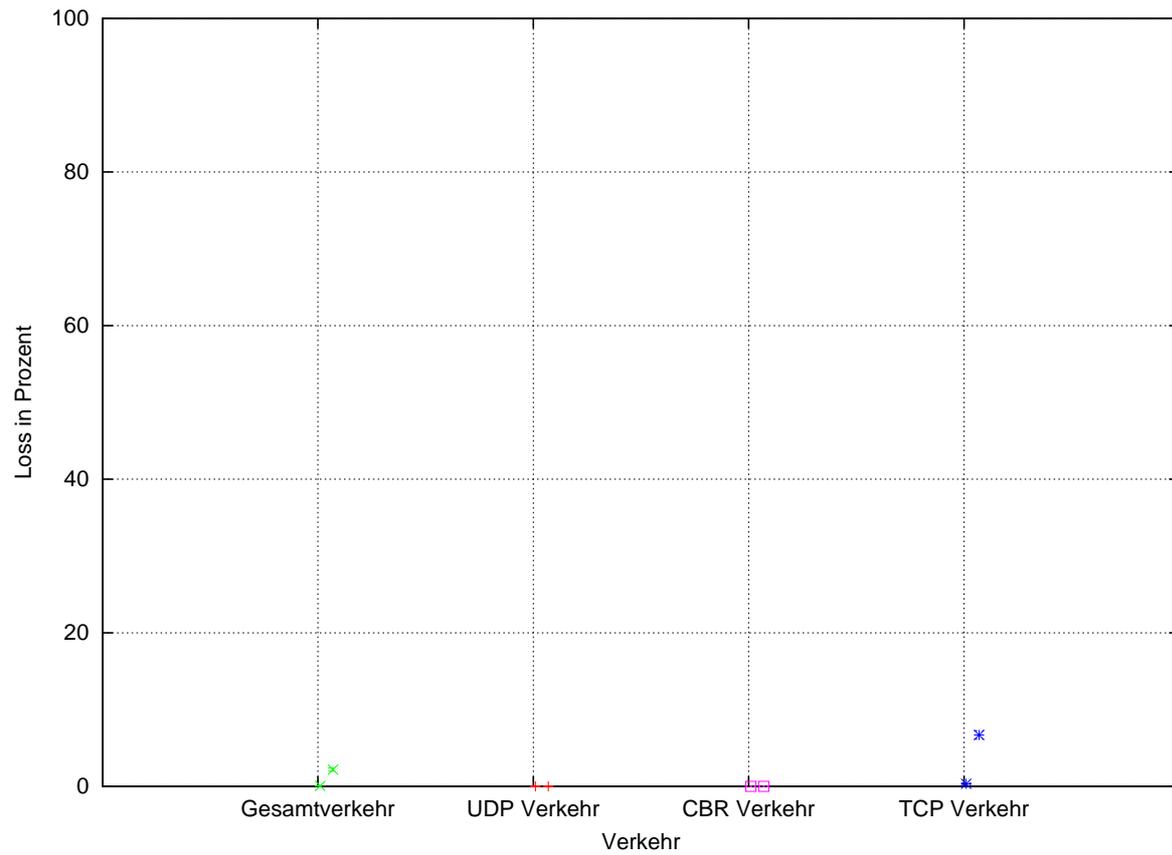
**Abbildung C.25:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



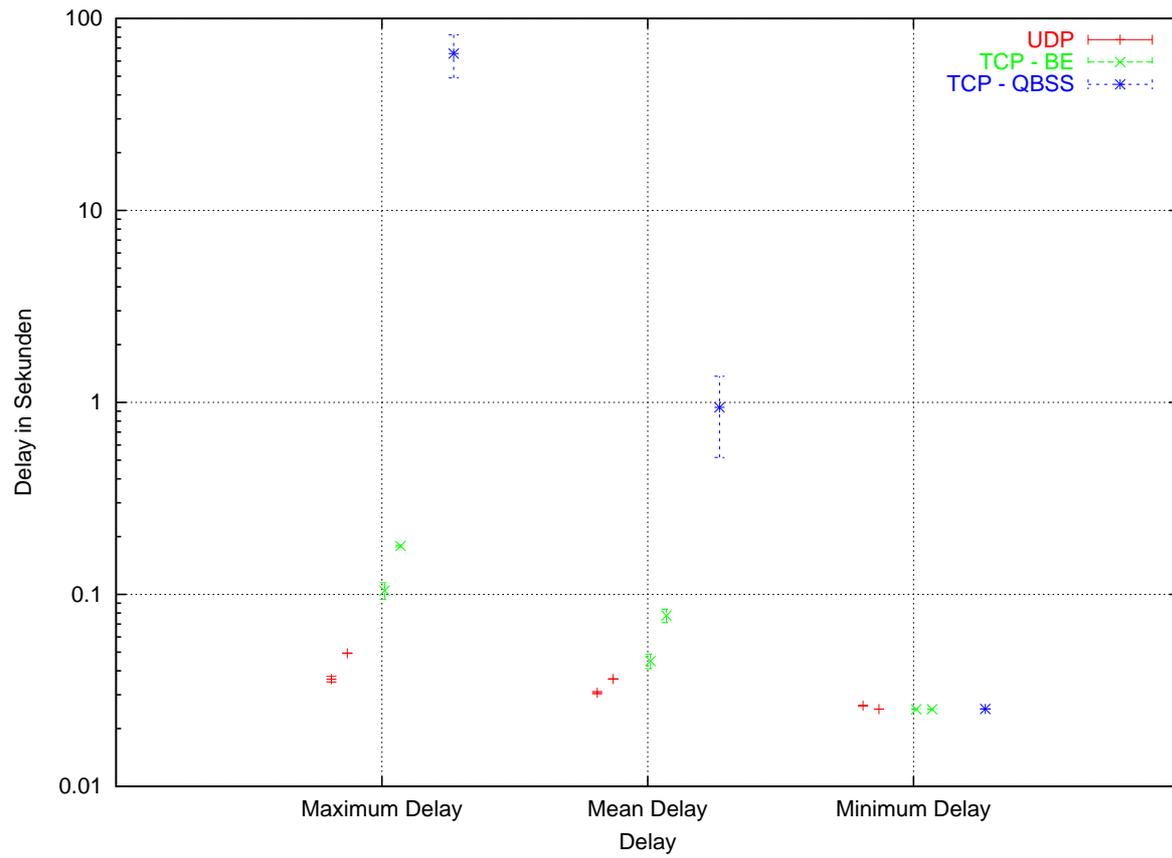
**Abbildung C.26:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



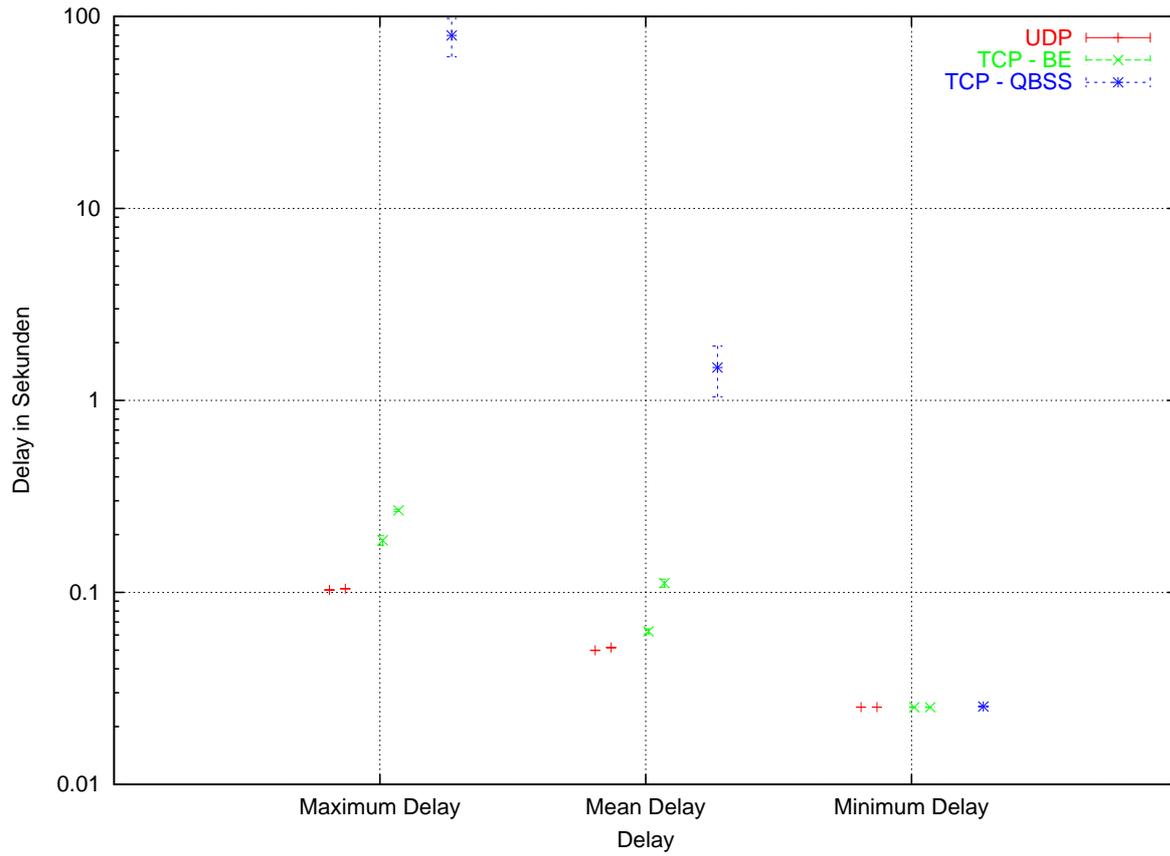
**Abbildung C.27:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



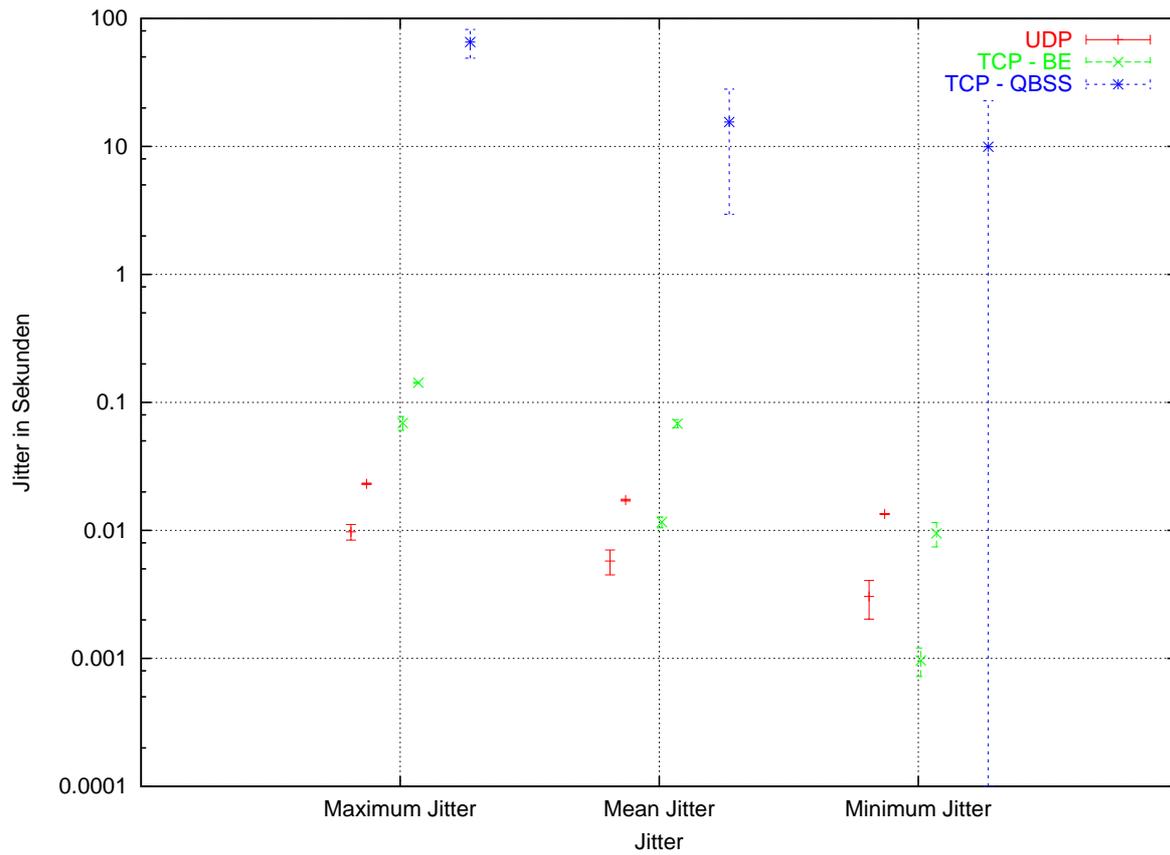
**Abbildung C.28:** IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



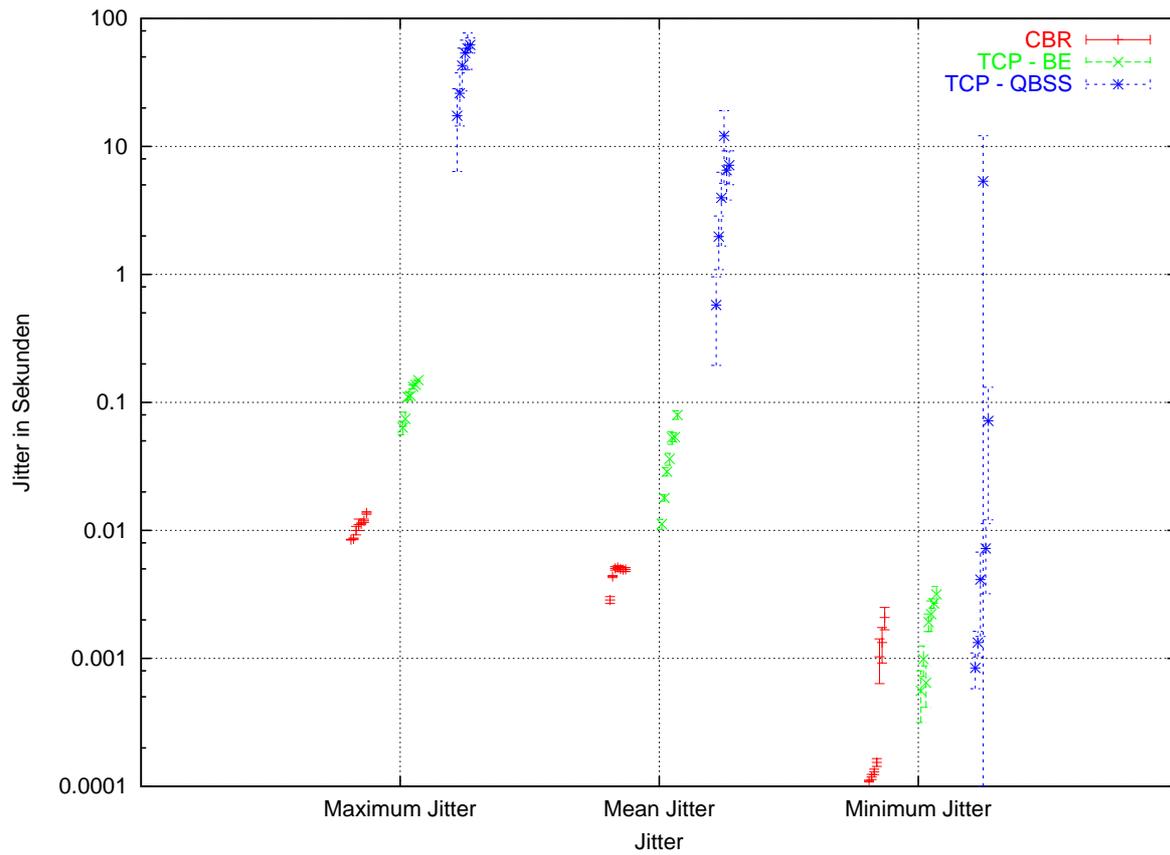
**Abbildung C.29:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



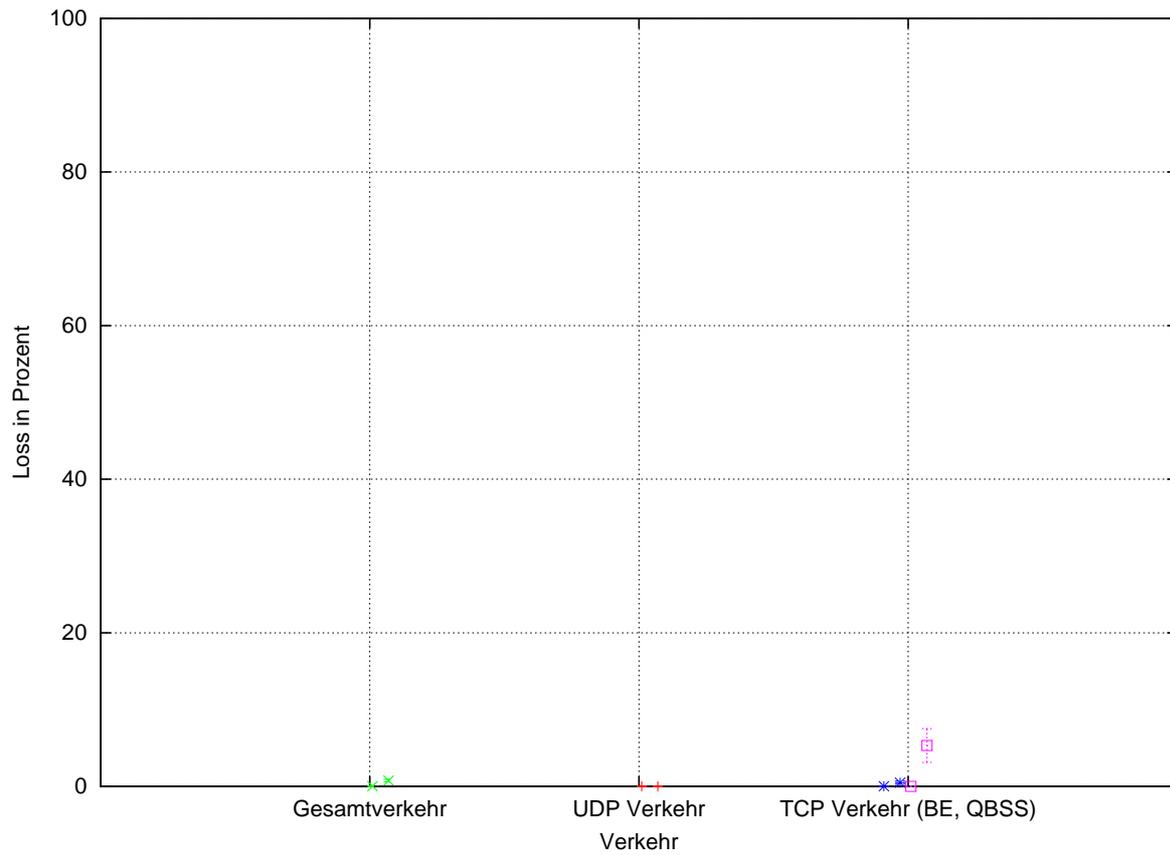
**Abbildung C.30:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



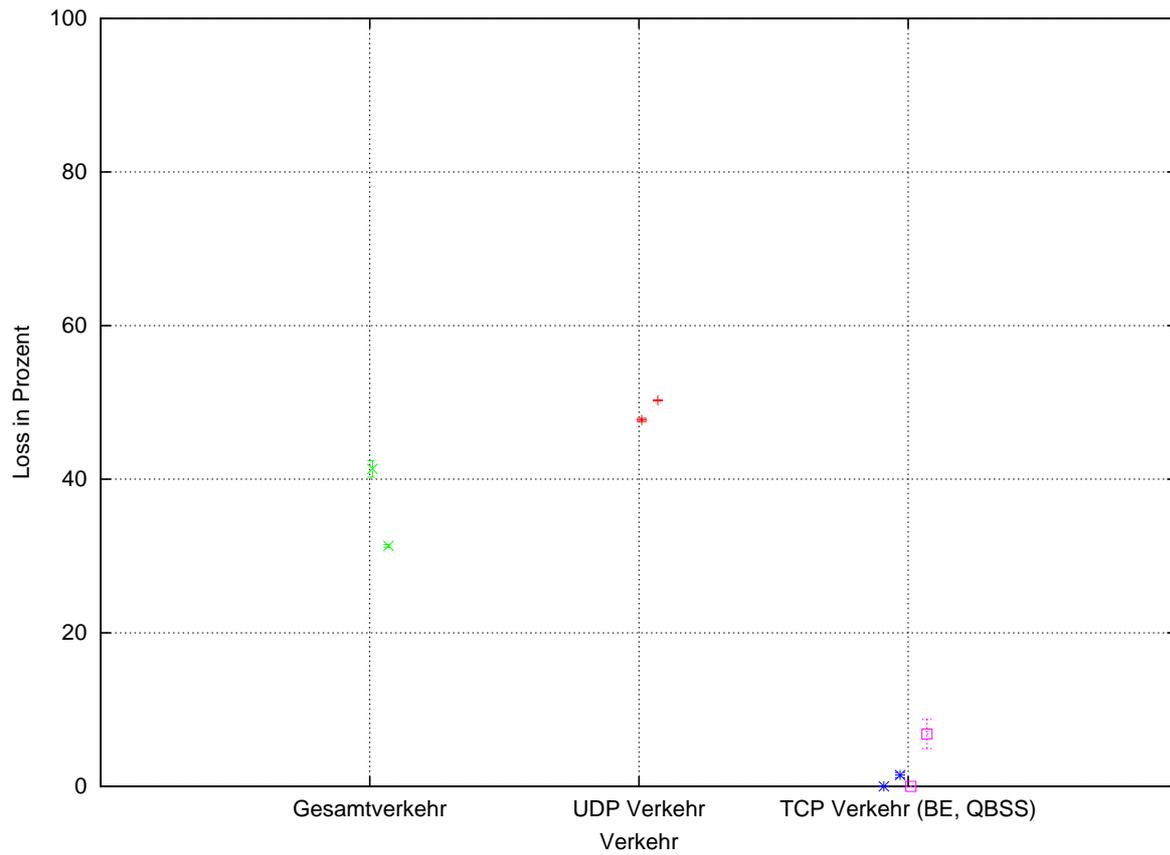
**Abbildung C.31:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



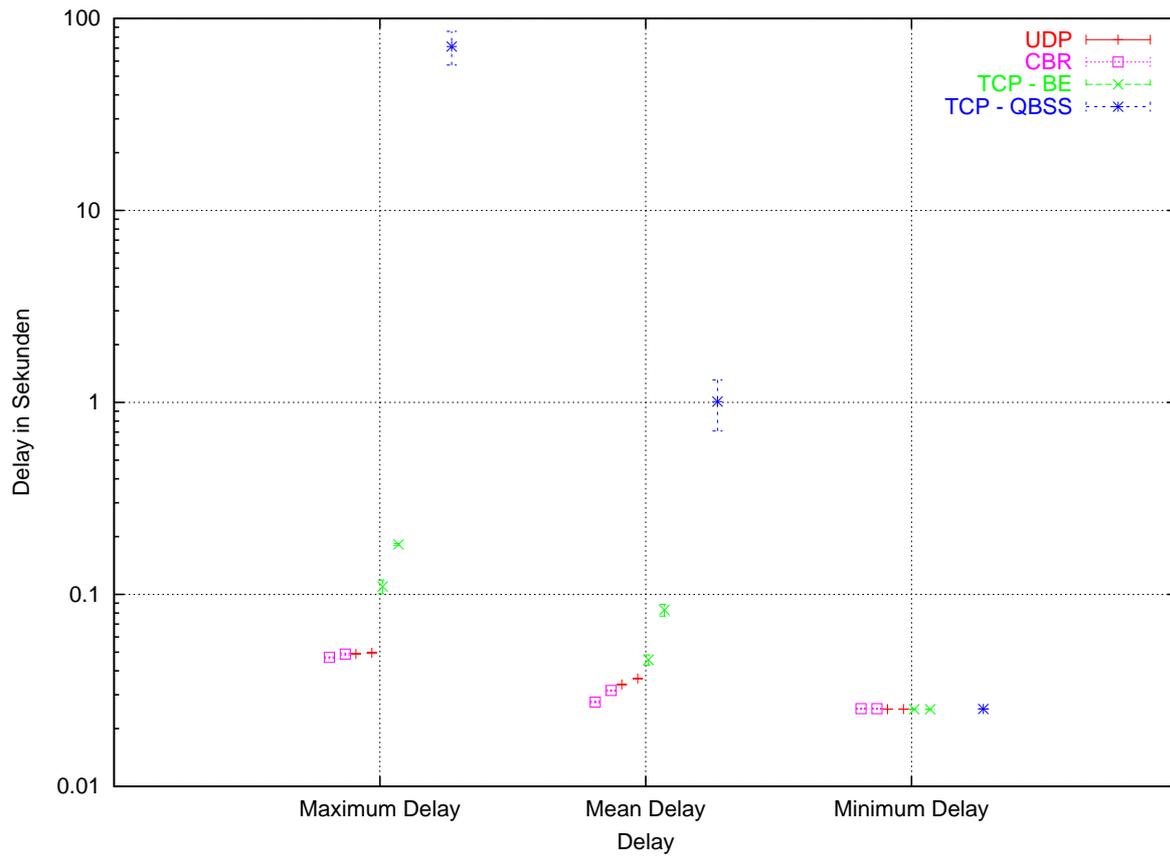
**Abbildung C.32:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,2, ..., 7)



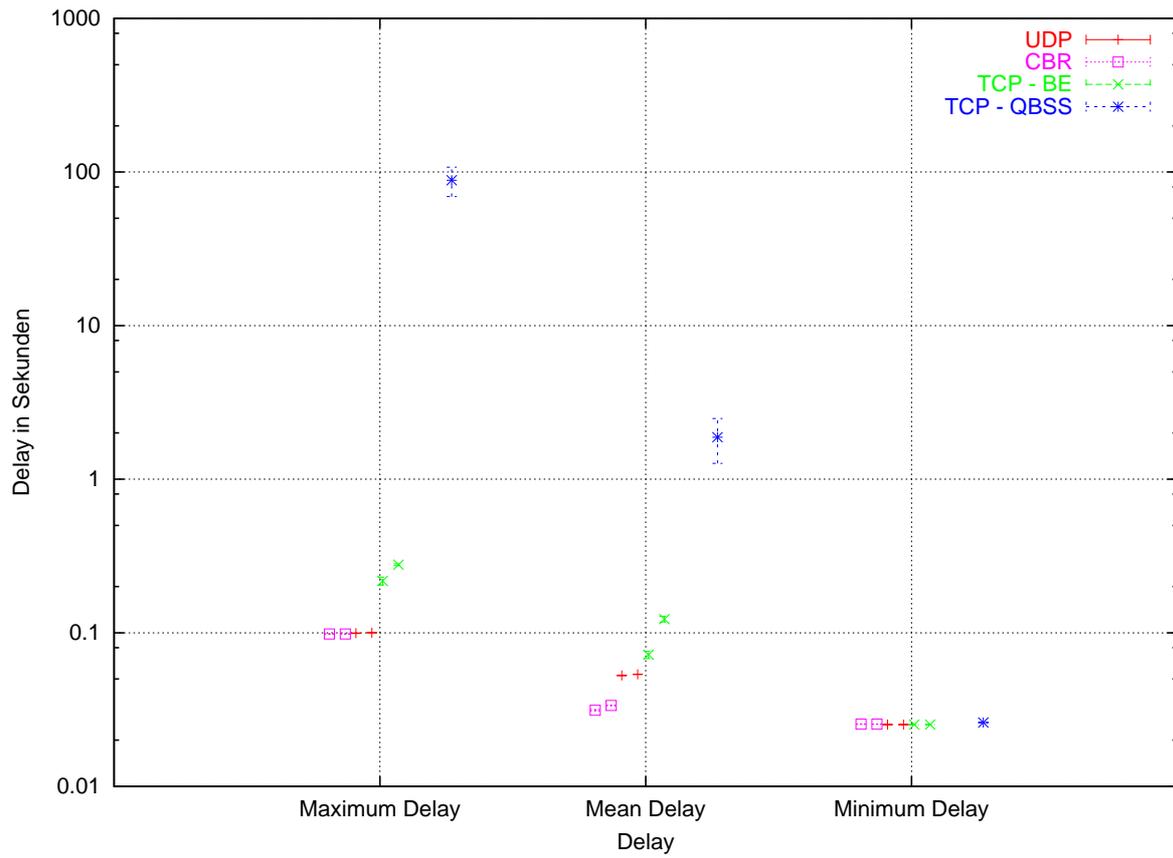
**Abbildung C.33:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



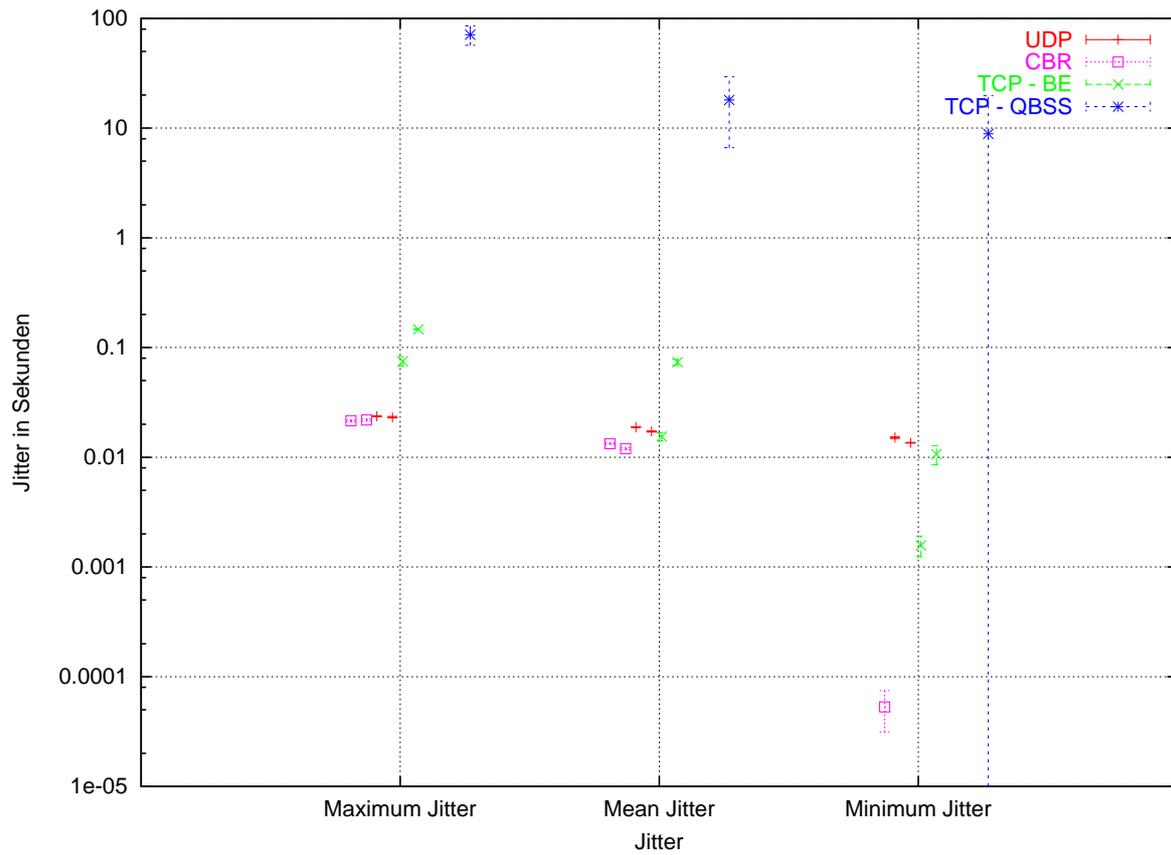
**Abbildung C.34:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



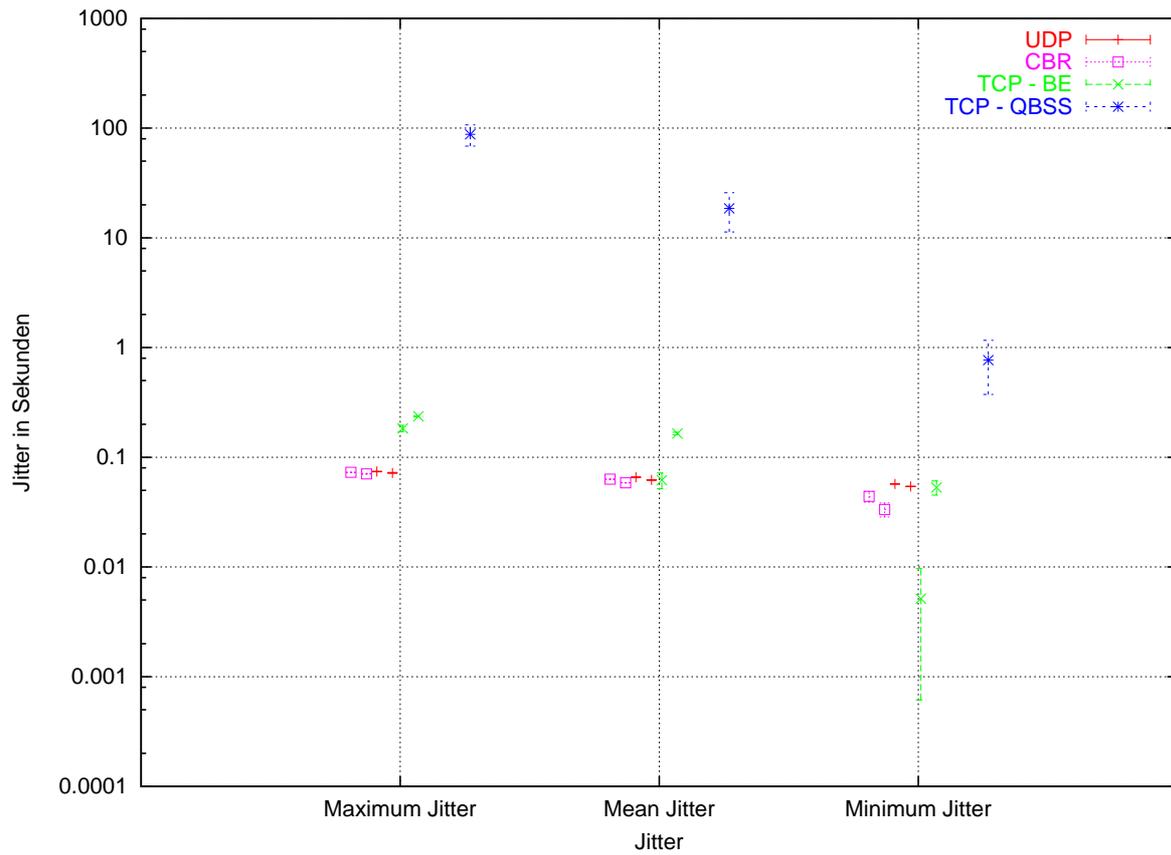
**Abbildung C.35:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



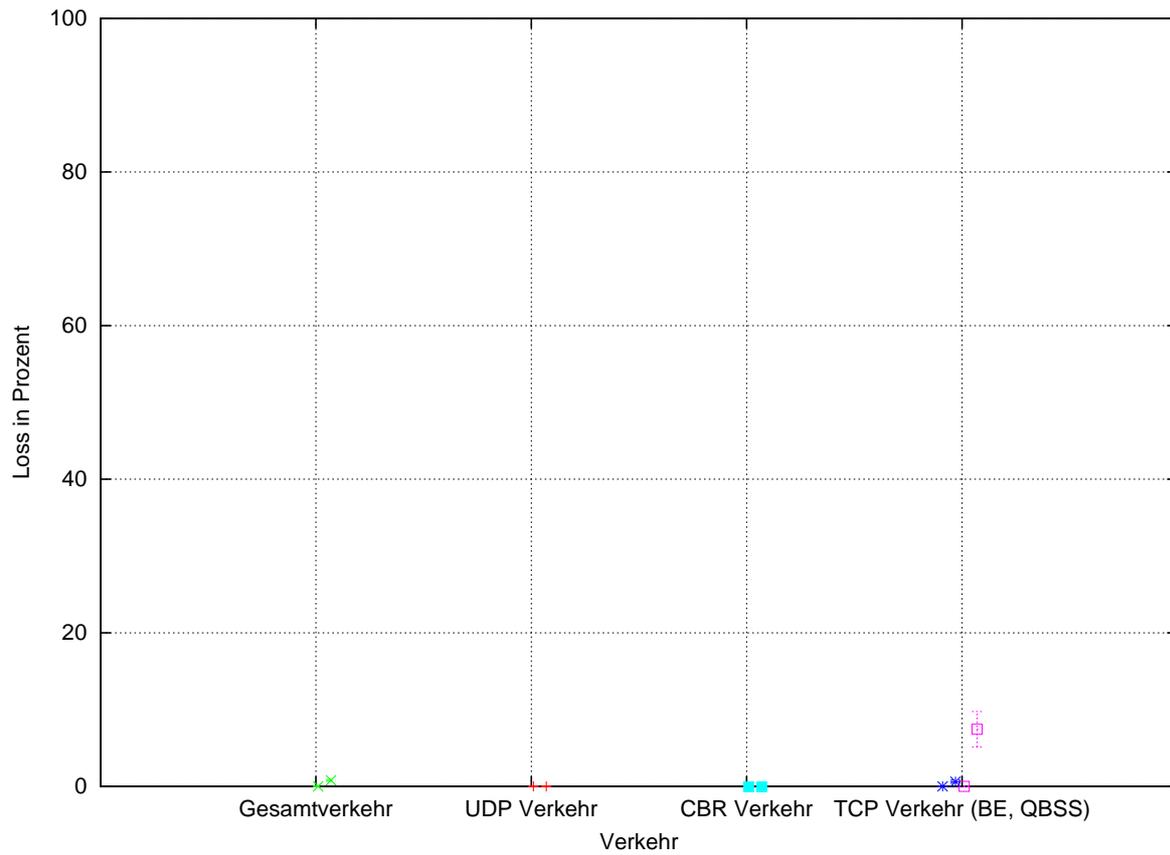
**Abbildung C.36:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



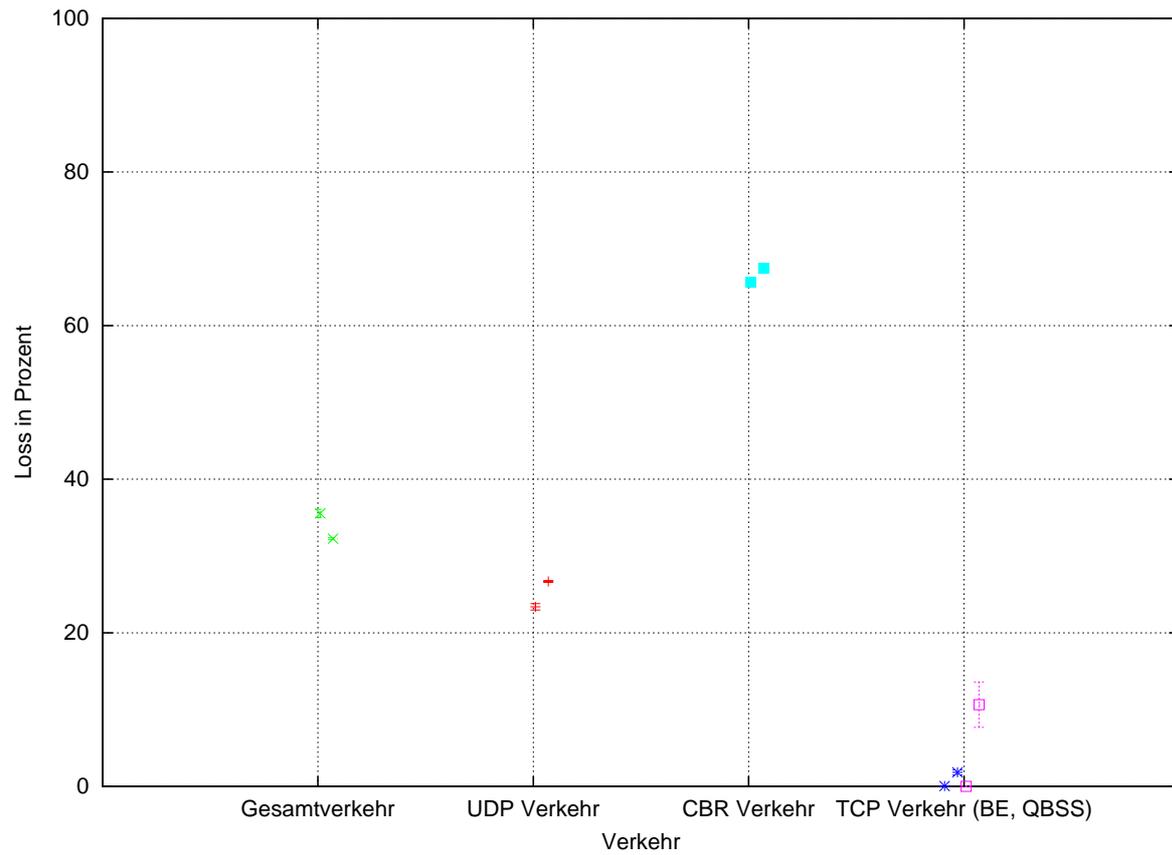
**Abbildung C.37:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



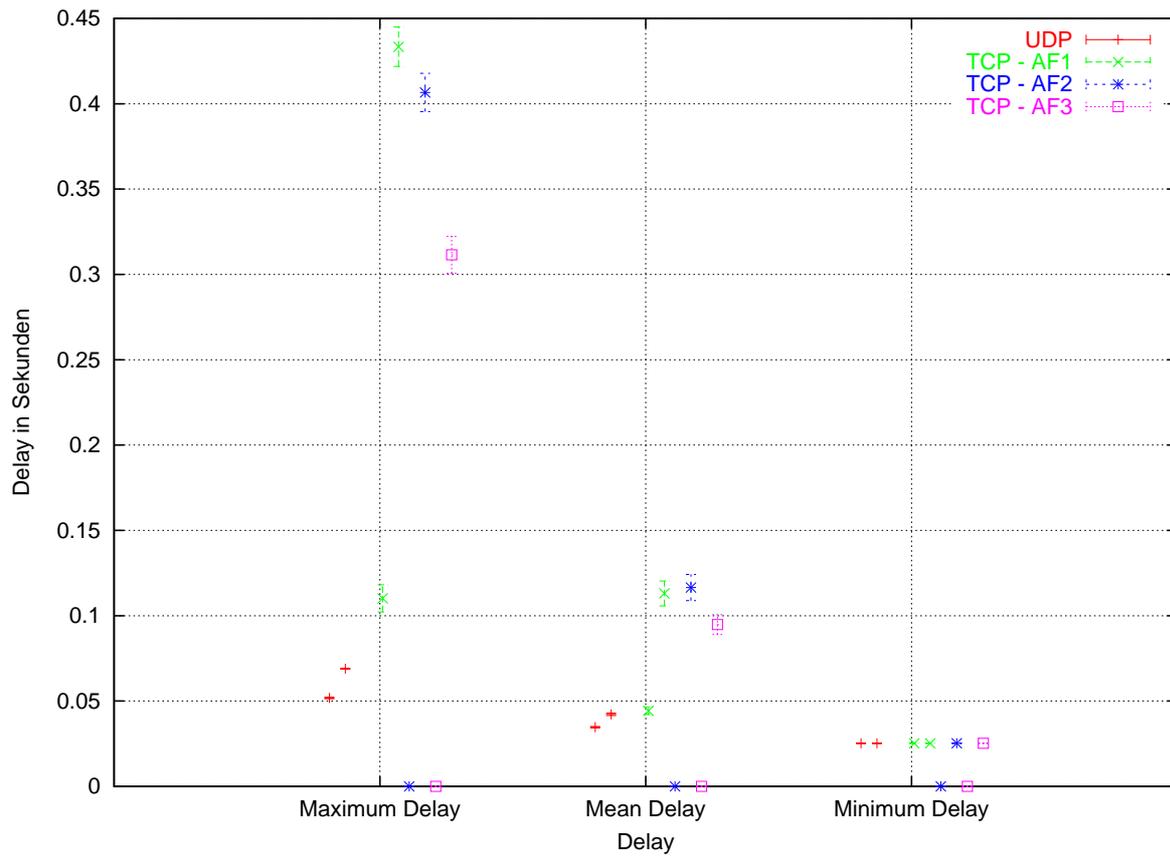
**Abbildung C.38:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



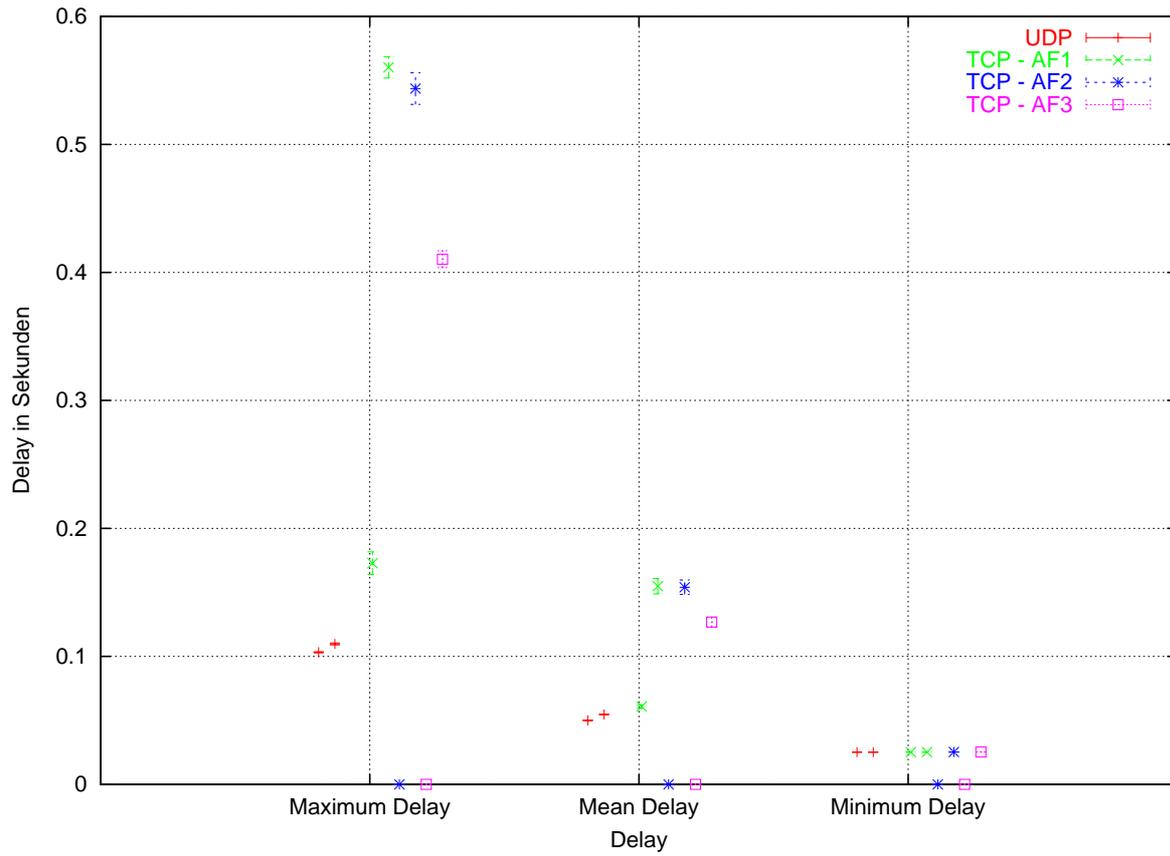
**Abbildung C.39:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



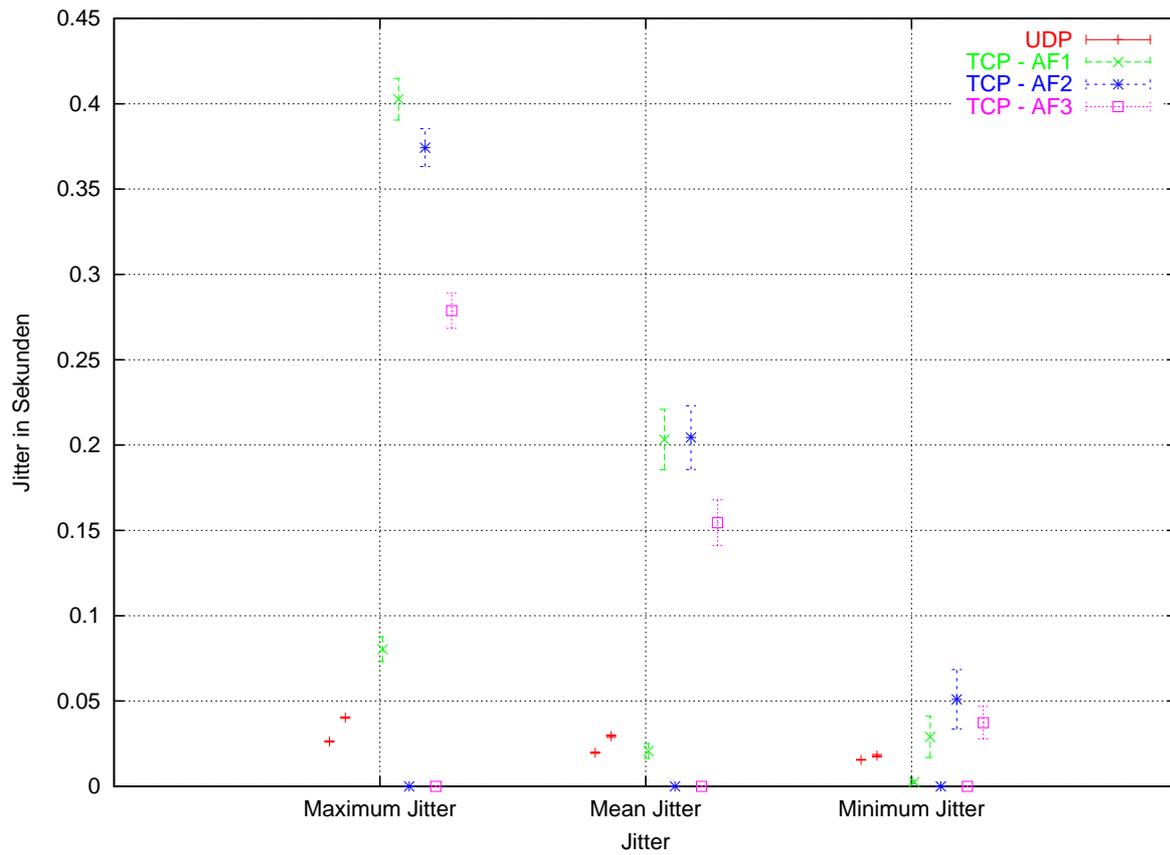
**Abbildung C.40:** DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



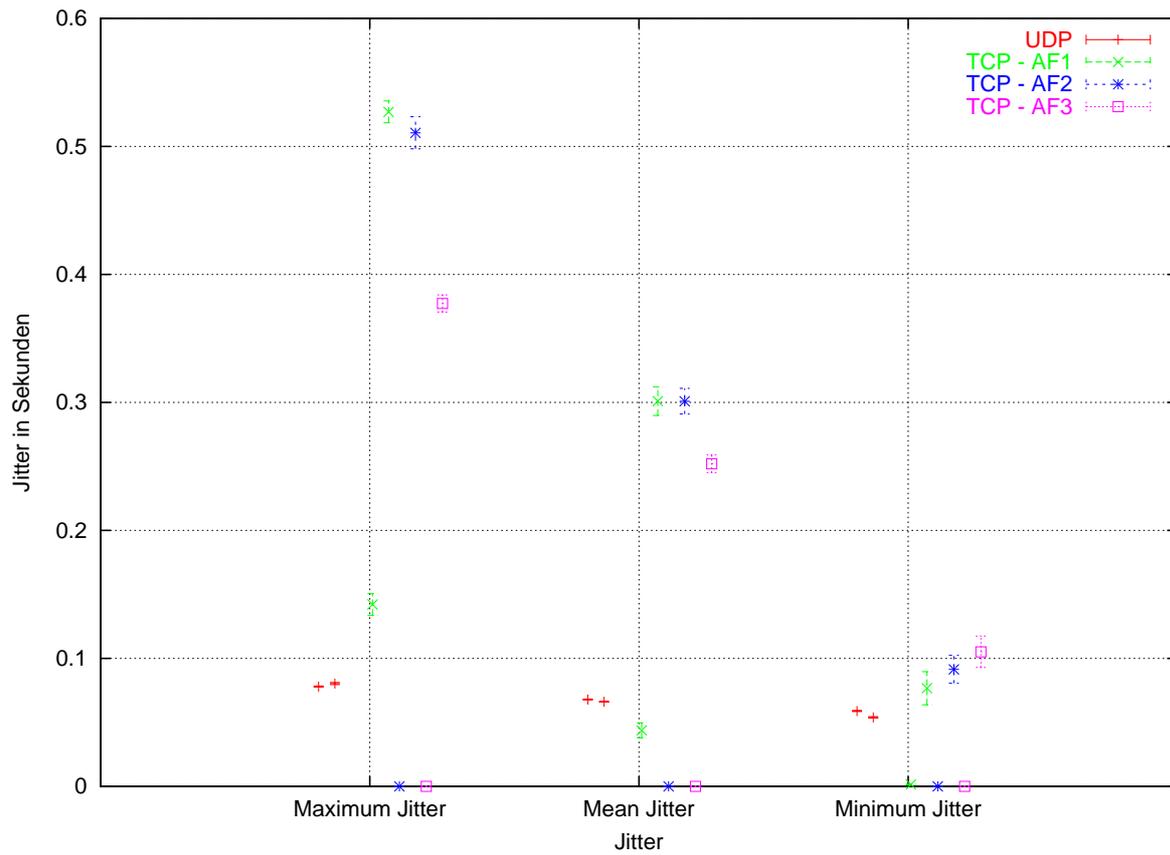
**Abbildung C.41:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



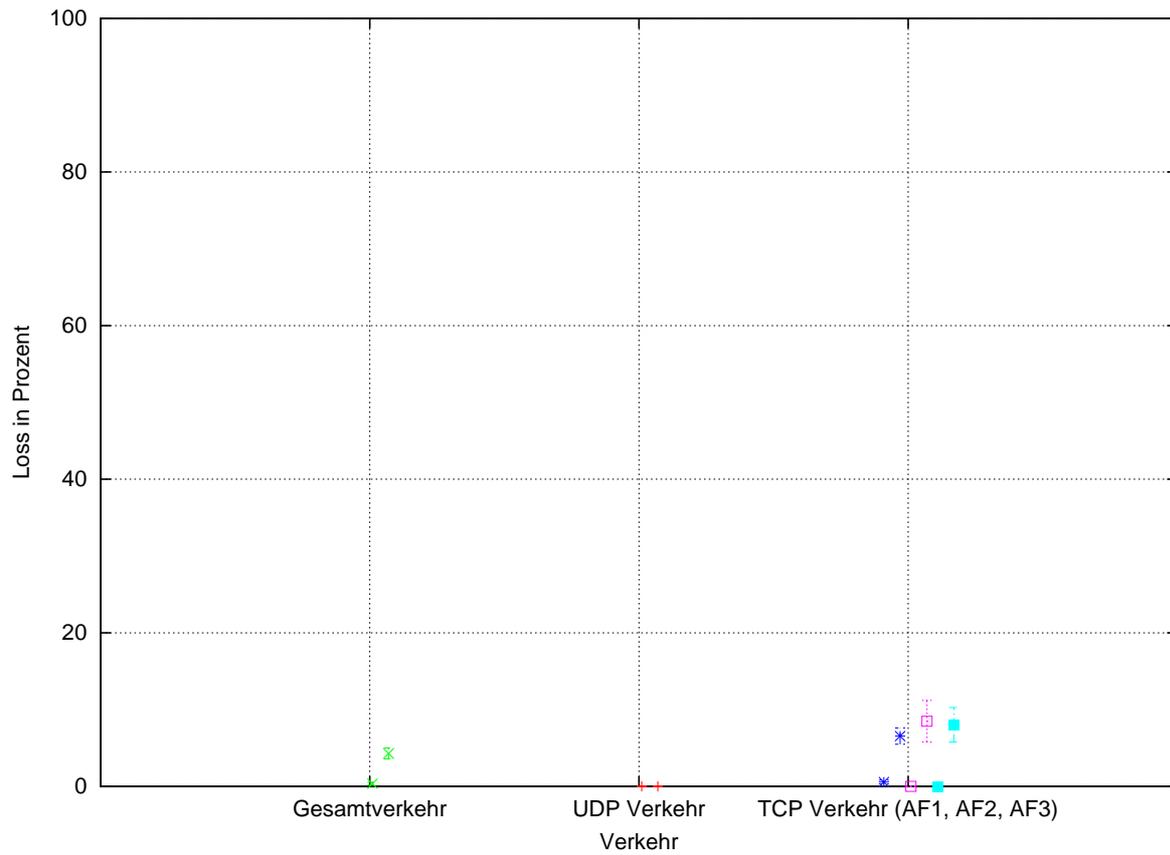
**Abbildung C.42:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



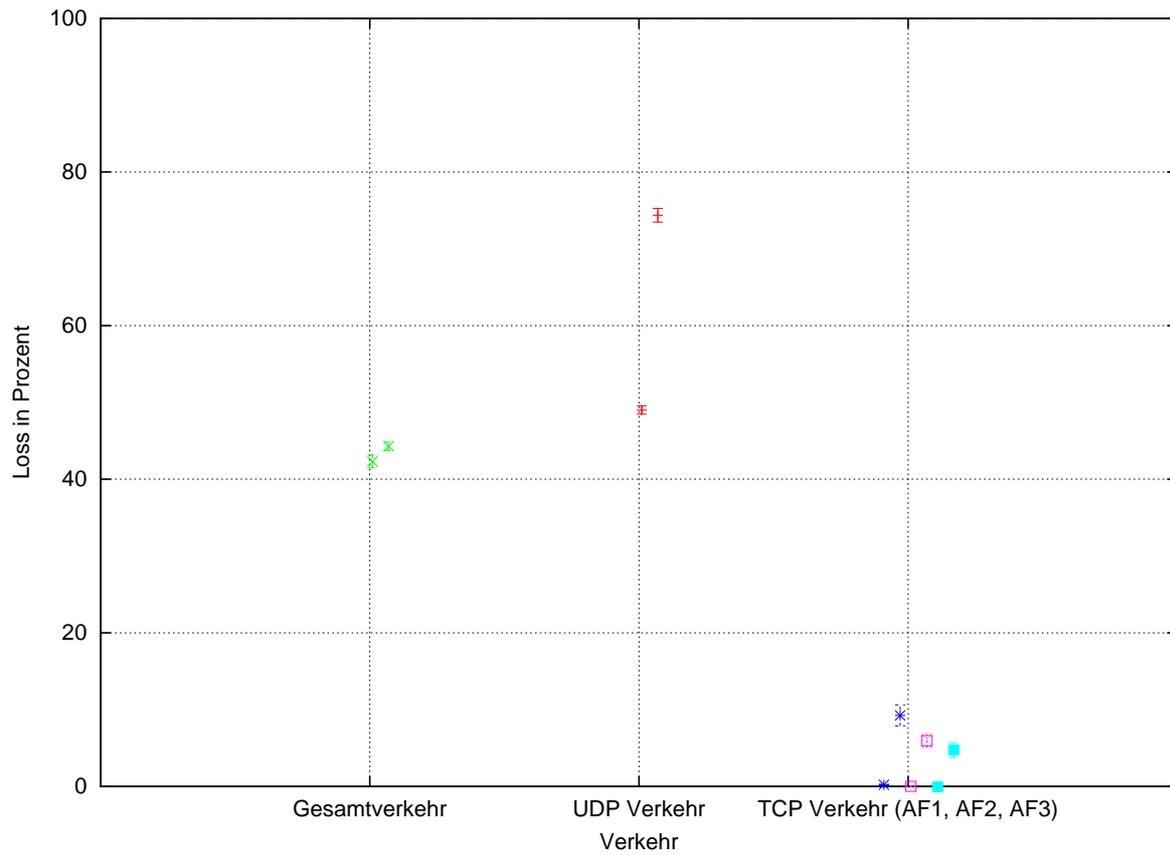
**Abbildung C.43:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



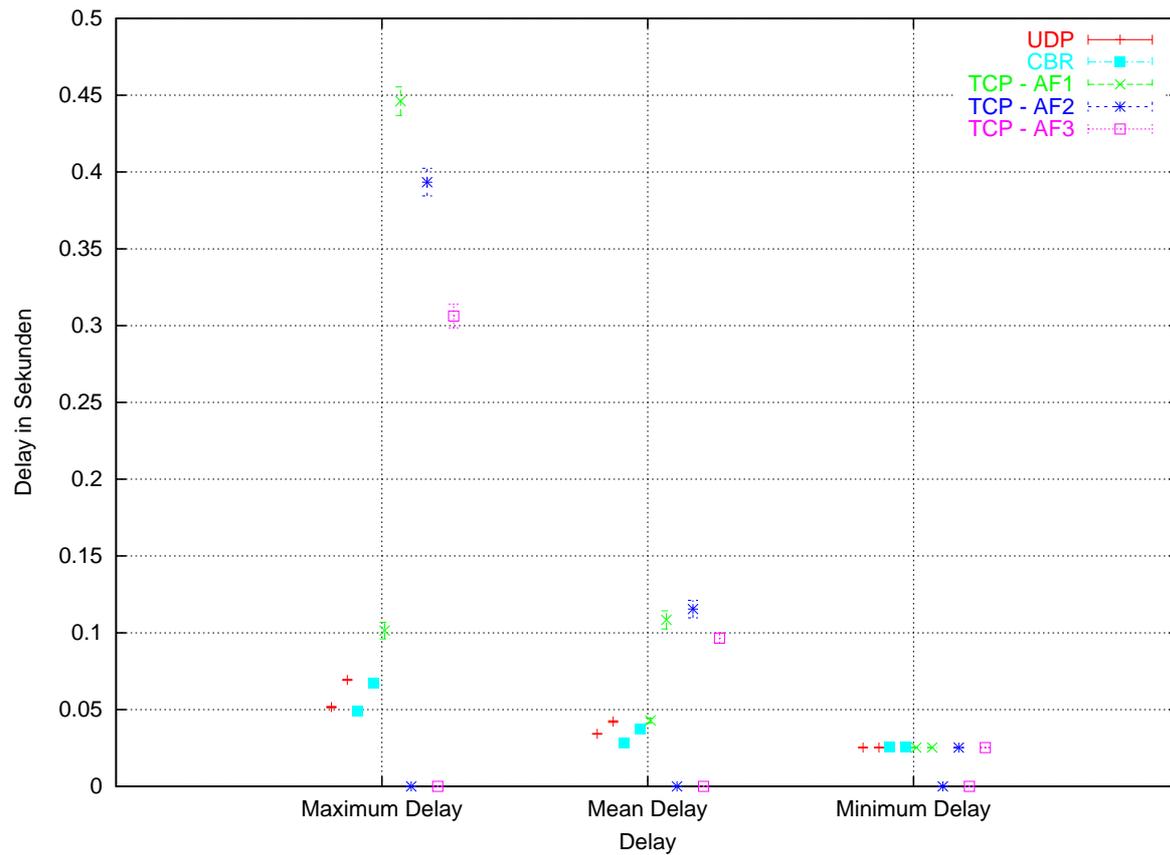
**Abbildung C.44:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



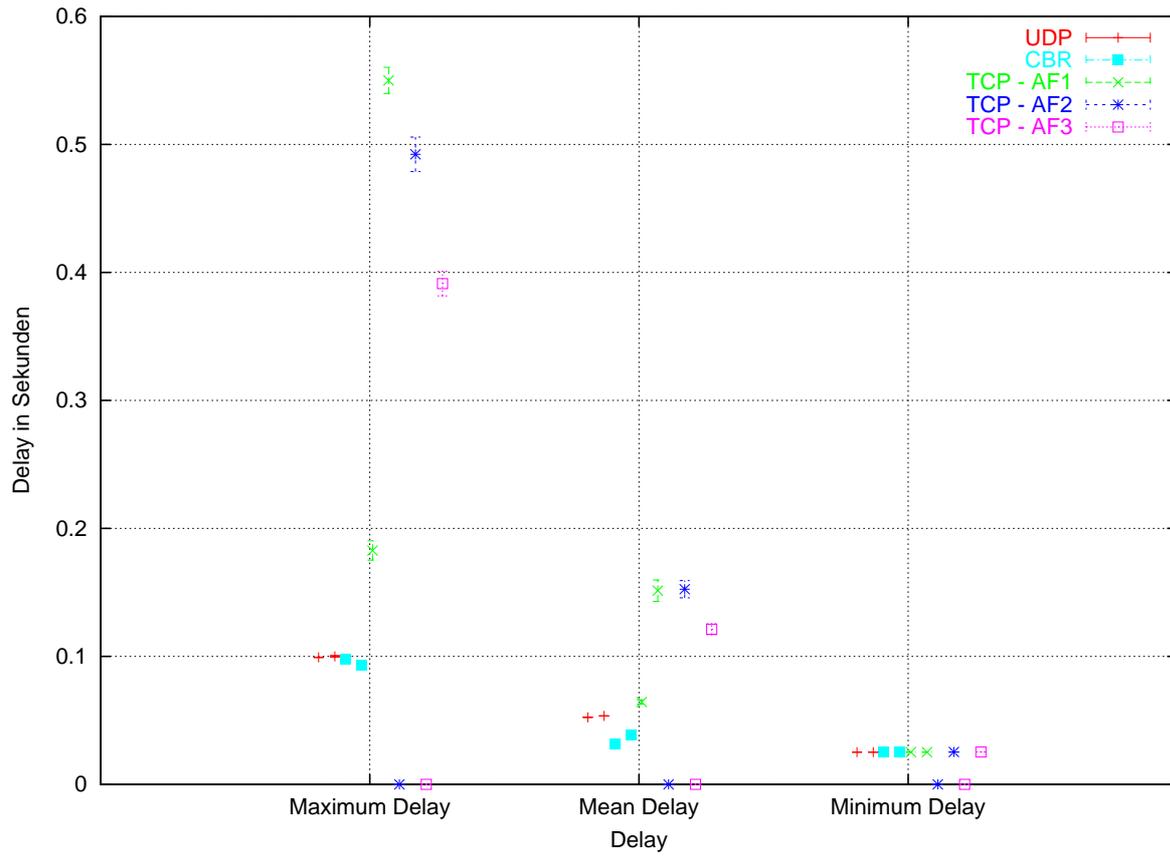
**Abbildung C.45:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7)



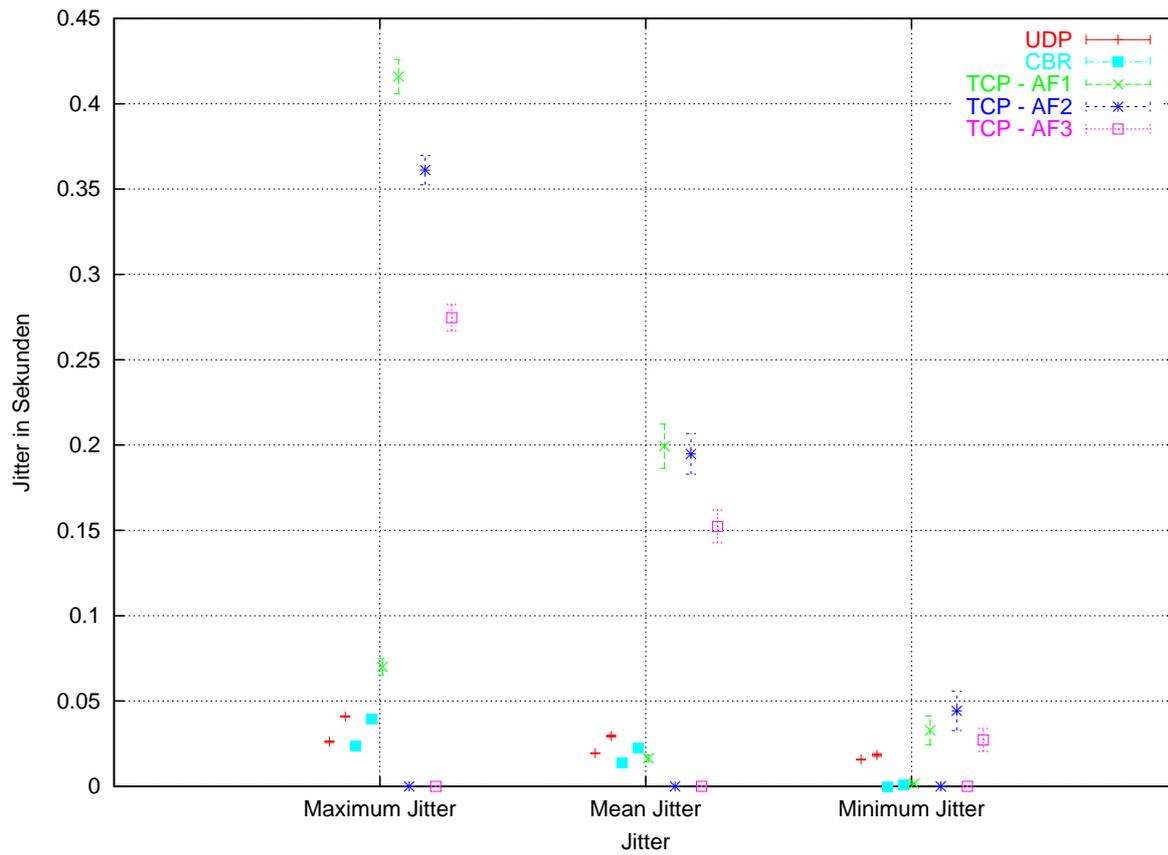
**Abbildung C.46:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7)



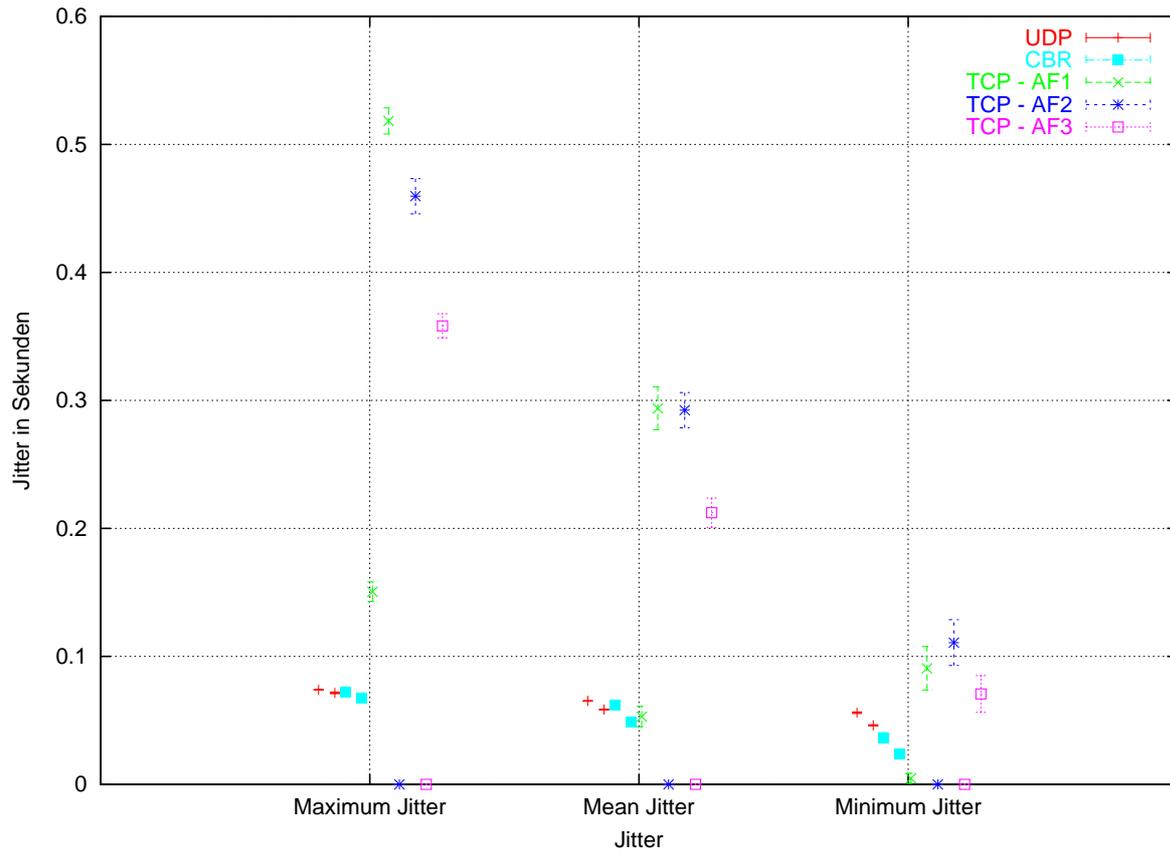
**Abbildung C.47:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



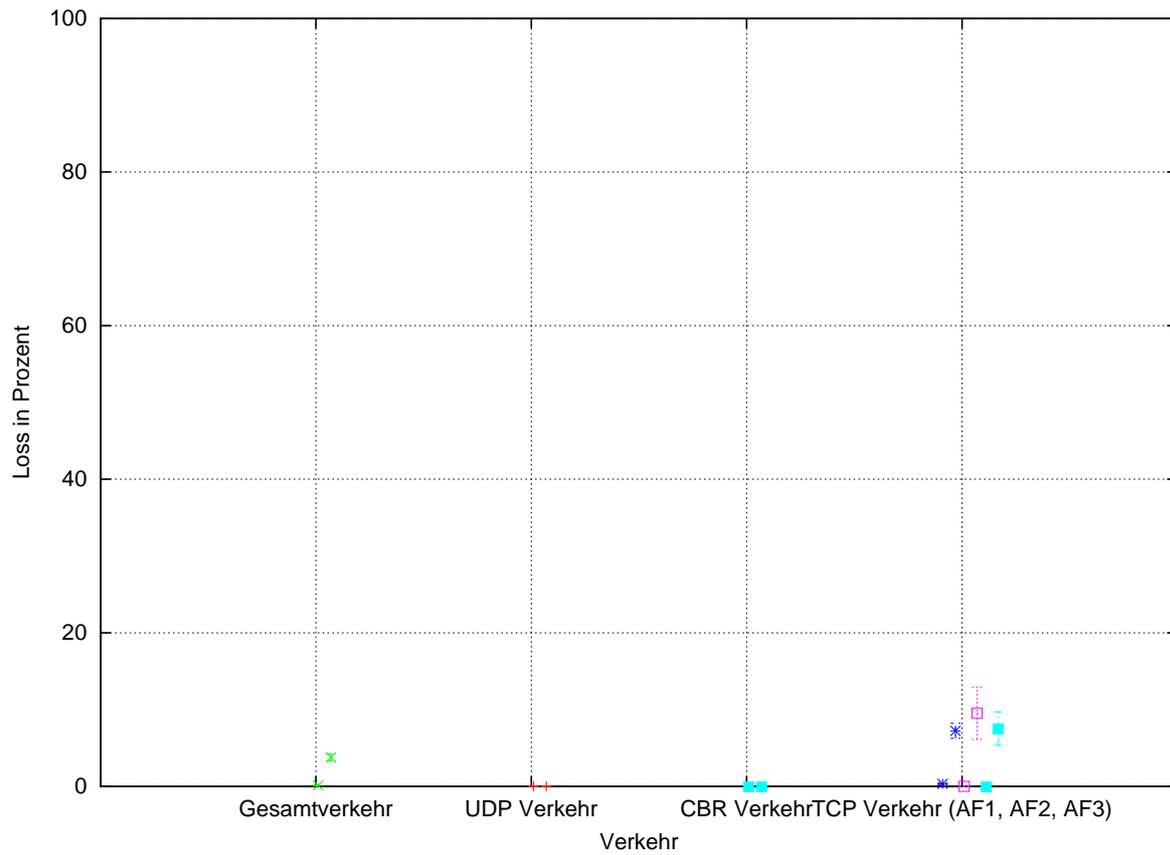
**Abbildung C.48:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



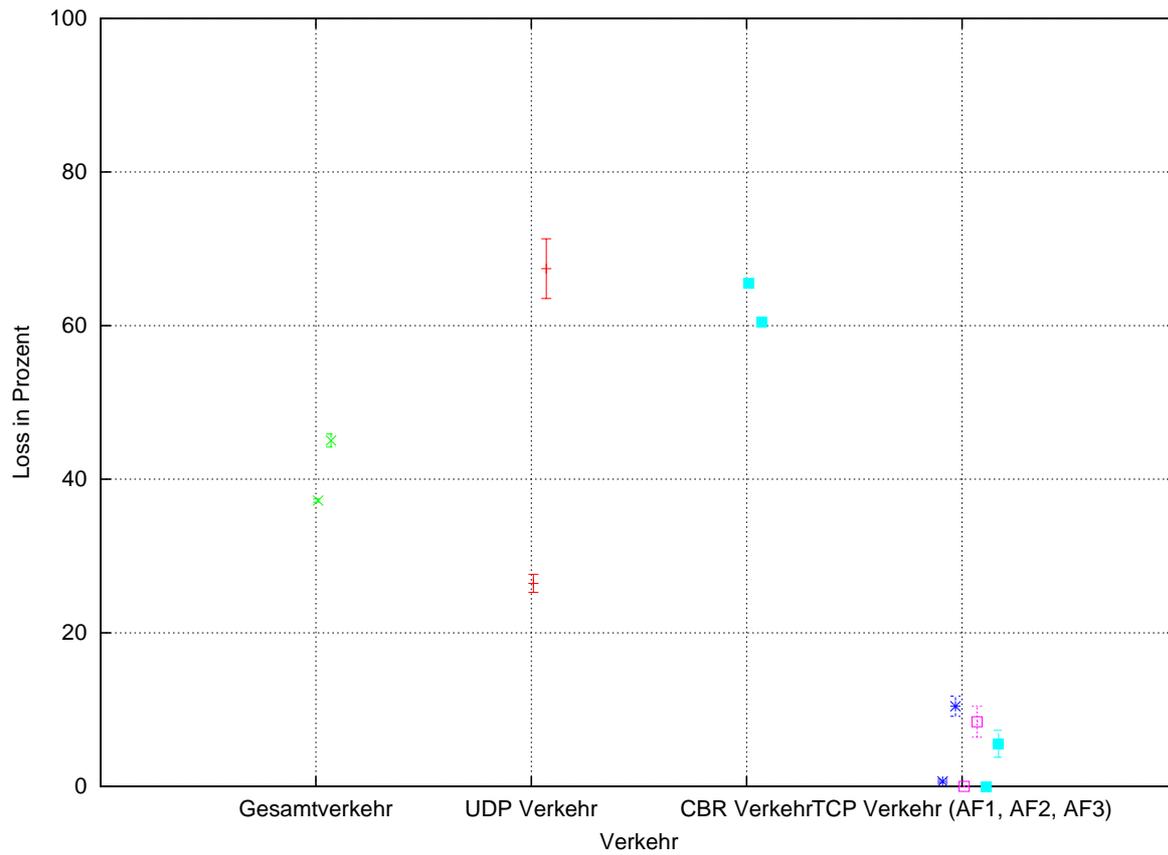
**Abbildung C.49:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



**Abbildung C.50:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)



**Abbildung C.51:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7)



**Abbildung C.52:** DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)

## Anhang D

# G-WiN Kernnetztopologie

Standort	Level	Cisco Router Typ
Berlin	1	12016
Erlangen	1	12016
Essen	1	12016
Frankfurt	1	12016
Hamburg	1	12016
Hannover	1	12016
Köln (St. Augustine)	1	12016
Leipzig	1	12016
München (Garching)	1	12016
Stuttgart	1	12016
Aachen	2	12008
Augsburg	2	12008
Bielefeld	2	12008
Braunschweig	2	12008
Dresden	2	12008
Göttingen	2	12008
Heidelberg	2	12008
Illmenau	2	12008
Kaiserslautern	2	12008
Karlsruhe	2	12008
Kiel	2	12008
Magdeburg	2	12008
Marburg	2	12008
Oldenburg	2	12008
Regensburg	2	12008
Rostock	2	12008
Würzburg	2	12008

**Tabelle D.1:** Standort, Level, und Routertyp

Verbindung	Kapazität	durchschnittliche Auslastung hin/rück	maximale Auslastung hin/rück
Berlin - Erlangen	2.4 Gbit/s	2.25 % / 0.96 %	10.92 % / 10.51 %
Berlin - Frankfurt	2.4 Gbit/s	5.37 % / 7.67 %	13.11 % / 24.65 %
Berlin - Hannover	2.4 Gbit/s	3.44 % / 1.59 %	12.62 % / 21.08 %
Berlin - Hamburg	2.4 Gbit/s	4.47 % / 2.52 %	8.44 % / 8.19 %
Erlangen - München	2.4 Gbit/s	0.97 % / 2.19 %	18.28 % / 20.12 %
Erlangen - Stuttgart	2.4 Gbit/s	5.45 % / 3.72 %	29.14 % / 30.49 %
Essen - Frankfurt	2.4 Gbit/s	6.51 % / 8.53 %	12.17 % / 29.45 %
Essen - Hamburg	2.4 Gbit/s	5.58 % / 2.32 %	10.17 % / 5.67 %
Essen - München	2.4 Gbit/s	0.51 % / 2.30 %	8.22 % / 10.22 %
Frankfurt - Hamburg	2.4 Gbit/s	6.11 % / 4.96 %	15.88 % / 10.94 %
Frankfurt - Hannover	2.4 Gbit/s	5.74 % / 7.58 %	26.88 % / 16.00 %
Frankfurt - Köln	2.4 Gbit/s	6.25 % / 7.35 %	14.42 % / 28.21 %
Frankfurt - Leipzig	10 Gbit/s	3.22 % / 4.13 %	8.39 % / 7.72 %
Frankfurt - Stuttgart	2.4 Gbit/s	9.15 % / 7.29 %	19.47 % / 24.94 %
Hamburg - Hannover	2.4 Gbit/s	4.62 % / 1.89 %	9.33 % / 7.77 %
Hamburg - Leipzig	2.4 Gbit/s	1.65 % / 1.30 %	6.11 % / 5.09 %
Hamburg - Köln	2.4 Gbit/s	2.74 % / 2.18 %	6.63 % / 12.67 %
Hannover - Leipzig	2.4 Gbit/s	1.26 % / 3.15 %	5.46 % / 7.68 %
Köln - München	2.4 Gbit/s	0.66 % / 0.67 %	3.21 % / 6.97 %
Köln - Stuttgart	2.4 Gbit/s	4.74 % / 1.02 %	10.96 % / 8.56 %
Leipzig - München	2.4 Gbit/s	4.44 % / 8.20 %	11.65 % / 19.25 %

**Tabelle D.2:** Level1 - Level1 Verbindungen im GWiN-Kernnetz mit ihren Kapazitäten, durchschnittlichen und maximalen Auslastungen gemessen über den November 2003 in 15 Minuten Intervallen

Verbindung	Kapazität	durchschnittliche Auslastung hin/rück	maximale Auslastung hin/rück
Aachen - Köln	2.4 Gbit/s	3.72 % / 1.92 %	24.45 % / 6.06 %
Augsburg - München	622 Mbit/s	2.28 % / 1.29 %	10.79 % / 6.63 %
	622 Mbit/s	2.14 % / 1.23 %	9.87 % / 5.84 %
Berlin - Magdeburg	2.4 Gbit/s	0.94 % / 0.84 %	2.35 % / 2.31 %
	2.4 Gbit/s	0.93 % / 0.86 %	4.51 % / 2.47 %
Berlin - Rostock	2.4 Gbit/s	1.62 % / 2.18 %	4.46 % / 3.86 %
	2.4 Gbit/s	1.73 % / 2.16 %	6.50 % / 5.87 %
Bielefeld - Essen	2.4 Gbit/s	3.77 % / 2.80 %	7.56 % / 24.10 %
	2.4 Gbit/s	3.97 % / 2.47 %	7.64 % / 6.23 %
Braunschweig - Hannover	2.4 Gbit/s	1.87 % / 1.20 %	3.57 % / 3.57 %
	2.4 Gbit/s	1.90 % / 1.06 %	3.75 % / 3.08 %
Dresden - Leipzig	2.4 Gbit/s	5.65 % / 2.87 %	15.32 % / 10.25 %
	2.4 Gbit/s	5.58 % / 2.77 %	10.40 % / 6.65 %
Erlangen - Regensburg	2.4 Gbit/s	0.77 % / 0.42 %	3.08 % / 1.75 %
	2.4 Gbit/s	0.77 % / 0.42 %	3.00 % / 1.36 %
Erlangen - Würzburg	2.4 Gbit/s	0.33 % / 0.25 %	2.29 % / 1.78 %
	2.4 Gbit/s	0.35 % / 0.23 %	2.20 % / 1.78 %
Frankfurt - Karlsruhe	2.4 Gbit/s	1.11 % / 0.34 %	40.44 % / 29.78 %
Frankfurt - Marburg	2.4 Gbit/s	0.79 % / 0.56 %	12.64 % / 6.20 %
	2.4 Gbit/s	0.74 % / 0.57 %	2.64 % / 1.85 %
Göttingen - Hannover	2.4 Gbit/s	0.00 % / 0.00 %	1.07 % / 0.73 %
	2.4 Gbit/s	5.31 % / 3.04 %	7.24 % / 6.93 %
Hamburg - Kiel	622 Mbit/s	1.64 % / 1.06 %	11.33 % / 5.12 %
	622 Mbit/s	1.56 % / 1.25 %	9.96 % / 6.39 %
Hamburg - Oldenburg	2.4 Gbit/s	2.23 % / 3.76 %	5.22 % / 5.58 %
	2.4 Gbit/s	2.24 % / 3.80 %	5.23 % / 6.07 %

**Tabelle D.3:** Level1 - Level2 Verbindungen im GWiN-Kernnetz mit ihren Kapazitäten, durchschnittlichen und maximalen Auslastungen gemessen über den November 2003 in 15 Minuten Intervallen

<b>Verbindung</b>	<b>Kapazität</b>	<b>durchschnittliche Auslastung hin/rück</b>	<b>maximale Auslastung hin/rück</b>
Heidelberg - Stuttgart	622 Mbit/s	0.49 % / 0.80 %	4.00 % / 5.25 %
	622 Mbit/s	0.49 % / 0.61 %	2.84 % / 4.56 %
Illmenau - Leipzig	2.4 Gbit/s	2.79 % / 1.74 %	6.19 % / 5.50 %
	2.4 Gbit/s	2.61 % / 1.78 %	5.14 % / 6.02 %
Kaiserslautern - Stuttgart	2.4 Gbit/s	1.24 % / 0.62 %	2.04 % / 1.73 %
	2.4 Gbit/s	1.96 % / 0.85 %	3.88 % / 2.76 %
Karlsruhe - Stuttgart	2.4 Gbit/s	0.11 % / 0.08 %	40.37 % / 31.97 %

**Tabelle D.4:** Fortsetzung: Level1 - Level2 Verbindungen im GWiN-Kernnetz mit ihren Kapazitäten, durchschnittlichen und maximalen Auslastungen gemessen über den November 2003 in 15 Minuten Intervallen

# Abbildungsverzeichnis

3.1	Einteilung von Anwendungen nach QoS Sensitivität . . . . .	21
4.1	Video Paketgrößenverteilung . . . . .	23
6.1	Anbindung des NS2 an den Szenario-Generator . . . . .	27
8.1	Extrapolationsschema . . . . .	36
9.1	IntServ Architektur . . . . .	39
9.2	IntServ Calculus Grafik: Kumulativer Fluß bei IntServ als Funktion der Zeit . . . . .	42
9.3	IntServ - Scheduling . . . . .	43
9.4	DiffServ Edge-Knoten Architektur nach dem Olympic Ansatz . . . . .	45
9.5	DiffServ Core-Knoten Architektur nach dem Olympic Ansatz . . . . .	46
9.6	DiffServ Olympic Ansatz - Scheduling . . . . .	47
9.7	DiffServ Edge-Knoten Architektur nach dem Default Ansatz . . . . .	48
9.8	DiffServ Core-Knoten Architektur nach dem Default Ansatz . . . . .	49
9.9	DiffServ Default Ansatz - Scheduling . . . . .	50
9.10	Vergleich der Berechnungsergebnisse der M/G/1 Formel mit den Simulationen des Single Queue Simulators bei der Verteilungskombination nexp-hyp2 . . . . .	54
9.11	Vergleich der Berechnungsergebnisse der M/G/1 Formel mit den Simulationen des Single Queue Simulators bei der Verteilungskombination nexp-tpt . . . . .	55
9.12	Bewertung der Krämer Formel und des Whitt Ansatzes bei 20 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen . . . . .	57
9.13	Bewertung der Krämer Formel und des Whitt Ansatzes bei 50 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen . . . . .	58
9.14	Bewertung der Krämer Formel und des Whitt Ansatzes bei 80 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen . . . . .	59
9.15	Bewertung der Krämer Formel und des Whitt Ansatzes bei 90 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen . . . . .	60
9.16	Bewertung der Krämer Formel und des Whitt Ansatzes bei 95 % Auslastung unter Verwendung der angegebenen Verteilungskombinationen . . . . .	61
10.1	ETS Topologie . . . . .	67
10.2	Best Effort: Auslastung bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) innerhalb eines 1 s Intervalls (Int.) . . . . .	70

10.3	Best Effort: Auslastung bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) innerhalb eines 1 s Intervalls (Int.) . . . . .	71
10.4	Best Effort: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	72
10.5	Best Effort: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ...,7) . . . . .	73
10.6	Best Effort: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ...,7) . . . . .	74
10.7	Best Effort: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ...,7) . . . . .	75
10.8	Best Effort: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ...,7) . . . . .	76
10.9	Best Effort: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ...,7) . . . . .	77
10.10	IntServ: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ...,7) . . . . .	82
10.11	IntServ: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	83
10.12	IntServ: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	84
10.13	IntServ: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	85
10.14	IntServ: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	86
10.15	IntServ: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	87
10.16	ETS DiffServ Topologie . . . . .	90
10.17	DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	91
10.18	DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	92
10.19	DiffServ - Olympic Ansatz: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	93
10.20	DiffServ - Olympic Ansatz: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	94
10.21	DiffServ - Olympic Ansatz: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	95
10.22	DiffServ - Olympic Ansatz: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	96
10.23	DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von CBR Flows und einem TCP Flow . . . . .	99
10.24	DiffServ - Olympic Ansatz: Loss bei variierender Anzahl von CBR Flows und einem TCP Flow . . . . .	100
10.25	DiffServ - Olympic Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von Video Traces und einem TCP Flow . . . . .	101
10.26	DiffServ - Olympic Ansatz: Loss bei variierender Anzahl von Video Traces und einem TCP Flow . . . . .	102

10.27	DiffServ - Default Ansatz: Ende-zu-Ende Delay bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	104
10.28	DiffServ - Default Ansatz: Ende-zu-Ende Delay bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	105
10.29	DiffServ - Default Ansatz: Jitter bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	106
10.30	DiffServ - Default Ansatz: Jitter bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	107
10.31	DiffServ - Default Ansatz: Loss bei einem CBR Flow und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	108
10.32	DiffServ - Default Ansatz: Loss bei 10 CBR Flows und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	109
10.33	DiffServ - Default Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von CBR Flows und einem TCP Flow . . . . .	112
10.34	DiffServ - Default Ansatz: Loss bei variierender Anzahl von CBR Flows und einem TCP Flow . . . . .	113
10.35	DiffServ - Default Ansatz: Ende-zu-Ende Delay bei variierender Anzahl von Video Traces und einem TCP Flow . . . . .	114
10.36	DiffServ - Default Ansatz: Loss bei variierender Anzahl von Video Traces und einem TCP Flow . . . . .	115
10.37	Overprovisioning: Vergleich des Ende-zu-Ende Delays bei 10 CBR und 7 TCP Flows mit und ohne Overprovisioning . . . . .	116
10.38	Overprovisioning: Vergleich des Ende-zu-Ende Delays bei 3 Video Traces und 1 TCP Flow mit und ohne Overprovisioning . . . . .	117
10.39	Overprovisioning: Vergleich des Jitters bei 10 CBR und 7 TCP Flows mit und ohne Overprovisioning . . . . .	118
10.40	Overprovisioning: Vergleich des Jitters bei 3 Video Traces und 1 TCP Flow mit und ohne Overprovisioning . . . . .	119
10.41	Overprovisioning: Vergleich des Loss' bei 10 CBR und 7 TCP Flows mit und ohne Overprovisioning . . . . .	120
10.42	Overprovisioning: Vergleich des Loss' bei 3 Video Traces und 1 TCP Flow mit und ohne Overprovisioning . . . . .	121
10.43	ETS unter Einsatz von BE: Wartezeit in den Warteschlangen bei einem CBR Flow und variierender Anzahl von TCP Flows (1, 2, ..., 7) . . . . .	150
10.44	ETS unter Einsatz von BE: Wartezeit in der Warteschlange bei 10 CBR Flows und variierender Anzahl von TCP Flows (1, 2, ..., 7) . . . . .	151
10.45	Ablaufdiagramm 1 . . . . .	166
10.46	Ablaufdiagramm 2 . . . . .	167
10.47	Ablaufdiagramm 3 . . . . .	168
10.48	Ablaufdiagramm 4 . . . . .	169
10.49	Ablaufdiagramm 5 . . . . .	170

10.50	Ablaufdiagramm 6 . . . . .	171
11.1	G-WiN Kernnetztopologie (Stand: 31.11.2003) . . . . .	173
11.2	Beispiel einer Kernnetzverbindung: G-WiN Kernnetzleitungsauslastung über einen Monat im 15 min - Mittel (Stand: 31.11.2003) . . . . .	175
11.3	Durchsatz einer 155 Mb/s - G-WiN Zugangsleitung über einen Tag im 5 min - Mittel (Stand: 31.11.2003) . . . . .	176
11.4	Durchsatz derselben 155 Mb/s - G-WiN Zugangsleitung wie in Abb. 11.3 über einen Monat im 15 min - und 2 h - Mittel (Stand: 31.11.2003) . . . . .	176
11.5	G-WiN Beispielpfad . . . . .	178
11.6	G-WiN: Vergleich von Ende-zu-Ende Delay unter Verwendung von BE, Overprovisioning und Kombiansatz . . . . .	183
11.7	G-WiN: Vergleich von Jitter unter Verwendung von BE, Overprovisioning und Kombiansatz . . . . .	184
11.8	G-WiN: Vergleich von Loss unter Verwendung von BE, Overprovisioning und Kombiansatz . . . . .	185
B.1	Auslastung 20 %: Heavy Traffic Formel . . . . .	198
B.2	Auslastung 50 %: Heavy Traffic Formel . . . . .	199
B.3	Auslastung 80 %: Heavy Traffic Formel . . . . .	200
B.4	Auslastung 90 %: Heavy Traffic Formel . . . . .	201
B.5	Auslastung 95 %: Heavy Traffic Formel . . . . .	202
C.1	BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	220
C.2	BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	221
C.3	BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	222
C.4	BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	223
C.5	BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	224
C.6	BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	225
C.7	BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	226
C.8	BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	227
C.9	BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	228

C.10	BE Simulationsergebnisse von Kapitel 10.2.1: Auslastung des Bottleneck Links bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	229
C.11	BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	230
C.12	BE Simulationsergebnisse von Kapitel 10.2.1: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	231
C.13	BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	232
C.14	BE Simulationsergebnisse von Kapitel 10.2.1: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	233
C.15	BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	234
C.16	BE Simulationsergebnisse von Kapitel 10.2.1: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	235
C.17	IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	236
C.18	IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	237
C.19	IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	238
C.20	IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	239
C.21	IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	240
C.22	IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	241
C.23	IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	242
C.24	IntServ Simulationsergebnisse von Kapitel 10.2.2: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	243
C.25	IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	244
C.26	IntServ Simulationsergebnisse von Kapitel 10.2.2: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	245
C.27	IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	246
C.28	IntServ Simulationsergebnisse von Kapitel 10.2.2: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	247
C.29	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	248
C.30	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	249

C.31	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	250
C.32	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,2, ..., 7) . . . . .	251
C.33	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	252
C.34	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	253
C.35	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	254
C.36	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	255
C.37	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . .	256
C.38	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	257
C.39	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . .	258
C.40	DiffServ (Olympic) Simulationsergebnisse von Kapitel 10.2.3: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	259
C.41	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	260
C.42	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	261
C.43	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	262
C.44	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	263
C.45	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei einem Video Trace (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	264
C.46	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei 10 Video Traces (UDP) und variierender Anzahl von TCP Flows (1,7) . . . . .	265
C.47	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . . .	266
C.48	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Ende-zu-Ende Delay bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7)	267
C.49	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . .	268
C.50	DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Jitter bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . .	269

- 
- C.51 DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei einem Video Trace (UDP) und einem CBR Flow und variierender Anzahl von TCP Flows (1,7) . . . 270
  - C.52 DiffServ (Default) Simulationsergebnisse von Kapitel 10.2.4: Loss bei 5 Video Traces (UDP) und 5 CBR Flows und variierender Anzahl von TCP Flows (1,7) . . . . . 271

# Tabellenverzeichnis

1.1	Variationsmöglichkeiten . . . . .	9
2.1	AF DS Codepoints - binär (dezimal) . . . . .	15
3.1	Obere Grenzwerte der Dienstgütemetriken . . . . .	19
3.2	Einteilung von Anwendungen nach QoS Sensitivität . . . . .	20
8.1	Auftretende Fälle durch Variation der 3 Freiheitsgrade . . . . .	33
9.1	Parameter zur Ende-zu-Ende Delayberechnung . . . . .	41
9.2	Parameter der Wartezeitformeln . . . . .	52
9.3	Verteilungen und Variation der Variationskoeffizienten innerhalb der M/G/1 Untersuchungen . . . . .	53
9.4	Verteilungen und Variation der Variationskoeffizienten innerhalb der GI/G/1 Untersuchungen . . . . .	56
10.1	Variation der Anzahl von CBR und TCP Flows innerhalb der Simulationsexperimente . . . . .	69
10.2	Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente . . . . .	78
10.3	Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente . . . . .	79
10.4	Variation der Anzahl von CBR Flows und TCP Flows innerhalb der Simulationsexperimente . . . . .	80
10.5	Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente . . . . .	88
10.6	Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente . . . . .	89
10.7	Variation der Anzahl von CBR Flows und TCP Flows innerhalb der Simulationsexperimente . . . . .	91
10.8	Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente . . . . .	97
10.9	Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente . . . . .	98
10.10	Variation der Anzahl von CBR Flows und TCP Flows innerhalb der Simulationsexperimente . . . . .	103
10.11	Variation der Anzahl von Video Traces und TCP Flows innerhalb der Simulationsexperimente . . . . .	110

---

10.12	Variation der Anzahl von Video Traces, CBR Flows und TCP Flows innerhalb der Simulationsexperimente . . . . .	111
10.13	ETS Fallstudie für BE und IntServ . . . . .	123
10.14	ETS Fallstudie für DiffServ (Olympic und Default) . . . . .	124
10.15	Verkehrsspezifische Variablenwerte . . . . .	127
10.16	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR und TCP Verkehr . . . . .	128
10.17	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR und TCP Verkehr . . . . .	129
10.18	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus Video und TCP Verkehr . . . . .	130
10.19	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus Video und TCP Verkehr . . . . .	130
10.20	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR, Video und TCP Verkehr . . . . .	131
10.21	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR, Video und TCP Verkehr . . . . .	132
10.22	Verkehrsspezifische Variablenwerte . . . . .	135
10.23	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR und TCP Verkehr . . . . .	136
10.24	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR und TCP Verkehr . . . . .	137
10.25	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus Video und TCP Verkehr . . . . .	138
10.26	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus Video und TCP Verkehr . . . . .	138
10.27	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR, Video und TCP Verkehr . . . . .	139
10.28	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR, Video und TCP Verkehr . . . . .	140
10.29	Verkehrsspezifische Variablenwerte . . . . .	144
10.30	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR und TCP Verkehr . . . . .	145
10.31	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR und TCP Verkehr . . . . .	146
10.32	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus Video und TCP Verkehr . . . . .	147
10.33	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus Video und TCP Verkehr . . . . .	147
10.34	Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten für einen Verkehrsmix aus CBR, Video und TCP Verkehr . . . . .	148

10.35	Vergleich von berechneten und durch Simulation erzeugten Werten von Jitter für einen Verkehrsmix aus CBR, Video und TCP Verkehr . . . . .	149
10.36	Extrapolation von BE auf Olympic bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	153
10.37	Extrapolation von BE auf Olympic bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	153
10.38	Extrapolation von BE auf Default bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	154
10.39	Extrapolation von BE auf Default bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	154
10.40	Extrapolation von BE auf BE bei einer veränderten Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	156
10.41	Extrapolation von BE auf BE bei einer veränderten Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	156
10.42	Extrapolation von Olympic auf Olympic bei einer veränderten Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	156
10.43	Extrapolation von Olympic auf Olympic bei einer veränderten Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	157
10.44	Extrapolation von Default auf Default bei einer veränderten Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	157
10.45	Extrapolation von Default auf Default bei einer veränderten Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	157
10.46	Extrapolation von BE auf BE mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	159
10.47	Extrapolation von BE auf BE mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	159
10.48	Extrapolation von BE auf BE mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	160
10.49	Extrapolation von BE auf BE mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	160

10.50	Extrapolation von Olympic auf Olympic mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	160
10.51	Extrapolation von Olympic auf Olympic mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	161
10.52	Extrapolation von Olympic auf Olympic mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	161
10.53	Extrapolation von Olympic auf Olympic mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	161
10.54	Extrapolation von Default auf Default mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	162
10.55	Extrapolation von Default auf Default mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus CBR und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	162
10.56	Extrapolation von Default auf Default mit doppelter Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	162
10.57	Extrapolation von Default auf Default mit halbiertes Bandbreite aller Links bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechneten und durch Simulation erzeugten Ende-zu-Ende Delaywerten . . . . .	163
11.1	G-WiN: Vergleich von berechneten und simulierten Ende-zu-Ende Delaywerten . . . . .	181
11.2	G-WiN: Vergleich von berechneten und simulierten Ende-zu-Ende Jitterwerten . . . . .	182
11.3	Extrapolation von BE auf Kombiansatz bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechnetem und durch Simulation erzeugtem Ende-zu-Ende Delay . . . . .	187
11.4	Extrapolation von BE auf BE mit Overprovisioning bei einer Verkehrszusammensetzung aus Video und TCP Verkehr: Vergleich von berechnetem und durch Simulation erzeugtem Ende-zu-Ende Delay . . . . .	188
B.1	Verteilungen und Variation der Variationskoeffizienten innerhalb der GI/G/1 Untersuchungen . . . . .	203
B.2	nexp-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	204
B.3	nexp-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	204
B.4	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit identischem $C_A^2$ und $C_B^2$ . . . . .	205
B.5	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit identischem $C_A^2$ und $C_B^2$ . . . . .	206
B.6	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem $C_A^2$ und $C_B^2$ . . . . .	207
B.7	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem $C_A^2$ und $C_B^2$ . . . . .	208
B.8	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem $C_A^2$ und $C_B^2$ . . . . .	209

---

B.9	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem $C_A^2$ und $C_B^2$ .	210
B.10	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem $C_A^2$ und $C_B^2$ .	211
B.11	hyp2-hyp2 Singlequeue Simulator Untersuchungen mit unterschiedlichem $C_A^2$ und $C_B^2$ .	212
B.12	tpt-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	213
B.13	hyp2-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	214
B.14	tpt-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	215
B.15	erl-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	216
B.16	erl-hyp2 Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	217
B.17	erl-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	218
B.18	erl-tpt Singlequeue Simulator Untersuchungen: Mittelwert der Wartezeit . . . . .	219
D.1	Standort, Level, und Routertyp . . . . .	272
D.2	Level1 - Level1 Verbindungen im GWiN-Kernnetz mit ihren Kapazitäten, durchschnittlichen und maximalen Auslastungen gemessen über den November 2003 in 15 Minuten Intervallen . . . . .	273
D.3	Level1 - Level2 Verbindungen im GWiN-Kernnetz mit ihren Kapazitäten, durchschnittlichen und maximalen Auslastungen gemessen über den November 2003 in 15 Minuten Intervallen . . . . .	274
D.4	Fortsetzung: Level1 - Level2 Verbindungen im GWiN-Kernnetz mit ihren Kapazitäten, durchschnittlichen und maximalen Auslastungen gemessen über den November 2003 in 15 Minuten Intervallen . . . . .	275

# Literaturverzeichnis

- [ADLER 02] H.-M. Adler: *Neues im G-WiN*. 36. Betriebstagung, 2002
- [ALLEN 90] A.O. Allen: *Probability, Statistics, and Queueing Theory with Computer Science Applications*. Academic Press, Inc., 1990
- [BAUMGARTNER ET AL. 96] F. Baumgartner, T. Braun and Pascal Habegger: *Differentiated Services: A New Approach for Quality of Service in the Internet*. Chapman & Hall, 1996.
- [BLAKE ET AL. 98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss: *An Architecture for Differentiated Services*. Network Working Group, Category: Informational, RFC 2475, December 1998.
- [BRADEN ET AL. 94] R. Braden, D. Clark and S. Shenker: *Integrated Services in the Internet Architecture: an Overview*. Network Working Group, Category: Informational, RFC 1633, June 1994.
- [BRADEN ET AL. 97] R. Braden, L. Zang, S. Berson, S. Herzog and S. Jamin: *Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification*. Network Working Group, Category: Standards Track, RFC2205, September 1997.
- [BURGSTAHLER ET AL. 01] L. Burgstahler, J. Jaehnert, L. Mark, J. Tiemann and R. Roth: *QUASAR - Quality of Service Architekturen - Meilenstein 1: Technologieuebersicht*. DFN Projekt, Start 2001.
- [BURGSTAHLER ET AL. 02] L. Burgstahler et al.: *QUASAR - Quality of Service Architekturen - Meilenstein 2: Quasar QoS Reference Model*. DFN-Projekt, Start 2001.
- [BURGSTAHLER ET AL. 03] L. Burgstahler et al.: *QUASAR - Quality of Service Architekturen - Meilenstein 3: Preliminary Quasar QoS Architecture*. DFN-Projekt, Start 2001.
- [BURGSTAHLER ET AL. 04] L. Burgstahler et al.: *QUASAR - Quality of Service Architekturen - Meilenstein 4: First Test Results*. DFN-Projekt, Start 2001.
- [BURGSTAHLER ET AL. 05] L. Burgstahler et al.: *QUASAR - Quality of Service Architekturen - Meilenstein 5: QUASAR QoS-Architektur*. DFN-Projekt, Start 2001.
- [BURGSTAHLER ET AL. 06] L. Burgstahler et al.: *QUASAR - Quality of Service Architekturen - Meilenstein 6: Projektabschluss mit Abschlußbericht*. DFN-Projekt, Start 2001.
- [CHO 01] K. Cho: *The Design and Implementation of the AltQ Traffic Management System*. Doctoral Dissertation, Keio University, January 2001
- [CISCO SYSTEMS 12008] Cisco Systems, Inc.: *Product Overview*.  
< URL : <http://www.cisco.com/univercd/cc/td/doc/product/core/cis12000/cis12008/icg/product1.htm> >, 2001
- [CISCO SYSTEMS 12016] Cisco Systems, Inc.: *Product Overview*.  
< URL : <http://www.cisco.com/univercd/cc/td/doc/product/core/cis12000/cis12016/icg/hfricgpo.htm> >, 2001

- [CISCO SYSTEMS 02] Cisco Systems, Inc.: *Implementing QoS Solutions for H.323 Video Conferencing Over IP*.  
< URL : <http://www.cisco.com/warp/public/105/video-qos.html> >, 2002
- [CONSTABLE & PRICE 01] G. Constable and D. Price: *Options for Quality of Service Mechanisms, Network Quality of Service and Resource Issues, H.323 IP Videoconferencing demonstrator*. University of Wales, Aberystwyth, 3/2001
- [DAVIE ET AL. 02] B. Davie et al.: *An Expedited Forwarding PHB*. Network Working Group, Category: Standards Track, RFC 3246, March 2002.
- [DFN 01] DFN: *Gigabit-Wissenschaftsnetz G-WiN*. < URL : <http://www.dfn.de/win> >, 2001
- [FALL & VARADHAN 02] K. Fall und K. Varadhan: *The ns Manual*.  
< URL : <http://www.isi.edu/nsnam/ns/ns-documentation.html> >, 2002
- [FERGUSON & HUSTON 98] Paul Ferguson and Geoff Huston: *Quality of Service - Delivering QoS on the Internet and in Corporate Networks*. WILEY COMPUTER PUBLISHING, 1998.
- [FERGUSON & HUSTON 98] P. Ferguson and G. Huston: *Quality of Service in the Internet: Fact, Fiction, or Compromise?*  
< URL : [http://www.employees.org/ferguson/inet\\_qos.htm](http://www.employees.org/ferguson/inet_qos.htm) >, Cisco Systems, Inc., 1998
- [GLASMANN ET AL.] J. Glasmann et al.: *Estimation of Token Bucket Parameters for Videoconferencing Systems in Corporate Networks*. Lehrstuhl für Kommunikationsnetze, Technische Universität München
- [GLASMANN 03] J. Glasmann: *Ressourcenmanagement für Echtzeitverkehre in Intranets*. Dissertation, Lehrstuhl für Kommunikationsnetze, TU München, 2003
- [GRANGES 00] E. A. Granges: *Bandwidth Broker for Differentiated Services*. Diplomarbeit, Forschungsgruppe Rechnernetze und Verteilte Systeme (RVS), Institut für Informatik und angewandte Mathematik, Universität Bern, 2000
- [GREINER & TINHOFFER 96] M. Greiner and G. Tinhofer: *Stochastik fuer Studienanfänger der Informatik*. Carl Hanser, 1996.
- [GREIS 02] M. Greis: *Tutorial for the Network Simulator ns*.  
< URL : <http://www.isi.edu/nsnam/ns/tutorial/index.html> >, 2002
- [HEINANEN ET AL. 99] J. Heinanen et al: *Assured Forwarding PHB Group*. Network Working Group, Category: Standard Track, RFC 2597, June 1999.
- [HOFFMANN 04] M. Hoffmann: *Programming Code of the LETSQoS Project*. DFN LETSQoS Project, 2004
- [HOLLECZEK ET AL.] P. Holleczeck et al.: *Bestimmung der Dienstqualität im G-WiN mittels IPPM*. Vortrag DFN-Mitgliederversammlung, Dezember 2003
- [HU & LOW] J. Hu and N. Low: *Investigation of Telephonic Voice Traffic Network Performance on Integrated Wireless LAN and Wired Internet*. Royal Melbourne Institute of Technology, Melbourne
- [ITU 00] ITU-T: *Internet Protocol Communication Service - IP Performance and Availability Objectives and Allocations*. Draft ITU-T Recommendation Y.1541, November 2000

- [ITU 02] International Telecommunication Union: *List of ITU-T Recommendations*.  
< URL : <http://www.itu.int/publibase/itu-t/ItutDetailsOnSubject.asp?serie=g&subjectid=2100> >, 2002
- [JAISWAL 68] N. K. Jaiswal: *Priority Queues*. ACADEMIC PRESS, 1968.
- [JESSEN 01] E. Jessen: *Vorlesung Sommersemester 2001 - Rechnernetze II*. Lehrstuhl fuer Informatik, Technische Universitaet Muenchen, 2001.
- [JOBMANN 01] M. Jobmann: *Quantitative Modelle fuer Rechensysteme - Ausarbeitung der Folien zur Vorlesung WS 01/02*. Lehrstuhl fuer Informatik, Technische Universitaet Muenchen, 2001.
- [KARSTEN 03] M. Karsten: *KOM RSVP Engine* . < URL : <http://www.kom.e-technik.tu-darmstadt.de/rsvp/> >, 2003
- [KESHAV 98] S. Keshav: *An Engineering Approach to Computer Networking. ATM Networks, the Internet, and the Telephone Network*. Addison-Wesley, 1998
- [KLEINROCK 75] L. Kleinrock: *Queueing Systems - Volume I: Theory*. Wiley-Interscience, 1975.
- [KLEINROCK 76] L. Kleinrock: *Queueing Systems - Volume 2: Computer Applications*. Wiley-Interscience, 1976.
- [LIPSKY ET AL.] L. Lipsky et al.: *Simulationstechniken und Module*. Bericht zum Ergänzungspaket 1 des Projektes - Pufferdimensionierung in ATM-Weitverkehrsnetzen mit selbstähnlichen Verkehrsströmen unter besonderer Berücksichtigung des transienten Verhaltens, Juni 1999
- [NICHOLS 99] K. Nichols et al.: *A Two-bit Differentiated Services Architecture for the Internet*. Network Working Group, Category: Standards Track, RFC 2638, July 1999
- [PAREKH & GALLEGER] A.K. Parekh and R.G. Gallager: *A generalized processor sharing approach to flow control in integrated services networks: The single node case*. IEEE/ACM Trans. on Networking, 2(2): 137-150, April 1994
- [SCHMITT 01] J. B. Schmitt: *Heterogeneous Network Quality of Service Systems*. Kluwer Academic Publisher, 2001.
- [SCHMITT ET AL. 02] J. Schmitt et al.: *Network QoS An Up-to-Date Overview*. DFN LETSQoS Project, Milestone 1, 2002
- [SCHMITT ET AL. 03-1] J. Schmitt et al.: *Scenario Generation*. DFN LETSQoS Project, Milestone 2, 2003
- [SCHMITT ET AL. 04-1] J. Schmitt et al.: *Final Report*. DFN LETSQoS Project, Milestone 4, 2004
- [SHENKER ET AL. 97] S. Shenker, C. Partridge and R. Guerin: *Specification of Guaranteed Quality of Service*. Network Working Group, Category: Standards Track, RFC 2212, September 1997.
- [STARDUST.COM, INC. 99/1] Stardust.com, Inc.: *White Paper - The Need for QoS*. July 1999.
- [STARDUST.COM, INC. 99/2] Stardust.com, Inc.: *White Paper - QoS protocols and architectures*. July 1999.
- [STARDUST.COM, INC. 99/3] Stardust.com, Inc.: *White Paper - Introduction to QoS Policies*. July 1999.

- [STEINMETZ 99] R. Steinmetz: *Multimedia-Technologie - Grundlagen, Komponenten und Systeme*. Springer, 1999.
- [STOYAN 83] D. Stoyan: *Comparison Methods for Queues and Other Stochastic Models*. John Wiley & Sons: 1983
- [TANENBAUM 98] A.S. Tanenbaum: *Computernetzwerke*. Prentice Hall, 1998
- [TEITELBAUM 01] B. Teitelbaum: *Future Priorities for Internet2 QoS*. October 2001.
- [TRAN-GIA 96] P. Tran-Gia: *Analytische Leistungsbewertung verteilter Systeme - Eine Einfuehrung*. Springer, 1996.
- [VAN DER ZEE 99] M. van der Zee and G. Heijenk: *Quality of Service Routing - State of the Art Report*. ERICSSON, Open REPORT, August 1999.
- [VARSCHNEY ET AL. 02] U. Varschney et al.: *VOICE OVER IP COMMUNICATIONS OF THE ACM.*, Vol. 45, 2002
- [WELCH 97] B. B. Welch: *Practical Programming in Tcl and Tk*. Prentice Hall Ptr, 1997
- [WHITE & CROWCROFT] P. White and J. Crowcroft: *The Integrated Services in the Internet: State of Art*. Department of Computer Science, University College London
- [NICHOLS 99] K. Nichols et al.: *A Two-bit Differentiated Services Architecture for the Internet*. Network Working Group, Category: Standards Track, RFC 2638, July 1999
- [WHITT 88] W. Whitt and S. Suresh: *Arranging Queues in Series*. 1988
- [WROCLAWSKI 97] J. Wroclawski: *The Use of RSVP with IETF Integrated Services*. Network Working Group, Category: Standards Track, RFC 2210, September 1997.
- [WROCLAWSKI 97] J. Wroclawski: *Specification of Controlled-Load Network Element Service*. Network Working Group, Category: Standards Track, RFC 2211, September 1997.
- [ZHANG 95] H. Zhang: *Service Disciplines for Guaranteed Performance Service in Packet Switching Networks*. Proc. IEEE,S. 1373-1396, Oktober 1995

# Index

- Admission Control, 16
- AdSpec, 13
- Alternate Queueing, 29
- Ankunftskurve, 41
- Ankunftsprozess, 55
- Assured Forwarding, 15
  
- Bandbreite, 19
- Bandwidth Broker, 17
- Bedienkurve, 41
- Bedienprozess, 53
- Bedienstrategie, 38
- Berechnung, 35
- Best Effort, 11, 37
- Best/Worst Case Analyse, 37
  
- Class-Based Queueing, 15
- Classifier, 13
- Codecs, 22
- Constant Bit Rate, 25
- Controlled Load Service, 12
- Customer Network Management, 174
  
- Default Ansatz, 48
- Delay, 19
- DFN, 172
- Dienstgütemetrik, 19
- Differentiated Services, 14, 44
- DiffServ Codepoint, 14
- Dropper, 14
  
- Ein-Stationen-Wartesysteme, 63
- Erlang Verteilung, 55
- Expedited Forwarding, 15
- Extrapolations-Szenario, 31
- Extrapolations-Testszenario, 67
- Extrapolationsdurchführungsbeschreibung, 165
- Extrapolationsverfahren, 33
  
- First Come First Serve, 11
- Flowspec, 13
- Fluid Model, 41
- Framework, 12
- Freiheitsgrade, 33
  
- G-WiN, 172
- G.723.1, 22
  
- GÉANT, 172
- GI/G/1, 51
- Graph Modeling Language, 24
- Guaranteed Service, 12
  
- H-FSC, 29
- H.323, 22
- Heavy-Traffic Formel, 52
- hyperexponentielle Verteilung, 53
  
- Integrated Services, 11, 38
- IS Model, 12
  
- Jitter, 19
- JoBS, 29
  
- Kernnetzauslastung, 174
- Kernnetztopologie, 172
- Krämer, 52
  
- Langenbach-Belz, 52
- Last-Generator, 24
- Lastsituation, 174
- Lastspektrum, 31
- Loss, 20
  
- M/G/1, 51
- Marker, 14
- Meter, 14
  
- negativ exponentielle Verteilung, 53
- Netzwerkkonfiguration, 31
- NS2, 26
  
- Olympic Ansatz, 44
- OTcl, 26
- Overprovisioning, 11
  
- P2P, 24
- Packet Scheduler, 13
- PATH-Message, 13
- Per-Hop Behaviour, 14
- Policy Control, 14
- Pollaczek-Khintchine, 52
- Priority Queueing, 15
- Propagation Delay, 43
  
- QBone Scavenger Service, 16

QoS, 1  
QoS Spektrum, 32  
QoS Technologien, 32, 194  
QoS-sensitive Anwendungen, 19

Random Early Detection, 27  
Reservation Setup Protocol, 12  
Restbedienzeit, 43  
RESV-Message, 13  
RIO, 29  
RSpec, 13  
RSVP Engine, 27

Sender Template, 13  
Sensitivitätsanalyse, 68  
Sensitivitätsklasse, 20  
Service Level Agreement, 14  
Shaper, 14  
Simulationen, 33  
Standardabweichung, 53  
Stochastisches Verfahren, 51  
Synchronous Digital Hierarchy, 173

TclCl, 26  
Testbett, 29  
Throughput, 19  
Token Bucket, 40  
Topologie-Generator, 24  
Topology File Format, 24  
Traffic Conditioning Agreement, 14  
Truncated Powertail Verteilung, 53  
TSpec, 13  
Type of Service, 14

Validierung, 35  
Variable Bit Rate, 22  
Variationskoeffizient, 55  
Verkehrsmodell, 24  
Verkehrszusammensetzung, 176  
Vertrauensintervall, 53  
Videokonferenz, 22  
VoIP, 22

Wartezeitformel, 51  
Wavelength Division Multiplexing, 173  
Weighted Fair Queueing, 27, 43  
Weighted Round Robin, 15  
Whitt, 55  
WWW, 24

Zugangsleitung, 175