

Universität Rostock

Fakultät für Ingenieurwissenschaften

Fachbereich Elektrotechnik und Informationstechnik

Institut für Nachrichtentechnik und Informationselektronik



Studienarbeit

Analyse und Test einer VoIP Umgebung

cand. ing. Thomas Vergin

5. März 2001

Betreuer

Dr.-Ing. H.-D. Melzer

Dipl.-Ing. T. Kessler

Inhaltsverzeichnis

Abbildungsverzeichnis	5
Tabellenverzeichnis	6
Abkürzungsverzeichnis	9
1 Vorwort	12
2 Motivation	14
3 Theoretische Grundlagen	19
3.1 Einleitung	22
3.2 End-to-End QoS-Modelle	23
3.2.1 Best-Effort Service	24
3.2.2 Integrated Services (IntServ)	24
3.2.3 Differentiated Services (DiffServ)	25
3.3 Quality of Service	26
3.3.1 Classification	27
3.3.1.1 IP Precedence	27
3.3.1.2 Policy-Based Routing (PBR)	29
3.3.1.3 Committed Access Rate (CAR) - Packet Classification	29
3.3.2 Congestion Management	30
3.3.2.1 First In First Out Queueing (FIFO)	31
3.3.2.2 Weighted Fair Queueing (WFQ)	32
3.3.2.3 Class-Based Weighted Fair Queueing (CBWFQ) . . .	34
3.3.2.4 Priority Queueing (PQ)	35
3.3.2.5 Custom Queueing (CQ)	36
3.3.2.6 IP RTP Priority	37
3.3.2.7 Low Latency Queueing (LLQ)	38

3.3.3	Congestion Avoidance	38
3.3.3.1	Verfahren der Überlastvermeidung	39
3.3.3.2	Weighted Random Early Detection (WRED)	40
3.3.3.3	Flow-Based WRED	41
3.3.4	Policing and Shaping	42
3.3.4.1	Committed Access Rate (CAR) - Rate Limiting	43
3.3.4.2	Generic Traffic Shaping (GTS)	44
3.3.5	Signalling	44
3.3.5.1	Resource reSerVation Protocol (RSVP)	44
3.3.6	Link Efficiency Mechanisms	45
3.3.6.1	Compressed Real-Time Protocol (CRTP)	45
3.3.6.2	Link Fragmentation and Interleaving (LFI)	47
3.4	Codierung des Audiosignals	48
3.4.1	G.711 A law	49
3.4.2	G.723.1	50
3.4.3	G.726	51
3.4.4	G.728	51
3.4.5	G.729	51
3.4.6	Zusammenfassung	51
3.5	Prinzipien des Benchmarking	52
4	Definition der Testszenarien und praktische Versuche	55
4.1	Testszenarien	55
4.2	Testwerkzeuge	55
4.3	Meßwerterfassung	61
4.4	Meßergebnisse	65
4.4.1	Vergleich der QoS-Algorithmen	65
4.4.1.1	FIFO-Queueing	65
4.4.1.2	Weighted Fair Queueing	68
4.4.1.3	IP RTP Priority	70
4.4.1.4	Class-Based Weighted Fair Queueing	71
4.4.1.5	Priority Queueing	79
4.4.1.6	Compressed Real-Time Protocol	83
4.4.1.7	Zusammenfassung	83

4.4.2	Vergleich der Codecs	89
4.4.2.1	G.711 A law	89
4.4.2.2	G.723.1	90
4.4.2.3	G.726	92
4.4.2.4	G.728	92
4.4.2.5	G.729	93
4.4.2.6	Zusammenfassung	94
5	Zusammenfassung	97
A	Dokumentation der Tests	103
A.1	Router-Konfiguration	103
A.1.1	FIFO-Queueing	103
A.1.2	Weighted Fair Queueing	108
A.1.3	IP RTP Priority	113
A.1.4	Class-Based Weighted Fair Queueing	118
A.1.4.1	1 Klasse: Voice - TD	118
A.1.4.2	2 Klassen: Voice - TD, Data - TD	123
A.1.4.3	2 Klassen: Voice - TD, Data - WRED	129
A.1.4.4	2 Klassen: Voice - LLQ, Data - WRED	134
A.1.5	Priority Queueing	140
A.1.6	Priority Queueing + CAR	145
A.1.7	Compressed Real-Time Protocol	151
A.1.8	Codecs	156
A.1.8.1	G.711 A law	156
A.1.8.2	G.723.1	156
A.1.8.3	G.726	157
A.1.8.4	G.728	158
A.1.8.5	G.729	158
A.2	Anlagenbelegung Hicom 300	159
B	Detaillierte Meßergebnisse	160
	Literaturverzeichnis	213

Abbildungsverzeichnis

2.1	Testumgebung	18
3.1	Type of Service Feld einer IPv4 PDU (nach [MM98])	27
3.2	IPv4 Paket (nach [MM98])	28
3.3	Funktionsweise des FIFO-Queueing (nach [Cis00])	32
3.4	Funktionsweise des Weighted Fair Queueing (nach [Cis00])	33
3.5	Funktionsweise des Priority Queueing (nach [Cis00])	35
3.6	Funktionsweise des Custom Queueing (nach [Cis00])	36
3.7	Funktionsweise des Weighted Random Early Detection (nach [Cis00]) .	41
3.8	Funktionsweise der RTP Header Compression (nach [Cis00])	46
3.9	Kaskadierung von Codecs	50
4.1	Netperf Beispielausgabe UDP_STREAM	57
4.2	Netperf Beispielausgabe TCP_STREAM	58
4.3	Netperf Beispielausgabe TCP_CRR	59
4.4	Revidierter Testaufbau	64
4.5	Übersicht Netperf: UDP_STREAM	84
4.6	Übersicht Callanalyse: UDP_STREAM Teil 1	86
4.7	Übersicht Callanalyse: UDP_STREAM Teil 2	86
4.8	Übersicht Netperf: TCP_STREAM	87
4.9	Übersicht Callanalyse: TCP_STREAM	87
4.10	Übersicht Netperf: TCP_CRR	88
4.11	Übersicht Callanalyse: TCP_CRR	88
4.12	Übersicht Codecs	94
A.1	Anlagenbelegung Hicom 300	159

Tabellenverzeichnis

3.1	Zusammenfassung End-to-End QoS-Modelle	26
3.2	IP Precedence Werte (nach [RFC 0791])	28
3.3	Zusammenfassung Congestion Management (nach [Cis00])	39
3.4	Reduzierung der Paketgröße durch RTP Header Compression (nach [Cis00])	47
3.5	Übersicht der verwendeten Codecs (aus [Wal00])	52
4.1	Getestete Netperf Makros	56
4.2	Meßpunkte Callanalyse	60
4.3	Testdurchlauf	61
4.4	Ergebnisse Netperf - unbelasteter Link	65
4.5	Ergebnisse Netperf - FIFO	66
4.6	Ergebnisse Call-Analyse - FIFO	67
4.7	Ergebnisse Netperf - WFQ	68
4.8	Ergebnisse Call-Analyse - WFQ	69
4.9	Ergebnisse Netperf - IP RTP Priority	70
4.10	Ergebnisse Call-Analyse - IP RTP Priority	71
4.11	Ergebnisse Netperf - CBWFQ (1 Klasse: Voice - TD)	71
4.12	Ergebnisse Call-Analyse - CBWFQ (1 Klasse: Voice - TD)	72
4.13	Ergebnisse Netperf - CBWFQ (2 Klassen: Voice - TD, Data - TD) . . .	73
4.14	Ergebnisse Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - TD)	74
4.15	Ergebnisse Netperf - CBWFQ (2 Klassen: Voice - TD, Data - WRED) .	75
4.16	Ergebnisse Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - WRED)	76
4.17	Ergebnisse Netperf - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED)	77
4.18	Ergebnisse Call-Analyse - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED)	78
4.19	Ergebnisse Netperf - PQ	79
4.20	Ergebnisse Call-Analyse - PQ	80
4.21	Ergebnisse Netperf - PQ + CAR	81

4.22	Ergebnisse Call-Analyse - PQ + CAR	82
4.23	Reduzierung der Bandbreite durch CRTP	83
4.24	Bandbreite - Codec G.711 A law	89
4.25	Ergebnisse Call-Analyse - Codec G.711 A law	90
4.26	Bandbreite - Codec G.723.1 5,3k	90
4.27	Ergebnisse Call-Analyse - Codec G.723.1 5,3k	91
4.28	Bandbreite - Codec G.723.1 6,3k	91
4.29	Ergebnisse Call-Analyse - Codec G.723.1 6,3k	91
4.30	Bandbreite - Codec G.726	92
4.31	Ergebnisse Call-Analyse - Codec G.726	92
4.32	Bandbreite - Codec G.728	93
4.33	Ergebnisse Call-Analyse - Codec G.728	93
4.34	Bandbreite - Codec G.729	93
4.35	Ergebnisse Call-Analyse - Codec G.729	94
4.36	Zusammenfassung der Codec-Ergebnisse	95
B.1	Meßwerte Netperf - unbelasteter Link	161
B.2	Meßwerte Netperf - FIFO	162
B.3	Meßwerte Call-Analyse - FIFO (Clock Rate: 125000 bps)	163
B.4	Meßwerte Call-Analyse - FIFO (Clock Rate: 256000 bps)	164
B.5	Meßwerte Netperf - WFQ	167
B.6	Meßwerte Call-Analyse - WFQ (Clock Rate: 125000 bps)	168
B.7	Meßwerte Call-Analyse - WFQ (Clock Rate: 256000 bps)	169
B.8	Meßwerte Netperf - IP RTP Priority	172
B.9	Meßwerte Call-Analyse - IP RTP Priority (Clock Rate: 125000 bps)	173
B.10	Meßwerte Netperf - CBWFQ (1 Klasse: Voice - TD)	174
B.11	Meßwerte Call-Analyse - CBWFQ (1 Klasse: Voice - TD; Clock Rate: 125000 bps)	175
B.12	Meßwerte Call-Analyse - CBWFQ (1 Klasse: Voice - TD; Clock Rate: 256000 bps)	176
B.13	Meßwerte Netperf - CBWFQ (2 Klassen: Voice - TD, Data - TD)	179
B.14	Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - TD; Clock Rate: 125000 bps)	180

B.15 Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - TD; Clock Rate: 256000 bps)	181
B.16 Meßwerte Netperf - CBWFQ (2 Klassen: Voice - TD, Data - WRED) .	184
B.17 Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - WRED; Clock Rate: 125000 bps)	185
B.18 Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - WRED; Clock Rate: 256000 bps)	186
B.19 Meßwerte Netperf - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED) .	189
B.20 Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED; Clock Rate: 125000 bps)	190
B.21 Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED; Clock Rate: 256000 bps)	191
B.22 Meßwerte Netperf - PQ	194
B.23 Meßwerte Call-Analyse - PQ (Clock Rate: 125000 bps)	195
B.24 Meßwerte Call-Analyse - PQ (Clock Rate: 256000 bps)	196
B.25 Meßwerte Netperf - PQ + CAR	199
B.26 Meßwerte Call-Analyse - PQ + CAR (Clock Rate: 125000 bps)	200
B.27 Meßwerte Call-Analyse - PQ + CAR (Clock Rate: 256000 bps)	201
B.28 Meßwerte Netperf - PQ + CRTP	204
B.29 Meßwerte Netperf - Codec G.711 A law	205
B.30 Meßwerte Call-Analyse - Codec G.711 A law	205
B.31 Meßwerte Netperf - Codec G.723.1 5,3k	206
B.32 Meßwerte Call-Analyse - Codec G.723.1 5,3k	206
B.33 Meßwerte Netperf - Codec G.723.1 6,3k	207
B.34 Meßwerte Call-Analyse - Codec G.723.1 6,3k	207
B.35 Meßwerte Netperf - Codec G.726	208
B.36 Meßwerte Call-Analyse - Codec G.726	208
B.37 Meßwerte Netperf - Codec G.728	209
B.38 Meßwerte Call-Analyse - Codec G.728	209
B.39 Meßwerte Netperf - Codec G.729	210
B.40 Meßwerte Call-Analyse - Codec G.729	210
B.41 Übersicht Faxübertragungen	211

Abkürzungsverzeichnis

ACELP	Algebraic Codebook-Excited Linear Predictive Coding
ACM	Association for Computing Machinery
ACL	Access Control List
ACR	Absolute Category Rating
A/D(C)	Analog Digital (Converter)
ADPCM	Adaptive Difference PCM
ATM	Asynchronous Transfer Mode
CAR	Committed Access Rate
CaTV	CableTV
CBWFQ	Class-Based Weighted Fair Queueing
CELP	Codebook-Excited Linear Predictive Coding
Codec	Coder/Decoder
COPS	Common Open Policy Services
CoS	Classes of Service
CQ	Custom Queueing
CRC	Cyclic Redundancy Check
CRTP	Compressed Real-Time Protocol
CS-ACELP	Conjugate Structure ACELP
CTI	Computer Telephony Integration
D/A(C)	Digital Analog (Converter)
DGL	Differentialgleichung
DiffServ	Differentiated Services
DoS	Denial of Service
DTMF	Dial Tone Multi Frequency
ETSI	European Telecommunications Standards Institute
FB-WRED	Flow-Based WRED
FIFO	First In First Out
FR	Frame Relay
FTP	File Transfer Protocol
GSM	Global System for Mobile communication
HTTP	HyperText Transfer Protocol

IETF	Internet Engineering Task Force
IMTC	International Multimedia Teleconferencing Consortium
IntServ	Integrated Services
IP	Internet Protocol
IPv4	IP Version 4
IPv6	IP Version 6
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ITG	Informationstechnische Gesellschaft
ITSP	Internet Telefonie Service Provider
ITU	International Telecommunications Union
ITU-T	International Telecommunications Union – Telecommunication Standardization
LAN	Local Area Network
LCR	Least Cost Router
LD-CELP	Low Delay CELP
LFI	Link Fragmentation and Interleaving
MAC	Media Access Control
MCML	Multi-Class extension to Multi-Link PPP
MCU	Multipoint Control Unit
MLP	Multilink Point-to-Point Protocol
Modem	Modulator/Demodulator
MOS	Mean Opinion Score
MP-MLQ	Multi Pulse excitation with a Maximum Likelihood Quantizer
MTU	Maximum Transmission Unit
NVRAM	Non-Volatile Random Access Memory
OSI	Open Systems Interconnection
PBR	Policy-Based Routing
PCM	Pulse-Code Modulation
PDU	Protocol Data Unit
PPP	Point-to-Point Protocol
PQ	Priority Queueing
PRI	Primary Rate

QoS	Quality of Service
QSIG	Unified International corporate network signalling standard
RED	Random Early Detection
RFC	Request for Comments
RSVP	Resource reSerVation Protocol
RTCP	RTP Control Protocol
RTP	Real-Time Protocol
RTSP	Real-Time Streaming Protocol
SNR	Signal to Noise Ratio
SS7	Signalisierungssystem Nr. 7
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
Tiphon	Telecommunications und Internet Protocol Harmonization Over Network
TK	Telekommunikation
ToS	Type of Service
UDP	User Datagram Protocol
VAD	Voice Activity Detection
VoATM	Voice over ATM
VoFR	Voice over Frame Relay
VoIP	Voice over IP
WAN	Wide Area Network
WFQ	Weighted Fair Queueing
WRED	Weighted Random Early Detection
WWW	World Wide Web
xDSL	Digital Subscriber Line Technologien
B_c	Committed Burst Size (Burst Größe)
CIR	Committed Information Rate (mittlere Datenrate)
T_c	Time Interval (Zeitintervall)
t_x	Zeitpunkt t

Kapitel 1

Vorwort

In der heutigen Zeit ist eine bestmögliche Kommunikation innerhalb eines Unternehmens Voraussetzung, um am aggressiven Weltmarkt zu überleben. Unter dem Schlagwort VoIP gewinnt dabei eine Technologie zur Integration von Sprache und Video in bestehende Datennetze stark an Bedeutung. Es werden derzeit große Anstrengungen unternommen, um VoIP am Markt zu etablieren, verspricht es doch enorme Einsparungen bei den Kommunikationskosten einzelner Unternehmensstandorte sowie eine vielfach höhere Funktionalität, die vollkommen neue Anwendungsgebiete erschließt.

Inhalt dieser Arbeit ist der Test und die Analyse einer VoIP Umgebung. Nach einigen einleitenden Worten zu den Vorteilen und Nutzen der Technik werden Anforderungen an bestehende Datennetze zu Integration von VoIP wiedergegeben. Es werden theoretische Grundlagen (Service-Modelle, Queueing-Algorithmen, Codecs usw.) für den darauf folgenden Test erläutert. Der detaillierten Beschreibung der Versuche und erzielten Ergebnisse schließt sich die Zusammenfassung an, die zeigt, ob VoIP den hohen Erwartungen gerecht wird.

Abstract

Today only the company with the best communication between its teams can survive on the aggressive market. A new technology for the integration of voice and video in existing data networks attracts interest. It is known as VoIP and is pushed into the market at the moment, because it promises huge savings of costs in the fields of interior communication of a firm and a higher functionality, which is leading to new fields of application.

This document contains the testing and analysis of a VoIP environment. After presenting the advantages of this technique, this paper will focus on the demands on data networks to integrate VoIP. Theoretical basics like service models, queueing-algorithms,

codecs etc. will be explained. The document continues with a description of the experiment. The summary shows whether VoIP will come up to its expectations.

Kapitel 2

Motivation

Zur Zeit erfährt die Unternehmenslandschaft einen Umbruch zum verteilten Arbeiten (einschließlich Homeworking) – dem Internetworking. Hierbei spielt die Kommunikation untereinander eine große Rolle. Da zusammengehörende Teams weit voneinander entfernt arbeiten können, ist eine leistungsfähige Kommunikationsinfrastruktur vonnöten.

Das Telefonnetz ist ein System von verbindungsorientierten Punkt- zu-Punkt Verbindungen zwischen zwei Endgeräten (z.B. Telefonen). Die Entwicklung ging von komplett physikalischen Verbindungen zu logischen Verbindungen über die Telefoninfrastruktur über. Die Vorteile der leitungsorientierten Technik sind fehlende Bandweitenengpässe und relativ geringe Verzögerungen. Nachteilig ist die ursprüngliche einseitige Ausrichtung auf Sprachübertragungen. Somit ist das Telefonnetz nicht in der Lage, andere Verkehrstypen effizient zu übermitteln.

Heute erfolgt bei der Sprachdatenübertragung aus wirtschaftlichen Gründen ein Wandel vom leitungsvermittelnden Sprachnetz zum paketvermittelnden Sprach-/ Datennetz, da in paketorientierten Netzen die zur Verfügung stehende Bandbreite effektiver ausgenutzt wird. Die Sprachübertragung in paketvermittelnden Netzen birgt neue Probleme (näheres in Kapitel 3), die bei der traditionellen, leitungsgebundenen Übertragungsform nicht auftreten.

Aktuelle Netzwerke stützen sich auf das Internet Protokoll (IP). Mit Hilfe von IP und deren übergeordneten Transportprotokollen TCP und UDP können heute sämtliche Rechner in einem Netzverbund (LAN und WAN) miteinander kommunizieren. Das IP-Protokoll hat die Kommunikationswelt derart durchdrungen, daß andere Protokolle nur noch eine untergeordnete Rolle spielen. Die Sprachvermittlung über IP-Netze gibt dieser Technik ihren Namen – Voice over IP (VoIP).

Welche Möglichkeiten und Vorteile bietet die Integration von Daten und Sprache in

einem Netz? Unternehmen können (je nach Einsatzgebiet) Kosten einsparen. VoIP ermöglicht eine Doppelnutzung vorhandener Infrastruktur – die Netzstruktur des Intranets – für den Informationsaustausch einerseits und der Kommunikation zwischen den Mitarbeitern und dem Umfeld der Firma andererseits. Da Telefongespräche zwischen einzelnen Standorten über das unternehmenseigene Datennetz übertragen werden können, fallen keine zusätzlichen Gesprächsgebühren (Toll Bypass) an eine Telefongesellschaft an.

Die Ersparnis fällt umso höher aus, je größer die Zahl der Firmenstandorte ist, die bereits über Datenleitungen miteinander verbunden sind. Bei weltweit tätigen Unternehmen sind davon auch Auslandsgespräche betroffen. Installiert ein Unternehmen weiterhin Gateways¹ an seinen Standorten, lassen sich zusätzlich die Kosten für Gespräche ins öffentliche Fernsprechnetz senken. Dabei wird ein Gespräch zunächst über das unternehmenseigene Datennetzwerk vom Arbeitsplatz des Mitarbeiters einer Filiale bis zum Gateway eines anderen Standortes geführt. Das Gateway ruft dann den Gesprächspartner im Ortsnetz an und verbindet den Teilnehmer mit dem Mitarbeiter der entfernten Filiale. Das Gespräch kostet das Unternehmen demnach nur die Verbindungsgebühren zum Ortstarif (Flächendeckung vorausgesetzt), plus die Kosten der eigenen Datenleitungen. Ein Least Cost Router (LCR) Server bestimmt hierbei, wo sich der Austrittspunkt aus dem Netz mit dem geringsten Kostenaufkommen befindet. VoIP führt im dargestellten Beispiel folglich zum teilweisen oder kompletten Wegfall von Ferngesprächen innerhalb des Landes und bei internationalen Unternehmen zur Drosselung der Kosten von Auslandsgesprächen. Ein vorhandenes eigenes Sprachnetz läßt sich durch Intranet-Telefonie ersetzen. In Neubauten läßt sich eine TK-Anlage komplett einsparen. Weiterhin sind die Installationskosten bei der Verkabelung der hausinternen Kommunikationsinfrastruktur geringer, da nur noch ein Anschluss zum Arbeitsplatz geführt werden muß.

Der zusätzliche Kostenaufwand durch Einrichtung und Betrieb einer VoIP-Lösung soll nicht verborgen werden. Neben Anschaffung von IP-Telefonie-Gateways, entsprechender Ausrüstung der Arbeitsplätze sowie Wartungs- und Administrationskosten für zusätzliches Equipment muß vor allem der Erweiterung der Bandbreite² des Datennetzes Aufmerksamkeit geschenkt werden.

¹Übergang zwischen firmeninternen Datennetz und dem öffentlichen Telefonnetz

²Dabei ist zu beachten, daß die Bandbreite eines einzelnen ISDN-B-Kanals zur Übertragung von bis zu 10 Gesprächen bei entsprechender Komprimierung (siehe dazu Kapitel 3.4) ausreicht.

In Anbetracht stetig sinkender Preise für Telefonverbindungen ist eine individuelle Analyse der Unternehmenskommunikation und der zur Verfügung stehenden Infrastruktur für eine Wirtschaftlichkeitsrechnung der Intranet-Telefonie unbedingt zu empfehlen. Die durch Einsatz von VoIP verursachten Kosten werden durch die genannten Synergieeffekte in den meisten Fällen wieder eingespielt.

Dem aufkommenden Trend verstärkt Heimarbeiter einzusetzen, kann ebenfalls mit VoIP Rechnung getragen werden. So sind zusätzlich zur Telefonie Videokommunikation, Application Sharing und Konferenzschaltungen möglich. Implementiert ein Gateway zusätzlich klassische Remote-Access- und Router-Funktionen, läßt es sich für die Anbindung von Home-Offices und Außendienstmitarbeitern einsetzen. Ein Homeworker kann somit auf eine virtuelle Büroumgebung zugreifen. Neben dem reinen Abruf von Daten erfolgt die Kommunikation mit dem Team per VoIP. Über das in der Zentrale des Unternehmens installierte System kann der Anwender von seinem Rechner aus mit allen Firmenressourcen kommunizieren. Telefonkonferenzen können zwischen Computer-Telefonie-Arbeitsplätzen in der Firma, Heimarbeitsplätzen und normalen Telefonapparaten geschaltet werden. Durch den Ausbau der Access-Systeme mit xDSL- oder CaTV-Technologien steht für die genannten Anwendungen genügend Bandbreite zur Verfügung.

Die letzten Jahre waren dadurch geprägt, daß die alten dienstspezifischen Netze durch ein digitales dienstintegrierendes Netz (ISDN) ersetzt wurden. Es war nun nicht mehr nötig, für die verschiedenen elektronischen Dienste verschiedene Endgeräte an verschiedene Telekommunikations- oder Datennetze anzuschließen, sondern ein universelles Terminal reichte dafür aus. ISDN stellte den ersten Schritt der Dienstintegration dar. VoIP ist nun eine weitere Möglichkeit die Integration mehrerer Dienste voranzubringen. In diesem Zusammenhang ist der Mehrwertdienst Unified Messaging zu erwähnen. Es steht für die Gleichbehandlung aller eingehenden Nachrichten wie Telefon, Fax, E-Mail. Eingehende Gespräche werden durch ein Gateway an den jeweiligen Arbeitsplatz³ verteilt. Bei Nichterreichen des Teilnehmers wird ihm intern eine elektronische Nachricht zugestellt.

Die Sprache wird im Netz wie eine Sonderform der Daten behandelt. VoIP-Lösungen sind durch die verwendeten Sprachkompressions-Algorithmen in der Lage, Sprache bei gleicher Qualität mit wesentlich geringeren Bitraten als traditionelle Systeme zu über-

³Dessen Standort ist variabel.

tragen. Die Vereinigung der Sprachkommunikation mit der Datenverarbeitung führt zu vollkommen neuen Anwendungsgebieten. Als Beispiele seien Sprachübermittlung über das Internet bzw. Intranets, Computer Telephony Integration (CTI), Web-basierte Call Center, Messaging via Web, Collaborative Whiteboarding und das Remote Teleworking genannt.

Durch die Kombination verschiedener Medientypen wird die Kommunikation vielfältiger. Die komplizierte Beschreibung eines Sachverhaltes in Textform kann durch die Übertragung von anschaulicheren Bildern oder Videos vereinfacht werden.

Mit den Internet Telefonie Service Providern (ITSP) etablieren sich hoch spezialisierte Service Provider. Merkmal solcher Provider ist die Vermarktung von High Speed IP Netzen mit einer geringen Ende-zu-Ende-Verzögerung zur Übermittlung von Sprache und Fax-Informationen.

Neue Technologien müssen vor dem Praxiseinsatz verifiziert werden. Die Firma Siemens ist deswegen an das ComLab, das Labor für Kommunikationssysteme im Institut für Nachrichtentechnik und Informationselektronik der Universität Rostock, herangetreten. Im Rahmen dieser Arbeit soll mit einer von Siemens und der Firma Cisco bereitgestellten Testumgebung die Analyse und der Test einer VoIP Umgebung erfolgen. Die Teststellung (siehe Abbildung 2.1) ist auf ihre Möglichkeiten und Grenzen in Bezug auf die parallele Übertragung von Sprache und Daten über eine gemeinsame Übertragungsstrecke zu untersuchen.

Die Siemens Hicom 300 ist eine herkömmliche Telefonanlage. Die Ausstattung der Hicom kann dem Anhang A.2 entnommen werden. Sie wird dual-ported betrieben, d.h. sie stellt logisch 2 voneinander unabhängige TK-Anlagen dar, die über eine VoIP-Strecke gekoppelt sind. Die Verbindung besteht aus 2 voice-fähigen Routern und 2 Siemens WAN-Modems. Zwischen den Modems wird eine herkömmliche E1 Standardleitung mit einer Bruttobitrate von $2,048 \frac{\text{Mbit}}{\text{s}}$ nachgebildet.

Die Telefon-Gateways sind wichtige Komponenten in der gesamten VoIP-Lösung. Gateways sorgen für die Anpassung der verbindungsorientierten Telefonnetze an die paketorientierten Datennetze. Ein VoIP-Gateway bietet die folgenden Funktionen: Interfaces zum Anschluß von Telefonnebenstellenanlagen bzw. des öffentlichen Telefonnetzes, Signalisierungsfunktionen (Verbindungsauf- bzw. -abbau, z.B. per QSIG), Rufumleitung, Sprachkompression/Dekompression in Echtzeit, Verpacken bzw. Entpacken der komprimierten Sprache, Interfaces zum IP-Netz und Funktionen zur Abrechnung. Diese

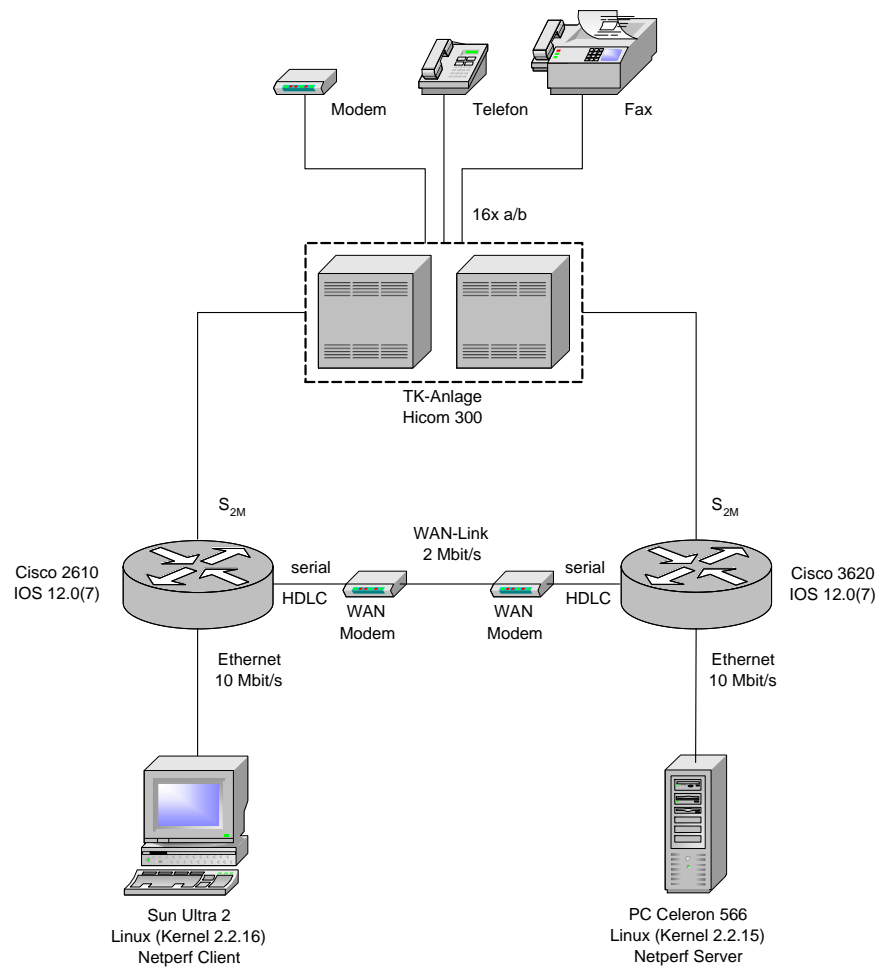


Abbildung 2.1: Testumgebung

Dienste werden in der Teststellung durch die Cisco Router C2610 und C3620 bereitgestellt.

Die Router sind per S_{2M} -Anschluß mit den beiden Ports der Hicom verbunden. An den Ethernet-Interfaces der Router sind 2 Rechner angeschlossen, die ein Unternehmensnetzwerk simulieren.

Kapitel 3

Theoretische Grundlagen

Wie die leitungsvermittelnden Netze ursprünglich auf die Übertragung von Sprache ausgerichtet waren, sind paketvermittelnde Netze von sich aus nicht in der Lage, Telefongespräche in ansprechender Qualität zu übermitteln. Der Grund dafür ist in den unterschiedlichen Übertragungscharakteristika zu suchen.

Bei der herkömmlichen Telefonie wird das Sprachsignal 8000 mal pro Sekunde mit einer Auflösung von 8 bit^1 abgetastet. Daraus ergibt sich die Datenrate von $64 \frac{\text{kbit}}{\text{s}}$. Es entsteht folglich ein Datensignal konstanter Bitrate. Weiterhin ist bekannt, daß die Qualität der Sprachübermittlung rapide mit steigender Verzögerung und Variation der Verzögerung (Jitter) abnimmt. Die Übertragung von Audiosignalen ist somit als isochroner oder zeitkritischer Prozeß einzuordnen. [Wal00]

Demgegenüber steht die Übertragungscharakteristik der Paketvermittlung. Daten werden periodisch (Bursts) versendet. Es wechseln sich somit Zeitabschnitte niedriger und hoher Netzlast ab. Bei den paketerorientierten Netzen werden die Informationen in diskrete digitale Pakete zerlegt und mit Hilfe von zusätzlichen Übertragungsprotokollen über das Netzwerk übertragen. Anhand der in den Headern der Pakete enthaltenen Adressinformationen übermitteln die Netzknoten die Pakete über das Netz. Bei der Paketübermittlung wird für jedes Paket zwischen Sender und Empfänger die optimale Route ermittelt und das Paket anhand dieser Vorgaben übertragen. Somit können in einem Übertragungsweg bestimmte Störfaktoren wie beispielsweise ungenügende Bandbreite, Ausfälle des Paketweges usw. berücksichtigt werden. Aufgrund der verschiedenen Wege treten unterschiedliche Laufzeiten bzw. Verzögerungszeiten zwischen Sender und Empfänger auf. Da die Daten nicht mehr sequentiell über eine bestimmte Verbindung übertragen werden, müssen die Datenströme beim Empfänger wieder zu einem kontinuierlichen Datenstrom zusammengesetzt werden.

¹Entsprechend 256 Auflösungsstufen

Bei der Sprachübertragung über paketvermittelnde Netze wird das Audiosignal in Blöcke zerlegt. Deren Größe² liegt im *ms* Bereich. Eine Gruppe von Samples wird mit einem Header (Adreßinformationen usw.) versehen und das ganze Paket über das Netzwerk verschickt. Aufgrund der asynchronen Datenübertragung werden die Sprachpakete mit unterschiedlicher und zeitlich schwankender Verzögerung beaufschlagt, die der Empfänger durch Puffer kompensieren muß, mit denen wiederum die Signalverzögerung erhöht wird. Der Einsatz von VAD³, mit denen sich in den Sprachpausen zwischen den Talkspurts die Datenrate auf dem Übertragungskanal stark reduzieren läßt, führen zusammen mit der Signalpufferung in fast allen Fällen zu zeitlichen Verschiebungen der Talkspurts untereinander. Derartige Verschiebungen mindern die Sprachqualität stark. Blockverluste infolge periodisch auftretender hoher Netzauslastung rufen beim Einsatz von komprimierenden Sprachcodecs (Kapitel 3.4) starke Signalverzerrungen hervor.

Unter Berücksichtigung dieser Gegebenheiten ist bei paralleler Übertragung von synchronen und asynchronen Daten eine Priorisierung der synchronen Daten (Sprache) gefordert. Mittels Quality of Service (QoS) kann eine bestimmte Dienstleistungsgüte zwischen Sender und Empfänger garantiert werden. Mit Voice over ATM (VoATM) und Voice over Frame Relay (VoFR) sind bereits Techniken etabliert, die QoS implementieren. Mit der immer größer werdenden Verbreitung von IP vollzieht sich ein Übergang von den verbindungsorientierten VoATM und VoFR zum verbindungslosen VoIP. Die Funktionalität von VoATM und VoFR befinden sich innerhalb des ISO/OSI Referenzmodells vor allem in Schicht 2. Durch die Konzentration auf das in Layer 3 angesiedelte IP werden darunter liegende Protokolle und Techniken hier nicht berücksichtigt.

Das in den Netzwerken vorherrschende Routed-Protokoll IP unterstützt von sich aus kein QoS, sondern bietet einen reinen Datagrammdienst, d.h. IP übernimmt nur den Transport von Paketen. Daher existieren keinerlei Flußkontrolle, Sicherheiten bezüglich Bandbreite und Verzögerung der Sprachpakete oder eine Fehlerkorrektur. Diese Aufgaben müssen höhere Protokollschichten übernehmen. [RFC 0768], [RFC 0791], [RFC 0793], [RFC 1889], [RFC 2700]

Für die Sprachübertragung müssen dementsprechend alle beteiligten Verkehrsknoten den folgenden Punkten genügen:

²Die Wahl der Größe ist ein schwieriger Prozeß, da zwischen Effektivität (Größe Header – Größe Sprachblock) und Verzögerung (zeitliche Diskretisierung des Signals \Rightarrow großer Sprachblock bedeutet hohe Verzögerung) ein Kompromiss eingegangen wird.

³In Sprachpausen werden keine Daten übertragen.

- Unterstützung garantierter Bandbreite
- Verbesserung der Verlustcharakteristik
- Vermeidung bzw. Management von Netzüberlastung
- Regulierung des Netzverkehrs (policing and shaping)
- Netzwerkweites Priorisieren von Verkehrsklassen

Bei der Netzwerkplanung ist der Rechenaufwand für die QoS-Algorithmen in den Routern usw. zu berücksichtigen.

Standards bilden die Grundlage für Interoperabilität und die Entwicklung eines Marktes im Bereich VoIP. In den heute verfügbaren Lösungen für die IP-Telefonie wurde von allen Herstellern der ITU H.323 Standard integriert. Dieser Standard definiert Funktionen und Dienste zur Übertragung von Informationen (per Terminal) über lokale Netze, die über keinerlei Quality of Service-Mechanismen verfügen. Die H.323 Terminals und Systeme sind in der Lage, sowohl Sprache, Daten als auch Video zu übermitteln. Dabei ist es unerheblich, ob die Informationen über nur ein LAN-Segment oder einer Kombination von Netzen (LAN, WAN, Internet) übertragen werden. Die H.323 Terminals können in PCs wie auch Videotelefonen integriert werden. Der H.323 Standard beinhaltet eine Reihe anderer Standards, wie z.B. H.225.0 (Paketsynchronisation), H.245 (Kontrollfunktionen), H.261 und H.263 (Video Codecs), G.711, G.723, G.726, G.728 und G.729 (Audio Codecs) und die T.120 Reihe der Multimedia Protokolle. Innerhalb dieser Spezifikationen werden eine Reihe neuer Netzkomponenten (H.323 Terminals, H.323 Multipoint Control Units (MCU), H.323 Gatekeeper und H.323 Gateways) definiert, die mit anderen spezifikationskonformen Komponenten über ein H.323 Gateway kommunizieren. Zusätzlich zur ITU werden von folgenden Organisationen weitere Standards für die IP-Telefonie erarbeitet:

- International Multimedia Teleconferencing Consortium (IMTC) ⇒ Vorschläge gingen in die ITU H.320 Standard ein
- European Telecommunications Standards Institute (ETSI) ⇒ „Telecommunications und Internet Protocol Harmonization Over Network“ (Tiphon) sorgt für die Anpassung der unterschiedlichen Sprachnetze (VoIP - ISDN/GSM)

- Internet Engineering Task Force (IETF) \Rightarrow Type of Service (ToS, siehe Abschnitt 3.3.1.1) Funktionen, Common Open Policy Services (COPS) Dienste für regelbasierte Netze, Abbildung der Protokolle des Signalisierungssystems Nr. 7 (SS7) in IP-Funktionen

Die folgenden Abschnitte behandeln zum Verständnis der im Kapitel 4 eingesetzten QoS-Techniken deren theoretische Grundlagen.

3.1 Einleitung

Innerhalb der Wirtschaftlichkeitsrechnung können mit der Analyse der Unternehmenskommunikation die benötigten Ressourcen für den Einsatz von VoIP bestimmt werden. Die Nachrichtenverkehrstheorie stellt die dazu erforderlichen Rechenmittel bereit. Sie beschäftigt sich mit der Berechnung und Bewertung von Verkehrsbeziehungen in Kommunikationssystemen. Hierbei nimmt die Wahrscheinlichkeitstheorie einen großen Stellenwert ein. Die folgenden Erkenntnisse gehen auf Studien der Mathematiker Erlang und Markow zurück.

Hauptwerkzeug der Nachrichtenverkehrstheorie ist die Bedienungstheorie. Die Anforderungen an ein Kommunikationssystem sind abhängig vom Verkehrsaufkommen, welches bestimmt werden muß. Im Allgemeinen kann ein Kommunikationssystem in eine Anforderung (z.B. ein Anruf oder ein Paket) und den Bedienprozeß (eine Telefonanlage nimmt den Ruf an oder das Paket wird durch einen Router bearbeitet) unterteilt werden. Solche Systeme werden als reine Verlustsysteme bezeichnet, da z.B. eine ausgelastete TK-Anlage keine Anrufe mehr annimmt. Enthält ein System einen zusätzlichen (endlichen) Wartespeicher (z.B. zur Pufferung des Paketes bei Vollausslastung des Routers) wird es als Warte-(Verlust-) System charakterisiert. Zur Bewertung eines Kommunikationssystems können die nachstehenden Fragen herangezogen werden:

- Wieviele Bedienprozesse kann ein gemeinsam genutztes Bediensystem bei angemessener Antwortzeit aufnehmen?
- Wie hoch ist die Blocking Rate eines solchen Systems? Welche Verbesserung bringt eine Erweiterung des Systems? Welche Art der Erweiterung bringt den meisten Gewinn?
- Welches ist die beste Architektur für ein gemeinsam genutztes Bediensystem?

Derartige Queueing-Systeme werden anhand des Kendallschen Klassifikationsschemas für einstufige Bediensysteme unterschieden. In der erweiterten Form A/B/N/S stehen A für den stochastischen Typ des Ankunftsprozesses, B für den stochastischen Typ des Bedienprozesses, N für die Anzahl der Bedieneinheiten und S für die Anzahl der Warteplätze (optional). Für A (Abstandsverteilungen der ankommenden Forderungen) und B (Bedienzeitverteilungen) können z.B. stehen:

- D : Deterministisch
- G : Beliebig (General)
- M : Negativ-exponentiell (Markow)

Markow-Systeme werden durch einen Poisson-Ankunftsprozeß⁴ und einer Exponentialverteilung als Wartedauerverteilung beschrieben. Sie sind außerdem durch ihr fehlendes Gedächtnis charakterisiert, da der aktuelle Zustand im Zeitpunkt t_0 nur vom Zustand im Zeitpunkt t_{-1} und nicht von weiter zurückliegenden Zuständen abhängt.

VoIP Systeme sind als geschlossenes Warte-Verlust-System (M/M/N/S) mit Prioritäten zu klassifizieren, da nur endlicher Speicher zur Pufferung in Überlastungssituationen zur Verfügung steht. Geschlossen bedeutet, daß die Zahl der Teilnehmer am System endlich ist. Die Priorisierung ist bedingt durch das von VoIP benötigte QoS.

Die Bestimmung der Zustandswahrscheinlichkeiten erfolgt im Allgemeinen per DGL oder Markow Graphen. Die Berechnungsgrundlagen sind komplex und können der Literatur entnommen werden. [ITG97] [Rob94] [Roc00]

3.2 End-to-End QoS-Modelle

Die Fähigkeit eines Netzwerkes, gewünschte Dienstmerkmale von einem Ende zum anderen zu liefern, wird als End-to-End QoS bezeichnet. Dessen Möglichkeiten werden durch Service-Modelle charakterisiert. Es existieren 3 Service-Modelle: Best-Effort Service, Integrated Service, Differentiated Service.

⁴Der Poisson Prozeß ist ein rein zufälliger Ankunftsprozeß.

3.2.1 Best-Effort Service

Best-Effort Service stammt aus den Anfangszeiten des Internet, in der Datenübertragung und Fernzugriff (remote access) die Hauptanwendungen waren. Diesen sind die geringen Anforderungen an zeitliche Bedingungen gemein, die zu einem simplen und skalierbaren Netzwerkdesign führten. Das Modell besitzt nur ein Dienstmerkmal: das Übertragen von Daten. Anwendungen können Daten ohne Einschränkungen in Hinblick auf Uhrzeit, Dauer, Menge oder Zugriffskontrollen senden. Das Netz überträgt die Daten, wenn möglich, ohne jegliche Zusicherung bezüglich Verlässlichkeit, maximale Verzögerungszeiten oder Bandbreite.

Mit der Zeit wuchs das Internet und somit die Zahl der Nutzer, die einen immer größeren Bereich an Anwendungen nutzen wollten. Hier stößt Best-Effort Service an seine Grenzen, da die Vielzahl möglicher Anwendungen an das Netzwerk unterschiedliche Anforderungen bei der Übertragung stellen. Zudem lassen sich die entstehenden Massen von Paketen nicht mehr effizient übermitteln. Deshalb wurden Wege gesucht, um einerseits eine stark wachsenden Zahl von Nutzern und zu übertragenden Datenmengen zu bewältigen. Andererseits muß es innerhalb des Modells möglich sein, verschiedene Datentypen (z.B. Sprache und Daten im klassischen Sinn) oder Nutzer voneinander unterscheiden und bestimmte Daten/Nutzer priorisieren zu können. Daher wurden neue Ansätze entwickelt, die diese Gegebenheiten berücksichtigen.

3.2.2 Integrated Services (IntServ)

Die Integrated Services-Architektur stellt Dienstklassen zur Verfügung, die an die Anforderungen verschiedener Anwendungen angepaßt wurden. IntServ läßt sich in Bezug auf eine maximale Verzögerung (delay-bound) in 3 Klassen unterteilen:

1. Garantierte Dienstklasse (ermöglicht Dienstvereinbarungen mit definierter maximaler Latenz \Rightarrow für zeitkritische und gegenüber Verzögerung empfindlich reagierende Applikationen)
2. Dienstklasse kontrollierter Belastung (ermöglicht Dienstvereinbarungen mit mittlerer Verzögerung, d.h. Überschreitungen der vereinbarten maximalen Latenzzeit finden nicht häufiger statt als im unbelasteten Zustand \Rightarrow für zeitkritische aber gegenüber Verzögerung robustere Anwendungen)

3. Best-Effort Service mit weiterer Unterteilung (für adaptive und zeitunkritische Applikationen):

- burst-artiger Verkehr (z.B. Web)
- bulk-artiger Verkehr (Streaming, FTP)
- asynchroner Verkehr (z.B. E-Mail)

Die Punkte 1. und 2. benötigen Signalisierung und Zugangskontrolle (admission control) in den Netzwerkknoten, wie es z.B. durch RSVP (siehe Kapitel 3.3.5.1) bereitgestellt wird.

Im IntServ-Modell fordert eine Anwendung ein spezielles Dienstmerkmal⁵ im Netz an (signaling), bevor es seine Daten sendet. Von der Anwendung wird erwartet, daß es die Daten erst sendet, wenn es eine Bestätigung durch das Netzwerk erhalten hat und die Daten dem innerhalb der Signalisierungsphase angegebenen Profil entsprechen. Sind diese Punkte erfüllt, führt das Netzwerk aufbauend auf den Informationen der Anwendung und den vorhandenen Netzwerkressourcen eine Zugangskontrolle (admission control) durch. Das Netz verpflichtet sich, die QoS-Anforderungen solange zu erfüllen, wie der Verkehr innerhalb seiner Spezifikation verbleibt.

Vorteile der Integrated Services-Architektur ist die Unterteilung in mehrere Klassen, die für bestimmte Anwendungstypen optimiert sind sowie der integrierte Best-Effort Service. So können existierende Anwendungen ohne Veränderung weiter verwendet werden. Gegen IntServ spricht, daß End-to-End Merkmale nur nutzbar sind, wenn alle beteiligten Knoten das Service-Modell integrieren. [MEH00], [RFC 1633]

3.2.3 Differentiated Services (DiffServ)

DiffServ ist ebenfalls ein Service-Modell mit mehreren Dienstklassen. Im Unterschied zu IntServ wird keine explizite Signalisierung durchgeführt, bevor eine Anwendung Daten sendet. Stattdessen wird (i.A. durch einen Provider) den einzelnen IP-Paketen ihre Priorität zugewiesen. Das kann z.B. anhand der IP-Precendence-Bits (siehe Abschnitt 3.3.1.1) im IP-Header oder durch Quell- und/oder Zieladresse geschehen. Auf diese Weise lassen sich viel feiner unterteilte Dienstklassen bewerkstelligen. DiffServ wird

⁵Dienstmerkmale sind z.B. die benötigte Bandbreite oder Anforderungen an die maximale Verzögerung eines Paketes.

vor allem für ein End-to-End QoS sowie die Übertragung kritischer Daten, wie synchrone oder Steuer-Daten verwendet. Gegenüber IntServ bietet DiffServ die folgenden Vorteile:

1. Die fehlende Signalisierungsphase bewirkt einen schnelleren Aufbau einer DiffServ-Verbindung.
2. Durch die hohe Anzahl von Dienstklassen läßt sich eine Anwendung eher in eine bestimmte Klasse einordnen.

Aus dem letztgenannten Punkt ergibt sich auch ein entscheidender Nachteil gegenüber den genannten Verfahren, da es ein komplexer Vorgang ist, derart viele Dienstklassen mit verschiedenen QoS-Merkmalen simultan im selben Netzwerk bereitzustellen. [MEH00], [RFC 2474], [RFC 2475]

Merkmal	Best-Effort Service	IntServ	DiffServ
QoS Unterstützung	nein	ja	ja
Anzahl Dienstklassen	1	3	> 6
Signalisierung	nicht erforderlich	erforderlich	nicht erforderlich
Administrierbarkeit	einfach	mittel	komplex

Tabelle 3.1: Zusammenfassung End-to-End QoS-Modelle

3.3 Quality of Service

Für die gleichzeitige Übertragung von synchronen Daten⁶ und burst-artigem Verkehr ist eine Priorisierung der synchronen Daten nötig. Quality of Service (QoS) stellt Mechanismen bereit, um bestimmten Verkehrsklassen Garantien in Bezug auf Bandbreite und Verzögerung zu geben. Verkehrsklassen stellen verschiedene Prioritätsebenen dar. Sie werden auch als Classes of Service (CoS) bezeichnet.

In den nächsten Kapiteln wird ein Überblick über die QoS-Algorithmen gegeben, die in den Routern C2610 und C3620 implementiert worden sind. Alle QoS-Features

⁶z.B. Sprache oder Video

sind auf den gestellten Routern in Software umgesetzt. Das hat zwar einen höheren Rechenaufwand zur Folge, bietet aber den Vorteil der schnellen Beseitigung von Fehlern (z.B. entdeckte Sicherheitslücken) sowie die einfache Einbringung neuer Funktionen. [Cis00]

3.3.1 Classification

Der Datenverkehr lässt sich mittels Classification in verschiedene Classes of Service unterteilen. Den CoS können dann unterschiedliche QoS-Merkmale zugeordnet werden. Verkehr kann anhand folgender Merkmale klassifiziert werden:

- durch IP Precedence gesetzte Priorität
- physischer Port am Router
- Quell- und/oder Zieladresse (IP oder MAC)
- Protokolltyp und Portnummer
- durch Access-Listen definierte Merkmale

3.3.1.1 IP Precedence

Mit IP Precedence ist es möglich, Daten in 6 verschiedene Verkehrsklassen zu unterteilen. Die Pakete werden mittels der 3 Precedence-Bits im Type of Service (ToS) Feld eines IP-Paketes (IP Version 4) unterschiedlich priorisiert. Der Aufbau des ToS-Feldes und eines IP-Paketes ist in Abbildung 3.1 und 3.2 zu erkennen.

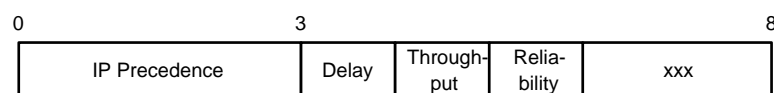


Abbildung 3.1: Type of Service Feld einer IPv4 PDU (nach [MM98])

Die weiteren Bits im ToS-Feld werden gesetzt, wenn höhere Anforderungen hinsichtlich Verzögerungen (delay), Bandbreite (throughput) und Zuverlässigkeit (reliability) gestellt werden. Die letzten beiden Bits sind nicht definiert. Die möglichen Precedence-Werte sind in Tabelle 3.2 zusammengefaßt.

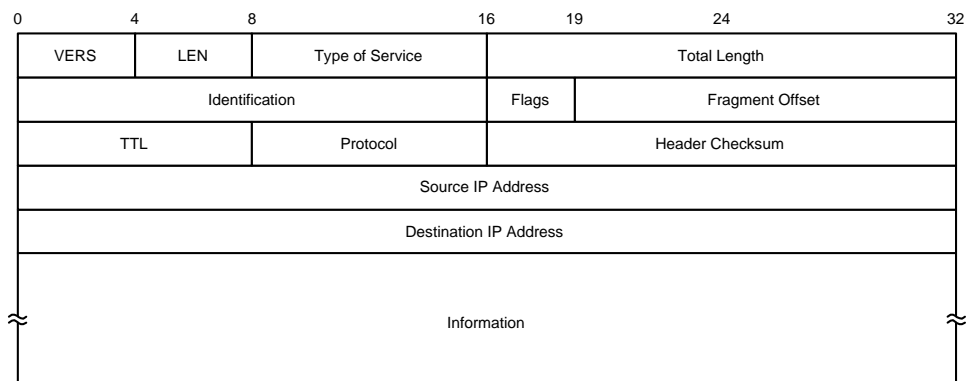


Abbildung 3.2: IPv4 Paket (nach [MM98])

Nummer	Precedence-Bits	Name
0	000	routine
1	001	priority
2	010	immediate
3	011	flash
4	100	flash override
5	101	critic/ecp
6	110	internetwork control
7	111	network control

Tabelle 3.2: IP Precedence Werte (nach [RFC 0791])

Beim Einsatz von IP Precedence ist darauf zu achten, daß die Klassen 6 und 7 für Kontrollnachrichten, wie z.B. Updates von Routing-Tabellen, reserviert sind und nicht für normales Datenaufkommen verwendet werden können. Andere QoS-Mechanismen können mittels der Precedence-Bits passende Verkehrsregeln (Reihenfolge der Übermittlung, Wahrscheinlichkeit des Verwerfens von Paketen bei Überlast usw.) aufstellen. Somit lassen sich Bandbreitenreservierungen (näheres in Abschnitt 3.3.4) durch Committed Access Rate (CAR) realisieren. Weiterhin können die Informationen des ToS-Feldes für Strategien in Überlastsituationen (siehe Abschnitt 3.3.2 und 3.3.3) durch Techniken wie Weighted Fair Queueing (WFQ) und Weighted Random Early Detection (WRED) verwendet werden.

Durch die genannten Funktionen lässt sich mit IP Precedence ein Differentiated Services (DiffServ) QoS-Modell realisieren. Die nachfolgend beschriebenen Funktionen des Cisco IOS erlauben das Setzen der IP Precedence-Bits von durchgeleiteten Paketen.

Im Test werden die Voice-Pakete in den Routern erzeugt, wobei sich der Wert der IP Precedence direkt vorgeben lässt. Deshalb sind die Funktionen Policy-Based Routing und Committed Access Rate - Packet Classification nicht im Versuchsumfang enthalten.

3.3.1.2 Policy-Based Routing (PBR)

Die 3 wichtigsten Funktionen des Policy-Based Routing sind:

- Klassifizierung des Datenaufkommens anhand von Kontrolllisten (Access Control Lists - ACL)
- Setzen der IP Precedence-Bits
- Routen von Paketen entlang spezifizierter Pfade

Damit gibt PBR einem Netzwerkadministrator eine bessere Kontrolle über das Routen durch die Erweiterung bestehender Routing-Mechanismen. Pakete werden anhand definierter Policies geroutet. Bei einer Paketfilterung am Eingangsinterface werden die ankommenden Daten auf Übereinstimmung mit vorher aufgestellten Access-Listen überprüft. Diese enthalten Regeln für den Umgang von Paketen mit bestimmten IP-Adressen, Port-Nummern, Protokollen und/oder Paketgrößen. Mögliche Einsatzgebiete von PBR sind das Bereitstellen von gleichberechtigten Zugriff und das Routen abhängig vom Protokoll, Quell-, Zieladresse, Datenmenge oder Eingangsport. Damit ist es möglich, Pakete abhängig von ihrer Klassifikation auf einen Standard-Link zu legen, bzw. bei Paketen mit höherer Priorität auf eine (teurere) Verbindung mit garantierter Bandbreite auszuweichen. [RFC 1104]

3.3.1.3 Committed Access Rate (CAR) - Packet Classification

CAR kann ebenso wie PBR zur Einordnung und Neueinordnung des Verkehrs genutzt werden. Somit ist es möglich, den Netzwerkverkehr in verschiedene Prioritätsebenen oder Classes of Service aufzuteilen, die von den Netzelementen entsprechend behandelt werden. Die Policies für die Klassifizierung können auf dem physischen Port, Quell- und

Zieladresse (IP oder MAC), Protokolltyp und Portnummer oder Access-Listen beruhen. [Cis00]

CAR kann nur auf IP-Pakete angewendet werden. Schnittstellen, die andere Protokolle verwenden, wie z.B. Fast Ether Channel oder Tunnel, werden nicht durch CAR unterstützt. CAR beinhaltet weiterhin eine Bandbreitenbeschränkung für einzelne Verkehrsströme. Auf diese Funktion wird in Abschnitt 3.3.4.1 näher eingegangen.

3.3.2 Congestion Management

Die hier berücksichtigten Netzwerkszenarien weisen eine typische Eigenschaft auf: die Verbindung zwischen den Routern einzelner Standorte besitzt eine geringere Bitrate als die der einzelnen lokalen Netze. Dadurch entstehen Überlastsituationen, d.h. es gehen mehr Datenpakete ein, als der Ausgangsport zu übertragen imstande ist.

Durch Stau (congestion) auf der Übertragungsstrecke werden Pakete verworfen und erreichen den Empfänger somit nicht mehr. Wegen der hohen Variabilität der Datenraten sind weder genaue Voraussagen noch allgemeingültige Statistiken möglich. Anhand der Verlustwahrscheinlichkeit kann dargelegt werden, daß sich ein überlastetes Netzwerk nicht durch einen Poisson-Prozeß beschreiben läßt und somit im Falle eines Staus, Datenverkehr nicht bzw. schlecht vorhersagbar ist. Aus diesen Gründen ist es eine komplexe Aufgabe, ein Netzwerk effizient zu planen und Stausituationen zu verringern. [PF94]

Der Einsatz von Congestion Management Mechanismen sollte erfolgen, wenn einer der folgenden Punkte zutrifft:

- Übertragung von interaktivem, verzögerungsempfindlichem Verkehr (z.B. VoIP), der eine Priorisierung gegenüber anderen Daten benötigt
- Kombinierte Übertragung von burst-artigem Verkehr und solchem mit geringer Bitrate, die eine zeitweilige Überlastung zur Folge haben kann
- Datenübertragung über Verbindungen mit Geschwindigkeiten von $2 \frac{Mbit}{s}$ und darunter
- Steigende Antwortzeiten im Netzwerk

Trifft keiner der genannten Punkte zu, gibt es keinen Grund, Congestion Management einzusetzen. Congestion Management kann zudem keine fehlende Bandbreite in ständig überlasteten Netzen ersetzen. [Cis00]

Das Congestion Management verwaltet Stauungen durch Umsortierung der eingehenden Pakete anhand ihnen zugewiesener Prioritäten mittels Queueing-Algorithmen. Dabei werden im Falle einer Überlastung ein oder mehrere Puffer (Queues) erzeugt und die eingehenden Pakete entsprechend ihrer Priorität bzw. Klassifizierung in die Queues geschrieben (queueing). Nach einem festzulegenden Plan werden die Queues geleert (scheduling), in den Ausgangspuffer kopiert (dequeueing) und übertragen, wenn die Leitung wieder Pakete aufnehmen kann. Die Anzahl der Queues, die Klassifizierung und die Reihenfolge des Dequeueing-Prozesses werden durch Queueing-Algorithmen bestimmt. Es existieren verschiedene Queueing-Algorithmen, die für bestimmte Verkehrsfluss-Probleme optimiert sind.

Im folgenden werden einige der Mechanismen sowie ihre prinzipielle Eignung für die Übertragung von synchronen Daten vorgestellt. Es ist zu beachten, daß alle hier vorgestellten Algorithmen nur in Stauungssituationen wirksam werden. Unter geringer Last werden alle Pakete in der Reihenfolge des Eingangs ohne Zwischenspeicherung versandt. Queueing-Algorithmen sind nicht miteinander kombinierbar, da einem Interface immer nur ein Mechanismus zugewiesen werden kann. [RFC 2309], [RFC 2581]

3.3.2.1 First In First Out Queueing (FIFO)

FIFO-Queueing ist der einfachste Algorithmus, denn er besitzt keine Möglichkeiten zur Priorisierung oder Klassifizierung von Datenverkehr. Es existiert nur ein Puffer, d.h. alle Pakete werden gleich behandelt und in der selben Reihenfolge verschickt, wie sie eintreffen. FIFO-Queueing bietet somit nur Best-Effort Service. Dadurch ist der Algorithmus sehr schnell, da keinerlei Rechenzeit für die Klassifizierung und das Scheduling verloren gehen. Er wird daher vor allem in schnellen und wenig belasteten Verbindungen eingesetzt. Allerdings kann somit eine Anwendung die gesamte Bandbreite aufbrauchen sowie burstartiger Verkehr Wartezeiten bei zeitkritischen Verbindungen hervorrufen. Deshalb ist FIFO-Queueing für langsame Verbindungen und gleichzeitige Übertragung von asynchronen und synchronen Daten nicht geeignet. In Abbildung 3.3 ist der Queueing-Algorithmus schematisch dargestellt.

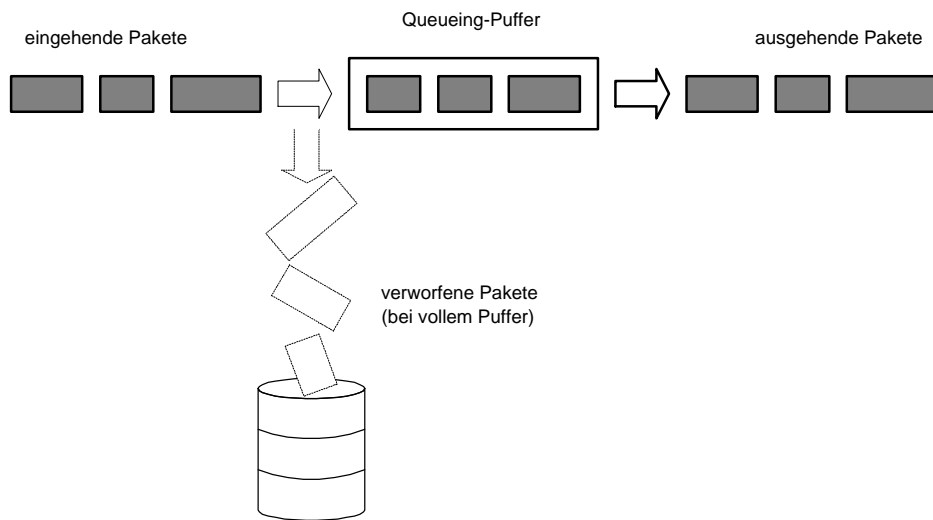


Abbildung 3.3: Funktionsweise des FIFO-Queueing (nach [Cis00])

3.3.2.2 Weighted Fair Queueing (WFQ)

Fair Queueing ist eine Methode, die Bandbreite „fair“ auf die einzelnen Datenpakete zu verteilen. WFQ gibt dazu identifiziertem Verkehr Gewicht (Priorität), um ihn zu klassifizieren und anhand einer solchen Bewertung zu bestimmen, wie viel Bandbreite einzelnen Verbindungen relativ zu anderen zusteht. Der Verkehr wird anhand von Flows klassifiziert. Flows sind charakterisiert durch die selbe Quell- und Zieladresse (IP- oder MAC-Adresse), verwendetem Protokoll, Quell- und Zielport, der Socket-Nummer und ihrem Type of Service (ToS) Wert. WFQ priorisiert interaktiven Verkehr⁷ (meist mit geringem Bandbreitebedarf), d.h. dieser wird als erstes dequeued, um Antwortzeiten zu verringern. Die restliche Bandbreite wird zwischen Verbindungen mit hohen Bandbreitenanforderungen entsprechend ihren Gewichten aufgeteilt. Damit überwindet es eine Einschränkung des FIFO-Queueing, welches Flows ohne Rücksicht auf deren Bandbreitenkonsum überträgt. Jedem Flow wird eine Queue zugeordnet. Neu eintreffende Pakete für Flows mit hohem Datenvolumen werden nach Erreichen einer bestimmten Schwelle verworfen, während Pakete für niedervolumige Flows immer noch gepuffert werden. Durch WFQ ist es möglich, Datenströmen mit hohen oder niedrigen Bitraten konstante Antwortzeiten zu liefern, ohne die Bandbreite stark erhöhen zu müssen. Außerdem

⁷Im Allgemeinen handelt es sich um Request/Response Anwendungen, wie z.B. Telnet, WWW usw.

passt es sich den wechselnden Netzwerkbedingungen an. Startet oder endet ein Flow, wird die jedem Paketstrom zugewiesene Bandbreite neu berechnet. Abbildung 3.4 zeigt das Prinzip des Weighted Fair Queueing.

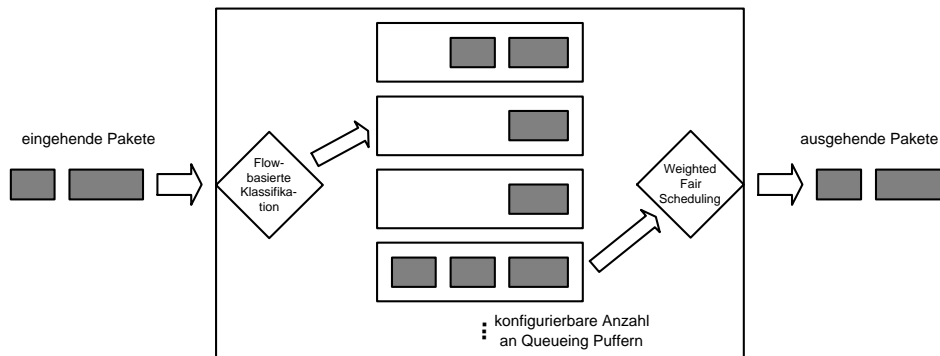


Abbildung 3.4: Funktionsweise des Weighted Fair Queueing (nach [Cis00])

WFQ wertet das ToS-Feld eines IP-Paketes aus. Pakete mit höherer Priorität werden schneller aus dem Wartepuffer entnommen, um die Antwortzeiten gering zu halten. Mit steigender Priorität eines Flows wird somit im Falle eines Staus mehr Bandbreite für ihn freigehalten. Dazu teilt Weighted Fair Queueing jedem Flow ein Gewicht entsprechend der benutzten Bandbreite zu. Die Datenströme mit dem geringsten Gewicht werden als erste aus der Queue entnommen und übertragen. Der Queueing-Algorithmus teilt nun das Gewicht jedes Flows durch dessen Precedence + 1 und verringert damit dessen Gewicht je nach Priorität.

Prinzipiell ist WFQ geeignet, synchrone Daten zu übertragen, da es Flows mit hohem Bandbreitebedarf abwertet. Interaktiver Verkehr findet konstante Bedingungen (Bandbreite, Delay) vor. Durch die fehlende Konfigurationsmöglichkeit (abgesehen von der Zuweisung unterschiedlicher Prioritäten mit IP Precedence) ist dagegen nicht die volle Kontrolle über einen Voice-Stream vorhanden. Im Extremfall könnte eine VoIP-Flow mehr Bandbreite benötigen als eine Reihe von interaktiven Flows (Telnet, Remote Control) und würde dann in einer Stausituation nicht zuerst dequeued und Pakete verlieren.

3.3.2.3 Class-Based Weighted Fair Queueing (CBWFQ)

Class-Based Weighted Fair Queueing erweitert die Möglichkeiten des Flow-Based WFQ. Es unterstützt definierbare Verkehrsklassen mit bestimmbar Eigenschaften. Für jede erzeugte Klasse wird eine Queue reserviert. Pakete werden durch Übereinstimmung von Merkmalen den Klassen zugeteilt. Als Kriterien dienen verwendetes Protokoll, Eingangsinterface oder ACL's. Folgende Eigenschaften lassen sich einer Klasse zuweisen:

- Bandbreite (wird im Staufall garantiert)
- Gewicht (wird aus der Bandbreite bestimmt)
- Maximal akkumulierbare Anzahl an Paketen in der Queue
- Verhalten bei Erreichen des Paket-Limit (Tail Drop oder WRED, siehe Abschnitt 3.3.3)

Alle Pakete, die sich nicht einer definierten Klasse zuordnen lassen, werden in die Klasse *class-default* geschrieben, der wiederum Eigenschaften zugewiesen werden können. [Cis00]

Beim Class-Based Weighted Fair Queueing wird jedem Paket ein aus der zugewiesenen Bandbreite (einer Klasse) abgeleitetes Gewicht zugeteilt. Somit ist das Gewicht frei konfigurierbar. Ein Paket wird in die Warteschlange eingereiht, nachdem ihm sein Gewicht zugeordnet wurde. Die Gewichte sollen sicherstellen, daß die Warteschlange fair bedient wird.

CBWFQ bringt damit 2 Vorteile mit sich: Es erlaubt die Zuteilung von Bandbreite für eine bestimmte Klasse, die unter Last garantiert wird. Die verfügbare Bandbreite kann auf bis zu 64 Klassen aufgeteilt und im Gegensatz zu WFQ kontrolliert werden. WFQ kann praktisch nur 6 (nutzbare) verschiedene „Klassen“ mit den IP Precedence Bits unterscheiden.

Durch die Fähigkeit, synchronen Daten eine eigene Klasse zuzuordnen und diesen die benötigte Bandbreite zuzuteilen, ist CBWFQ für VoIP geeignet. Allerdings fehlt eine strikte Priorisierung gegenüber anderem Verkehr, um eine akzeptable Verzögerungszeit zu erzielen.

3.3.2.4 Priority Queueing (PQ)

Priority Queueing bietet dem Nutzer strikte Priorisierung. Dazu besitzt der Algorithmus 4 Warteschlangen mit unterschiedlich hohen Prioritäten. In der Abbildung 3.5 ist PQ schematisch dargestellt.

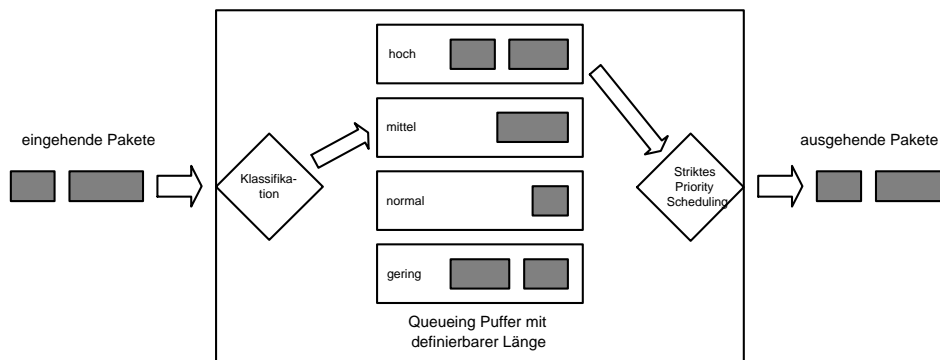


Abbildung 3.5: Funktionsweise des Priority Queueing (nach [Cis00])

Pakete können anhand von Protokolltyp und Eingangsschnittstelle klassifiziert werden. Die Warteschlange mit der höchsten Priorität wird immer zuerst bedient. Erst wenn die Queue leer ist, folgen die anderen Queues entsprechend ihrer Wertigkeit. Dadurch sind schnelle Antwortzeiten bedingt. PQ ist daher ideal für die Übertragung synchroner Daten. Netzwerkinterner Verkehr füllt ebenfalls die Queue mit der höchsten Priorität. Zur Optimierung kann die Queue-Länge für jede Warteschlange eingestellt werden. [Cis00]

Ein Schwachpunkt dieses Konzepts ist die Tatsache, daß Pakete aus den Queues mit geringeren Prioritäten unter Umständen nie gesendet werden, wenn die Leitung mit Paketen höherer Priorität ausgelastet ist. Das läßt sich indes durch eine Limitierung der Bandbreite für einzelne Queues umgehen. Dafür bietet sich die Rate Limiting Funktion des Committed Access Rate-Algorithmus (siehe Abschnitt 3.3.4.1) an.

Priority Queueing produziert extra Overhead und ist daher für schnelle Interfaces mit hohen Bitraten nur bedingt geeignet. Aus diesem und oben genannten Gründen ist der Einsatz von PQ vor allem für überlastete langsamere Verbindungen gedacht.

Durch die strikte Priorisierung mit der daraus resultierenden garantierten Bandbreite und geringen Verzögerungszeit (und -variation) ist Priority Queueing gut für die Über-

tragung synchroner Daten über gestaute Verbindungen geeignet. Im Zusammenspiel mit CAR kann, wenn nötig, der Sprachverkehr eingeschränkt werden.

3.3.2.5 Custom Queueing (CQ)

Custom Queueing erlaubt das Anlegen von bis zu 16 Queues, die zirkular abgearbeitet werden. Einstellbar sind die maximal pro Warteschlange in einem Durchgang zu übertragenden Pakete (bytecount oder packetcount) und die Länge der Queue. Aus diesen Angaben leitet sich die Bandbreite einer Queue ab. So springt der Algorithmus von Warteschlange zu Warteschlange und leert sie. Der Vorgang dauert so lange an, bis entweder die vereinbarte Anzahl an Paketen erreicht oder die Queue leer ist. Dann spricht CQ die nächste Warteschlange an. In Überlastsituationen stellt Custom Queueing so sicher, daß jede Warteschlange nur den vereinbarten prozentualen Anteil der Bandbreite erhält. Pakete können anhand des verwendeten Protokolls oder des Eingangsinterface den Custom Queues zugeordnet werden. CQ ist ein statischer Algorithmus, d.h. er passt sich nicht den wechselnden Netzwerkbedingungen an. In der Abbildung 3.6 wird die Funktionsweise des Custom Queueing dargestellt. [Cis00]

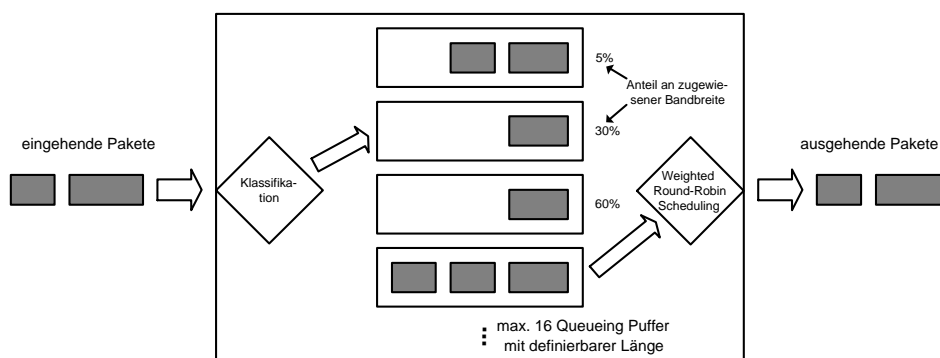


Abbildung 3.6: Funktionsweise des Custom Queueing (nach [Cis00])

Aufgrund der aufgeführten Eigenschaften und der fehlenden Priorisierung ist Custom Queueing für die Übertragung von synchronen Daten nicht geeignet. CQ gewährleistet zwar eine bestimmte Bandbreite, es ist dagegen nur schlecht möglich, Voraussetzungen über Verzögerungen zu treffen. Im schlimmsten Fall sind mehrere Classes of Service (Custom Queues) definiert worden, die der Reihe nach abzuarbeiten sind. Sind

alle Queues gefüllt, dauert es lange (je nach eingestelltem Bytecount), bis die Warteschlange mit den Voice-Paketen an der Reihe ist. Die kann nun ihre Pakete auf einen Schlag abschicken. Es wechseln sich also Perioden mit hoher und geringer Verzögerung ab. Solch starken Variationen beeinträchtigen das Hörempfinden erheblich. Weiterhin ist durch die prozentuale Bandbreitenzuteilung keine Aussage über die tatsächlich für Sprachdaten zur Verfügung stehende Kapazität der Verbindung möglich.

3.3.2.6 IP RTP Priority

IP RTP Priority stellt Priority Queueing (siehe Kapitel 3.3.2.4) für synchrone Datenströme zur Verfügung, die anhand des benutzten Real-Time Transport Protocol (RTP) identifiziert werden. IP-RTP-Flows werden immer vor anderen Daten aus der Warteschlange entnommen und übertragen. Dadurch sind kurze Verzögerungszeiten und eine garantierte Übertragung sichergestellt. IP RTP Priority arbeitet portbasiert, d.h. es können alle RTP-Ports (16384 - 32767) oder gezielt einzelne Ports oder Teilbereiche priorisiert werden.

Dieses Verfahren ist kombinierbar mit WFQ und CBWFQ. Im Zusammenspiel mit Weighted Fair Queueing wird Sprache strikt priorisiert. Die anderen Queues werden entsprechend dem WFQ-Schema dequeued. Die Verknüpfung von CBWFQ mit IP RTP Priority führt ebenso auf eine strikte Priorisierung von Voice-Paketen. Allerdings sind Klassen für andere Datenströme definierbar, die eine garantierte Bandbreite benötigen. Alle weiteren Pakete werden in die Queue der Default Class geschrieben und können entsprechend ihren Gewichten fair aus der Warteschlange entnommen werden.

IP RTP Priority erwartet bei der Konfiguration die Eingabe einer Bandbreite, die garantiert werden soll. Bei der Planung ist darauf zu achten, daß jeglicher Voice-Traffic, der den angegebenen Wert überschreitet, verworfen wird, d.h. die garantierte Bandbreite ist gleichzeitig die maximal übertragbare. Compressed Real-Time Protocol (näheres in Abschnitt 3.3.6.1) wird automatisch mit IP RTP Priority aktiviert. Daher kann für die Planung der zugewiesenen Bandbreite auf die Werte der durch CRTP komprimierten Voice-Calls zurückgegriffen werden. Weiterhin ist die Bandbreite vom gebrauchten Codec (siehe Kapitel 3.4) abhängig. Für den Ausgleich von Jitter und Verzögerungen sollten ebenfalls ein paar Prozent eingeplant werden.

3.3.2.7 Low Latency Queueing (LLQ)

Low Latency Queueing ermöglicht Priority Queueing innerhalb des Class-Based Weighted Fair Queueing. Dadurch wird die Verzögerung von Paketen verringert, was vor allem für Sprachdienste, die sensibel auf Delay reagieren, sinnvoll ist.

CBWFQ kann ohne LLQ nur auf Klassen basiertes WFQ erreichen. Es unterstützt kein Prioritätsqueueing, sondern kann den Klassen nur eine bestimmte Bandbreite garantieren. Das Gewicht eines Paketes hängt von der Bandbreite ab, die einer Klasse zugeordnet wurde. Daher bestimmt die zugewiesene Bandbreite die Reihenfolge der Übertragung der Pakete. Das ist vor allem für Sprachübertragung ungünstig, da eine zu große Verzögerung oder Variation in der Verzögerung als störend empfunden wird.

Beim Einsatz von LLQ wird eine einzelne Priority Queue angelegt. Alle Pakete einer mit LLQ ausgestatteten Klasse werden in diese Queue umgeleitet und streng gegenüber anderen Paketen priorisiert.

Vom den in seiner Wirkungsweise ähnlichen IP RTP Priority unterscheidet sich Low Latency Queueing dahingehend, als daß es auf alle zu einer Klasse gehörenden Pakete wirkt. Die Konfigurationsmöglichkeiten sind somit nicht nur auf RTP-Portnummern beschränkt, denn Pakete lassen sich anhand von ACL's, Protokollen oder Eingangsschnittstellen den Klassen zuordnen. Weiterhin kann die zugeteilte Bandbreite ohne Paketverlust überschritten werden, wenn keine Überlastung auf dem Link herrscht. Selbst wenn alle priorisierten Pakete in eine Queue dirigiert werden, geschieht die Überwachung der Bandbreite einzeln für jede Klasse. IP RTP Priority und LLQ lassen sich kombinieren. Dabei besitzt IP RTP Priority Vorrang gegenüber LLQ. Zu beachten ist, daß LLQ nicht auf Frame Relay Links einsetzbar ist.

In der Tabelle 3.3 sind alle Algorithmen des Congestion Management und deren Eigenschaften zusammengefaßt.

3.3.3 Congestion Avoidance

Die vorangegangenen Kapitel beschreiben die Möglichkeiten zum Management von Stausituationen. In diesem Abschnitt wird dargestellt, wie Überlastungen der Verbindung vermieden werden können. Verkehrsstauungen im Netzwerk sind im Kapitel 3.3.2 beschrieben worden. Die Vermeidung von Congestion beinhaltet die Überwachung der Netzlast sowie das gezielte Verwerfen von Paketen, um der Stauung entgegenzuwirken.

Merkmal	FIFO	WFQ	CBWFQ	PQ	CQ
QoS Unterstützung	nein	ja	ja	ja	ja
Anzahl Queues	1 (fest)	max. 256 (einstellbar)	max. 64 (einstellbar)	4 (fest)	max. 16 (einstellbar)
Konfiguration	nicht erforderlich	nicht erforderlich	erforderlich	erforderlich	erforderlich
Merkmal	keine Verkehrs-priorisierung	„faire“ Zu- teilung der Bandbreite für Flows anhand ihrer Gewichte, strikte Prio- risierung mittels IP RTP Priority möglich	ermöglicht garantierte Bandbreite für defi- nierbare Verkehrs- klassen, strikte Prio- risierung mittels IP RTP Priority oder LLQ möglich	strikte Prio- risierung der Queue mit der höheren Priorität	zirkulare Abarbeitung der Queues

Tabelle 3.3: Zusammenfassung Congestion Management (nach [Cis00])

3.3.3.1 Verfahren der Überlastvermeidung

Im einfachsten Fall wird bei einer Überlastung alles verworfen, wenn die Queue gefüllt ist. Die Methode wird als Tail Drop bezeichnet. Tail Drop arbeitet ohne Berücksichtigung von CoS, d.h. alle Pakete werden gleich behandelt (Best-Effort Service).

Daneben existiert seit Anfang der 90er Jahre mit Random Early Detection (RED) eine Methode, die es ermöglicht, bei Netzwerkstauungen zu agieren anstatt zu reagieren. Das wird auf folgendem Weg erreicht: Grundlage von RED sind Transportprotokolle (z.B. TCP), die auf Zellverlust mit Herabsetzung der Bitrate reagieren. RED besitzt eine eigene Queue. Übersteigt deren Füllgrad eine bestimmte Schwelle, verwirft RED zufällig Pakete. Ein TCP-Sender verringert nun solange seine Bitrate (slow start), bis alle Pakete beim Empfänger ankommen. Das bedeutet, die Stauung ist beseitigt. RED

läßt sich somit als Integrierglied verdeutlichen. Es „verwischt“ die Spitzen, die durch den burst-artigen Charakter der asynchronen Datenübertragung entstehen und passt ihn der Form synchronen Verkehrs an. TCP eignet sich wegen seiner Fähigkeit, Übertragungen kurzzeitig stoppen sowie die Datenrate an die verfügbare Bandbreite anpassen zu können, gut für den Einsatz von Random Early Detection. Unidirektionale oder nicht quittierende Transportprotokolle (z.B UDP) sind prinzipbedingt nicht für RED verwendbar. [Cis00]

RED verwirft Pakete früh, anstatt darauf zu warten, daß die Warteschlange voll ist und vermeidet damit, viele Pakete auf einen Schlag fallen zu lassen. Durch die zufällige Auswahl der zu verwerfenden Pakete verringert RED weiterhin die Chance einer Synchronisation der einzelnen Flows. Wird Tail Drop benutzt, gehen Pakete mehrerer TCP-Verbindungen bei Überlastung verloren. Diese verringern alle daraufhin simultan ihre Datenrate. Die Verkehrsdichte nimmt schlagartig ab. Wegen der nun optimalen Bedingungen versuchen alle TCP-Sitzungen wiederum, gleichzeitig ihre Paketrage zu erhöhen. Die entstehende Wellenbewegung reduziert die Effektivität des Netzes unnötig.

3.3.3.2 Weighted Random Early Detection (WRED)

Weighted Random Early Detection ergänzt RED um die Fähigkeit, den Datenverkehr zu klassifizieren. WRED wertet das ToS-Feld eines IP-Paketes aus. Droht ein Interface zu verstopfen, werden vorzugsweise Pakete niedriger Priorität verworfen. WRED bietet somit ein differenziertes Service-Modell. Je höher die Priorität eines Paketes ist, desto größer ist die Wahrscheinlichkeit, verschickt zu werden. Diese Funktion läßt sich bei Bedarf abschalten. Im Zusammenspiel mit RSVP (ein Signalisierungsverfahren, siehe Abschnitt 3.3.5.1) werden zuerst Pakete verworfen, die nicht zu einem RSVP-Flow gehören.

Weighted Random Early Detection wichtet die einzelnen Flows, wie in Kapitel 3.3.2.2 beschrieben. Pakete von Datenströmen mit einem hohen Gewicht (hoher Bitrate) werden zuerst verworfen. Flows mit einem hohen Datenaufkommen müssen daher eher ihre Geschwindigkeit herabsetzen als Flows, die nur wenig Verkehr erzeugen. Die Funktionsweise von WRED wird in Abbildung 3.7 dargestellt. [Cis00]

WRED vermeidet ebenso wie RED die Synchronisation einzelner Flows. Dieser Algorithmus ist vor allem für Verbindungen gedacht, in der der größte Teil am Verkehrsauf-

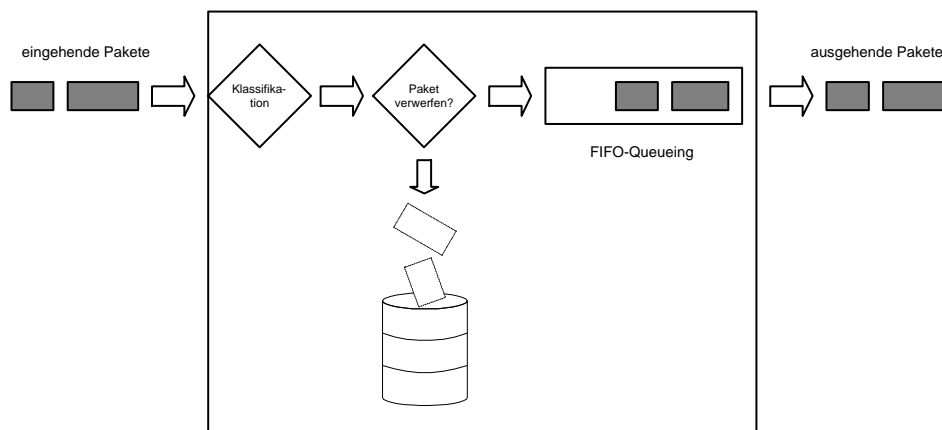


Abbildung 3.7: Funktionsweise des Weighted Random Early Detection (nach [Cis00])

kommen durch TCP/IP-Flows verursacht wird. Protokolle ohne Verkehrskontrolle oder Rückkanal (wie UDP) können nicht erkennen, daß Pakete verworfen wurden. Daraus resultiert, daß weder verworfene Pakete erneut gesendet werden, noch eine Verringerung des Verkehrsstaus eintritt. Nicht auf IP beruhende Pakete werden wie IP Traffic mit einer IP Precedence 0 behandelt und deswegen häufiger fallengelassen als IP Verkehr. WRED läßt sich nicht mit CQ, PQ oder WFQ kombinieren.

Für die Übertragung von VoIP Paketen ist WRED aus den genannten Gründen nicht geeignet. Das Unvermögen von UDP (als Transportprotokoll für VoIP) auf Paketverlust zu reagieren und die fehlende Priorisierung der Voice-Pakete in der nachgeschalteten FIFO-Queue von WRED verbieten den Einsatz von Weighted Random Early Detection. Alternativ läßt sich WRED innerhalb einer Klasse von CBWFQ aktivieren. Sind vor allem TCP-Flows einer mit WRED ausgestatteten Klasse zugeordnet, passt sich die benutzte Bandbreite der Klasse der aktuellen Netzauslastung an.

3.3.3.3 Flow-Based WRED

Flow-Based WRED bringt gegenüber dem normalen WRED eine gerechtere Behandlung aller Flows hinsichtlich dem Verwerfen von Paketen. Flows lassen sich allgemein in 3 Kategorien unterteilen: nichtadaptive Flows, z.B. UDP/RTP Datenströme, reagieren nicht auf Überlastung. Sie werden nicht weiter betrachtet. Adaptive Ströme (z.B. auf TCP basierende) verringern die Geschwindigkeit im Fall eines Staus. Sie lassen

sich in robusten und anfälligen Verkehr unterteilen. Robuste Flows besitzen eine im Durchschnitt konstante Datenrate. Anfälliger Verkehr besitzt weniger Pakete im Ausgangspuffer als ein robuster Strom. Im Falle eines Staus werden genauso viele Pakete der robusten und fragilen Flows verworfen, obwohl fragile Flows weniger Pakete gepuffert haben. Dadurch entsteht eine Benachteiligung der anfälligen Flows, da prozentual mehr von ihnen fallengelassen wird.

Flow-Based WRED stellt sicher, daß adaptive vor nichtadaptiven Strömen gesichert werden. Weiterhin verhindert der Mechanismus, daß ein einzelner Flow den gesamten Puffer für sich vereinnahmen kann. Das wird durch 2 Ansätze sichergestellt:

- Eingehender Verkehr wird unterteilt in Datenströme anhand von Quell- und Zieladressen sowie dem benutzten Port.
- Der Status aktiver Flows (solche, die Pakete im Ausgangspuffer haben) wird überwacht.

Flow-Based WRED stellt somit sicher, daß einzelne Flows nicht mehr als die ihnen zugeteilten Ressourcen in der Ausgangsqueue verbrauchen. Diese werden anhand der Anzahl der Flows und der Größe der Queue bestimmt. Reißt ein Flow zuviele Puffer an sich, wird er durch Verwerfen von Paketen „bestraft“.

3.3.4 Policing and Shaping

Policing and shaping bezeichnet zwei Verfahren zur Regulierung des Verkehrs. Sie werden eingesetzt, um sicherzustellen, daß ein Flow oder ein Sender sich an einen vereinbarten Verkehrsvertrag hält. Die beiden Verfahren unterscheiden sich im Verhalten im Falle einer Überschreitung der Vereinbarungen. Ein Policer verwirft im Allgemeinen Pakete, wogegen ein Shaping-Algorithmus derartige Pakete mittels eines Puffers verzögert.

Die nachfolgenden Algorithmen verwenden den Token Bucket Mechanismus, eine abstrakte Beschreibung der Übertragungsgeschwindigkeit CIR . Laut Gleichung 3.1 wird diese aus der Größe der Bursts B_c bestimmt, die innerhalb einer bestimmten Zeit T_c gesendet werden können, ohne festgelegte Limits zu überschreiten.

$$CIR = \frac{B_c}{T_c} \quad (3.1)$$

Der Token Bucket besitzt somit einen integrierenden Charakter. Er wird entsprechend einer zugestandenen Bandbreite kontinuierlich mit Token gefüllt. Für jedes zu übertra-

gende Paket wird gemäß seiner Größe eine bestimmte Anzahl an Token entnommen. Die Tiefe des Buckets entspricht somit der Burst Size B_c . Sind nicht ausreichend Token für das Verschicken eines bestimmten Paketes im Token Bucket, wird es entweder verworfen (policing) oder gepuffert (shaping), bis genügend Token vorhanden sind. Ist dagegen die Kapazität des Eimers erschöpft, werden die überschüssigen Token verworfen und können somit nicht für zukünftig eintreffende Pakete genutzt werden. [Cis00]

3.3.4.1 Committed Access Rate (CAR) - Rate Limiting

Committed Access Rate (CAR) wacht – neben seiner klassifizierenden Funktion (siehe Kapitel 3.3.1.3) – über die Einhaltung von Bandbreiten auf Eingangs- oder Ausgangsschnittstellen. Policies können anhand des Eingangsinterfaces, auf jedweden IP-Verkehr oder eingeschränkt auf Access Listen angewandt werden. Es ist möglich, Policies zu kaskadieren, um eine feinere Unterteilung der Flows zu erreichen. Je nachdem, ob ausreichend Token vorhanden sind, lassen sich verschiedene Aktionen zur Behandlung der Pakete definieren:

- Transmit – Paket wird gesendet
- Drop – Paket wird verworfen
- Set precedence & transmit – Priorität des Paketes wird geändert und das Paket dann verschickt
- Continue – Paket wird an die nächste verfügbare Policy in der Kaskade weitergegeben; existiert keine weitere Policy, wird das Paket gesendet
- Set precedence & continue – Priorität des Paketes wird geändert und an die nächste Policy in der Kette übergeben

Es existiert außerdem die Möglichkeit, eine Extended Burst Size zu definieren, mit der Token geborgt werden können. [Cis00]

Die in Abschnitt 3.3.1.3 genannten Einschränkungen hinsichtlich der Einsetzbarkeit von CAR gelten auch hier.

3.3.4.2 Generic Traffic Shaping (GTS)

Generic Traffic Shaping (GTS) erlaubt den Ausgangsverkehr zu steuern, um die Übertragungsgeschwindigkeit an das gegenüberliegende Interface anzupassen. So können Engpässe durch unterschiedliche Datenraten vermieden werden.

Die Datenrate kann auf eine bestimmte Bandbreite oder einen Staulevel limitiert werden. Hier kommt wiederum der Token Bucket Mechanismus zum Einsatz. Im Unterschied zu einem Policer wird das Paket in eine Queue geschrieben. Generic Traffic Shaping verwendet dafür WFQ. GTS kann für ein Interface oder eine Access-Liste konfiguriert werden.

3.3.5 Signalling

QoS Signalisierung kann als eine Form der Kommunikation zwischen Endpunkten oder Netzwerkknoten verstanden werden, mit der Anforderungen für einen Verkehrsstrom ausgehandelt werden können. Jedes Netzwerkelement muß QoS Signalling beherrschen, um ein End-to-End QoS zu garantieren. Benötigte Prioritäten können durch 2 Techniken signalisiert werden. Zum einen kann durch das in Abschnitt 3.3.1.1 vorgestellte IP Precedence eine in-band Signalisierung (Differentiated QoS) realisiert werden. Mit RSVP läßt sich garantiertes Quality of Service verwirklichen. RSVP arbeitet mit out-of-band Signalisierung.

3.3.5.1 Resource reSerVation Protocol (RSVP)

RSVP ermöglicht die dynamische Einrichtung eines End-to-End QoS über ein heterogenes Netzwerk. Eine Anwendung kann durch das Resource reSerVation Protocol ein bestimmtes QoS Level anfordern und somit dynamisch Bandbreite zugeteilt bekommen. RSVP ist das einzige Signalisierungsprotokoll, daß Bandbreite zwischen 2 Endpunkten garantiert. Die Kommunikation ähnelt einem Client/Server-Modell. RSVP kennt 5 verschiedene Standardnachrichtentypen. [MM98], [RFC 2205]

Das Resource reSerVation Protocol besitzt keine eigenen Routingalgorithmen, sondern baut auf Routed-Protokolle niederer Schichten auf. So kann RSVP bei Änderungen des Routing-Pfades die Reservierungen adaptiv an den neuen Pfad anpassen. Weiterhin arbeitet RSVP transparent auf Routern, die dieses Protokoll nicht unterstützen. RSVP kann keine Queueing Mechanismen ersetzen, sondern nur ergänzen. So kann RSVP

zwar ein spezielles QoS-Merkmal anfordern, die Umsetzung obliegt jedoch dem jeweiligen Queueing-Algorithmus. RSVP nutzt Weighted Fair Queueing, um Platz in einer Queue zu reservieren und die Reihenfolge der Übermittlung der Pakete zu bestimmen. RSVP ist ferner mit CBWFQ kombinierbar. Dabei arbeiten beide Verfahren jedoch unabhängig voneinander. Mit WFQ läßt sich so eine spezielle Datenrate garantieren. Im Zusammenspiel mit WRED kann die Auslastung eines Links kontrolliert werden.

Auf dem Weg von einem End-Host zum anderen passiert RSVP alle Netzknoten und versucht, eine bestimmte Bandbreite exklusiv für einen bestimmten Flow zu reservieren. Sind genügend Ressourcen vorhanden und besitzt die Anwendung das Recht, diese zu reservieren, werden die Pakete des Flows durch RSVP für die Klassifizierung und den Scheduler⁸ des Queueing-Algorithmus gekennzeichnet. Sind eine der beiden Bedingungen nicht erfüllt, erhält die Anwendung eine Fehlermeldung zurück.

Die durch RSVP garantierte Bandbreite kann durch eine Anwendung überschritten werden, wenn der Link noch nicht ausgelastet ist. Im Falle eines Staus werden alle Pakete, die außerhalb des Bereiches liegen, verworfen.

RSVP löst nicht alle Probleme im Bereich des Quality of Service. Zudem benötigt es Zeit für die Einrichtung einer Reservierung zwischen 2 Endpunkten (Signalisierungsphase).

3.3.6 Link Efficiency Mechanisms

Die nachfolgend beschriebenen Verfahren verringern den Overhead bei der Datenübertragung und erhöhen somit die Effizienz eines Links.

3.3.6.1 Compressed Real-Time Protocol (CRTP)

Als Transportprotokoll zur Übertragung von Echtzeit-Daten (z.B. VoIP) findet das Real-Time Protocol (RTP) Verwendung. RTP und seine Ergänzungen (RTCP – RTP Control Protocol, RTSP – Real-Time Streaming Protocol) beinhalten vielfältige Funktionen für Videokonferenzen, Video-Multicasting usw. [MM98], [RFC 1889]

RTP befindet sich im ISO/OSI Referenzmodell oberhalb von UDP. Damit erhält ein Voice-Paket (20 – 160 *byte* lang) zuerst einen RTP Header, dann einen UDP Header und auf der Netzwerkebene noch den IP Header. Der Prozeß der Verschachtelung wird

⁸Die Begriffe Klassifizierung und Scheduler werden in Abschnitt 3.3.2 erläutert.

Encapsulation genannt. Ein kombinierter IP/UDP/RTP Header ist 40 *byte* groß. Dem Mißverhältnis zwischen der Größe von Header und Datenteil begegnet das Compressed Real-Time Protocol (CRTP), indem es sich den Umstand zu Nutze macht, daß die meisten Pakete einen bis auf wenige Bytes identischen Kopf besitzen. Diese Tatsache bildet die Grundlage für eine verlustfreie Komprimierung⁹, da nur geringe Differenzen der einzelnen Header übertragen werden müssen. CRTP schaltet sich vor die Ausgangsqueue eines Systems. Nicht-RTP Pakete werden sofort weitergeleitet. Die Pakete eines RTP Datenstroms werden dann entsprechend der Abbildung 3.8 verkleinert. Ein IP/UDP/RTP Zellkopf läßt sich somit im Schnitt auf 2 bis 5 *byte* reduzieren. CRTP wird

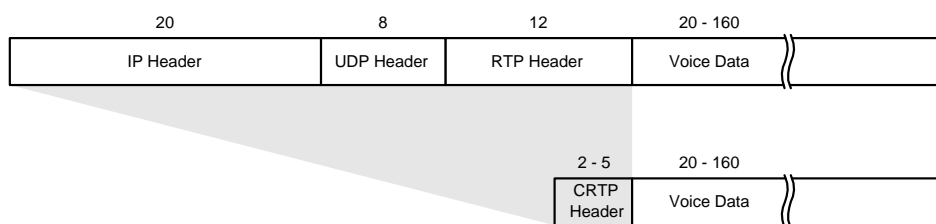


Abbildung 3.8: Funktionsweise der RTP Header Compression (nach [Cis00])

in jedem Netzknoten komprimiert/dekomprimiert ähnlich der TCP Header Compression. [RFC 1144]

Die Einsparungsmöglichkeiten von CRTP sind in Tabelle 3.4 dargestellt. Die Werte basieren auf einer angenommenen Größe des CRTP Kopfes von 5 *byte*. Wie zu erkennen ist, reduziert sich die Größe eines Paketes bei einer Payload von 160 *byte* um 21,1 %. Zusätzlich verringert sich die Verzögerung eines Paketes durch seine geringere Größe bei einer Übertragungsrate von $64 \frac{\text{kbit}}{\text{s}}$ um $\approx 5 \text{ ms}$. Das Compressed Real-Time Protocol (CRTP) ist damit vor allem für den Einsatz in WAN Interfaces mit begrenzter Bandbreite und hohem RTP Aufkommen sowie geringer Nutzlastgröße (20 bis 50 *byte*) der RTP Pakete prädestiniert. Hier wird eine bessere Ausnutzung der Verbindung durch den verringerten Overhead und kürzere Latenzzeiten durch die kürzere Übertragungsdauer erreicht. Auf High-Speed Links bringt CRTP dagegen nahezu keine Vorteile und sollte daher dort nicht eingesetzt werden. [RFC 2507], [RFC 2508], [RFC 2509]

⁹Der Empfänger der Pakete oder ein beliebiger Knotenpunkt im Netzwerk können den Original-Header ohne Informationsverlust rekonstruieren.

Nutzlast	Größe der Nutzlast <i>byte</i>	Reduzierung der Paketgröße %
VoIP	20	240
	160	21,2
SQL	256	13,4
FTP	1500	2,3

Tabelle 3.4: Reduzierung der Paketgröße durch RTP Header Compression (nach [Cis00])

3.3.6.2 Link Fragmentation and Interleaving (LFI)

Auf Multilink PPP Interfaces gibt es eine weitere Möglichkeit, die Verzögerungszeiten bei Echtzeit-Daten zu verringern: Link Fragmentation and Interleaving (LFI). LFI ist auch unter dem Namen Multi-Class Extension to Multi-Link PPP (MCML) bekannt. [RFC 1661], [RFC 1717], [RFC 2686]

Die Funktionalität von LFI ist innerhalb des Layers 2 des ISO/OSI Referenzmodells anzusiedeln. Da sich diese Arbeit vor allem auf Mechanismen der Schichten 3 und 4 konzentriert, wird LFI nicht in die Versuchsreihen aufgenommen. Eine Beschreibung des Link Fragmentation and Interleaving ist der Vollständigkeit halber im Folgenden wiedergegeben.

Den Vorteil von LFI verdeutlicht folgendes Problem. Zur Verringerung von Overhead werden Datenpakete zu sogenannten Maximum Transmission Units (MTU) zusammengefaßt. Auf langsamen WAN Links (z.B. ISDN mit $64 \frac{kbit}{s}$) werden durch die Übertragung solcher großen Datagramme kleinere Pakete stark behindert, da alle Pakete der Reihe nach übertragen werden. Eine MTU mit einer Größe von 1500 *byte* braucht zur Übertragung auf einem ISDN B-Kanal $\approx 190 ms$. Erst nach dieser Zeit steht die Verbindung für ein anderes Paket zur Verfügung. Eine derart hohe Verzögerung ist deutlich hörbar. Es wird daher ein Verfahren benötigt, daß die übergroßen Zellen in mehrere Teile aufsplittet und die auf Delay empfindlich reagierenden Pakete dazwischen einsortiert.

Das Multilink Point-to-Point Protocol (MLP) ist die Basis für LFI, indem es Funktionen zum Aufsplitten und zum Zusammenfügen von Paketen in der richtigen Rei-

henfolge bei der Übertragung über mehrere logische Verbindungen bereitstellt. Link Fragmentation and Interleaving versieht „Jumbogramme“ mit einem Multilink Header und fragmentiert diese danach. Die Größe der Teilstücke wird durch die Anforderung an die maximal zulässige Verzögerung bestimmt. Kleinere Voice-Pakete werden ohne den Multilink Header zwischen den Bruchstücken eingeordnet. Den einzelnen logischen Links kann unterschiedlich viel Bandbreite zugeordnet werden.

LFI bezieht WFQ ein. Dadurch werden Jumbogramme in einer Queue und Voice-Pakete in einer anderen Queue gepuffert. Die Jumbogramme in der einen Queue werden fragmentiert. Die Gewichte werden anhand der Paketgrößen und nicht durch die Größe der Fragmente bestimmt. Entsprechend ihren Gewichten werden die Pakete und Fragmente dann nacheinander der Ausgangsqueue zugeführt und übertragen.

Somit führt LFI zu einer Verringerung der Latenzzeit für kleine Pakete (Voice, Telnet usw.). Der Einsatz ist bei der Übertragung synchroner Daten auf MLP Verbindungen zu empfehlen.

3.4 Codierung des Audiosignals

Bei der Umsetzung eines analogen Signals in ein äquivalentes Digitalsignal wird es unter anderem codiert. Die Codierungsmechanismen sind an den jeweiligen Einsatzzweck angepaßt und lassen sich in 3 Gruppen einteilen: Signalform-Codierer, Parametrische Codierer und Hybride Codierer, die die Vorteile der beiden Erstgenannten in sich vereinen.

Bei der Signalform-Codierung durchläuft ein analoges Signal die 3 Stufen der klassischen Digitalisierung:

1. Überführung in ein amplitudenmoduliertes Impulssignal durch Abtastung
2. Quantisierung durch Unterteilung in diskrete Amplitudenstufen
3. Codierung

Hierbei werden in einem A/D-Wandler die abgetasteten Werte mittels einer Quantisierungskennlinie den zu übertragenden digitalen Codeworten zugeordnet.

Parametrische Codierer verfolgen einen völlig anderen Ansatz, da sie anstelle des Signals nur dessen mathematische Beschreibung übertragen. Solche Vocoder modellieren den menschlichen Stimmapparat und versuchen ihn elektrisch nachzubilden. Das

Ergebnis sind geringe Bitraten¹⁰, aber auch eine unnatürlich wirkende Wiedergabe. Sie werden deshalb nicht in Telefonieanwendungen genutzt. [Höy99], [MM98]

Hybride Codierer arbeiten nach dem Analyse/Synthese-Verfahren. Dabei wird das analoge Signal in Blöcke unterteilt und in einer Optimierungsschleife ein sogenannter Anregungsvektor mit dem kleinsten gewichteten Fehler bestimmt und zusammen mit einem Verstärkungsfaktor übertragen. Anregungsvektoren stammen aus der parametrischen Codierung und bezeichnen eine Einflußgröße des Modells des menschlichen Stimmorgans. Aufgrund der Optimierung auf die Stimme eines Menschen läßt sich Musik o.ä. nur begrenzt über solche Codecs wiedergeben. Die wesentlichen Unterschiede zwischen den einzelnen Codecs dieser Klasse bestehen in der Bestimmung der Anregungsvektoren. [Höy99]

Hybride Codecs bieten den besten Kompromiss aus erzielbarer Sprachqualität und geringem Bandbreitenbedarf. Sie bieten sich daher für Telefonieanwendungen wegen der geringeren Netzlast an.

Die im Folgenden aufgeführten Codecs sind von der ITU-T standardisiert und in den Geräten des Versuchsaufbaus implementiert. Sie unterscheiden sich hinsichtlich Sprachqualität, benötigter Rechenleistung und durch Umrechnung bedingte Verzögerungszeiten (delay).

3.4.1 G.711 A law

G.711 A law basiert auf PCM (Puls Code Modulation), dem am häufigsten eingesetzten Verfahren zur Digitalisierung¹¹. PCM gehört zu den Signalform-Codierern.

Um ein über den gesamten Dynamikbereich gleichbleibendes SNR zu erzielen, wird das Signal an einer 13-stufigen logarithmischen Kennlinie quantisiert, d.h. leise Signale werden höher aufgelöst als solche mit großen Amplituden. Man spricht dabei auch von einer Dynamikkompression/-dekompression. Bei der Rückwandlung in einem D/A-Wandler wird die inverse Kennlinie genutzt. Der Zusatz „A law“ bezeichnet die in Europa verwendete A-Komprimierungskennlinie.

Das Verfahren zeichnet sich durch seine gute Sprachqualität, geringeren Rechenaufwand und geringe Verzögerungszeiten aus. Demgegenüber stehen der hohe Bandbreitenbedarf von $64 \frac{kbit}{s}$.

¹⁰Üblich sind $2 \frac{kbit}{s}$.

¹¹PCM findet z.B im ISDN Verwendung.

PCM dient den meisten Codecs als Eingangssignal in einer Kaskadenschaltung. Im Rahmen dieser Arbeit wird ebenfalls eine Kaskade eingesetzt, wie in Abbildung 3.9 zu erkennen ist. Die TK-Anlagen liefern auf dem S_{2M} -Interface (PRI) ein PCM-codiertes

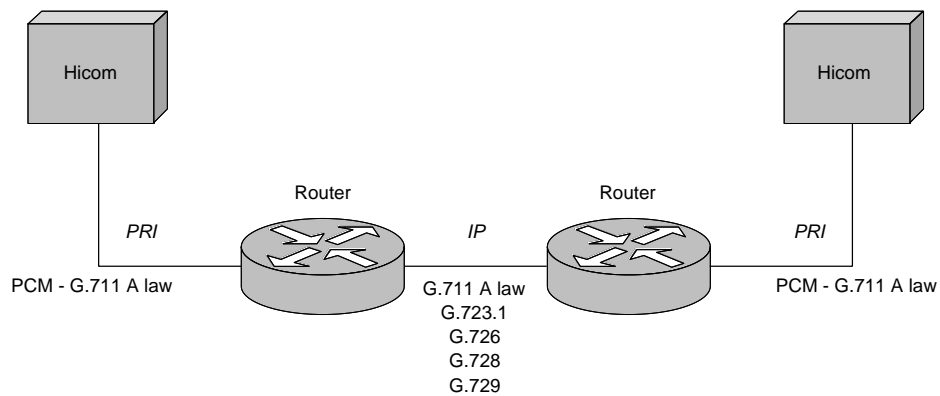


Abbildung 3.9: Kaskadierung von Codecs

Signal, welches in einem Router umcodiert und über den WAN-Link geschickt wird. Im 2. Router erfolgt die Rückwandlung in einen PCM-Strom.

3.4.2 G.723.1

G.723.1 bezeichnet einen hybriden Codierer. Er unterstützt 2 Bitraten, die durch unterschiedliche Methoden zur Gewinnung der Anregungsvektoren erreicht werden. Bei einer Geschwindigkeit von $5,3 \frac{kbit}{s}$ wird das ACELP-Verfahren (Algebraic Codebook-Excited Linear Predictive Coding) verwendet, bei dem der optimale Anregungsvektor aus einem adaptiven Codebuch entnommen wird. Für eine Bitrate von $6,4 \frac{kbit}{s}$ benutzt der Codierer zur Anregung MP-MLQ (Multi Pulse excitation with a Maximum Likelihood Quantizer), bei dem die Anregungsvektoren aus verschiedenen gewichteten Impulsen variabler Abstände generiert werden. Die Sprachqualität ist gut. Nachteilig sind eine hohe erforderliche Rechenleistung und hohe Verzögerungszeiten.

3.4.3 G.726

G.726 beruht auf einem Signalform-Coder-Verfahren. Es wird im Gegensatz zu PCM eine Datenkompression durchgeführt. Die Verringerung der Datenrate geht aus der Korrelation benachbarter Werte eines Signals hervor. Der Coder schätzt die Folgewerte eines Signals mit Hilfe eines Prädiktorfilters (lineare Prädiktion) und zieht diese vom Eingangswert ab. Das Differenzsignal besitzt eine geringere Varianz und läßt sich somit effizienter codieren. Adaptive Difference PCM beschreibt ein Verfahren, bei dem zusätzlich der Prädiktor adaptiv an die Signalstatistik angeglichen wird. ADPCM ist Grundlage für den Codec des G.726 Standards. Er arbeitet mit variablen Bitraten zwischen $16 \frac{kbit}{s}$ und $40 \frac{kbit}{s}$ bei guter Sprachqualität. Durch die Prädiktion entsteht eine geringe Verzögerung.

3.4.4 G.728

Der Standard G.728 stellt einen hybriden Codierer dar, der auf sehr kurze Verzögerungszeiten optimiert ist. Dazu werden kurze Blocklängen eingesetzt ($0,625 ms$ gegenüber $30 ms$ bei G.723.1). Die Anregungsvektoren werden einem Codebuch entnommen. Daher wird der Algorithmus LD-CELP (Low Delay CELP) genannt. Er arbeitet mit einer Datenrate von $16 \frac{kbit}{s}$ und ist wegen seiner geringen Verzögerungszeiten vor allem zur Übertragung synthetischer Signale, wie DTMF¹²- oder Modem-Signalen gedacht.

3.4.5 G.729

G.729 benutzt einen hybriden Codierer auf Basis von Codebüchern. Der optimale Anregungsvektor wird aus einer gewichteten Überlagerung zweier verschiedener Codebücher, z.B. eines algebraischen und eines adaptiven Codebuches bestimmt. Das Verfahren heißt CS-ACELP (Conjugate Structure ACELP). Es arbeitet mit einer Bitrate von $8 \frac{kbit}{s}$ mit relativ geringen Verzögerungszeiten.

3.4.6 Zusammenfassung

In der Tabelle 3.5 sind die oben genannten Codecs zusammengefaßt. Die angegebene Sprachqualität wurde nach ITU-T SG12 in einem ACR (Absolute Category Rating) Test

¹²Dial Tone Multi Frequency – Wähltöne eines Telefons mit Mehrfrequenzwahl

bestimmt. Dabei wird die Qualität subjektiv durch eine hohe Zahl an Probanden bestimmt. Im Allgemeinen wird dazu eine 5 Punkte Skala (5 = excellent bis 1 = schlecht) verwendet. Anschließend werden die Werte zu einem MOS (Mean Opinion Score) gemittelt. Die Spalte Verzögerung zeigt die Zeit, die ein Signal braucht, durch den Codec zu gelangen. In der Literatur sind unterschiedliche Werte für Rechenlast und Verzögerung

Codec	Verfahren	Bitrate $10^3 \frac{bit}{s}$	Sprachqualität <i>MOS</i>	Rechenlast <i>MIPS</i>	Verzögerung <i>ms</i>
G.711	PCM	64	4,3	0	0,125
G.723.1	ACELP/MP-MLQ	5,3/6,4	4,1	25	70
G.726	ADPCM	16, 24, 32, 40	2,0 - 4,3	6,5	0,125
G.728	LD-CELP	16	4,1	37,5	2
G.729	CS-ACELP	8	4,1	34	20

Tabelle 3.5: Übersicht der verwendeten Codecs (aus [Wal00])

für die einzelnen Codecs zu finden sind. Die in der Tabelle angegebenen Werte dienen daher nur als Richtwerte.

3.5 Prinzipien des Benchmarking

Um aussagekräftige Testergebnisse zu erhalten, müssen bestimmte Regeln eingehalten werden. An erster Stelle steht die Reproduzierbarkeit eines Tests. Damit ist gemeint, daß die Wiederholung einer Testreihe bei gleichen Umgebungsparametern zu den selben Ergebnissen führt. Dazu gehört vor allem eine ausführliche Dokumentation der Testumgebung und -durchführung. Die IETF hat Standards definiert, denen entnommen werden kann, wie ein Versuch durchzuführen und zu dokumentieren ist. [RFC 1242], [RFC 2330], [RFC 2544]

Weitere Punkte, auf die zu achten ist, sind nachfolgend wiedergegeben. So sind zur Durchführung einer Meßreihe nur eine überschaubare Anzahl an unbestätigten Annahmen oder Voraussetzungen zuzulassen, um nicht dadurch unbeachtet Fehler einzustreuen. Auftretende Meßabweichungen (z.B. endliche Meßgenauigkeit eines Meßgerätes) und/oder Fehler sind zu quantisieren und detailliert zu beschreiben. Gehören zum Test

verschiedene Meßreihen, ist nach Möglichkeit jeweils nur ein Parameter zwischen 2 Messungen zu verändern. So ist gewährleistet, daß bei komplexen Umgebungen der Zusammenhang von Ursache und Wirkung bestimmbar bleibt. Bei jedem Versuch sind mehrere Meßwerte aufzunehmen und anschließend die Ergebnisse zu mitteln (z.B. arithmetisch). So können Ausreißer eliminiert werden und nicht die Bewertung verfälschen. Zum Testen von Netzwerkgeräten sind Standardpakete und -größen zu verwenden. Weiterhin ist statt des Namens die zugehörige IP-Adresse eines Hosts zu verwenden, da ein DNS-Server keinen Einfluß auf das Benchmark-Resultat haben soll. Außerdem kann eine Maschine an mehrere Netze angeschlossen sein. Bei der Verwendung des Hostnamens ist nicht ausgeschlossen, daß das falsche Netz getestet wird. Meßwerte sind in Dezimalform anzugeben, d.h. 1 *kbyte* entspricht 1000 *byte* und nicht $2^{10} = 1024$ *byte*. Weiterhin sind bei der Planung eines Tests Überlegungen zur Sicherheit in Netzwerken wichtig, denn viele Benchmark-Szenarien ähneln einem Denial of Service (DoS) Angriff. Entsprechend den genannten RFC's befindet sich die gesamte Dokumentation der Tests inklusive der Konfigurationsdaten der Router im Anhang dieser Arbeit. [RFC 2330]

Im Folgenden wird auf Einschränkungen der obigen Punkte bei der Testdurchführung eingegangen. Sie beziehen sich vor allem darauf, daß nicht die Netzwerkperformance getestet wird, sondern ein Nachweis für die Funktionsfähigkeit der Queueing Algorithmen erbracht werden soll und somit das Hauptaugenmerk bei der Sprachübertragung liegt. Die Kriterien für die Durchsatztests, wie z.B. Meßfolgen mit verschiedenen Ethernet-Framegrößen finden hier keine Anwendung. [RFC 2544]

Die in Kapitel 4.2 beschriebenen Versuchsreihen haben eins gemein: Sie enthalten synthetische Benchmarks. Zur Verwertbarkeit der erhaltenen Meßwerte in Real-World Umgebungen befinden sich im genannten Abschnitt einige Ausführungen. Hier wird der Funktionsnachweis von Queueing Mechanismen für VoIP-Lösungen behandelt. Dementsprechend finden keine Vergleichstests zwischen Geräten verschiedener Hersteller statt. Im Kapitel 5 werden allerdings weitere Hersteller voice-fähiger Geräte aufgeführt. Zudem muß für einen Vergleich mehrerer Router die Testumgebung ausreichend berücksichtigt werden.

Die vorgeschlagenen Verfahren zur Bestimmung der Verzögerung von Paketen sind nicht durchführbar, da ein externes Meßgerät nicht zur Verfügung steht. Für die Versuche wird auf die Möglichkeit der Callanalyse durch die Router zurückgegriffen. Wei-

terhin ist darauf verzichtet worden, pro Meßreihe 20 Samples à 120 s aufzunehmen, da nach Meinung des Autors die in Abschnitt 4.3 vorgeschlagene Anzahl an Samples und die Meßdauer für den Nachweis der VoIP-Funktionalität ausreichen. [RFC 2544]

Kapitel 4

Definition der Testszenarien und praktische Versuche

In diesem Kapitel steht die Durchführung der Tests im Vordergrund. Es werden die Einzelheiten der praktischen Versuche dargestellt, Ergebnisse gezeigt, bewertet und mit den Erwartungen aus den theoretischen Grundlagen verglichen.

4.1 Testszenarien

Für die Analyse der Qualität der Queueing Algorithmen werden verschiedene Szenarien einer Büroumgebung getestet. Für den Telefonieverkehr sind 4 analoge Telefone der Firma Siemens vorhanden. Faxübertragungen werden durch einen PC mit einem externen Modem Dr. Neuhaus Smarty 56k und einem Siemens Thermopapierfaxgerät als Gegenstelle realisiert. Auf dem PC wird die Fax-Software PCphone Fax-Office 3.8 unter Windows 98 eingesetzt. Modemverbindungen werden mittels eines PC mit externem Modem Dr. Neuhaus Smarty 56k und eines PC mit externem Modem Elsa MicroLink 28.8TQV verwirklicht. Die Software HyperTerminal aus dem Lieferumfang von Windows 98 wird zur Verbindung der Modems benutzt. Mit diesen Geräten läßt sich eine Büroumgebung simulieren.

4.2 Testwerkzeuge

Eine reale VoIP Umgebung teilt die zur Verfügung stehende Bandbreite zwischen dem normalen Datenverkehr und den Voicepaketen auf. Um aussagekräftige Ergebnisse für den Test eines solchen Szenarios zu erzielen, ist die Verbindung der Router mit geeigneten Mitteln auszulasten. Für die Simulation des normalen Datenverkehrs wird ein

Benchmark-Tool eingesetzt. Das Programm Netperf von Hewlett-Packard ist eine freie Software zur Bestimmung der Geschwindigkeit eines Netzwerkes. Die vorliegende Version trägt die Nummer 2.1pl3. Es stehen Binaries für Windows NT und diverse UNIX-Derivate sowie der Quellcode von Netperf zur Verfügung. Der Sourcecode von Netperf ist auf den beiden Testmaschinen mit den Standardparametern kompiliert worden. [Hew96]

Netperf arbeitet Client/Server basiert. Als Client wird eine Workstation SUN Ultra 2 unter Linux (Kernelversion 2.2.16) benutzt. Ein PC (Celeron 566 MHz, 128 MB RAM) mit Linux (Kernel 2.2.15) dient als Server. Beide Rechner sind über die Ethernet Interfaces mit den Routern verbunden worden. Netperf beinhaltet mehrere Makros bzw. Testmethoden. Jedes Makro deckt einen speziellen Anwendungsbereich ab. Aus diesem Spektrum werden 3 Tests ausgewählt, die einen Großteil des Datenverkehrs repräsentieren. In der Tabelle 4.1 sind sie gegenübergestellt. UDP_STREAM und TCP_STREAM

Makro	Beschreibung	Maßeinheit
UDP_STREAM	UDP-Streaming-Test	$\frac{\text{bit}}{\text{s}}$
TCP_STREAM	TCP-Streaming-Test	$\frac{\text{bit}}{\text{s}}$
TCP_CRR	TCP Connect/Request/Response	$\frac{\text{Transactions}}{\text{s}}$

Tabelle 4.1: Getestete Netperf Makros

messen den reinen Datendurchsatz für UDP- oder TCP-Pakete. Die ermittelten Ergebnisse geben an, wie schnell ein System Daten zu einem anderen senden bzw. wie schnell das andere die Daten empfangen kann. Mit UDP_STREAM kann Streaming Video/Audio oder ähnliche ungesicherte und verbindungslose Datenübertragung (z.B. TFTP¹) simuliert werden. Der Test wird mit dem folgenden Kommando aufgerufen:

```
netperf -H 10.254.252.2 -t UDP_STREAM -f k -l 10 -- -m 1024
```

Dabei haben die einzelnen Parameter nachstehende Bedeutung:

¹Trivial File Transfer Protocol

- H 10.254.252.2 bezeichnet den Remote Host
- t UDP_STREAM gibt das Testverfahren an
- f k weist Netperf an, das Ergebnis in $10^3 \frac{\text{bit}}{\text{s}}$ auszugeben
- l 10 Länge des Test in Sekunden
- m 1024 Größe der gesendeten Pakete in *byte* (testspezifischer Parameter, wird deshalb durch -- von den globalen Parametern abgegrenzt)

Optional kann die Größe der Socket Buffer für den lokalen Rechner und den Server angepasst werden. Diese Werte sind jedoch bei ihren Standardeinstellungen belassen worden. Netperf verwendet dann die Systemgrößen. In Abbildung 4.1 ist eine Beispielausgabe von Netperf mit der angegebenen Kommandozeile dargestellt. Der Abbildung

```
UDP UNIDIRECTIONAL SEND TEST to 10.254.252.2
```

Socket	Message	Elapsed	Messages		
Size	Size	Time	Okay	Errors	Throughput
bytes	bytes	secs	#	#	10^3bits/sec
65535	1024	10.00	114730	0	93998.26
65535		10.00	176		144.32

Abbildung 4.1: Netperf Beispielausgabe UDP_STREAM

können neben den erzielten Durchsatzraten auch die Testeinstellungen (Puffer- und Paketgrößen) entnommen werden. Netperf gibt die erreichbare Datenrate in Sende- und Empfangsrichtung an. In der Auswertung werden jeweils nur die Empfangsraten verarbeitet, da in Senderichtung nur die Geschwindigkeit des Netzwerkinterfaces des Clients gemessen wird.

Das TCP_STREAM-Makro dient der Simulation von TCP-Datenströmen, wie sie z.B. durch FTP hervorgerufen werden. Der Test wird durch Eingabe von

```
netperf -H 10.254.252.2 -t TCP_STREAM -f k -l 10
```

gestartet. Da TCP jedes Paket quittiert, ist die Datenrate in Sende- und Empfangsrichtung identisch. Die Abbildung 4.2 zeigt ein typisches Ergebnis für den TCP_STREAM Test. Die Einstellungen für die Paketgrößen sind ebenfalls ersichtlich.

```

TCP STREAM TEST to 10.254.252.2
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time  Throughput
bytes bytes bytes secs.  10^3bits/sec

87380 65535 65535 10.00 120.03

```

Abbildung 4.2: Netperf Beispielausgabe TCP_STREAM

Das letztgenannte Makro TCP_CRR dient der Simulation des HTTP-Protokolls, welches das Rückgrat des WWW bildet. Hierbei erzeugt der Client eine Anforderung (request). Der Server verschickt daraufhin eine Antwort (response). Eine Transaction ist definiert als Austausch einer Anforderung und einer Antwort. TCP_CRR errichtet für jede Transaction eine neue Verbindung (connect). Aus der Transactionrate kann die durchschnittliche Verzögerung des Übertragungsweges hergeleitet werden, wenn die Antwortzeit des Servers vernachlässigt wird. Der Test wird durch Eingabe von:

```
netperf -H 10.254.252.2 -t TCP_CRR -l 10 -- -r 32,1024
```

gestartet. Der testspezifische Parameter `-r 32,1024` steuert die Größe der Pakete für die Anforderung und Antwort. In der Dokumentation von Netperf wird empfohlen, den Test eine angemessene Zeit² laufen zu lassen, da die Ergebnisse sonst überhöhte Werte aufweisen können. Trotzdem wird der Test eine kürzere Zeit gefahren, da er hauptsächlich zur Belastung des Links dient und nicht zur Bestimmung der maximalen Geschwindigkeit der Verbindung. [Hew96]

In der Abbildung 4.3 ist eine Beispielausgabe für den TCP_CRR Test zu sehen. Die Einstellungen für die Puffer- und Paketgrößen können ebenfalls der Abbildung entnommen werden.

Bei Betrachtung der einzelnen Makros existiert folgende Einschränkung hinsichtlich der Relevanz in realen Umgebungen: durch ein Makro wird gleichzeitig lediglich ein Teilbereich des Datenaufkommens abgedeckt. Der „normale“ Verkehr zwischen den Routern besteht aus einem Gemisch von UDP- oder TCP-Strömen und HTTP-Requests.

²Vorgeschlagen werden 120 s.

```

TCP Connect/Request/Response TEST to 10.254.252.2
Local /Remote
Socket Size   Request  Resp.   Elapsed  Trans.
Send   Recv   Size     Size     Time     Rate
bytes  Bytes  bytes    bytes    secs.    per sec

131070 131070 32        1024     9.99      9.40
16384  87380

```

Abbildung 4.3: Netperf Beispielausgabe TCP_CRR

Bei der Bewertung der Ergebnisse oder dem Vergleich mit anderen Umgebungen (auch den realen VoIP Umgebungen) ist das entsprechend zu berücksichtigen. Sollen aussagekräftige Ergebnisse für Real-World Szenarien erzielt werden, muß die Verbindung der Router durch echtes Verkehrsaufkommen belastet werden. Solch ein Test ist dann allerdings nicht mehr deterministisch und reproduzierbar, sondern spiegelt nur die Resultate für genau jene Umgebung wider. Weiterhin muß der Umstand berücksichtigt werden, daß das Stressen der Verbindung mittels Netperf nur einen Grenzfall darstellen kann, weil die hundertprozentige Auslastung des Links atypisch ist.

Die qualitative Bewertung der VoIP Funktionalität stützt sich auf eine subjektive Hörwertung bei den Telefonverbindungen. Werte unter 80 % werden als ungeeignet für den Einsatz in realen Umgebungen angesehen. Für Fax- und Modemübertragungen ist ohne Hilfsmittel keine Beurteilung möglich, da beide Verfahren Fehlerkorrekturmechanismen implementieren. Allerdings erlauben die Router C2610 und C3620 eine Callanalyse per

```
show call active voice
```

Kommando. Hierbei machen die Router Informationen, wie Dauer der Verbindung, Connection-ID, Rufnummern der Teilnehmer usw. sichtbar. Die hier interessierenden Punkte sind in Tabelle 4.2 zusammengefasst.

Für die Simulation von Telefongesprächen ist Sprache³ in Form einer Nachrichtensendung aufgenommen und über eine Lautsprecher-Telefonhörer-Konstruktion ein Quellensignal erzeugt worden. Am anderen Teilnehmerapparat ist die Aufzeichnung

³Musik ist nach Abschnitt 3.4 nur bedingt zur Übertragung geeignet.

Callanalyse	Beschreibung	Maßeinheit
ReceiveDelay	Verzögerung der Übertragungsstrecke	<i>ms</i>
LostPackets	Anzahl/Dauer der verworfenen Pakete	<i>ms</i>
EarlyPackets	Anzahl/Dauer zu früh eingetroffener Pakete	<i>ms</i>
LatePackets	Anzahl/Dauer zu spät eingetroffener Pakete	<i>ms</i>

Tabelle 4.2: Meßpunkte Callanalyse

abgehört worden. Da sich der Lautsprecher auch in Hörreichweite befindet, ist es möglich, die Sprachqualität und Verzögerung genauer zu bestimmen, da Abweichungen vom Original leicht zu erkennen sind. Im Normalfall ist das Referenzsignal nicht zu hören, so daß es nicht nötig ist, den Maßstab für die Latenz so hoch anzusetzen.

Eine solche Testumgebung kommt einer Simplexverbindung gleich. Es ist daher bei den unidirektionalen Netperf-Tests (UDP_STREAM) darauf zu achten, daß die Richtung beider Flows (Netperf und Voice) übereinstimmt. Die Callanalyse der Telefonverbindungen ist durch die definierte Laufzeit von Netperf reproduzierbar. Dabei wird jedoch angenommen, daß sich mehrere gleichzeitig geführte Verbindungen nicht gegenseitig infolge mangelnder Bandbreite beeinflussen. In einem solchen Fall wäre die Analyse der Telefongespräche aufgrund der manuell gesteuerten Verbindungen und der dadurch hervorgerufenen unterschiedlich langen betrachteten Zeiträume nur bedingt reproduzierbar.

Zum Faxen ist ein einseitiges Dokument verwendet worden, welches vorrangig Text enthält. Bei dem Versuch wird nach Ablauf der Initialisierungsphase Netperf gestartet. Die Callanalyse wird nach erfolgtem Faxversand (bzw. Abbruch) durchgeführt und berücksichtigt somit nur die parallel geführten Telefongespräche, da die Router keine Untersuchung bereits beendeter Telefon- oder Faxverbindungen erlauben. Eine Callanalyse während der Übertragung eines Faxes ist nicht zweckmäßig, weil sich kein vollständiger Fax-Call (Initialisierung und Übermittlung der Daten) betrachten läßt. Da durch Paketverlust hervorgerufene Übertragungsfehler von einer Fehlerkorrektur weitgehend beseitigt werden, kann eine Beurteilung der QoS Algorithmen lediglich anhand der Übertragungsdauer erfolgen.

Die Modemverbindung soll durch den Transfer einer 1000000 *byte* langen Datei ausgelastet werden. Hinsichtlich der Callanalyse und der Richtung der Flows gelten

die selben Überlegungen wie bei den Telefonverbindungen. Übertragungsfehler werden durch ein CRC-Verfahren korrigiert.

4.3 Meßwerterfassung

In jeder Meßreihe werden die Funktionen der Tabelle 4.3 abgearbeitet. Jede Meßprobe enthält die Ergebnisse des Netperf-Benchmarks, den subjektiven Höreindruck der/des Telefongespräche(s) sowie die Callanalyse der Router. Entsprechend Kapitel 3.5 werden für jeden Meßpunkt 3 Samples aufgenommen und arithmetisch gemittelt. Bei den Faxübertragungen ist die Meßwerterfassung auf ein Sample beschränkt worden, da der Test einen unverhältnismäßig hohen Zeitbedarf besitzt. Aus dem selben Grund wird nur ein Makro getestet. Dafür ist UDP_STREAM ausgewählt worden, da es die Voice-Calls am stärksten beeinträchtigt. Der Funktionsnachweis kann trotzdem erbracht werden.

Nummer	Beschreibung	Anzahl Samples
1	1 Telefongespräch	3
2	2 Telefongespräche	3
3	1 Telefongespräch und 1 Faxübertragung	1
4	2 Telefongespräche und 1 Faxübertragung	1
5	1 Telefongespräch und 1 Modemverbindung	3
6	2 Telefongespräche und 1 Modemverbindung	3
7	1 Telefongespräch, 1 Faxübertragung und 1 Modemverbindung	1
8	2 Telefongespräche, 1 Faxübertragung und 1 Modemverbindung	1

Tabelle 4.3: Testdurchlauf

Die Dauer der Beeinflussung durch Netperf beträgt 10 s. Bei den Faxtests wird die Testdauer auf 30 s verlängert, da nur ein Versuch stattfindet und somit eine höhere Meßdauer zu mehr Genauigkeit führt. Die Tatsache, daß die Testdauer 3 mal so lang wie bei Telefon- und Modemversuchen ist, muß in der Auswertung entsprechend berücksichtigt werden.

Um den Vergleich einzelner Meßreihen nicht zu komplex zu gestalten, wird die Anzahl der zu bewertenden Ergebnisse verringert. Dazu werden die Resultate von Lost-

Packets, EarlyPackets und LatePackets zu einem gewichteten Mittelwert zusammengefasst, der die Anzahl/Dauer der nicht übertragenen bzw. zum falschen Zeitpunkt eingetroffenen Pakete angibt. In den Mittelwert gehen die Ergebnisse von LostPackets zu 50 % sowie EarlyPackets und LatePackets zu jeweils 25 % ein. Die Wichtung hat ihre Berechtigung in der Tatsache, daß verloren gegangene Pakete den Höreindruck eher schmälern als zu früh oder spät eingetroffene.

Aus dem Spektrum an verfügbaren Mechanismen (siehe Kapitel 3) sind die folgenden Algorithmen zum Testen ausgewählt worden:

1. FIFO-Queueing
2. Weighted Fair Queueing
3. Weighted Fair Queueing kombiniert mit IP RTP Priority
4. Class-Based Weighted Fair Queueing (Klasse: Voice/Tail Drop)
5. Class-Based Weighted Fair Queueing (Klassen: Voice/Tail Drop, Daten/Tail Drop)
6. Class-Based Weighted Fair Queueing (Klassen: Voice/Tail Drop, Daten/WRED)
7. Class-Based Weighted Fair Queueing (Klassen: Voice/LLQ, Daten/WRED)
8. Priority Queueing
9. Priority Queueing bandbreitenlimitiert mit Committed Access Rate
10. Compressed Real-Time Protocol

FIFO-Queueing dient hierbei nur zur Darstellung des Worst Case, da der Algorithmus keinerlei Mittel zur Priorisierung von Sprachdaten beinhaltet. Die Konfiguration der beiden Router für diesen Test kann dem Anhang (siehe Abschnitt A.1.1) entnommen werden. Die Einstellungen für WFQ sind im Abschnitt A.1.2 zu finden. In der nächsten Versuchsreihe wird mittels IP RTP Priority dem Voice-Verkehr eine Bandbreite von $100 \frac{\text{kbit}}{\text{s}}$ garantiert (siehe Kapitel A.1.3).

CBWFQ wird in 4 Variationen getestet, um die verschiedenen Optionen auf ihre Eignung zur Sprachübermittlung zu untersuchen. Zuerst ist nur eine Klasse für Voice-Pakete definiert worden. Läuft die Queue der Klasse voll, werden die dann eintreffenden

Pakete verworfen (Tail Drop). Alle anderen Daten sind in der immer vorhandenen *class-default* enthalten. In einem zweiten Schritt wird eine 2. Klasse etabliert, die die Pakete des Ethernet-Interfaces aufnimmt. Hier ist wiederum Tail Drop aktiv. Im nächsten Versuch wird Weighted Random Early Detection für die Datenklasse eingestellt, um die Auslastung des Links besser kontrollieren zu können. Zuletzt wird innerhalb der Voice-Klasse Low Latency Queueing aktiviert, um eine strikte Priorisierung zu ermöglichen. Der Klasse werden $200 \frac{kbit}{s}$ Bandbreite garantiert. Die Konfigurationsdaten dieser 4 Versuche kann dem Kapitel A.1.4 entnommen werden.

Die nächsten 3 Tests implementieren PQ. Sprachdaten erhalten die höchste Priorität, alle anderen die niedrigste. Der zweite Test beschränkt mittels der Rate Limiting Funktion von CAR die für Sprache zur Verfügung stehende Bandbreite auf $96 \frac{kbit}{s}$. Im letzten Test wird versucht, die Effizienz der Verbindung durch CRTP zu steigern. Die für den Tests nötigen Einstellungen sind in Abschnitt A.1.5 bis A.1.7 dargestellt.

RSVP ist ein Protokoll, daß es ermöglichen soll, QoS Merkmale in einem heterogenen Netzwerk zu reservieren. RSVP ist nicht mit in die Testreihen aufgenommen worden, da eine Verbindung von 2 Routern kaum Aussagekraft besitzt.

Weiterhin werden die im Kapitel 3.4 genannten Codecs hinsichtlich ihrer Sprachqualität (subjektiv) getestet und ihre Eignung zur Faxübertragung und Modemverbindung überprüft.

Bei ersten Probeversuchen ist festgestellt worden, daß es nicht möglich ist, die Sprachqualität durch Benutzung von Netperf zu verringern. Hierbei entspricht der Testaufbau dem Worst Case Szenario, d.h. als Queueing-Algorithmus kommt FIFO-Queueing zum Einsatz. Außerdem wird der Codec G.711 mit $64 \frac{kbit}{s}$ verwendet und zusätzlich Voice Activity Detection (VAD) abgeschaltet⁴. Es ist augenscheinlich, daß mehr Voice-Verbindungen nötig sind, um Einbußen im Höreindruck hervorzurufen. Mangels weiterer Endgeräte muß also die Übertragungskapazität verringert werden. Die Siemens WAN-Modems unterstützen dies nicht und sind daher entfernt worden. Die Router sind im weiteren Testverlauf direkt (seriell) gekoppelt worden. Am C3620 kann die Übertragungsgeschwindigkeit geändert werden. Für die folgenden Versuche wird mit Bitraten von $125 \frac{kbit}{s}$ und $256 \frac{kbit}{s}$ gearbeitet. Der revidierte Testaufbau ist in Abbildung 4.4 dargestellt.

⁴VAD ist während der gesamten Testzeit deaktiviert, da diese Versuche immer den Grenzfall einer VoIP Umgebung darstellen sollen.

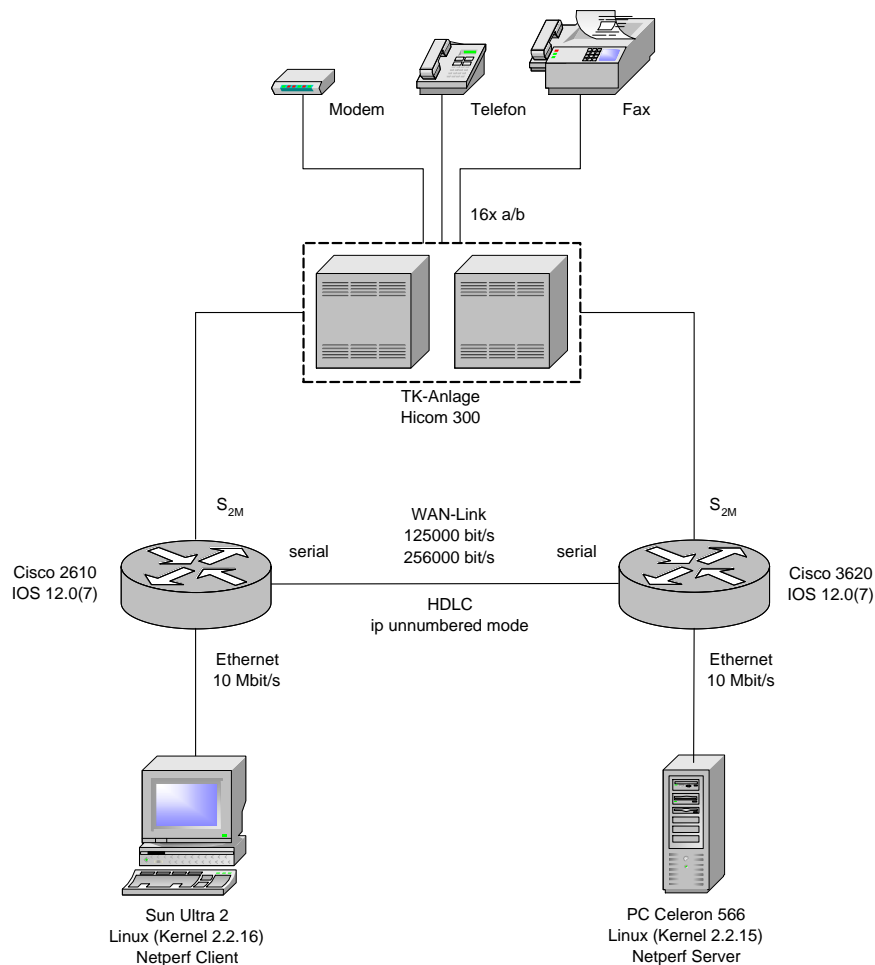


Abbildung 4.4: Revidierter Testaufbau

Ein weiteres Problem ergibt sich bei der Modemübertragung. Die beiden Rechner bauen eine Verbindung zueinander auf, eine Kommunikation innerhalb des Terminals ist jedoch prinzipiell nicht möglich. Eine Nachfrage bei Cisco ergab, daß Modemverbindungen grundsätzlich nicht möglich seien. Faxübertragungen sind dagegen durchführbar, da sie als solche vom Router erkannt und entsprechend behandelt werden. Da analoge Modems heute weitgehend an Bedeutung verloren haben, ist es kein schwerwiegendes Problem, daß Rechnerverbindungen mit Modems nicht funktionieren. Daraus ergibt sich eine Verkürzung des Versuchsablaufes (siehe Tabelle 4.3) auf die Punkte 1 bis 4.

Als Ergebnis der Versuchsreihen ist umfangreiches Datenmaterial entstanden. Für jeden Queueing Mechanismus sind bei einer Bitrate von $125 \frac{kbit}{s}$ die Punkte 1 und 2 der Tabelle 4.3 und bei $256 \frac{kbit}{s}$ die Punkte 1 bis 4 (jeweils mit den 3 Netperf-Makros) abgearbeitet und Werte aufgenommen worden. Das entstandene Material ist in den Anhang B ausgegliedert worden. Im Folgenden werden nur Zusammenfassungen der Ergebnisse wiedergegeben.

4.4 Meßergebnisse

In diesem Abschnitt werden die Ergebnisse der Versuchsreihen dargestellt und interpretiert. Auf Besonderheiten wird gezielt hingewiesen.

Zunächst ist der maximale Datendurchsatz der Testumgebung gemessen worden. Außer Netperf hat keine weitere Anwendung auf das Medium zugegriffen. Die Meßwerte geben daher die maximal erreichbare Übertragungsgeschwindigkeit an. In der Tabelle 4.4 sind die Ergebnisse und in Tabelle B.1 alle Meßwerte aufgelistet. Wie zu erkennen ist, liegen die UDP_STREAM Ergebnisse über den eingestellten Bitraten. Wahrscheinlich ist die Clock Rate tatsächlich höher, als im Router angegeben.

Clock Rate	Netperf Makro		
	UDP_STREAM	TCP_STREAM	TCP_CRR
$10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$
125	144,30	120,06	9,40
256	297,56	246,72	18,81

Tabelle 4.4: Ergebnisse Netperf - unbelasteter Link

4.4.1 Vergleich der QoS-Algorithmen

4.4.1.1 FIFO-Queueing

FIFO-Queueing stellt den Worst Case dar, da es keinerlei Priorisierungsmechanismen bereitstellt. Diese Tatsache spiegelt sich im Versuch wider. Beim Start von Netperf (UDP_STREAM) bricht der Voice-Call ab. Erst nach Beendigung des Netperf-Laufs setzt die Sprachübertragung wieder ein. In der Tabelle 4.5 ist die Ursache erkennbar:

die Meßwerte für das unbelastete Medium und für 1 bzw. 2 Calls stimmen überein, d.h. es steht keine Bandbreite für Sprache zur Verfügung. Das Verhältnis zeigt sich auch bei

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro		
		UDP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_CRR $\frac{\text{Trans.}}{\text{s}}$
125	–	144,30	120,06	9,40
	1 call	144,24	93,89	3,80
	1 call + 1 fax	128,66	/	/
256	–	297,56	246,72	18,81
	1 call	297,56	201,24	15,81
	1 call + 1 fax	264,68	/	/
	2 calls	296,64	198,76	8,20
	2 calls + 1 fax	263,78	/	/

Tabelle 4.5: Ergebnisse Netperf - FIFO

der Callanalyse (siehe Tabelle 4.6). Der Einsatz des UDP_STREAM Makros bedingt durchweg eine höhere Verzögerungszeit (ReceiveDelay) sowie einen hohen Wert für falsch übertragene Pakete. Dementsprechend ist die Verständlichkeit mit 0 % bewertet worden.

Die längere Testdauer bei der Faxübertragung ist bereits in den Meßwerten und Ergebnissen berücksichtigt, indem alle Werte durch 3 dividiert worden sind. Die Netperf-Werte der Faxversuche zeigen, daß ein Faxcall vom Router erkannt und auch mit FIFO-Queueing priorisiert wird.

Wie im Anhang (siehe Tabelle B.3) ersichtlich ist, sind nach Angabe der Router nur 699 ms an Sprachpaketen verloren gegangen, obwohl die Sprache mindestens 10 s ausgefallen ist. Die Werte der Callanalyse der Router stimmen offensichtlich nicht, jedoch erlauben sie den Vergleich innerhalb der Versuchsreihen. Weiterhin ist in der Tabelle ein Wert eingeklammert. Hierbei handelt es sich augenscheinlich um einen „Ausreißer“. Da Netperf nur 10 s aktiv ist, können nicht Pakete mit einer Gesamtlänge von 62299 ms \approx 62 s verloren gegangen sein. Solche offenkundig falschen Meßwerte sind durch Klammerung gekennzeichnet und nicht mit in die Bewertung aufgenommen worden.

Es sei darauf hingewiesen, daß das Wort „unbelastet“ (in den Tabellen durch „–“ dargestellt) bei den Ergebnissen der Netperf-Tests und der Callanalyse unterschiedliche Bedeutung hat. Bei den Netperf-Versuchen ist damit gemeint, daß kein Voice-Call aktiv ist. Bei der Callanalyse besagt es, daß die Kapazität der Leitung der Sprachübertragung exklusiv zur Verfügung steht.

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	53	0	100
		UDP_STREAM	1	64	382,5	0
		TCP_STREAM	1	64	335,8	10
		TCP_CRR	1	53	0	100
	1 call + 1 fax	UDP_STREAM	1	64	385,0	0
256	1 call	–	1	53	0	100
		UDP_STREAM	1	64	313,0	0
		TCP_STREAM	1	62	214,8	10
		TCP_CRR	1	53	0,3	100
	1 call + 1 fax	UDP_STREAM	1	64	313,3	0
	2 calls	–	1	53	0,3	100
			2	53	0,3	100
		UDP_STREAM	1	64	315,3	0
			2	64	314,3	0
		TCP_STREAM	1	53	289,5	10
			2	57	289,3	10
		TCP_CRR	1	53	0	100
			2	52	0,3	100
	2 calls + 1 fax	UDP_STREAM	1	64	335,0	0
			2	64	321,5	0

Tabelle 4.6: Ergebnisse Call-Analyse - FIFO

Bei Verwendung des TCP_STREAM Makros bricht die Verbindung für die Dauer des Netperf-Laufs bis auf einzelne Sprachfetzen ebenfalls ab. Diese leichte Verbesserung ist auf die Flußkontrolle von TCP zurückzuführen. Sinkt die Datenrate des Senders,

wie in Abschnitt 3.3.3.1 beschrieben, kurz ab, können vereinzelt Sprachpakete übertragen werden. Die Ergebnisse bekräftigen die Einschätzung.

Der Test mit dem TCP_CRR Makro konnte die Leitung nicht soweit auslasten, um die Voice-Calls zu beeinflussen. Normaler Webverkehr beeinträchtigt die Voicefunktionalität dementsprechend nur unwesentlich.

Die Ergebnisse belegen auch, daß ein Faksimile als solches erkannt und entsprechend bevorzugt behandelt wird. Der Tabelle B.41 sind die Übertragungszeiten bei erfolgreich abgeschlossenem Faxversand und die Zuordnung von Versuch und beigelegten Faxdokumenten zu entnehmen.

4.4.1.2 Weighted Fair Queueing

In den Ergebnissen (Tabelle 4.7 und 4.8) spiegelt sich wider, daß WFQ die zur Verfügung stehende Bandbreite auf die verschiedenen Flows fair aufteilt. Bei einer Clock Rate von

Clock Rate	Test	Netperf Makro		
		UDP_STREAM	TCP_STREAM	TCP_CRR
$10^3 \frac{bit}{s}$		$10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$
125	–	144,30	120,06	9,40
	1 call	62,88	50,56	3,80
	1 call + 1 fax	48,07	/	/
256	–	297,56	246,72	18,81
	1 call	198,31	164,93	15,88
	1 call + 1 fax	185,32	/	/
	2 calls	100,52	83,61	8,20
	2 calls + 1 fax	86,02	/	/

Tabelle 4.7: Ergebnisse Netperf - WFQ

$125 \frac{kbit}{s}$ ist der Voice-Flow größer als der des Datenverkehrs. Dementsprechend sinkt die Sprachqualität auf 50 %. Mit $256 \frac{kbit}{s}$ kehrt sich das Verhältnis. Es gehen nur noch vereinzelt Sprachpakete verloren. Das macht sich beim Hören jedoch nicht bemerkbar. Deswegen ist die Verständlichkeit mit 100 % bewertet worden. Die Hörqualität bricht mit 2 aufgebauten Telefongesprächen nicht ein, da jedes Gespräch einen eigenständigen Flow bildet. Somit wird die Bandbreite nicht zwischen Sprache und Datenverkehr

halbiert, sondern auf alle 3 Flows aufgeteilt.

Die Meßwerte für den Test 1 call + 1 fax @ 256 $\frac{kbit}{s}$ können nicht zugeordnet werden, da eigentlich genügend Bandbreite zur Verfügung steht. Wahrscheinlich handelt es sich bei dem Ergebnis um einen Einzelfall und nicht um ein allgemeines Problem. Daneben ist eine erhöhte Verzögerung bei der Verwendung des UDP_STREAM Makros auffällig.

Clock Rate $10^3 \frac{bit}{s}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	53	0	100
		UDP_STREAM	1	70	124,0	50
		TCP_STREAM	1	49	152,8	50
		TCP_CRR	1	53	0,3	100
	1 call + 1 fax	UDP_STREAM	1	56	327,3	20
256	1 call	–	1	53	0	100
		UDP_STREAM	1	60	1,0	100
		TCP_STREAM	1	57	3,0	100
		TCP_CRR	1	53	0,3	100
	1 call + 1 fax	UDP_STREAM	1	64	349,3	20
	2 calls	–	1	52	0	100
			2	52	0,3	100
		UDP_STREAM	1	52	54,8	100
			2	56	16,0	100
		TCP_STREAM	1	54	13,8	100
			2	53	28,5	100
		TCP_CRR	1	52	0	100
			2	52	0,3	100
	2 calls + 1 fax	UDP_STREAM	1	51	205,5	50
			2	45	158,0	50

Tabelle 4.8: Ergebnisse Call-Analyse - WFQ

Zusätzlich ist die Versuchsreihe mit veränderten IP Precedence Werten durchgeführt worden. Dazu ist der Router (im Dial Peer) derart konfiguriert worden, daß er im Header

der erzeugten IP Pakete den TOS-Wert von voreingestellten 0 = routine auf 5 = critic/ecp verändert. Die Ergebnisse sind entgegen der Annahme aus Kapitel 3.3.2.2 unverändert ausgefallen. Deswegen werden nur die Meßwerte der ersten Testreihe mit dem IP Precedence Wert 0 angegeben. Die Arbeit mit verschiedenen TOS erscheint im Moment nicht sinnvoll, da in heterogen Netzen die IP Precedence Bits von den meisten Netzknoten ignoriert werden.

4.4.1.3 IP RTP Priority

Mit IP RTP Priority wird bei Nutzung von Weighted Fair Queueing eine strikte Priorisierung erreicht, was durch die Ergebnisse in Tabelle 4.9 und 4.10 deutlich wird. Selbst bei einer Clock Rate von $125 \frac{kbit}{s}$ ist die Übertragungsqualität sehr gut. Allerdings ist wie in den vorangegangenen Kapiteln die ReceiveDelay bei Belastung mit einem UDP Datenstrom höher.

Clock Rate	Test	Netperf Makro		
		UDP_STREAM	TCP_STREAM	TCP_CRR
$10^3 \frac{bit}{s}$		$10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$
125	—	144,30	120,06	9,40
	1 call	45,91	37,98	3,50
	1 call + 1 fax	31,53	/	/

Tabelle 4.9: Ergebnisse Netperf - IP RTP Priority

Der Test 1 call + 1 fax zeigt, daß $125 \frac{kbit}{s}$ nicht für die Übertragung eines $64 \frac{kbit}{s}$ Calls, eines Faxes und zusätzlichem Datenverkehr ausreichen. In der Tabelle 4.9 ist der Grund zu erkennen: der Codec G.711 benötigt eine Bandbreite von knapp $100 \frac{kbit}{s}$.

WFQ weist schon ohne Zuhilfenahme von IP RTP Priority bei $256 \frac{kbit}{s}$ nur geringe Verluste auf. Daher wird hier auf die Tests mit dieser Clock Rate verzichtet.

Entgegen der Beschreibung in Kapitel 3.3.2.6 ist nicht zu erkennen, daß mit IP RTP Priority auch das Compressed Real-Time Protocol aktiviert worden ist, da die Ergebnisse nicht mit denen aus Abschnitt 4.4.1.6 übereinstimmen. Die Größenordnung gleicht dagegen eher des Priority Queueing (Ergebnisse in Kapitel 4.4.1.5) ohne aktiviertes CRTP.

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	52	0	100
		UDP_STREAM	1	66	1,3	100
		TCP_STREAM	1	50	1,0	100
		TCP_CRR	1	52	0	100
	1 call + 1 fax	UDP_STREAM	1	51	230,3	50

Tabelle 4.10: Ergebnisse Call-Analyse - IP RTP Priority

4.4.1.4 Class-Based Weighted Fair Queueing

Bei den nächsten Versuchen findet CBWFQ Verwendung. Für den ersten Test ist eine Klasse für Sprachdaten definiert worden. Den nachstehenden Tabellen kann entnommen werden, daß Sprache ohne Einstellung zur Priorisierung vom Router bevorzugt übertragen wird. Der Router gewährt Paketen der Klasse *class-default* schon in der Standardeinstellung keinen Vorrang. Der *class-default* gehören in der Konfiguration die Pakete des Datenverkehrs (Netperf) an.

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro		
		UDP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_CRR $\frac{\text{Trans.}}{\text{s}}$
125	–	144,30	120,06	9,40
	1 call	46,70	38,02	3,50
	1 call + 1 fax	35,89	/	/
256	–	297,56	246,72	18,81
	1 call	199,10	164,74	12,51
	1 call + 1 fax	141,73	/	/
	2 calls	100,76	83,13	7,10
	2 calls + 1 fax	56,54	/	/

Tabelle 4.11: Ergebnisse Netperf - CBWFQ (1 Klasse: Voice - TD)

Der Versuch zeigt (siehe Tabelle 4.12), daß die Verzögerung der Sprachpakete bei

Einsatz des UDP_STREAM Makros nicht prinzipiell hoch ausfallen muß. Bei ausreichend hoher Bandbreite ($256 \frac{kbit}{s}$) stimmen die Werte für alle 3 Makros mit dem unbelasteten Fall überein.

Allerdings ist eine Lösung mit nur einer Klasse äußerst unflexibel und führt die Idee des Class-Based Weighted Fair Queueing ad absurdum, da nur zwischen 2 verschiedenen Datenaufkommen unterschieden werden kann, von denen einer der beiden priorisiert wird.

Clock Rate $10^3 \frac{bit}{s}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	53	0	100
		UDP_STREAM	1	63	1,8	100
		TCP_STREAM	1	48	2,3	100
		TCP_CRR	1	53	0	100
	1 call + 1 fax	UDP_STREAM	1	62	216,3	50
256	1 call	–	1	53	0,3	100
		UDP_STREAM	1	53	0,3	100
		TCP_STREAM	1	54	0	100
		TCP_CRR	1	52	0	100
	1 call + 1 fax	UDP_STREAM	1	53	39,0	100
	2 calls	–	1	52	0,3	100
			2	53	3,0	100
		UDP_STREAM	1	53	2,3	100
			2	53	3,0	100
		TCP_STREAM	1	67	4,5	100
			2	54	4,0	100
		TCP_CRR	1	54	0	100
			2	53	0,3	100
	2 calls + 1 fax	UDP_STREAM	1	53	6,3	100
			2	52	0	100

Tabelle 4.12: Ergebnisse Call-Analyse - CBWFQ (1 Klasse: Voice - TD)

Mit der Definition einer 2. Klasse (*data_class*) für die Daten des Ethernet-Interfaces werden Voice und Daten wieder gleichrangig behandelt. Die Ergebnisse (siehe Tabelle 4.13 und 4.14) ähneln daher dem des Weighted Fair Queueing, d.h. bei $125 \frac{kbit}{s}$ ist der Höreindruck schlecht. Bei einer Bitrate von $256 \frac{kbit}{s}$ wird ein Telefongespräch dagegen

Clock Rate $10^3 \frac{bit}{s}$	Test	Netperf Makro		
		UDP_STREAM $10^3 \frac{bit}{s}$	TCP_STREAM $10^3 \frac{bit}{s}$	TCP_CRR $\frac{Trans.}{s}$
125	–	144,30	120,06	9,40
	1 call	69,66	50,62	4,93
	1 call + 1 fax	60,64	/	/
256	–	297,56	246,72	18,81
	1 call	199,22	165,46	15,81
	1 call + 1 fax	147,32	/	/
	2 calls	133,30	103,41	9,80
	2 calls + 1 fax	124,53	/	/

Tabelle 4.13: Ergebnisse Netperf - CBWFQ (2 Klassen: Voice - TD, Data - TD)

gut übertragen. Im Unterschied zum WFQ treten bei mehreren gleichzeitigen Gesprächen vermehrt Übertragungsfehler auf. Das kann darauf zurückgeführt werden, daß der Klasse *voice_class* bei Überlastung der Verbindung nur 50 % der verfügbaren Bandbreite zugestanden werden. Das reicht nicht aus, um 2 Gespräche mit jeweils $64 \frac{kbit}{s}$ zu übertragen. Der Versand eines Faxes verringert die Übertragungskapazität für die *voice_class* zusätzlich und bringt daher einen der höchsten Mittelwerte für falsch übertragene Sprachpakete mit sich. Es kann allerdings nicht erklärt werden, warum der subjektive Höreindruck so weit vom Ergebnis der Callanalyse abweicht. Die Verzögerungszeiten sind wiederum auffällig hoch.

Die Tabelle B.14 zeigt, daß der Router Probleme hat, bei der Callanalyse Verzögerung der Pakete (ReceiveDelay) und zu spät eingetroffene Pakete (LatePackets) zu unterscheiden.

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	54	0	100
		UDP_STREAM	1	64	109,3	50
		TCP_STREAM	1	55	152,0	50
		TCP_CRR	1	53	0	100
	1 call + 1 fax	UDP_STREAM	1	64	228,3	50
256	1 call	–	1	52	0	100
		UDP_STREAM	1	53	0,3	100
		TCP_STREAM	1	54	3,3	100
		TCP_CRR	1	53	0,3	100
	1 call + 1 fax	UDP_STREAM	1	53	231,8	50
	2 calls	–	1	53	0,3	100
			2	53	0,3	100
		UDP_STREAM	1	63	100,8	80
			2	58	104,0	80
		TCP_STREAM	1	64	98,5	80
			2	64	111,8	80
		TCP_CRR	1	54	0	100
			2	53	0	100
	2 calls + 1 fax	UDP_STREAM	1	52	413,8	40
			2	45	407,5	40

Tabelle 4.14: Ergebnisse Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - TD)

Im nächsten Test wird die Klasse *data_class* von Tail Drop auf Weighted Random Early Detection umgestellt. Die Ergebnisse (Tabelle 4.15 und 4.16) stimmen prinzipiell mit denen des vorangegangenen Versuches überein. Die dort getroffenen Aussagen sind übertragbar. Auffällig sind lediglich die hohen ReceiveDelay Werte, die bei Einsatz des UDP_STREAM Makros bis zu 24 *ms* über dem unbelasteten Zustand liegen.

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro		
		UDP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_CRR $\frac{\text{Trans.}}{\text{s}}$
125	–	144,30	120,06	9,40
	1 call	71,32	56,55	3,87
	1 call + 1 fax	77,03	/	/
256	–	297,56	246,72	18,81
	1 call	199,46	165,32	14,57
	1 call + 1 fax	154,86	/	/
	2 calls	145,91	117,98	8,21
	2 calls + 1 fax	145,04	/	/

Tabelle 4.15: Ergebnisse Netperf - CBWFQ (2 Klassen: Voice - TD, Data - WRED)

Bei der Übertragung von TCP Paketen (TCP_STREAM und TCP_CRR) sollte durch WRED eine Entlastung des Voice-Calls eintreten. Es kann dagegen nur beim Einsatz des TCP_CRR Makros ein Rückgang der Transactionrate beobachtet werden. Sowohl die Netperf-Werte von TCP_STREAM als auch UDP_STREAM erhöhen sich und mindern so die Hörqualität. Das ist bei der Übertragung von UDP-Paketen unverständlich, da WRED dort die gleiche Wirkung wie Tail Drop besitzt. Das der Versuch mit TCP_STREAM nicht die erwünschte Wirkung zeigt, ist wahrscheinlich darauf zurückzuführen, daß zur Auslastung der Verbindung ein Benchmark-Utility verwendet worden ist. Netperf kann keinen realen Sender simulieren, da es die maximale Geschwindigkeit eines Links bestimmen soll und somit die Datenrate immer so hoch wie möglich hält.

Die Meßergebnisse in den Tabellen B.17 und B.18 weisen wieder Ausreißer auf. Es ist deutlich ein weiteres ReceiveDelay-LatePackets Fehlerpaar zu erkennen. Zudem ist es unwahrscheinlich, daß eine unbelastete Verbindung derart hohe Verluste (LostPackets = 60843 *ms*) aufweist, da zudem die Verständlichkeit mit 100 % auf das Gegenteil hin-

weist.

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	53	0,3	100
		UDP_STREAM	1	62	154,5	50
		TCP_STREAM	1	55	169,0	50
		TCP_CRR	1	53	0	100
	1 call + 1 fax	UDP_STREAM	1	55	224,8	50
256	1 call	–	1	52	0,3	100
		UDP_STREAM	1	53	0	100
		TCP_STREAM	1	54	0,3	100
		TCP_CRR	1	54	0,3	100
	1 call + 1 fax	UDP_STREAM	1	46	136,5	80
	2 calls	–	1	52	0,3	100
			2	52	0,5	100
		UDP_STREAM	1	76	160,0	80
			2	73	113,0	80
		TCP_STREAM	1	65	152,8	80
			2	65	97,3	80
		TCP_CRR	1	53	0,3	100
			2	53	0	100
	2 calls + 1 fax	UDP_STREAM	1	69	345,8	40
			2	64	324,8	50

Tabelle 4.16: Ergebnisse Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - WRED)

In einem weiteren Versuch mit CBWFQ wird für die Klasse *voice_class* Low Latency Queueing aktiviert. Die Klasse kann über $200 \frac{kbit}{s}$ Bandbreite verfügen. Die Ergebnisse (siehe Tabelle 4.17 und 4.18) zeigen eine stark verbesserte Übertragungsqualität für Voice. In jedem Test liegt die Verständlichkeit bei 100 % (bis auf 1 call + 1 fax, da die Bandbreite nicht ausreicht). Der einzige Schwachpunkt sind lange Verzögerungszeiten bei geringen Bitraten, obwohl LLQ für eine kurze Delay sorgen soll. Anscheinend gelingt dies erst bei entsprechend verfügbarer Leitungskapazität. Die angegebenen Werte der ReceiveDelay bewegen sich dennoch in einem nicht wahrnehmbaren Rahmen.

Die Meßwerte im Anhang offenbaren wiederum Fehler in der Callanalyse des Routers.

Clock Rate $10^3 \frac{bit}{s}$	Test	Netperf Makro		
		UDP_STREAM $10^3 \frac{bit}{s}$	TCP_STREAM $10^3 \frac{bit}{s}$	TCP_CRR $\frac{Trans.}{s}$
125	–	144,30	120,06	9,40
	1 call	47,26	38,83	3,50
	1 call + 1 fax	33,60	/	/
256	–	297,56	246,72	18,81
	1 call	199,47	165,29	13,10
	1 call + 1 fax	146,38	/	/
	2 calls	101,34	83,64	7,20
	2 calls + 1 fax	56,26	/	/

Tabelle 4.17: Ergebnisse Netperf - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED)

Clock Rate $10^3 \frac{bit}{s}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	53	0,3	100
		UDP_STREAM	1	84	1,3	100
		TCP_STREAM	1	116	1,0	100
		TCP_CRR	1	52	0,3	100
	1 call + 1 fax	UDP_STREAM	1	55	220,5	80
256	1 call	–	1	52	0,3	100
		UDP_STREAM	1	54	0,3	100
		TCP_STREAM	1	56	1,0	100
		TCP_CRR	1	53	0	100
	1 call + 1 fax	UDP_STREAM	1	53	0,3	100
	2 calls	–	1	52	0	100
			2	55	0,3	100
		UDP_STREAM	1	52	0	100
			2	53	0	100
		TCP_STREAM	1	59	9,8	100
			2	55	5,3	100
		TCP_CRR	1	52	0	100
			2	53	0	100
	2 calls + 1 fax	UDP_STREAM	1	53	0,3	100
			2	53	2,3	100

Tabelle 4.18: Ergebnisse Call-Analyse - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED)

4.4.1.5 Priority Queueing

Priority Queueing ermöglicht eine strikte Priorisierung der Sprachdaten gegenüber anderen Daten. Das wird durch die Ergebnisse in den Tabellen 4.19 und 4.20 bestätigt. Die Sprachqualität ist überall sehr gut. Auffällig sind hingegen wieder hohe Verzögerungszeiten bei einer Clock Rate von $125 \frac{kbit}{s}$. Das spricht dafür, daß die Implementierung des Priority Queueing⁵ in den Cisco Routern erst ab einer bestimmten Mindestbitrate zufriedenstellend arbeitet. Es sei angemerkt, daß mit dem verwendeten Codec G.711 die ReceiveDelay nicht hörbar ist. Codecs mit hoher Eigenverzögerung sind in der Hinsicht problematischer (mehr dazu im Kapitel 4.4.2.2).

In einem weiteren Versuch werden bei einer Datenrate von $125 \frac{kbit}{s}$ 2 Gespräche gleichzeitig geführt. Die Verbindung ist damit total ausgelastet. Ein Probe mit Netperf zeigt den Nachteil des Priority Queueing: Flows niedriger Priorität sind in Überlastsituationen nicht in der Lage, ihre Daten zu versenden. Netperf bricht den Test unter solchen Umständen ab, da keine Daten ausgetauscht werden können.

Clock Rate	Test	Netperf Makro		
		UDP_STREAM	TCP_STREAM	TCP_CRR
$10^3 \frac{bit}{s}$		$10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$
125	–	144,30	120,06	9,40
	1 call	47,01	38,67	3,50
	1 call + 1 fax	32,78	/	/
256	–	297,56	246,72	18,81
	1 call	199,69	165,36	12,84
	1 call + 1 fax	146,66	/	/
	2 calls	100,80	83,31	7,14
	2 calls + 1 fax	59,27	/	/

Tabelle 4.19: Ergebnisse Netperf - PQ

⁵Das gilt auch für Low Latency Queueing, da es sich um einen PQ-Algorithmus handelt.

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	53	0,3	100
		UDP_STREAM	1	70	1,3	100
		TCP_STREAM	1	112	0,8	100
		TCP_CRR	1	53	0,3	100
	1 call + 1 fax	UDP_STREAM	1	46	240,0	60
256	1 call	–	1	53	0	100
		UDP_STREAM	1	54	0,3	100
		TCP_STREAM	1	58	3,0	100
		TCP_CRR	1	52	0	100
	1 call + 1 fax	UDP_STREAM	1	53	0,3	100
	2 calls	–	1	52	0	100
			2	53	0,3	100
		UDP_STREAM	1	52	1,0	100
			2	54	0	100
		TCP_STREAM	1	60	15,5	100
			2	59	8,8	100
		TCP_CRR	1	53	0,3	100
			2	52	0	100
	2 calls + 1 fax	UDP_STREAM	1	52	0	100
			2	53	0,3	100

Tabelle 4.20: Ergebnisse Call-Analyse - PQ

Im nächsten Versuch wird die Queue mit der höchsten Prioritätsstufe (Sprache) mittels der Rate Limiting Funktion von Committed Access Rate auf $96 \frac{kbit}{s}$ bandbreitenlimitiert. Mit der jetzt zur Verfügung stehenden Bandbreite kann damit nur ein Call ohne Einschränkungen bedient werden. Erwartungsgemäß stimmen die Ergebnisse (Tabelle 4.21 und 4.22) für ein Telefongespräch mit den Meßwerten aus dem letzten Test überein. Werden mehrere Voice-Calls aufgebaut, sinkt die Sprachqualität dementsprechend. Es ist weiterhin ersichtlich, daß sich die Werte von Netperf für die Tests 1 call bzw. 2 calls weitgehend gleichen. Mehrere Gespräche teilen sich somit die Bandbreite eines einzelnen Calls. Da beiden Verbindungen nur $96 \frac{kbit}{s}$ zur Verfügung stehen, werden wegen der gegenseitigen Behinderung Qualitätseinbußen hervorgerufen.

Auch in diesem Versuch sind Fehler in der Callanalyse auffällig. Wird mit 2 aktiven Calls gearbeitet, gibt der Router teilweise für beide Gespräche stark unterschiedliche Werte für verloren gegangene Pakete an, obwohl der subjektive Höreindruck beider Gespräche gleich ist. Solche Meßwerte sind deswegen aus der Bewertung ausgeklammert bzw. im unbelasteten Fall gemittelt worden.

Clock Rate $10^3 \frac{bit}{s}$	Test	Netperf Makro		
		UDP_STREAM $10^3 \frac{bit}{s}$	TCP_STREAM $10^3 \frac{bit}{s}$	TCP_CRR $\frac{Trans.}{s}$
125	–	144,30	120,06	9,40
	1 call	46,99	38,57	3,53
	1 call + 1 fax	24,58	/	/
256	–	297,56	246,72	18,81
	1 call	199,13	164,90	12,97
	1 call + 1 fax	160,57	/	/
	2 calls	181,94	150,70	11,77
	2 calls + 1 fax	147,87	/	/

Tabelle 4.21: Ergebnisse Netperf - PQ + CAR

Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Test	Netperf Makro	Call	Receive-Delay ms	gewichteter Mittelwert ms	Verständlichkeit %
125	1 call	–	1	53	0	100
		UDP_STREAM	1	81	4,5	100
		TCP_STREAM	1	118	1,3	100
		TCP_CRR	1	54	0,3	100
	1 call + 1 fax	UDP_STREAM	1	64	246,3	70
256	1 call	–	1	52	0	100
		UDP_STREAM	1	52	0	100
		TCP_STREAM	1	53	3,0	100
		TCP_CRR	1	52	0	100
	1 call + 1 fax	UDP_STREAM	1	45	181,3	80
	2 calls	–	1	45	289,5	50
			2	57	289,0	50
		UDP_STREAM	1	49	210,3	50
			2	50	244,0	50
		TCP_STREAM	1	68	170,3	50
			2	62	222,5	50
		TCP_CRR	1	45	162,3	50
			2	48	193,3	50
	2 calls + 1 fax	UDP_STREAM	1	45	335,8	40
			2	56	296,3	40

Tabelle 4.22: Ergebnisse Call-Analyse - PQ + CAR

4.4.1.6 Compressed Real-Time Protocol

Für die Tests des Compressed Real-Time Protocol wird als Queueing Algorithmus PQ verwendet, da somit sichergestellt werden kann, daß nur eine unbedeutende Menge an Voice-Paketen verloren geht. Weiterhin ist eine Clock Rate von $256 \frac{kbit}{s}$ und als Codec G.711 verwendet worden. Die Bandbreite (siehe Tabelle 4.23) läßt sich aus der Differenz zwischen unbelasteter und mit 1 oder mehreren Gesprächen belasteter Verbindung bestimmen. Die Meßwerte entstammen der Tabelle B.22 und B.28.

Test	Netperf-Macro	Bandbreite		Reduzierung
		ohne CRTP $10^3 \frac{bit}{s}$ bzw. $\frac{Trans.}{s} \cdot *$	mit CRTP $10^3 \frac{bit}{s}$ bzw. $\frac{Trans.}{s} \cdot *$	
1 call	UDP_STREAM	97,87	80,05	22,26
	TCP_STREAM	81,36	66,36	22,60
	TCP_CRR*	5,97	5,03	18,69
2 calls	UDP_STREAM	196,76	160,67	22,46
	TCP_STREAM	163,41	133,29	22,60
	TCP_CRR*	11,67	8,97	30,10
1 fax	UDP_STREAM	22,54	18,12	24,39

Tabelle 4.23: Reduzierung der Bandbreite durch CRTP

Der Gewinn an Bandbreite ist abhängig von der Größe der Nutzlast. Die Router haben Pakete mit einer Payload von 160 *byte* erzeugt. Das laut Kapitel 3.3.6.1 zu erwartende Ergebnis von 21,2 % deckt sich mit den gemessenen Werten.

Wird die Größe der Payload und damit die Verzögerungszeit verringert, kann durch das Compressed Real-Time Protocol Leitungskapazität eingespart werden.

4.4.1.7 Zusammenfassung

In den Abbildungen 4.5 und 4.6 wird deutlich, daß beim Test 1 call @ $256 \frac{kbit}{s}$ alle Codecs (bis auf FIFO-Queueing) die selben guten Ergebnisse erzielen, da genügend Bandbreite zur Verfügung steht. Unterschiede zwischen den Codecs werden erst unter hoher Auslastung der Verbindung (z.B. 2 calls @ $256 \frac{kbit}{s}$) sichtbar. Allen Codecs gemein ist die Tatsache, daß $125 \frac{kbit}{s}$ für die Übertragung einer Telefonverbindung, 1 Faxes und des

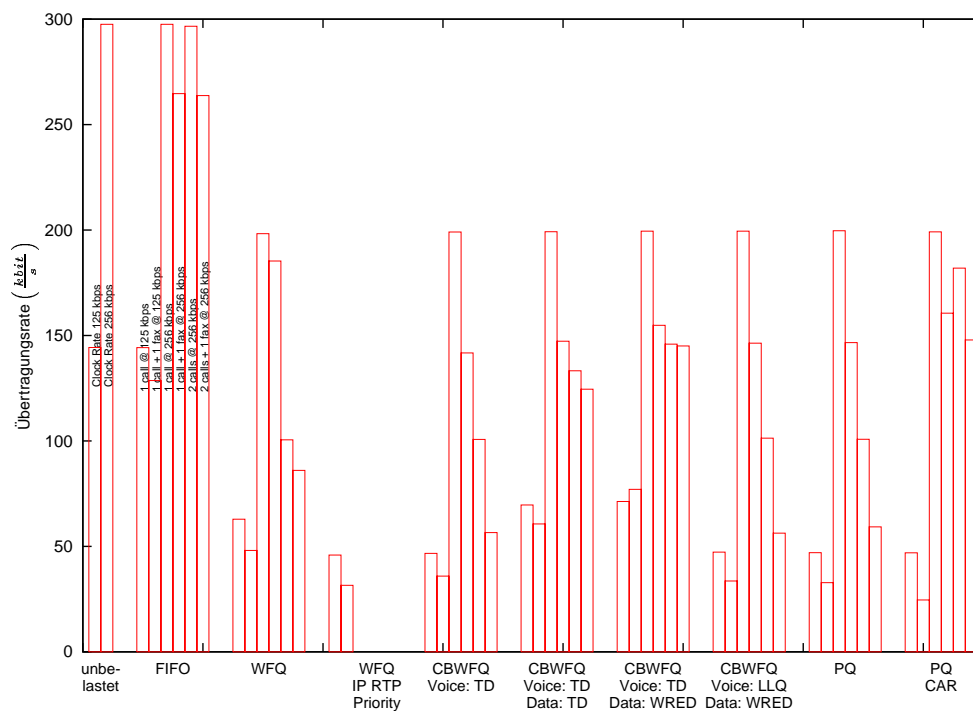


Abbildung 4.5: Übersicht Netperf: UDP_STREAM

UDP Datenstroms von Netperf nicht ausreichen. Damit kann belegt werden, daß die beschriebenen Verfahren zum Congestion Management und Congestion Avoidance keine fehlende Bandbreite ersetzen können. Der optimale Codec weist somit in Abbildung 4.6 und 4.7 bei allen Tests (bis auf 1 call + 1 fax @ 125 $\frac{kbit}{s}$) einen möglichst niedrigen gewichteten Mittelwert und eine Verständlichkeit von 100 % auf.

Die Abbildungen 4.8 und 4.9 veranschaulichen die Tests mit dem TCP_STREAM Makro. Die Hüllkurven entsprechen denen der UDP_STREAM Versuche. Lediglich der Wertebereich hat sich geändert. Die Werte der Callanalyse weisen geringfügig bessere Ergebnisse auf. Damit wird gezeigt, daß TCP-Datenströme in Überlastsituation besser beherrschbar sind.

Die Tests mit dem TCP_CRR (siehe Abbildung 4.10 und 4.10) offenbaren keine starken Unterschiede zwischen den Codecs (inclusive FIFO-Queueing). Die Sprachqualität wird nicht beeinflusst. TCP_CRR (und damit auf dem HTTP-Protokoll basierender Verkehr) lastet die Leitung nicht genügend aus, um die Verständlichkeit zu reduzieren. Der Ausreißer im Versuch PQ + CAR ist auf die künstlich beschränkte Bandbreite zurückzuführen.

Welcher Algorithmus ist für die gleichzeitige Übertragung von Sprache und Daten brauchbar? Im Folgenden wird eine Einschätzung zu jeder getesteten Konfiguration gegeben.

FIFO Queueing ist als Best-Effort Service nicht für VoIP gedacht und geeignet. Mit Weighted Fair Queueing erhält ein Sprachsignal genügend Bandbreite. Dafür darf der Voice-Flow nicht größer als die Daten-Flows werden. WFQ setzt entsprechend Leitungskapazität voraus, braucht allerdings nicht konfiguriert werden. Im Zusammenspiel mit IP RTP Priority ist eine strikte Priorisierung der VoIP Pakete möglich. Damit wird bei niedrigen Bitraten eine ansprechende Übertragungsqualität erzielt. IP RTP Priority ist beschränkt auf die RTP Ports (16384 - 32767). Zudem können VoIP Flows die garantierte Bandbreite bei vorhandenen Reserven des Links nicht überschreiten.

Class-Based Weighted Fair Queueing bietet die größte Anzahl an Optionen zur Konfiguration. Mit einer definierten Klasse ist die Sprachqualität sehr gut. Dafür ist die Lösung absolut unflexibel. Mit dem Anlegen weiterer Klassen kann die Flexibilität gesteigert werden. Sprache wird dann jedoch sowohl mit Tail Drop als auch WRED nicht mehr priorisiert. In Zusammenarbeit mit Low Latency Queueing erreicht CBWFQ das Optimum an Voicequalität und Flexibilität. Es lassen sich mehrere Verkehrsklassen unterscheiden, LLQ ist nicht auf RTP Ports beschränkt und die zugewiesene Bandbreite kann bei Nichtauslastung der Leitung überschritten werden.

PQ erreicht durch seine absolute Priorisierung eine sehr gute Sprachqualität. Ein wesentlicher Nachteil entsteht durch die völlige Auslöschung von Flows niederer Priorität bei voller Auslastung der Verbindung. Mit Hilfe von CAR können die Prioritätsklassen auf eine definierte Bandbreite limitiert werden. Committed Access Rate weist indes die selbe Beschränkung wie IP RTP Priority auf, indem Flows die angegebene Bandbreite bei vorhandener Leitungskapazität nicht überschreiten können.

Das Compressed Real-Time Protocol hilft den Overhead zu verringern, der bei Voice-Paketen mit geringen Nutzlasten (für kurze Verzögerungszeiten) entsteht. CRTP sollte daher immer aktiviert sein.

Diese Arbeit kann keine optimale Lösung bieten. Der Einsatz bestimmter Queueing-Algorithmen ist von der Umgebung, deren Anforderungen und den Parametern abhängig.

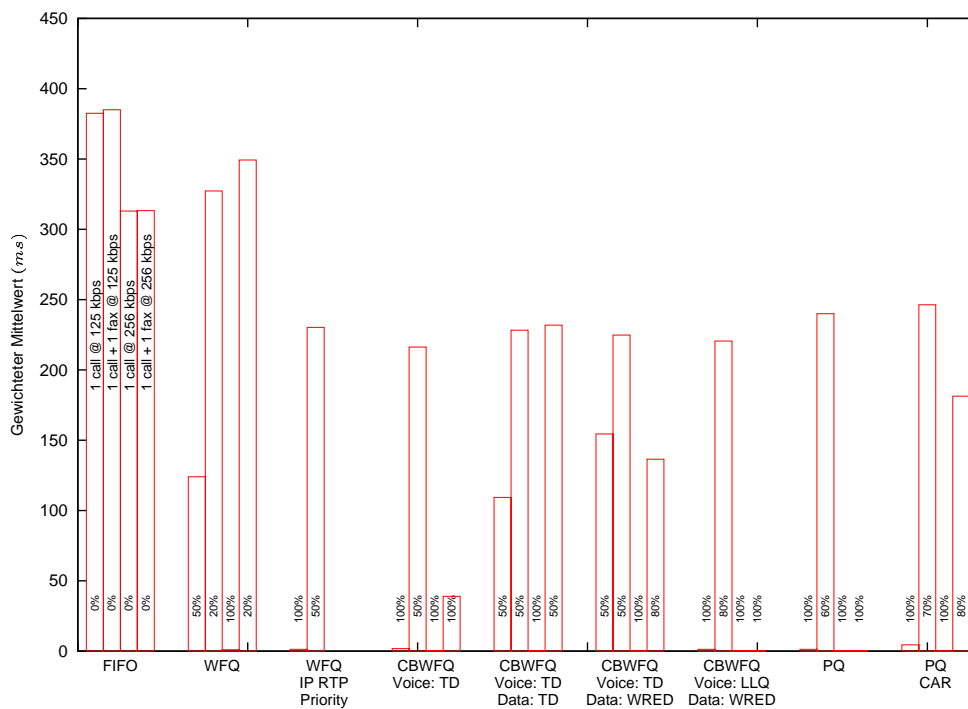


Abbildung 4.6: Übersicht Callanalyse: UDP_STREAM Teil 1

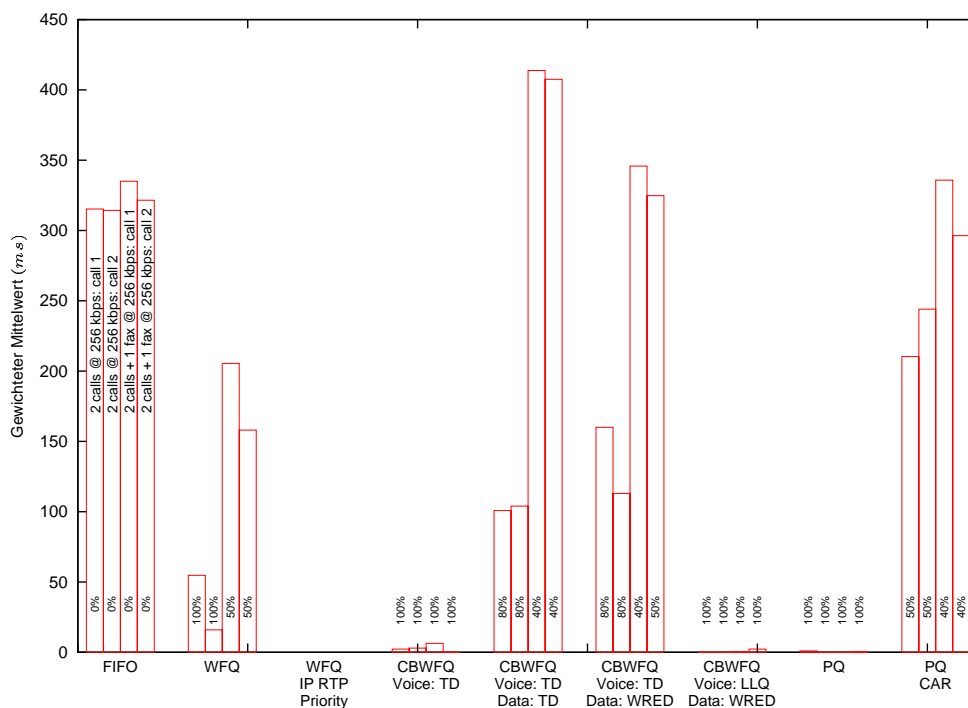


Abbildung 4.7: Übersicht Callanalyse: UDP_STREAM Teil 2

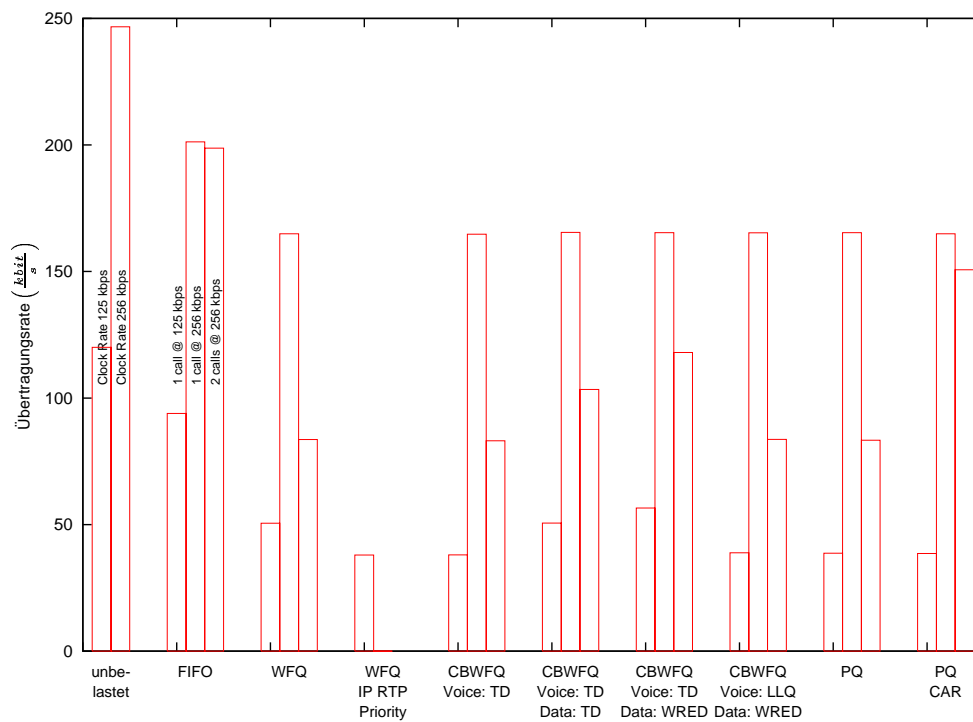


Abbildung 4.8: Übersicht Netperf: TCP_STREAM

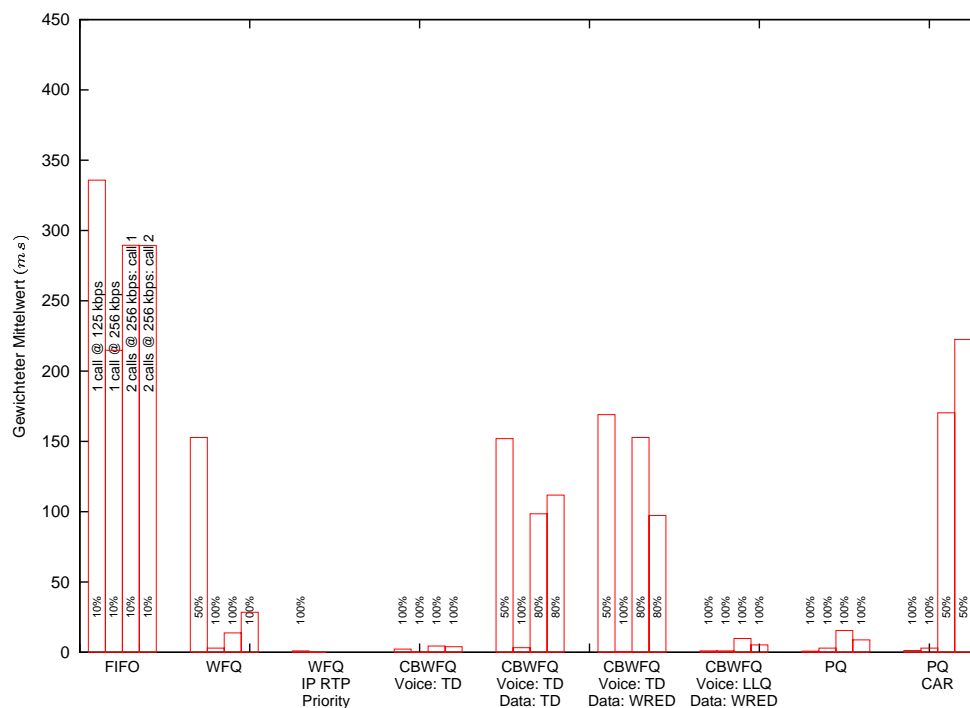


Abbildung 4.9: Übersicht Callanalyse: TCP_STREAM

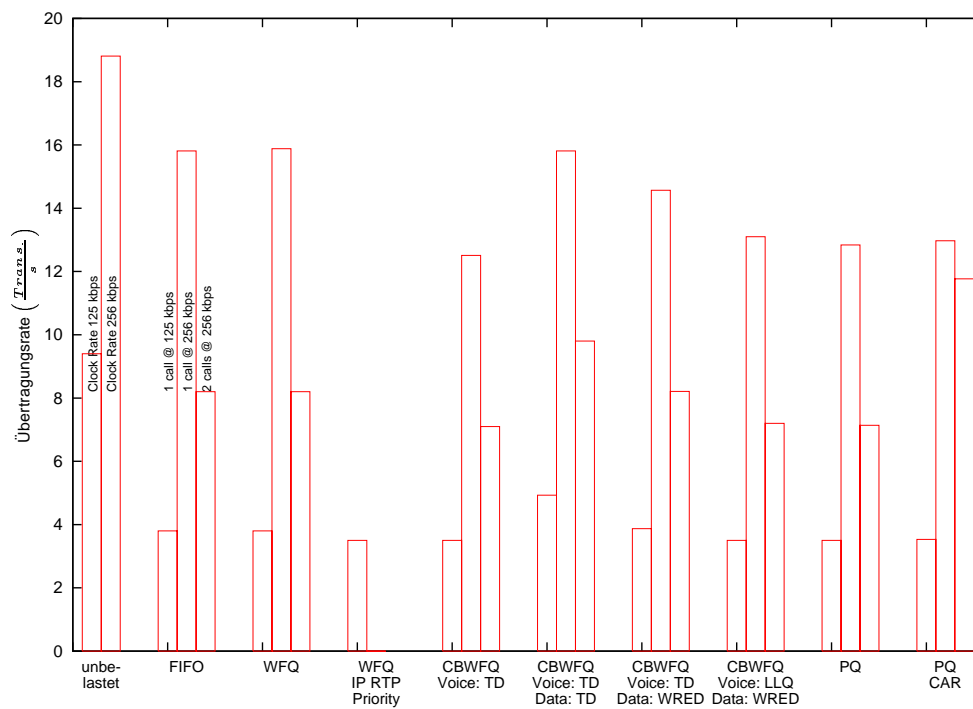


Abbildung 4.10: Übersicht Netperf: TCP_CRR

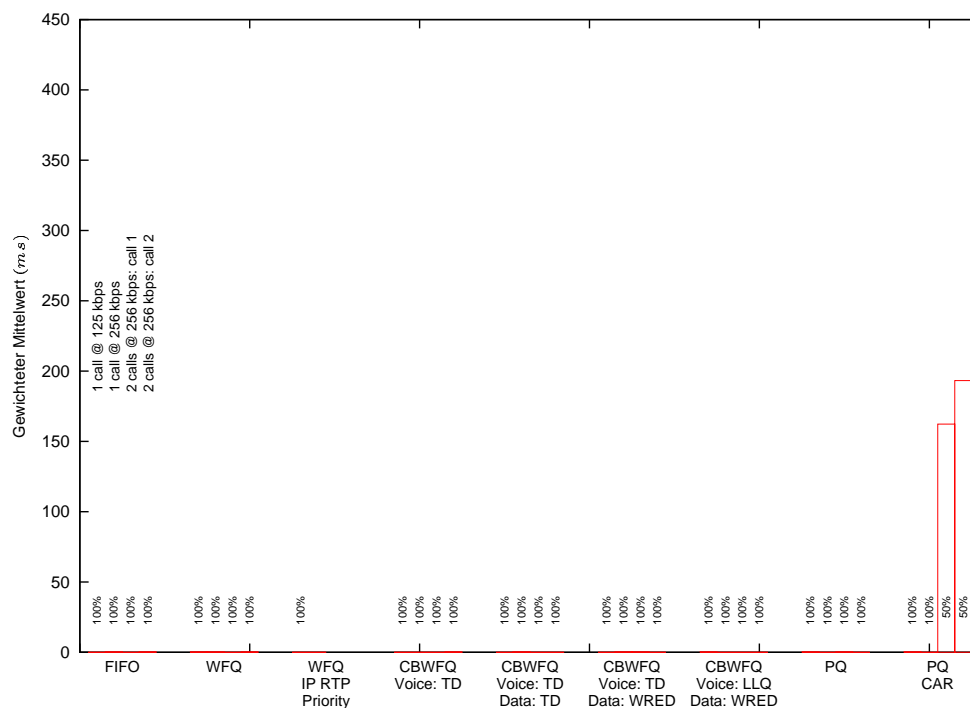


Abbildung 4.11: Übersicht Callanalyse: TCP_CRR

4.4.2 Vergleich der Codecs

Die qualitative Beurteilung der Sprachqualität stützt sich auf veröffentlichte, unabhängige Testergebnisse (siehe Kapitel 3.4.6). Die Arbeit beschränkt sich deshalb auf rekonstruierbare Meßergebnisse. Für die Bandbreiten- und Delay-Time Messung wird Priority-Queueing und CRTP bei einer Bitrate von $256 \frac{kbit}{s}$ verwendet. Priority Queueing priorisiert eine Queue (in diesem Fall alle Voice-Pakete) strikt gegenüber allen anderen. Somit können Beeinträchtigungen anderer Flows in Bezug auf die Bandbreite ausgeschlossen werden. CRTP findet Verwendung, da es erwiesenermaßen den Overhead verringert und somit in jeder Konfiguration eingeschaltet sein sollte. Die Meßwerte für die ReceiveDelay werden nur für den Test 1 call bestimmt, weil nur eine Aussage über die vom Codec hervorgerufene Verzögerung getroffen werden soll.

Weiterhin wird in den nächsten Abschnitten die (subjektive) Einschätzung des Autors zur Sprachqualität der Codecs wiedergegeben.

4.4.2.1 G.711 A law

Der Codec G.711 benötigt durch die PCM-Technik verhältnismäßig viel Bandbreite (siehe Tabelle 4.24). Das Einkapseln der Sprachdaten in IP-Pakete bedingt Overhead. Deswegen Anteil macht bei aktiviertem CRTP bis zu 25 % (UDP_STREAM) aus.

Da keine Datenkompression stattfindet, fallen die benötigte Rechenleistung und -zeit für die Umkodierung (siehe Abbildung 3.9) und die damit verbundene Verzögerung gering aus. Zusätzlich ist die Sprachqualität (Tabelle 4.25) sehr gut.

Test	UDP_STREAM $10^3 \frac{bit}{s}$	TCP_STREAM $10^3 \frac{bit}{s}$	TCP_CRR $\frac{Trans.}{s}$
1 call	80,05	66,36	5,03
2 calls	160,67	133,29	8,97
1 fax	18,12	/	/

Tabelle 4.24: Bandbreite - Codec G.711 A law

In einem weiteren Versuch ist der zeitliche Verlauf der übertragenen Datenmengen einer Faxübertragung durch wiederholten Aufruf von Netperf beobachtet worden. Dafür wurde die Laufzeit von Netperf von 30 s auf 10 s verringert. Hierbei ist zu erkennen,

daß während der Initialisierungsphase eine deutlich höhere Bitrate⁶ benötigt wird, als beim eigentlichen Versand. Während der Übertragung eines Dokumentes korrespondiert die Datenrate mit der ausgehandelten Geschwindigkeit. Die maximale Übertragungsgeschwindigkeit ist durch die beiden Endgeräte (siehe Abschnitt 4.1) auf 9,6 *kbaud* beschränkt. Somit ist der Versand eines Faksimiles nicht so kritisch, wie die Übermittlung eines G.711 Calls mit, da die Datenrate viel geringer ist.

Netperf-Macro	ReceiveDelay ms	Verständlichkeit %
UDP_STREAM	53	100
TCP_STREAM	56	100
TCP_CRR	53	100

Tabelle 4.25: Ergebnisse Call-Analyse - Codec G.711 A law

4.4.2.2 G.723.1

Die beiden behandelten Codecs des G.723.1 Standards weisen wegen des hohen Rechenaufwandes bei der Umkodierung eine deutlich höhere Verzögerung auf, als dies beim Codec G.711 der Fall ist. Zusätzliche Wartezeit entsteht bei der Erzeugung der IP-Pakete. Es nimmt einige Zeit in Anspruch, bis die Payload (im Versuch 160 *byte*) eines Voice-Paketes mit einer Datenrate von 5,3 $\frac{kbit}{s}$ bzw. 6,3 $\frac{kbit}{s}$ aufgefüllt ist. Dagegen ist mit einer Reduzierung der Payload zur Verringerung der ReceiveDelay eine Erhöhung des Overheads verbunden.

Test	UDP_STREAM $10^3 \frac{bit}{s}$	TCP_STREAM $10^3 \frac{bit}{s}$	TCP_CRR $\frac{Trans.}{s}$
1 call	10,98	9,07	0,30
2 calls	22,19	18,07	0,90
1 fax	9,13	/	/

Tabelle 4.26: Bandbreite - Codec G.723.1 5,3k

⁶Als Orientierung kann der Platzbedarf des eingestellten Voice-Codecs angesehen werden.

Netperf-Macro	ReceiveDelay	Verständlichkeit
	ms	%
UDP_STREAM	130	70
TCP_STREAM	130	70
TCP_CRR	130	70

Tabelle 4.27: Ergebnisse Call-Analyse - Codec G.723.1 5,3k

Die Unterschiede in der benötigten Bandbreite (Tabelle 4.26 und 4.28) zwischen $5,3 \frac{kbit}{s}$ und $6,3 \frac{kbit}{s}$ Codec sind zu vernachlässigen. Der Overhead beträgt bedingt durch die geringe Datenrate bis zu 107 % ($5,3 \frac{kbit}{s}$) bzw. 74 % ($6,3 \frac{kbit}{s}$). Die niedrige Bitrate begrenzt die maximale Faxgeschwindigkeit auf $4,8 kbaud$. Das schlägt sich in längeren Übertragungszeiten nieder.

Test	UDP_STREAM $10^3 \frac{bit}{s}$	TCP_STREAM $10^3 \frac{bit}{s}$	TCP_CRR $\frac{Trans.}{s}$
1 call	10,98	9,11	0,30
2 calls	21,92	18,28	0,90
1 fax	9,11	/	/

Tabelle 4.28: Bandbreite - Codec G.723.1 6,3k

Netperf-Macro	ReceiveDelay	Verständlichkeit
	ms	%
UDP_STREAM	131	70
TCP_STREAM	130	70
TCP_CRR	129	70

Tabelle 4.29: Ergebnisse Call-Analyse - Codec G.723.1 6,3k

Die Verzögerung des Sprachsignals liegt mit $130 ms$ im hörbaren Bereich. Aufgrund dessen wird der Codec in der Hörqualität abgewertet. Da das Referenzsignal während der Versuche in Hörreichweite liegt, kann keine Aussage darüber getroffen werden, wie störend diese Verzögerung in einer realen Umgebung ist.

4.4.2.3 G.726

G.726 liegt mit der benötigten Bandbreite zwischen G.711 und den hybriden Codecs (Tabelle 4.30). Der Anteil des Overhead an der Datenrate beträgt maximal 51 %. Im Versuch sind mit G.726 Faxübertragungen bis 9,6 *kbaud* möglich gewesen.

Test	UDP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_CRR $\frac{\text{Trans.}}{\text{s}}$
1 call	24,10	20,00	1,00
2 calls	48,41	39,88	2,20
1 fax	18,11	/	/

Tabelle 4.30: Bandbreite - Codec G.726

G.726 weist ebenfalls keine wahrnehmbare Verzögerung auf. Dafür ist die Wiedergabe der Sprache verzerrt. Das Referenzsignal klingt durch den Codec blechern und verrauscht. Deshalb erfolgt eine Abwertung der Sprachqualität (Tabelle 4.31).

Netperf-Macro	ReceiveDelay ms	Verständlichkeit %
UDP_STREAM	52	80
TCP_STREAM	54	80
TCP_CRR	53	80

Tabelle 4.31: Ergebnisse Call-Analyse - Codec G.726

4.4.2.4 G.728

Der Codec G.728 zeigt keinerlei Auffälligkeiten. Die Eckwerte stimmen mit denen des Standards G.726 überein. Dagegen bietet G.728 eine bessere Sprachqualität. Die maximal erreichbare Faxgeschwindigkeit liegt bei 9,6 *kbaud*. Der Codec produziert einen Overhead von bis 49 %.

Test	UDP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_CRR $\frac{\text{Trans.}}{\text{s}}$
1 call	23,80	19,59	1,00
2 calls	47,57	39,30	2,23
1 fax	18,10	/	/

Tabelle 4.32: Bandbreite - Codec G.728

Netperf-Macro	ReceiveDelay ms	Verständlichkeit %
UDP_STREAM	53	100
TCP_STREAM	53	100
TCP_CRR	52	100

Tabelle 4.33: Ergebnisse Call-Analyse - Codec G.728

4.4.2.5 G.729

Mittels G.729 läßt sich ein solch geringer Bandbreitenbedarf wie mit G.723.1 realisieren. Allerdings ist die Datenrate nicht so gering, daß die Verzögerung durch hohe Rechenzeiten und Wartezeiten beim Auffüllen der Payload stark ansteigt. Die Sprachqualität ist daher der Referenz G.711 gleichwertig. Der Codec sorgt dagegen mit seiner geringen Sprachdatenrate ($8 \frac{\text{kbit}}{\text{s}}$) für einen Overhead von 81 %. Als Einschränkung kann die maximale Faxgeschwindigkeit von $7,2 \text{ kbaud}$ angesehen werden, da sich die Übertragungszeiten um ein Drittel⁷ (siehe Tabelle B.41) verlängern.

Test	UDP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_STREAM $10^3 \frac{\text{bit}}{\text{s}}$	TCP_CRR $\frac{\text{Trans.}}{\text{s}}$
1 call	14,51	11,85	0,50
2 calls	28,75	23,59	1,50
1 fax	13,43	/	/

Tabelle 4.34: Bandbreite - Codec G.729

⁷Reine Übertragungsleistung. Die Initialisierung muß gesondert berücksichtigt werden.

Netperf-Macro	ReceiveDelay	Verständlichkeit
	ms	%
UDP_STREAM	54	100
TCP_STREAM	55	100
TCP_CRR	52	100

Tabelle 4.35: Ergebnisse Call-Analyse - Codec G.729

4.4.2.6 Zusammenfassung

Zur Veranschaulichung ist in der Abbildung 4.12 der Bandbreitenbedarf der einzelnen Codecs gegenübergestellt. In der Tabelle 4.36 sind die Codec-Kennndaten zusammengefaßt.

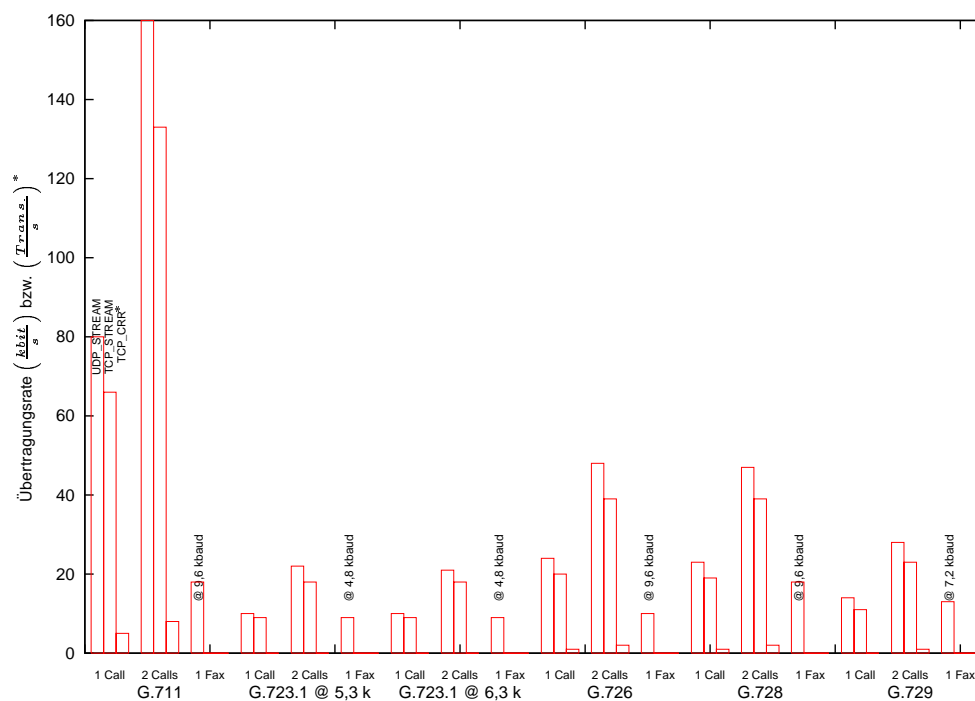


Abbildung 4.12: Übersicht Codecs

Aus der Bewertung der einzelnen Codecs lassen sich die folgenden Schlußfolgerungen zu deren Einsatz in einer VoIP-Umgebung ziehen: G.711 besitzt zwar eine gute Sprachqualität und eine geringe Verzögerung, erlaubt aber keine Voice-Kompression.

Codec	max. Overhead %	Verständlichkeit %	Fax-Datenrate <i>kbaud</i>
G.711 A law	25	100	9,6
G.723.1 5,3k	107	70	4,8
G.723.1 6,3k	74	70	4,8
G.726	51	80	9,6
G.728	49	100	9,6
G.729	81	100	7,2

Tabelle 4.36: Zusammenfassung der Codec-Ergebnisse

Damit geht ein wichtiger Vorteil der VoIP-Technik (Kapitel 2) verloren, da zusätzliche Leitungskapazitäten für die IP-Strecke nötig sind. Vom Einsatz des G.711 Codecs ist daher abzuraten.

Mit G.723.1 werden die niedrigsten Bitraten für Voice-Calls erzielt. Demgegenüber stehen die höchsten Verzögerungs- und Overhead-Werte. Die Übertragungszeiten von Faxen verdoppeln sich (9,6 *kbaud* zu 4,8 *kbaud*). G.723.1 ist daher nur bei ausreichender Rechenkapazität für die Kodierung und Verringerung der Payload-Size (bei weiter steigendem Overhead) zu empfehlen. Wird die Größe der Nutzlast verändert, muß überprüft werden, ob der Overhead nicht die Einsparungen durch die Kompression (gegenüber anderen Codecs wie G.729) wieder zunichte macht.

Der Einsatz von G.726 ist wegen der verminderten Sprachqualität nicht zu empfehlen. Von den Varianten mit 24, 32 oder 40 $\frac{kbit}{s}$ ist wegen des steigenden Bandbreitenbedarfs abzuraten.

Die Verwendung von G.728 ist in den Fällen zu befürworten, in denen es neben der Sprachkompression und guter Sprachqualität auf eine hohe Faxgeschwindigkeit ankommt.

Der Codec G.729 stellt nach Meinung des Autors in den meisten Fällen den besten Kompromiß aus Sprachqualität, Verzögerung und dafür benötigter Bandbreite dar. G.729 ist ebenfalls der Standard-Codec der getesteten Router. Innerhalb der Datenrate eines G.711 Calls lassen sich mit G.729 bis zu 5 Gespräche übermitteln.

Den „besten“ Codec gibt es nicht. Die Wahl eines Codecs kann nur spezifisch für die zu betrachtende VoIP-Umgebung erfolgen. Sie ist ein Kompromiß mit einer mög-

lichst geringen Bitrate auf der einen und einer hohen Verzögerung sowie Overhead und verminderten Sprachqualität und Faxgeschwindigkeit auf der anderen Seite.

Kapitel 5

Zusammenfassung

Eine leistungsstarke Kommunikationsinfrastruktur ist heutzutage unabdingbar. Mit Voice over IP können leistungsfähige und kostengünstige Kommunikationssysteme eingerichtet werden.

Eine Analyse der Unternehmenskommunikation und der zur Verfügung stehenden Infrastruktur ist für eine Wirtschaftlichkeitsrechnung der Intranet-Telefonie in Anbetracht stetig sinkender Preise für Telefonverbindungen unbedingt zu empfehlen.

Paketvermittelnde Datennetze sind von sich aus nicht in der Lage, Daten und Telefongespräche in ansprechender Qualität gleichzeitig zu übermitteln. Der Grund dafür ist in den unterschiedlichen Übertragungscharakteristika zu suchen. Während die Datenübermittlung im Allgemeinen asynchron abläuft, ist die Übertragung von Audiosignalen als isochroner oder zeitkritischer Prozeß einzuordnen. Die Qualität der Sprachübermittlung nimmt rapide bei Paketverlust sowie mit steigender Verzögerung des Signals und Variation der Verzögerung (Jitter) ab. Unter Berücksichtigung dieser Gegebenheiten ist bei paralleler Übertragung von synchronen und asynchronen Daten eine Priorisierung der synchronen Daten (Sprache) gefordert.

Mittels Quality of Service (QoS) kann eine bestimmte Dienstleistungsgüte zwischen Sender und Empfänger garantiert werden. Das in den Netzwerken vorherrschende Routed-Protokoll IPv4 unterstützt von sich aus kein QoS, sondern bietet einen reinen Datagrammdienst. Quality of Service muß daher durch höhere Protokollschichten bereitgestellt werden.

Im Rahmen dieser Arbeit ist eine VoIP Umgebung auf ihre Eignung zur parallelen Übertragung von Sprache und Daten untersucht worden. Die behandelten QoS-Features lassen sich in folgende Bereiche gliedern: Classification, Congestion Management, Congestion Avoidance, Policing and Shaping, Signalling und Link Efficiency Mechanisms. Das Hauptaugenmerk bei der Bewertung der Quality of Service Algorithmen liegt bei

der Einhaltung einer Mindestbandbreite und der strikten Priorisierung von Sprachdaten. Die minimale oder garantierte Bandbreite gibt an, wieviele Voice-Calls gleichzeitig ohne Paketverlust übertragen werden können. Die strikte Priorisierung ermöglicht geringe Latenzzeiten durch eine bevorzugte Übertragung synchroner Daten bei Überlastung der Leitung.

Die Versuchsreihen sind entsprechend den Standards der IETF durchgeführt worden. Die Tests mit einem synthetischen Benchmark weisen Einschränkungen hinsichtlich der Relevanz in realen Umgebungen auf, da ein Einzeltest lediglich ein Teilbereich des wirklichen Datenaufkommens abdeckt. Sollen aussagekräftige Ergebnisse für Real-World Szenarien erzielt werden, muß die Verbindung der Router durch echtes Verkehrsaufkommen belastet werden. Solch ein Test ist dann allerdings nicht mehr deterministisch und reproduzierbar, sondern spiegelt nur die Resultate für genau jene Umgebung wider. Es wird empfohlen, zur Bestimmung der optimalen Parameter der eigenen VoIP Umgebung (zumindest bei der Verwendung der Cisco Router C2610 und C3620) auf externe Meßgeräte zurückzugreifen, da die Callanalyse der Router teilweise falsche Ergebnisse ausgibt (siehe Kapitel 4.4.1).

Modemverbindungen sind über die untersuchte VoIP-Strecke grundsätzlich nicht möglich. Angesichts der sinkenden Bedeutung der analogen Übertragungstechnik ist solch ein Mangel zu verschmerzen. Der Versand von Faksimiles ist dagegen durchführbar. Ein Fax wird als solches vom Router erkannt und automatisch gegenüber dem asynchronen Datenverkehr priorisiert.

Die Eignung der QoS Algorithmen zur gleichzeitigen Übertragung von synchronen und asynchronen Daten wird im Folgenden wiedergegeben. FIFO-Queueing bietet einen reinen Best-Effort Service, d.h. es wird keinerlei Unterscheidung zwischen den verschiedenen Verkehrstypen gemacht. Der Algorithmus ist deshalb für VoIP Anwendungen nicht geeignet. Er wird vorzugsweise in gering belasteten High-Speed Netzen wegen seines geringen Rechenaufwandes eingesetzt.

Weighted Fair Queueing unterscheidet Flows anhand ihrer Gewichte (Verkehrsaufkommen). Verkehr mit einem geringen Gewicht wird gegenüber Traffic mit hohen Bandbreitenbedarf bevorzugt übertragen. WFQ ist schnell und einfach zu implementieren, da keine Konfiguration vonnöten ist. Somit ist jedoch keine direkte Kontrolle über die einzelnen Flows vorhanden. Entsprechende Leitungskapazität vorausgesetzt, bietet WFQ eine ausreichend hohe Qualität bei der Sprachübermittlung. Eine strikte Priorisierung

der VoIP Pakete ist nur im Zusammenspiel mit IP RTP Priority möglich. Damit wird auch bei niedrigen Bitraten eine ansprechende Übertragungsqualität erzielt. IP RTP Priority ist beschränkt auf die RTP Ports (16384 - 32767). Zudem können VoIP Flows die garantierte Bandbreite bei vorhandenen Reserven des Links nicht überschreiten.

Mit Class-Based Weighted Fair Queueing kann das Datenaufkommen auf bis zu 64 Klassen aufgeteilt werden, deren Eigenschaften gezielt an den jeweiligen Verkehrstyp angepaßt werden können. CBWFQ bietet die größte Anzahl an Optionen zur Konfiguration. Eine strikte Priorisierung ist mittels IP RTP Priority und Low Latency Queueing möglich. LLQ ist nicht auf RTP Ports beschränkt und die zugewiesene Bandbreite kann bei Nichtauslastung der Leitung überschritten werden. Ein Schwachpunkt sind lange Verzögerungszeiten bei geringen Bitraten, obwohl LLQ für eine kurze Latenz sorgen soll. Dies gelingt jedoch erst bei entsprechend verfügbarer Leitungskapazität. Im Kombination mit Weighted Random Early Detection kann die Auslastung des Links bei TCP Verbindungen gesteuert werden.

Priority Queueing unterstützt die Einteilung des Datenaufkommens in 4 Prioritätsklassen. Eine Klasse höherer Priorität hat bei der Übertragung absoluten Vorrang gegenüber einer Klasse mit einer geringeren Priorität. Somit können schnelle Antwortzeiten und eine bestimmte Bandbreite für einzelne Verkehrstypen garantiert werden. PQ wurde für den Einsatz in überlasteten langsamen Verbindungen konzipiert. Der Queueing-Algorithmus ist nur bedingt für schnelle Interfaces geeignet, da er einen zusätzlichen Overhead erzeugt. Ein wesentlicher Nachteil der absoluten Priorisierung entsteht durch die totale Auslöschung von Flows niedriger Priorität bei voller Auslastung der Verbindung mit Verkehr einer höheren Prioritätsstufe. Mit Hilfe von Committed Access Rate können die Prioritätsklassen auf eine definierte Bandbreite limitiert werden. CAR weist wiederum die selbe Beschränkung wie IP RTP Priority auf, indem Flows die angegebene Bandbreite bei vorhandener Leitungskapazität nicht überschreiten können. PQ erzeugt ebenso wie CBWFQ/LLQ bei sehr geringen Bitraten hohe Latenzzeiten. Das spricht dafür, daß die Implementierung des Priority Queueing in den getesteten Routern erst ab einer bestimmten Mindestbitrate zufriedenstellend arbeitet.

Custom Queueing kann den Verkehr in bis zu 16 Klassen aufteilen, die zirkular abgearbeitet werden. Aufgrund des Arbeitsprinzips entsteht eine hohe Variation der Verzögerungszeiten. Somit ist CQ für die Übertragung von VoIP Paketen nicht geeignet.

Weighted Random Early Detection nutzt die Fähigkeit eines TCP Senders, auf Zell-

verlust mit Herabsetzung der Bitrate zu reagieren. Dazu verwirft WRED ab einer bestimmten Füllgrad des Wartepuffers zufällig Pakete. Somit läßt sich bei TCP Verbindungen die Datenrate der Auslastung der Leitung anpassen. WRED sollte nicht auf einem Interface aktiviert werden, auf dem UDP Pakete (z.B. VoIP) übertragen werden. UDP erkennt den Zellverlust nicht und reagiert auch nicht mit Verringerung der Senderate. Weighted Random Early Detection kann nicht mit WFQ, PQ und CQ kombiniert werden.

Das Compressed Real-Time Protocol verringert durch eine Header-Kompression den Overhead von VoIP Paketen und hilft somit Bandbreite auf einem Link einzusparen. CRTP sollte in jeder Konfiguration aktiviert werden.

Während die Simulation von UDP- und TCP-Strömen die Verbindung je nach Queueing-Algorithmus vollständig blockieren konnte, ist es bei der Simulation von „normalem“ Webverkehr (HTTP Request/Response) nicht möglich gewesen, die Leitung auszulasten. Derartiger Traffic beeinträchtigt die Voicefunktionalität dementsprechend nur unwesentlich. Die Qualität der Sprachübertragung ist bei geringer und mittlerer Belastung bei allen getesteten Queueing-Mechanismen (außer FIFO-Queueing) annähernd gleich gut. Unterschiede werden nur unter hoher Auslastung der Verbindung sichtbar. Die Grenzen der QoS-Algorithmen konnten erst in einer atypischen Situation (Bitrate $125 \frac{kbit}{s}$, deaktiviertes VAD, Belastung durch $64 \frac{kbit}{s}$ Codec, Fax und Benchmark) aufgezeigt werden. Damit kann zudem belegt werden, daß die beschriebenen Verfahren zum Congestion Management und Congestion Avoidance keine fehlende Bandbreite ersetzen können.

Aus den Eigenschaften der Sprach-Codecs kann deren Eignung für den Einsatz in einer VoIP Umgebung abgeleitet werden. G.711 besitzt eine gute Sprachqualität und eine geringe Verzögerung, erlaubt aber keine Voice-Kompression. Da damit ein wichtiger Vorteil von VoIP verloren geht, ist vom Einsatz des G.711 Codec abzuraten. Mit G.723.1 werden die niedrigsten Bitraten für Voice-Calls erzielt. Demgegenüber stehen die höchsten gemessenen Verzögerungswerte. Die Übertragungszeiten von Faxen verdoppeln sich. G.723.1 ist daher nur bedingt zu empfehlen. Der Einsatz von G.726 ($16 \frac{kbit}{s}$) ist wegen der verminderten Sprachqualität nicht zu empfehlen. Von den Varianten mit 24, 32 oder $40 \frac{kbit}{s}$ ist wegen des steigenden Bandbreitenbedarfs abzuraten. Die Verwendung von G.728 ist in den Fällen zu befürworten, in denen es neben der Sprachkompression und guter Sprachqualität auf eine hohe Faxgeschwindigkeit ankommt. Der

Codec G.729 stellt nach Meinung des Autors in den meisten Fällen den besten Kompromiß aus Sprachqualität, Verzögerung und dafür benötigter Bandbreite dar. G.729 ist ebenfalls der Standard-Codec der getesteten Router. Innerhalb der Datenrate eines G.711 Calls lassen sich mit G.729 bis zu 5 Gespräche übermitteln. Bei der Planung einer VoIP Installation ist zu berücksichtigen, daß die Bruttobitrate eines Codecs durch Overhead teilweise erheblich höher ausfällt als die angegebene Nettobitrate (siehe Kapitel 4.4.2).

Diese Arbeit kann keine optimale Lösung oder vorgefertigte Konfiguration bieten. Die Benutzung bestimmter Queueing-Algorithmen ist von der Umgebung, deren Anforderungen und den Parametern abhängig. Ebenso gibt es nicht den „besten“ Codec. Die Wahl eines Codecs kann nur spezifisch für die zu betrachtende VoIP-Umgebung erfolgen. Sie ist ein Kompromiß mit einer möglichst geringen Bitrate auf der einen und einer hohen Verzögerung sowie Overhead und verminderten Sprachqualität und Faxgeschwindigkeit auf der anderen Seite.

Die nachfolgend beschriebenen Empfehlungen sind als Orientierung für das jeweilige Einsatzgebiet anzusehen. Die Anforderungen an die VoIP Umgebung sind entsprechend zu berücksichtigen.

1. Bei Verbindungen mit ausreichend hoher Bandbreite bzw. geringer Belastung sowie der Übertragung weniger Verkehrstypen bietet sich WFQ durch seinen geringen Administrationsaufwand an. Eine strikte Priorisierung durch IP RTP Priority ermöglicht bei höherer Belastung eine gute Sprachqualität. Die garantierte Bandbreite läßt sich jedoch bei vorhandenen Reserven des Links nicht überschreiten.
2. Ist ein Link mit ausreichend hoher Bandbreite bzw. geringer Belastung vorhanden und eine genaue Kontrolle über verschiedene Verkehrstypen erwünscht, sollte CBWFQ als Queueing-Algorithmus dienen. Mit LLQ ist eine strikte Priorisierung ohne die Nachteile von IP RTP Priority möglich (auch bei hoher Leitungsauslastung). WRED sollte in den Klassen aktiviert werden, die vorrangig TCP Pakete transportieren, damit die Datenrate dieser Klassen an die momentane Auslastung des Links angepaßt werden kann. Nach Meinung des Autors erreicht CBWFQ in Zusammenarbeit mit LLQ und WRED das Optimum an Voicequalität und Flexibilität.
3. Priority Queueing ist für Verbindungen mit geringen Leitungsgeschwindigkeiten

unter hoher Belastung optimal. Wird die gesamte Bandbreite durch Voice Pakete aufgebraucht, so daß keine Daten mehr übertragen werden können, ist eine Limitierung der Bandbreite durch CAR möglich. Vorhandene Leitungskapazität kann dann nicht genutzt werden.

In der Zukunft wird die Migration zu IP Version 6 (IPv6) Erleichterungen im Umgang mit zeitkritischen Daten bringen, da IPv6 QoS implementiert. [RFC 2460]

Neben Cisco haben sich eine ganze Reihe von Herstellern dem Thema VoIP angenommen. So sind Produktlösungen der Hersteller 3Com, Alcatel, Cabletron, Ericsson, Lucent Technologies, Nortel und Siemens geplant oder schon verfügbar. Auch bei Messaging-Produkten oder PC-Software für VoIP Lösungen sind viele Anbieter aktiv. Dazu gehören im Bereich Messaging Boston Technologie, CallWare, Centigram, Digital Sound, Lucent Technologies und Vodavi. Zu den Herstellern von PC-Lösungen sind Intel, Microsoft, Netscape, NetSpeak, Progressive Networks, VocalTec, Voxware zu zählen.

Anhang A

Dokumentation der Tests

A.1 Router-Konfiguration

Im Folgenden sind die Konfigurationen der beiden Router für die jeweiligen Tests zur Reproduzierbarkeit wiedergegeben. Dafür sind nach jeder Änderung der Systemdaten die Einstellungen per `write` in die *startup-config* kopiert worden. Die *startup-config* wird im NVRAM des Routers gehalten und beim Reboot ausgelesen. Die darin enthaltenen Daten werden durch den Befehl

```
copy nvram:startup-config ftp://host/directory/filename
```

auf einen der beiden angeschlossenen Rechner geschrieben. Auf Besonderheiten in der Konfiguration wird in den betreffenden Abschnitten eingegangen.

A.1.1 FIFO-Queueing

Konfiguration Router C3620:

```
!  
version 12.0  
service config  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-36-1  
!  
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0  
enable password cisco1  
!
```

```
!  
!  
!  
!  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.252.1 255.255.255.0  
    ip broadcast-address 10.254.252.255  
    no ip directed-broadcast  
    no ip mroute-cache  
    half-duplex  
    no cdp enable  
!  
interface Serial0/0  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    no fair-queue  
    serial restart-delay 15  
    clockrate 256000  
!  
interface Serial1/0:15
```



```
no ip address
no ip directed-broadcast
no logging event link-status
isdn switch-type primary-qsig
isdn incoming-voice voice
fair-queue 64 256 0
no cdp enable
!
interface Virtual-Template1
no ip address
no ip directed-broadcast
!
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
!
voice-port 1/0:15
compand-type a-law
!
dial-peer voice 2 pots
destination-pattern 91T
direct-inward-dial
port 1/0:15
forward-digits 3
no register e164
!
dial-peer voice 3 voip
destination-pattern 92T
session target ipv4:10.254.250.1
codec g711alaw
no vad
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
line vty 0 4
password cisco
```

```
login
!  
end
```

Konfiguration Router C2610:

```
!  
version 12.0  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-26-1  
!  
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/  
enable password cisco1  
!  
!  
!  
!  
!  
memory-size iomem 7  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!
```

```
interface Ethernet0/0
  ip address 10.254.250.1 255.255.255.0
  ip broadcast-address 10.254.250.255
  no ip directed-broadcast
  no ip mroute-cache
!
interface Serial0/0
  bandwidth 2048
  ip unnumbered Ethernet0/0
  no ip directed-broadcast
  no ip mroute-cache
  no fair-queue
  serial restart-delay 15
!
interface Serial1/0:15
  no ip address
  no ip directed-broadcast
  no logging event link-status
  isdn switch-type primary-qsig
  isdn incoming-voice voice
  fair-queue 64 256 0
  no cdp enable
!
ip classless
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1
no ip http server
!
tftp-server exit
!
voice-port 1/0:15
  compand-type a-law
!
dial-peer voice 1 pots
  destination-pattern 92T
  direct-inward-dial
  port 1/0:15
  forward-digits 3
!
dial-peer voice 2 voip
  destination-pattern 91T
```

```
session target ipv4:10.254.252.1
codec g711alaw
no vad
!
!
line con 0
  transport input none
line aux 0
line vty 0 4
  password cisco
  login
!
end
```

A.1.2 Weighted Fair Queueing

Konfiguration Router C3620:

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-36-1
!
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0
enable password cisco1
!
!
!
!
!
voice-card 1
  codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
```

```
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.252.1 255.255.255.0  
    ip broadcast-address 10.254.252.255  
    no ip directed-broadcast  
    no ip mroute-cache  
    half-duplex  
    no cdp enable  
!  
interface Serial0/0  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    serial restart-delay 15  
    clockrate 256000  
!  
interface Serial1/0:15  
    no ip address  
    no ip directed-broadcast  
    no logging event link-status  
    isdn switch-type primary-qsig  
    isdn incoming-voice voice  
    fair-queue 64 256 0  
    no cdp enable  
!  
interface Virtual-Template1  
    no ip address  
    no ip directed-broadcast  
!
```

```
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
!
voice-port 1/0:15
    compand-type a-law
!
dial-peer voice 2 pots
    destination-pattern 91T
    direct-inward-dial
    port 1/0:15
    forward-digits 3
    no register e164
!
dial-peer voice 3 voip
    destination-pattern 92T
    session target ipv4:10.254.250.1
    codec g711alaw
    no vad
!
!
line con 0
    exec-timeout 0 0
    transport input none
line aux 0
line vty 0 4
    password cisco
    login
!
end
```

Konfiguration Router C2610:

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-26-1
```

```
!  
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/  
enable password cisco1  
!  
!  
!  
!  
!  
memory-size iomem 7  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.250.1 255.255.255.0  
    ip broadcast-address 10.254.250.255  
    no ip directed-broadcast  
    no ip mroute-cache  
!  
interface Serial0/0  
    bandwidth 2048  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    serial restart-delay 15
```

```
!  
interface Serial1/0:15  
  no ip address  
  no ip directed-broadcast  
  no logging event link-status  
  isdn switch-type primary-qsig  
  isdn incoming-voice voice  
  fair-queue 64 256 0  
  no cdp enable  
!  
ip classless  
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1  
no ip http server  
!  
tftp-server exit  
!  
voice-port 1/0:15  
  compand-type a-law  
!  
dial-peer voice 1 pots  
  destination-pattern 92T  
  direct-inward-dial  
  port 1/0:15  
  forward-digits 3  
!  
dial-peer voice 2 voip  
  destination-pattern 91T  
  session target ipv4:10.254.252.1  
  codec g711alaw  
  no vad  
!  
!  
line con 0  
  transport input none  
line aux 0  
line vty 0 4  
  password cisco  
  login  
!  
end
```


A.1.3 IP RTP Priority

Im Abschnitt *interface Serial0/0* ist ersichtlich, daß die Flows auf den RTP Ports 16384-32767 strikt priorisiert werden. Derartigem Verkehr wird eine Bandbreite von $100 \frac{kbit}{s}$ zugewiesen. Weiterhin ist zu erkennen, daß die Bandbreite des Links auf $125 \frac{kbit}{s}$ beschränkt worden ist. Die Gründe dafür sind in Abschnitt 4.4.1.3 dargelegt.

Konfiguration Router C3620:

```
!  
version 12.0  
service config  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-36-1  
!  
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0  
enable password cisco1  
!  
!  
!  
!  
!  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31
```

```
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.252.1 255.255.255.0  
    ip broadcast-address 10.254.252.255  
    no ip directed-broadcast  
    no ip mroute-cache  
    half-duplex  
    no cdp enable  
!  
interface Serial0/0  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    serial restart-delay 15  
    clockrate 125000  
    ip rtp priority 16384 16383 100  
!  
interface Serial1/0:15  
    no ip address  
    no ip directed-broadcast  
    no logging event link-status  
    isdn switch-type primary-qsig  
    isdn incoming-voice voice  
    fair-queue 64 256 0  
    no cdp enable  
!  
interface Virtual-Template1  
    no ip address  
    no ip directed-broadcast  
!  
ip classless  
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1  
no ip http server  
!  
!  
voice-port 1/0:15  
    compand-type a-law  
!
```

```
dial-peer voice 2 pots
 destination-pattern 91T
 direct-inward-dial
 port 1/0:15
 forward-digits 3
 no register e164
!
dial-peer voice 3 voip
 destination-pattern 92T
 session target ipv4:10.254.250.1
 codec g711alaw
 ip precedence 5
 no vad
!
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password cisco
 login
!
end
```

Konfiguration Router C2610:

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-26-1
!
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/
enable password cisco1
!
!
!
!
```

```
!  
memory-size iomem 7  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.250.1 255.255.255.0  
    ip broadcast-address 10.254.250.255  
    no ip directed-broadcast  
    no ip mroute-cache  
!  
interface Serial0/0  
    bandwidth 2048  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    serial restart-delay 15  
    ip rtp priority 16384 16383 100  
!  
interface Serial1/0:15  
    no ip address  
    no ip directed-broadcast  
    no logging event link-status  
    isdn switch-type primary-qsig
```

```
    isdn incoming-voice voice
    fair-queue 64 256 0
    no cdp enable
!
ip classless
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1
no ip http server
!
tftp-server exit
!
voice-port 1/0:15
    compand-type a-law
!
dial-peer voice 1 pots
    destination-pattern 92T
    direct-inward-dial
    port 1/0:15
    forward-digits 3
!
dial-peer voice 2 voip
    destination-pattern 91T
    session target ipv4:10.254.252.1
    codec g711alaw
    ip precedence 5
    no vad
!
!
line con 0
    transport input none
line aux 0
line vty 0 4
    password cisco
    login
!
end
```

A.1.4 Class-Based Weighted Fair Queueing

A.1.4.1 1 Klasse: Voice - TD

Der Klasse *voice_class* werden alle Pakete zugeordnet, auf die die *access-list 1* zutrifft. Dies sind alle Pakete, außer denen, die von den beiden Rechnern am Ethernet-Interface stammen. Durch die Policy *voice_policy* werden die Eigenschaften der Klasse *voice_class* (verfügbare Bandbreite $50 \frac{kbit}{s}$, Tail Drop) bestimmt. Die Klasse *class-default* wird automatisch durch den Router erzeugt und enthält alle Pakete, die nicht anderen Klassen zugeordnet werden können. Die Policy wird dann dem WAN-Interface (nur für ausgehende Pakete) zugewiesen.

Konfiguration Router C3620:

```
!  
version 12.0  
service config  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-36-1  
!  
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0  
enable password cisco1  
!  
!  
class-map voice_class  
    match access-group 1  
!  
!  
policy-map voice_policy  
    class voice_class  
        bandwidth 50  
    class class-default  
        fair-queue  
!  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip
```

```
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
    pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
    ip address 10.254.252.1 255.255.255.0
    ip broadcast-address 10.254.252.255
    no ip directed-broadcast
    no ip mroute-cache
    half-duplex
    no cdp enable
!
interface Serial0/0
    ip unnumbered Ethernet0/0
    no ip directed-broadcast
    no ip mroute-cache
    service-policy output voice_policy
    serial restart-delay 15
    clockrate 256000
!
interface Serial1/0:15
    no ip address
    no ip directed-broadcast
    no logging event link-status
    isdn switch-type primary-qsig
    isdn incoming-voice voice
    fair-queue 64 256 0
    no cdp enable
!
interface Virtual-Template1
```

```
no ip address
no ip directed-broadcast
!
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
access-list 1 deny 10.254.252.2
access-list 1 deny 10.254.250.2
access-list 1 permit any
!
voice-port 1/0:15
compand-type a-law
!
dial-peer voice 2 pots
destination-pattern 91T
direct-inward-dial
port 1/0:15
forward-digits 3
no register e164
!
dial-peer voice 3 voip
destination-pattern 92T
session target ipv4:10.254.250.1
codec g711alaw
ip precedence 5
no vad
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
line vty 0 4
password cisco
login
!
end
```

Konfiguration Router C2610:

!


```
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-26-1
!
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/
enable password cisco1
!
!
class-map voice_class
    match access-group 1
!
!
policy-map voice_policy
    class voice_class
        bandwidth 50
    class class-default
        fair-queue
!
memory-size iomem 7
voice-card 1
    codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
    pri-group timeslots 1-31
!
!
```

```
!  
interface Ethernet0/0  
  ip address 10.254.250.1 255.255.255.0  
  ip broadcast-address 10.254.250.255  
  no ip directed-broadcast  
  no ip mroute-cache  
!  
interface Serial0/0  
  bandwidth 2048  
  ip unnumbered Ethernet0/0  
  no ip directed-broadcast  
  no ip mroute-cache  
  service-policy output voice_policy  
  serial restart-delay 15  
!  
interface Serial1/0:15  
  no ip address  
  no ip directed-broadcast  
  no logging event link-status  
  isdn switch-type primary-qsig  
  isdn incoming-voice voice  
  fair-queue 64 256 0  
  no cdp enable  
!  
ip classless  
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1  
no ip http server  
!  
access-list 1 deny 10.254.252.2  
access-list 1 deny 10.254.250.2  
access-list 1 permit any  
tftp-server exit  
!  
voice-port 1/0:15  
  compand-type a-law  
!  
dial-peer voice 1 pots  
  destination-pattern 92T  
  direct-inward-dial  
  port 1/0:15
```

```
forward-digits 3
!
dial-peer voice 2 voip
destination-pattern 91T
session target ipv4:10.254.252.1
codec g711alaw
ip precedence 5
no vad
!
!
line con 0
transport input none
line aux 0
line vty 0 4
password cisco
login
!
end
```

A.1.4.2 2 Klassen: Voice - TD, Data - TD

Die zweite Klasse für Datenpakete (*data_class*) enthält nun alle Pakete, die der *access-list* 2 entsprechen. Sie ist das Gegenteil der *access-list* 1. Beiden Klassen sind jeweils 50 % der verfügbaren Bandbreite zugeordnet. Weiterhin ist Tail Drop aktiv.

Konfiguration Router C3620:

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-36-1
!
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0
enable password cisco1
!
!
class-map data_class
```

```
    match access-group 2
class-map voice_class
    match access-group 1
!
!
policy-map voice_policy
    class class-default
        fair-queue
    class voice_class
        bandwidth percent 50
    class data_class
        bandwidth percent 50
!
voice-card 1
    codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
    pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
    ip address 10.254.252.1 255.255.255.0
    ip broadcast-address 10.254.252.255
    no ip directed-broadcast
    no ip mroute-cache
    half-duplex
    no cdp enable
!
```

```
interface Serial0/0
  ip unnumbered Ethernet0/0
  no ip directed-broadcast
  no ip mroute-cache
  service-policy output voice_policy
  serial restart-delay 15
  clockrate 256000
!
interface Serial1/0:15
  no ip address
  no ip directed-broadcast
  no logging event link-status
  isdn switch-type primary-qsig
  isdn incoming-voice voice
  fair-queue 64 256 0
  no cdp enable
!
interface Virtual-Template1
  no ip address
  no ip directed-broadcast
!
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
access-list 1 deny 10.254.252.2
access-list 1 deny 10.254.250.2
access-list 1 permit any
access-list 2 permit 10.254.252.2
access-list 2 permit 10.254.250.2
access-list 2 deny any
!
voice-port 1/0:15
  compand-type a-law
!
dial-peer voice 2 pots
  destination-pattern 91T
  direct-inward-dial
  port 1/0:15
  forward-digits 3
```

```
no register e164
!
dial-peer voice 3 voip
destination-pattern 92T
session target ipv4:10.254.250.1
codec g711alaw
no vad
!
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
line vty 0 4
password cisco
login
!
end
```

Konfiguration Router C2610:

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-26-1
!
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/
enable password cisco1
!
!
class-map data_class
match access-group 2
class-map voice_class
match access-group 1
!
!
policy-map voice_policy
class class-default
```

```
    fair-queue
  class voice_class
    bandwidth percent 50
  class data_class
    bandwidth percent 50
!
memory-size iomem 7
voice-card 1
  codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
  pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
  ip address 10.254.250.1 255.255.255.0
  ip broadcast-address 10.254.250.255
  no ip directed-broadcast
  no ip mroute-cache
!
interface Serial0/0
  bandwidth 2048
  ip unnumbered Ethernet0/0
  no ip directed-broadcast
  no ip mroute-cache
  service-policy output voice_policy
  serial restart-delay 15
!
```

```
interface Serial1/0:15
  no ip address
  no ip directed-broadcast
  no logging event link-status
  isdn switch-type primary-qsig
  isdn incoming-voice voice
  fair-queue 64 256 0
  no cdp enable
!
ip classless
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1
no ip http server
!
access-list 1 deny 10.254.252.2
access-list 1 deny 10.254.250.2
access-list 1 permit any
access-list 2 permit 10.254.252.2
access-list 2 permit 10.254.250.2
access-list 2 deny any
tftp-server exit
!
voice-port 1/0:15
  compand-type a-law
!
dial-peer voice 1 pots
  destination-pattern 92T
  direct-inward-dial
  port 1/0:15
  forward-digits 3
!
dial-peer voice 2 voip
  destination-pattern 91T
  session target ipv4:10.254.252.1
  codec g711alaw
  no vad
!
!
line con 0
  transport input none
line aux 0
```



```
line vty 0 4
  password cisco
  login
!
end
```

A.1.4.3 2 Klassen: Voice - TD, Data - WRED

Im Unterschied zu Abschnitt A.1.4.2 wird hier bei den Klassen *data_class* und *class-default* WRED aktiviert.

Konfiguration Router C3620:

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-36-1
!
enable secret 5 $1$zFZE$BLVNIJtrEzKNM8LZQeU0ym0
enable password cisco1
!
!
class-map data_class
  match access-group 2
class-map voice_class
  match access-group 1
!
!
policy-map voice_policy
  class class-default
    fair-queue
    random-detect
  class voice_class
    bandwidth percent 50
  class data_class
    bandwidth percent 50
    random-detect
```

```
!  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.252.1 255.255.255.0  
    ip broadcast-address 10.254.252.255  
    no ip directed-broadcast  
    no ip mroute-cache  
    half-duplex  
    no cdp enable  
!  
interface Serial0/0  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    service-policy output voice_policy  
    serial restart-delay 15  
    clockrate 256000  
!  
interface Serial1/0:15  
    no ip address  
    no ip directed-broadcast  
    no logging event link-status
```

```
isdn switch-type primary-qsig
isdn incoming-voice voice
fair-queue 64 256 0
no cdp enable
!
interface Virtual-Templat1
  no ip address
  no ip directed-broadcast
!
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
access-list 1 deny 10.254.252.2
access-list 1 deny 10.254.250.2
access-list 1 permit any
access-list 2 permit 10.254.252.2
access-list 2 permit 10.254.250.2
access-list 2 deny any
!
voice-port 1/0:15
  compand-type a-law
!
dial-peer voice 2 pots
  destination-pattern 91T
  direct-inward-dial
  port 1/0:15
  forward-digits 3
  no register e164
!
dial-peer voice 3 voip
  destination-pattern 92T
  session target ipv4:10.254.250.1
  codec g711alaw
  no vad
!
!
line con 0
  exec-timeout 0 0
  transport input none
```

```
line aux 0
line vty 0 4
  password cisco
  login
!
end
```

Konfiguration Router C2610:

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-26-1
!
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/
enable password cisco1
!
!
class-map data_class
  match access-group 2
class-map voice_class
  match access-group 1
!
!
policy-map voice_policy
  class class-default
    fair-queue
    random-detect
  class voice_class
    bandwidth percent 50
  class data_class
    bandwidth percent 50
    random-detect
!
memory-size iomem 7
voice-card 1
  codec complexity high
!
```

```
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
    pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
    ip address 10.254.250.1 255.255.255.0
    ip broadcast-address 10.254.250.255
    no ip directed-broadcast
    no ip mroute-cache
!
interface Serial0/0
    bandwidth 2048
    ip unnumbered Ethernet0/0
    no ip directed-broadcast
    no ip mroute-cache
    service-policy output voice_policy
    serial restart-delay 15
!
interface Serial1/0:15
    no ip address
    no ip directed-broadcast
    no logging event link-status
    isdn switch-type primary-qsig
    isdn incoming-voice voice
    fair-queue 64 256 0
    no cdp enable
!
ip classless
```

```
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1
no ip http server
!
access-list 1 deny 10.254.252.2
access-list 1 deny 10.254.250.2
access-list 1 permit any
access-list 2 permit 10.254.252.2
access-list 2 permit 10.254.250.2
access-list 2 deny any
tftp-server exit
!
voice-port 1/0:15
    compand-type a-law
!
dial-peer voice 1 pots
    destination-pattern 92T
    direct-inward-dial
    port 1/0:15
    forward-digits 3
!
dial-peer voice 2 voip
    destination-pattern 91T
    session target ipv4:10.254.252.1
    codec g711alaw
    no vad
!
!
line con 0
    transport input none
line aux 0
line vty 0 4
    password cisco
    login
!
end
```

A.1.4.4 2 Klassen: Voice - LLQ, Data - WRED

Nun wird innerhalb der Klasse *voice_class* eine Priority Queue mit einer Bandbreite von $200 \frac{\text{kbit}}{\text{s}}$ eingerichtet. Die restlichen Einstellungen entsprechen der Konfiguration in Ka-

pitel A.1.4.3.

Konfiguration Router C3620:

```
!  
version 12.0  
service config  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-36-1  
!  
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0  
enable password cisco1  
!  
!  
class-map data_class  
    match access-group 2  
class-map voice_class  
    match access-group 1  
!  
!  
policy-map voice_policy  
    class class-default  
        fair-queue  
        random-detect  
    class voice_class  
        priority 200  
    class data_class  
        bandwidth 50  
        random-detect  
!  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!
```

```
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
  pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
  ip address 10.254.252.1 255.255.255.0
  ip broadcast-address 10.254.252.255
  no ip directed-broadcast
  no ip mroute-cache
  half-duplex
  no cdp enable
!
interface Serial0/0
  ip unnumbered Ethernet0/0
  no ip directed-broadcast
  no ip mroute-cache
  service-policy output voice_policy
  serial restart-delay 15
  clockrate 256000
!
interface Serial1/0:15
  no ip address
  no ip directed-broadcast
  no logging event link-status
  isdn switch-type primary-qsig
  isdn incoming-voice voice
  fair-queue 64 256 0
  no cdp enable
!
interface Virtual-Template1
  no ip address
  no ip directed-broadcast
!
```



```
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
access-list 1 deny 10.254.252.2
access-list 1 deny 10.254.250.2
access-list 1 permit any
access-list 2 permit 10.254.252.2
access-list 2 permit 10.254.250.2
access-list 2 deny any
!
voice-port 1/0:15
    compand-type a-law
!
dial-peer voice 2 pots
    destination-pattern 91T
    direct-inward-dial
    port 1/0:15
    forward-digits 3
    no register e164
!
dial-peer voice 3 voip
    destination-pattern 92T
    session target ipv4:10.254.250.1
    codec g711alaw
    no vad
!
!
line con 0
    exec-timeout 0 0
    transport input none
line aux 0
line vty 0 4
    password cisco
    login
!
end
```

Konfiguration Router C2610:

!

```
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-26-1
!
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/
enable password cisco1
!
!
class-map data_class
    match access-group 2
class-map voice_class
    match access-group 1
!
!
policy-map voice_policy
    class class-default
        fair-queue
        random-detect
    class voice_class
        priority 200
    class data_class
        bandwidth 50
        random-detect
!
memory-size iomem 7
voice-card 1
    codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
```

```
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.250.1 255.255.255.0  
    ip broadcast-address 10.254.250.255  
    no ip directed-broadcast  
    no ip mroute-cache  
!  
interface Serial0/0  
    bandwidth 2048  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    service-policy output voice_policy  
    serial restart-delay 15  
!  
interface Serial1/0:15  
    no ip address  
    no ip directed-broadcast  
    no logging event link-status  
    isdn switch-type primary-qsig  
    isdn incoming-voice voice  
    fair-queue 64 256 0  
    no cdp enable  
!  
ip classless  
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1  
no ip http server  
!  
access-list 1 deny    10.254.252.2  
access-list 1 deny    10.254.250.2  
access-list 1 permit any  
access-list 2 permit 10.254.252.2  
access-list 2 permit 10.254.250.2  
access-list 2 deny    any
```

```
tftp-server exit
!
voice-port 1/0:15
    compand-type a-law
!
dial-peer voice 1 pots
    destination-pattern 92T
    direct-inward-dial
    port 1/0:15
    forward-digits 3
!
dial-peer voice 2 voip
    destination-pattern 91T
    session target ipv4:10.254.252.1
    codec g711alaw
    no vad
!
!
line con 0
    transport input none
line aux 0
line vty 0 4
    password cisco
    login
!
no scheduler allocate
end
```

A.1.5 Priority Queueing

Dem *interface Serial0/0* wird die *priority-group 1* zugeordnet. Alle Pakete vom *interface Serial1/0:15* (Voice-Interface) gelangen in die Queue mit der höchsten Priorität. Die Pakete des *Ethernet0/0*-Interfaces erhalten die niedrigste Priorität.

Konfiguration Router C3620:

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
!
hostname voip-router-36-1
!
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0
enable password cisco1
!
!
!
!
!
voice-card 1
    codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
    pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
    ip address 10.254.252.1 255.255.255.0
    ip broadcast-address 10.254.252.255
    no ip directed-broadcast
    no ip mroute-cache
    half-duplex
    no cdp enable
!
interface Serial0/0
    ip unnumbered Ethernet0/0
```

```
no ip directed-broadcast
no ip mroute-cache
priority-group 1
serial restart-delay 15
clockrate 256000
!
interface Serial1/0:15
no ip address
no ip directed-broadcast
no logging event link-status
isdn switch-type primary-qsig
isdn incoming-voice voice
fair-queue 64 256 0
no cdp enable
!
interface Virtual-Template1
no ip address
no ip directed-broadcast
!
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
priority-list 1 interface Serial1/0:15 high
priority-list 1 interface Ethernet0/0 low
!
voice-port 1/0:15
compand-type a-law
!
dial-peer voice 2 pots
destination-pattern 91T
direct-inward-dial
port 1/0:15
forward-digits 3
no register e164
!
dial-peer voice 3 voip
destination-pattern 92T
session target ipv4:10.254.250.1
codec g711alaw
```

```
no vad
!
!
line con 0
  exec-timeout 0 0
  transport input none
line aux 0
line vty 0 4
  password cisco
  login
!
end
```

Konfiguration Router C2610:

```
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname voip-router-26-1
!
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/
enable password cisco1
!
!
!
!
!
memory-size iomem 7
voice-card 1
  codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
```

```
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.250.1 255.255.255.0  
    ip broadcast-address 10.254.250.255  
    no ip directed-broadcast  
    no ip mroute-cache  
!  
interface Serial0/0  
    bandwidth 2048  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    priority-group 1  
    serial restart-delay 15  
!  
interface Serial1/0:15  
    no ip address  
    no ip directed-broadcast  
    no logging event link-status  
    isdn switch-type primary-qsig  
    isdn incoming-voice voice  
    fair-queue 64 256 0  
    no cdp enable  
!  
ip classless  
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1  
no ip http server  
!  
priority-list 1 interface Serial1/0:15 high  
priority-list 1 interface Ethernet0/0 low  
tftp-server exit  
!
```



```
voice-port 1/0:15
  compand-type a-law
!
dial-peer voice 1 pots
  destination-pattern 92T
  direct-inward-dial
  port 1/0:15
  forward-digits 3
!
dial-peer voice 2 voip
  destination-pattern 91T
  session target ipv4:10.254.252.1
  codec g711alaw
  no vad
!
!
line con 0
  transport input none
line aux 0
line vty 0 4
  password cisco
  login
!
end
```

A.1.6 Priority Queueing + CAR

Die Konfiguration des Priority Queueing entspricht dem in Abschnitt A.1.5. Für die Limitierung des Voice-Verkehrs kommt die in Kapitel A.1.4.1 vorgestellte *access-list 1* zum Einsatz. Jedweder Traffic auf den diese access-list zutrifft und der im Schnitt eine Datenrate von $96 \frac{kbit}{s}$ überschreitet oder Bursts mit mehr als 10 *kbytes* erzeugt, wird verworfen. Im Abschnitt *interface Serial0/0* ist bei der Konfiguration der Seitenumbruch zu berücksichtigen.

Konfiguration Router C3620:

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
!
hostname voip-router-36-1
!
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0
enable password cisco1
!
!
!
!
!
voice-card 1
    codec complexity high
!
ip subnet-zero
ip ftp username voip
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
    pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
    ip address 10.254.252.1 255.255.255.0
    ip broadcast-address 10.254.252.255
    no ip directed-broadcast
    no ip mroute-cache
    half-duplex
    no cdp enable
!
interface Serial0/0
    ip unnumbered Ethernet0/0
```

```
no ip directed-broadcast
rate-limit output access-group 1 96000 10000 10000 conform-action transmit
    exceed-action drop
no ip mroute-cache
priority-group 1
serial restart-delay 15
clockrate 256000
!
interface Serial1/0:15
no ip address
no ip directed-broadcast
no logging event link-status
isdn switch-type primary-qsig
isdn incoming-voice voice
fair-queue 64 256 0
no cdp enable
!
interface Virtual-Template1
no ip address
no ip directed-broadcast
!
ip classless
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1
no ip http server
!
access-list 1 deny    10.254.252.2
access-list 1 deny    10.254.250.2
access-list 1 permit any
priority-list 1 interface Serial1/0:15 high
priority-list 1 interface Ethernet0/0 low
!
voice-port 1/0:15
    compand-type a-law
!
dial-peer voice 2 pots
    destination-pattern 91T
    direct-inward-dial
    port 1/0:15
    forward-digits 3
    no register e164
```

```
!  
dial-peer voice 3 voip  
  destination-pattern 92T  
  session target ipv4:10.254.250.1  
  codec g711alaw  
  no vad  
!  
!  
line con 0  
  exec-timeout 0 0  
  transport input none  
line aux 0  
line vty 0 4  
  password cisco  
  login  
!  
end
```

Konfiguration Router C2610:

```
!  
version 12.0  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-26-1  
!  
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/  
enable password cisco1  
!  
!  
!  
!  
!  
memory-size iomem 7  
voice-card 1  
  codec complexity high  
!  
ip subnet-zero  
ip ftp username voip
```

```
ip ftp password voiptest
no ip domain-lookup
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 1/0
    pri-group timeslots 1-31
!
!
!
interface Ethernet0/0
    ip address 10.254.250.1 255.255.255.0
    ip broadcast-address 10.254.250.255
    no ip directed-broadcast
    no ip mroute-cache
!
interface Serial0/0
    bandwidth 2048
    ip unnumbered Ethernet0/0
    no ip directed-broadcast
    rate-limit output access-group 1 96000 10000 10000 conform-action transmit
        exceed-action drop
    no ip mroute-cache
    priority-group 1
    serial restart-delay 15
!
interface Serial1/0:15
    no ip address
    no ip directed-broadcast
    no logging event link-status
    isdn switch-type primary-qsig
    isdn incoming-voice voice
    fair-queue 64 256 0
    no cdp enable
!
ip classless
```

```
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1
no ip http server
!
access-list 1 deny 10.254.252.2
access-list 1 deny 10.254.250.2
access-list 1 permit any
priority-list 1 interface Serial1/0:15 high
priority-list 1 interface Ethernet0/0 low
tftp-server exit
!
voice-port 1/0:15
    compand-type a-law
!
dial-peer voice 1 pots
    destination-pattern 92T
    direct-inward-dial
    port 1/0:15
    forward-digits 3
!
dial-peer voice 2 voip
    destination-pattern 91T
    session target ipv4:10.254.252.1
    codec g711alaw
    no vad
!
!
line con 0
    transport input none
line aux 0
line vty 0 4
    password cisco
    login
!
end
```

A.1.7 Compressed Real-Time Protocol

Wie schon in Kapitel 4.4.1.6 beschrieben findet als Queueing-Mechanismus PQ Verwendung. Dessen Konfiguration entspricht dem in Abschnitt A.1.5.

Konfiguration Router C3620:

```
!  
version 12.0  
service config  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-36-1  
!  
enable secret 5 $1$zFZE$BLVNJtrEzKNM8LZQeU0ym0  
enable password cisco1  
!  
!  
!  
!  
!  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!
```

```
!  
interface Ethernet0/0  
  ip address 10.254.252.1 255.255.255.0  
  ip broadcast-address 10.254.252.255  
  no ip directed-broadcast  
  no ip mroute-cache  
  half-duplex  
  no cdp enable  
!  
interface Serial0/0  
  ip unnumbered Ethernet0/0  
  no ip directed-broadcast  
  no ip mroute-cache  
  priority-group 1  
  serial restart-delay 15  
  clockrate 256000  
  ip rtp header-compression  
!  
interface Serial1/0:15  
  no ip address  
  no ip directed-broadcast  
  no logging event link-status  
  isdn switch-type primary-qsig  
  isdn incoming-voice voice  
  fair-queue 64 256 0  
  no cdp enable  
!  
interface Virtual-Template1  
  no ip address  
  no ip directed-broadcast  
!  
ip classless  
ip route 10.254.250.0 255.255.255.0 Serial0/0 10.254.250.1  
no ip http server  
!  
priority-list 1 interface Serial1/0:15 high  
priority-list 1 interface Ethernet0/0 low  
!  
voice-port 1/0:15  
  compand-type a-law
```



```
!  
dial-peer voice 2 pots  
  destination-pattern 91T  
  direct-inward-dial  
  port 1/0:15  
  forward-digits 3  
  no register e164  
!  
dial-peer voice 3 voip  
  destination-pattern 92T  
  session target ipv4:10.254.250.1  
  codec g711alaw  
  no vad  
!  
!  
line con 0  
  exec-timeout 0 0  
  transport input none  
line aux 0  
line vty 0 4  
  password cisco  
  login  
!  
end
```

Konfiguration Router C2610:

```
!  
version 12.0  
service timestamps debug uptime  
service timestamps log uptime  
no service password-encryption  
!  
hostname voip-router-26-1  
!  
enable secret 5 $1$xEk6$JENBnUM.fXySLYNAQLlSE/  
enable password cisco1  
!  
!  
!  
!
```

```
!  
memory-size iomem 7  
voice-card 1  
    codec complexity high  
!  
ip subnet-zero  
ip ftp username voip  
ip ftp password voiptest  
no ip domain-lookup  
!  
isdn switch-type primary-qsig  
isdn voice-call-failure 0  
!  
!  
!  
!  
controller E1 1/0  
    pri-group timeslots 1-31  
!  
!  
!  
interface Ethernet0/0  
    ip address 10.254.250.1 255.255.255.0  
    ip broadcast-address 10.254.250.255  
    no ip directed-broadcast  
    no ip mroute-cache  
!  
interface Serial0/0  
    bandwidth 2048  
    ip unnumbered Ethernet0/0  
    no ip directed-broadcast  
    no ip mroute-cache  
    priority-group 1  
    serial restart-delay 15  
    ip rtp header-compression  
!  
interface Serial1/0:15  
    no ip address  
    no ip directed-broadcast  
    no logging event link-status
```

```
isdn switch-type primary-qsig
isdn incoming-voice voice
fair-queue 64 256 0
no cdp enable
!
ip classless
ip route 10.254.252.0 255.255.255.0 Serial0/0 10.254.252.1
no ip http server
!
priority-list 1 interface Serial1/0:15 high
priority-list 1 interface Ethernet0/0 low
tftp-server exit
!
voice-port 1/0:15
  compand-type a-law
!
dial-peer voice 1 pots
  destination-pattern 92T
  direct-inward-dial
  port 1/0:15
  forward-digits 3
!
dial-peer voice 2 voip
  destination-pattern 91T
  session target ipv4:10.254.252.1
  codec g711alaw
  no vad
!
!
line con 0
  transport input none
line aux 0
line vty 0 4
  password cisco
  login
!
no scheduler allocate
end
```

A.1.8 Codecs

Auch für die Bewertung der Codecs ist PQ als Queueing-Algorithmus verwendet worden. Die Konfiguration entspricht der in Abschnitt A.1.5 gezeigten. Für eine bessere Übersicht werden nur noch die relevanten Teile wiedergegeben.

A.1.8.1 G.711 A law

Konfiguration Router C3620:

```
!  
dial-peer voice 3 voip  
  destination-pattern 92T  
  session target ipv4:10.254.250.1  
  codec g711alaw  
  no vad  
!
```

Konfiguration Router C2610:

```
!  
dial-peer voice 2 voip  
  destination-pattern 91T  
  session target ipv4:10.254.252.1  
  codec g711alaw  
  no vad  
!
```

A.1.8.2 G.723.1

Konfiguration Router C3620 (Bandbreite $5,3 \frac{kbit}{s}$):

```
! dial-peer voice 3 voip  
  destination-pattern 92T  
  session target ipv4:10.254.250.1  
  codec g723r53  
  no vad  
!
```

Konfiguration Router C2610 (Bandbreite $5,3 \frac{kbit}{s}$):

```
! dial-peer voice 2 voip  
  destination-pattern 91T
```

```
session target ipv4:10.254.252.1
codec g723r53
no vad
!
```

Konfiguration Router C3620 (Bandbreite $6,3 \frac{kbit}{s}$):

```
! dial-peer voice 3 voip
destination-pattern 92T
session target ipv4:10.254.250.1
codec g723r63
no vad
!
```

Konfiguration Router C2610 (Bandbreite $6,3 \frac{kbit}{s}$):

```
! dial-peer voice 2 voip
destination-pattern 91T
session target ipv4:10.254.252.1
codec g723r63
no vad
!
```

A.1.8.3 G.726

Konfiguration Router C3620 (Bandbreite $16 \frac{kbit}{s}$):

```
! dial-peer voice 3 voip
destination-pattern 92T
session target ipv4:10.254.250.1
codec g726r16
no vad
!
```

Konfiguration Router C2610 (Bandbreite $16 \frac{kbit}{s}$):

```
! dial-peer voice 2 voip
destination-pattern 91T
session target ipv4:10.254.252.1
codec g726r16
no vad
!
```

A.1.8.4 G.728

Konfiguration Router C3620:

```
! dial-peer voice 3 voip
  destination-pattern 92T
  session target ipv4:10.254.250.1
  codec g728
  no vad
!
```

Konfiguration Router C2610:

```
! dial-peer voice 2 voip
  destination-pattern 91T
  session target ipv4:10.254.252.1
  codec g728
  no vad
!
```

A.1.8.5 G.729

Konfiguration Router C3620:

```
! dial-peer voice 3 voip
  destination-pattern 92T
  session target ipv4:10.254.250.1
  codec g729r8
  no vad
!
```

Konfiguration Router C2610:

```
! dial-peer voice 2 voip
  destination-pattern 91T
  session target ipv4:10.254.252.1
  codec g729r8
  no vad
!
```

A.2 Anlagenbelegung Hicom 300

Die Hicom 300 (Anlagennummer: L31610-E504) im Comlab weist die Software-Version 3.6-10 auf. Die aktuelle Anlagenbelegung kann der Abbildung A.1 entnommen werden.

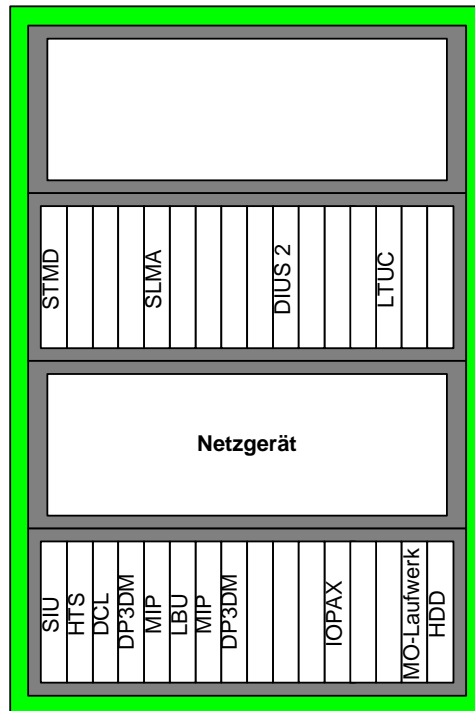


Abbildung A.1: Anlagenbelegung Hicom 300

Anhang B

Detaillierte Meßergebnisse

Im Folgenden sind die Meßergebnisse der Versuchsreihen wiedergegeben. Die Zusammenfassungen der Ergebnisse befinden sich im Kapitel 4. Eingeklammerte Werte sind offensichtlich Ausreißer bzw. fehlerhafte Angaben der Router bei der Callanalyse (siehe auch Abschnitt 4.4.1). Solche Meßwerte sind nicht mit in die Bewertung aufgenommen worden.

Tabelle B.1: Meßwerte Netperf - unbelasteter Link

Clock Rate 125000 bps						Clock Rate 256000 bps					
UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\bigcirc $\frac{Trans.}{s}$	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\bigcirc $\frac{Trans.}{s}$
144,32		120,03		9,40		297,63		246,73		18,81	
144,29	144,30	120,06	120,06	9,40	9,40	297,55	297,56	246,71	246,72	18,80	18,81
144,30		120,08		9,40		297,49		246,72		18,81	

Tabelle B.2: Meßwerte Netperf - FIFO

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\emptyset $\frac{Trans.}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\emptyset $\frac{Trans.}{s}$
—		144,30		120,06		9,40		297,56		246,72		18,81
1 call	144,22		93,88		3,80		297,43		201,40		15,91	
	144,27	144,24	93,98	93,89	3,80	3,80	297,61	297,56	200,97	201,24	15,91	15,81
	144,23		93,80		3,80		297,63		201,36		15,61	
1 call + 1 fax	128,66	128,66					264,68	264,68				
2 calls							296,60		198,26		8,20	
							296,73	296,64	199,14	198,76	8,20	8,20
							296,58		198,87		8,21	
2 calls + 1 fax							263,78	263,78				

Tabelle B.3: Meßwerte Call-Analyse - FIFO (Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
1 call	–	53	53	0	0	0	0	0	0	0	100
	UDP_STREAM	64		699		62		67		382,5	0
		64	64	698	699	65	63	71	69		
		64		699		63		69			
	TCP_STREAM	64		538		118		148		335,8	10
		63 [121]	64	539 522	533	169 113	133	138 145	144		
	TCP_CRR	55		0		0		0		0	100
		52	53	0	0	0	0	0	0		
		53		0		0		1			
1 call + 1 fax	UDP_STREAM	64	64	702	702	67	67	69	69	385,0	0

Tabelle B.4: Meßwerte Call-Analyse - FIFO (Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	%
1 call	–		53	53	0	0	0	0	0	0	0	100
	UDP_STREAM		64	64	582	583	22	25	62	61	313,0	0
			64		582		25		58			
			63		585		29		62			
	TCP_STREAM		63	62	342	350	84	70	81	89	214,8	10
			60		352		62		96			
			63		355		64		91			
	TCP_CRR		52	53	0	0	0	0	1	1	0,3	100
			53		0		0		1			
			53		0		0		0			
1 call + 1 fax	UDP_STREAM		64	64	583	583	24	24	63	63	313,3	0

Tabelle B.4: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	%
2 calls	–	1	53	53	0	0	0	0	1	1	0,3	100
		2	53	53	0	0	0	0	1	1	0,3	100
	UDP_STREAM	1	64	64	580	592	23	32	71	45	315,3	0
			64		599		37		31			
			64		596		35		33			
		2	64	64	600	599	10	27	32	32	314,3	0
			64		599		36		32			
			64		597		36		33			
	TCP_STREAM	1	47	53	[62299]	512	68	67	68	67	289,5	10
			53		514		63		64			
			58		510		71		69			
		2	56	57	512	509	75	71	71	68	289,3	10
			58		505		75		72			
			56		510		64		61			

Tabelle B.4: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	%
2 calls	TCP_CRR	1	53		0		0		1			
			53	53	0	0	0	0	0	0	0	100
			52		0		0		0			
		2	52		0		0		1			
			53	52	0	0	0	0	1	1	0,3	100
			52		0		0		0			
2 calls + 1 fax	UDP_STREAM	1	64	64	611	611	53	53	65	65	335,0	0
		2	64	64	600	600	45	45	41	41	321,5	0

Tabelle B.5: Meßwerte Netperf - WFQ

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$
–		144,30		120,06		9,40		297,56		246,72		18,81
1 call	62,81		50,57		3,80		198,38		165,06		15,91	
	62,81	62,88	50,63	50,56	3,81	3,80	198,28	198,31	165,45	164,93	15,81	15,88
	63,03		50,48		3,80		198,28		164,29		15,91	
1 call + 1 fax	48,07	48,07					185,32	185,32				
2 calls							99,95		82,43		8,20	
							100,77	100,52	82,87	83,61	8,20	8,20
							100,85		82,54		8,20	
2 calls + 1 fax							86,02	86,02				

Tabelle B.6: Meßwerte Call-Analyse - WFQ (Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
1 call	–	53	53	0	0	0	0	0	0	0	100
	UDP_STREAM	74	70	127	139	69	71	145	147	124,0	50
		71		155		87		138			
		65		134		56		159			
	TCP_STREAM	50	49	96	109	128	131	265	262	152,8	50
		45		104		161		248			
		53		128		105		273			
	TCP_CRR	53	53	0	0	0	0	1	1	0,3	100
		52		0		0		1			
		53		0		0		1			
1 call + 1 fax	UDP_STREAM	56	56	467	467	154	154	221	221	327,3	20

Tabelle B.7: Meßwerte Call-Analyse - WFQ (Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
1 call	–		53	53	0	0	0	0	0	0	0	100
	UDP_STREAM		53	60	0	0	0	4	0	0	1,0	100
			64		0		5		0			
			63		0		7		1			
	TCP_STREAM		52	57	0	0	21	11	0	1	3,0	100
			56		0		0		1			
			64		0		13		1			
	TCP_CRR		53	53	0	0	0	0	1	1	0,3	100
			53		0		0		1			
			53		0		0		0			
1 call + 1 fax	UDP_STREAM		64	64	398	398	228	228	373	373	349,3	20

Tabelle B.7: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
2 calls	–	1	52	52	0	0	0	0	0	0	0	100
		2	52	52	0	0	0	0	1	1	0,3	100
	UDP_STREAM	1	52	52	0	0	0	117	0	102	54,8	100
			[146]		0		214		149			
			[164]		0		137		157			
		2	52	56	0	5	34	54	1	0	16,0	100
			48		16		127		0			
			68		0		2		0			
	TCP_STREAM	1	53	54	0	0	20	54	1	1	13,8	100
			53		0		55		1			
			55		0		86		1			
		2	53	53	0	0	149	114	0	0	28,5	100
			52		0		117		0			
			55		0		75		0			

Tabelle B.7: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
2 calls	TCP_CRR	1	52		0		0		1			
			52	52	0	0	0	0	0	0	0	100
			53		0		0		0			
		2	52		0		0		0			
			53	52	0	0	0	0	1	1	0,3	100
			52		0		0		1			
2 calls + 1 fax	UDP_STREAM	1	51	51	335	335	79	79	73	73	205,5	50
		2	45	45	240	240	79	79	73	73	158,0	50

Tabelle B.8: Meßwerte Netperf - IP RTP Priority

Test	Clock Rate 125000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\emptyset $\frac{Trans.}{s}$
—		144,30		120,06		9,40
1 call	45,91		38,06		3,50	
	45,86	45,91	37,91	37,98	3,50	3,50
	45,96		37,98		3,51	
1 call + 1 fax	31,53	31,53				

Tabelle B.9: Meßwerte Call-Analyse - IP RTP Priority (Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
1 call	–	52	52	0	0	0	0	0	0	0	100
	UDP_STREAM	64	66	0	0	4	5	0	0	1,3	100
		71		0		3		0			
		62		0		9		1			
	TCP_STREAM	45	50	0	0	4	3	1	1	1,0	100
		53		0		3		1			
		52		0		3		0			
	TCP_CRR	52	52	0	0	0	0	1	0	0	100
		53		0		0		0			
		52		0		0		0			
1 call + 1 fax	UDP_STREAM	51	51	411	411	43	43	56	56	230,3	50

Tabelle B.10: Meßwerte Netperf - CBWFQ (1 Klasse: Voice - TD)

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\emptyset $\frac{Trans.}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\emptyset $\frac{Trans.}{s}$
–		144,30		120,06		9,40		297,56		246,72		18,81
1 call	46,70		37,91		3,50		199,20		164,56		12,51	
	46,56	46,70	38,13	38,02	3,50	3,50	198,97	199,10	165,12	164,74	12,51	12,51
	46,84		38,03		3,50		199,12		164,53		12,50	
1 call + 1 fax	35,89	35,89					141,73	141,73				
2 calls							100,80		83,13		7,10	
							100,57	100,76	83,22	83,13	7,10	7,10
							100,91		83,03		7,10	
2 calls + 1 fax							56,54	56,54				

Tabelle B.11: Meßwerte Call-Analyse - CBWFQ (1 Klasse: Voice - TD;Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
1 call	–	53	53	0	0	0	0	0	0	0	100
	UDP_STREAM	63	63	0	0	2	7	0	0	1,8	100
		64		0		12		1			
		61		0		6		0			
	TCP_STREAM	45	48	0	0	4	9	0	0	2,3	100
		53		0		13		1			
		46		0		11		0			
	TCP_CRR	53	53	0	0	0	0	0	0	0	100
		52		0		0		0			
		53		0		0		1			
1 call + 1 fax	UDP_STREAM	62	62	367	367	59	59	72	72	216,3	50

Tabelle B.12: Meßwerte Call-Analyse - CBWFQ (1 Klasse: Voice - TD;
Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms		
1 call	–		53	53	0	0	0	0	1	1	0,3	100
	UDP_STREAM		53	53	0	0	0	0	1	1	0,3	100
			53		0		0		1			
			53		0		0		1			
	TCP_STREAM		55	54	0	0	0	0	0	0	0	100
			53		0		0		1			
			55		0		0		0			
	TCP_CRR		52	52	0	0	0	0	1	0	0	100
			52		0		0		0			
			53		0		0		0			
1 call + 1 fax	UDP_STREAM		53	53	38	38	37	37	43	43	39,0	100

Tabelle B.12: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms	ms	%
2 calls	–	1	52	52	0	0	0	0	1	1	0,3	100
		2	53	53	0	0	9	9	3	3	3,0	100
	UDP_STREAM	1	52	53	0	0	0	8	1	1	2,3	100
			53		0		18		1			
			53		0		7		0			
		2	53	53	0	0	9	9	3	3	3,0	100
			54		0		3		5			
			52		0		16		2			
	TCP_STREAM	1	73	67	0	0	4	17	1	1	4,5	100
			64		0		17		3			
			64		0		29		0			
		2	52	54	0	0	25	13	0	3	4,0	100
			55		0		14		4			
			55		0		1		5			

Tabelle B.12: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms	ms	%
2 calls	TCP_CRR	1	55		0		0		0		0	100
			53	54	0	0	0	0	1	0		
			53		0		0		0			
		2	53		0		0		0		0,3	100
			54	53	0	0	0	0	1	1		
			53		0		0		1			
2 calls + 1 fax	UDP_STREAM	1	53	53	0	0	24	24	1	1	6,3	100
		2	52	52	0	0	0	0	0	0	0	100

Tabelle B.13: Meßwerte Netperf - CBWFQ (2 Klassen: Voice - TD, Data - TD)

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$
–		144,30		120,06		9,40		297,56		246,72		18,81
1 call	69,70		50,68		4,90		199,25		165,21		15,71	
	69,64	69,66	52,08	50,62	5,00	4,93	199,15	199,22	166,03	165,46	15,90	15,81
	69,64		49,11		4,90		199,25		165,15		15,81	
1 call + 1 fax	60,64	60,64					147,32	147,32				
2 calls							132,79		104,00		9,80	
							133,55	133,30	104,55	103,41	9,81	9,80
							133,56		101,69		9,80	
2 calls + 1 fax							124,53	124,53				

Tabelle B.14: Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - TD; Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
1 call	–	54	54	0	0	0	0	0	0	0	100
	UDP_STREAM	64	64	128	118	58	57	144	144	109,3	50
		63		126		50		143			
		[184]		99		64		[12]			
	TCP_STREAM	59	55	132	111	127	127	250	259	152,0	50
		56		112		121		253			
		51		89		132		274			
	TCP_CRR	53	53	0	0	0	0	1	0	0	100
		52		0		0		0			
		55		0		0		0			
1 call + 1 fax	UDP_STREAM	64	64	406	406	62	62	39	39	228,3	50

Tabelle B.15: Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD,
Data - TD; Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
1 call	–		52	52	0	0	0	0	0	0	0	100
	UDP_STREAM		52	53	0	0	0	0	0	1	0,3	100
			55		0		0		1			
			52		0		0		1			
	TCP_STREAM		52	54	0	0	25	12	1	1	3,3	100
			57		0		0		1			
			53		0		11		0			
	TCP_CRR		52	53	0	0	0	0	1	1	0,3	100
			53		0		0		1			
			55		0		0		1			
1 call + 1 fax	UDP_STREAM		53	53	285	285	174	174	183	183	231,8	50

Tabelle B.15: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
2 calls	–	1	53	53	0	0	0	0	1	1	0,3	100
		2	53	53	0	0	0	0	1	1	0,3	100
	UDP_STREAM	1	45	63	188	168	19	17	70	50	100,8	80
			80		156		17		9			
			64		159		15		71			
		2	64	58	155	159	32	27	69	71	104,0	80
			64		147		32		72			
			45		174		18		72			
	TCP_STREAM	1	64	64	100	107	43	53	140	127	98,5	80
			64		128		56		139			
			64		93		60		101			
		2	64	64	142	133	21	37	138	144	111,8	80
			[108] [124]		137 121		45 45		141 152			

Tabelle B.15: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms	ms	%
2 calls	TCP_CRR	1	52		0		0		1			
			55	54	0	0	0	0	0	0	0	100
			55		0		0		0			
		2	52		0		0		0			
			53	53	0	0	0	0	0	0	0	100
			53		0		0		0			
2 calls + 1 fax	UDP_STREAM	1	52	52	787	787	27	27	54	54	413,8	40
		2	45	45	773	773	26	26	58	58	407,5	40

Tabelle B.16: Meßwerte Netperf - CBWFQ (2 Klassen: Voice - TD, Data - WRED)

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\varnothing $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\varnothing $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\varnothing $\frac{Trans.}{s}$	$10^3 \frac{bit}{s}$	\varnothing $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\varnothing $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\varnothing $\frac{Trans.}{s}$
–		144,30		120,06		9,40		297,56		246,72		18,81
1 call	[99,19]		59,08		3,80		199,23		165,09		15,20	
	71,30	71,32	51,41	56,55	3,90	3,87	199,13	199,46	164,98	165,32	14,20	14,57
	71,33		59,16		3,90		200,02		165,89		14,30	
1 call + 1 fax	77,03	77,03					154,86	154,86				
2 calls							145,93		117,90		8,20	
							145,85	145,91	117,84	117,98	8,21	8,21
							145,95		118,20		8,21	
2 calls + 1 fax							145,04	145,04				

Tabelle B.17: Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD, Data - WRED; Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	⊘ ms	ms	%
1 call	–	53	53	0	0	0	0	1	1	0,3	100
	UDP_STREAM	64 [164] 59	62	[450] 223 223	223	55 58 58	57	121 92 132	115	154,5	50
	TCP_STREAM	55 54 [126]	55	239 238 191	223	75 86 74	78	143 160 [41]	152	169,0	50
	TCP_CRR	52 54 52	53	0 0 0	0	0 0 0	0	0 0 0	0	0	100
1 call + 1 fax	UDP_STREAM	55	55	297	297	142	142	163	163	224,8	50

Tabelle B.18: Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - TD,
Data - WRED; Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms		
1 call	–		52	52	0	0	0	0	1	1	0,3	100
	UDP_STREAM		52	53	0	0	0	0	1	0	0	100
			52		0		0		0			
			54		0		0		0			
	TCP_STREAM		53	54	0	0	0	0	1	1	0,3	100
			55		0		0		0			
			53		0		[122]		1			
	TCP_CRR		53	54	0	0	0	0	1	1	0,3	100
			55		0		0		1			
			55		0		0		1			
1 call + 1 fax	UDP_STREAM		46	46	271	271	3	3	1	1	136,5	80

Tabelle B.18: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
2 calls	–	1	52	52	0	0	0	0	1	1	0,3	100
		2	52	52	[60843]	– (0)	0	0	2	2	0,5	100
	UDP_STREAM	1	[176]		202		33		53		160,0	80
			64	76	306	273	31	37	67	57		
			88		310		46		52			
		2	[166]		259		14		51		113,0	80
			64	73	144	182	35	31	70	57		
			81		142		44		51			
	TCP_STREAM	1	45		244		21		74		152,8	80
			85	65	258	256	34	26	70	73		
			64		265		22		75			
		2	47		156		27		76		97,3	80
			84	65	132	142	43	30	75	75		
			64		137		19		74			

Tabelle B.18: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms	ms	%
2 calls	TCP_CRR	1	53		0		0		0		0,3	100
			53	53	0	0	0	0	1	1		
			53		0		0		1			
		2	53		0		0		0		0	100
			52	53	0	0	0	0	0	0		
			53		0		0		1			
2 calls + 1 fax	UDP_STREAM	1	69	69	531	531	187	187	134	134	345,8	40
		2	64	64	518	518	146	146	117	117	324,8	50

Tabelle B.19: Meßwerte Netperf - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED)

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$
–		144,30		120,06		9,40		297,56		246,72		18,81
1 call	47,52		38,50		3,50		199,19		164,87		13,20	
	47,55	47,26	39,00	38,83	3,50	3,50	200,05	199,47	165,44	165,29	13,50	13,10
	46,71		39,00		3,50		199,17		165,55		12,60	
1 call + 1 fax	33,60	33,60					146,38	146,38				
2 calls							100,79		83,32		7,50	
							102,42	101,34	83,39	83,64	7,10	7,20
							100,82		84,21		7,01	
2 calls + 1 fax							56,26	56,26				

Tabelle B.20: Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - LLQ, Data - WRED; Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
1 call	–	53	53	0	0	0	0	1	1	0,3	100
	UDP_STREAM	83	84	0	0	3	4	1	1	1,3	100
		85		0		2		0			
		84		0		6		1			
	TCP_STREAM	107	116	0	0	2	3	1	1	1,0	100
		110		0		3		1			
		131		0		5		1			
	TCP_CRR	52	52	0	0	0	0	0	1	0,3	100
		53		0		0		1			
		52		0		0		1			
1 call + 1 fax	UDP_STREAM	55	55	391	391	42	42	58	58	220,5	80

Tabelle B.21: Meßwerte Call-Analyse - CBWFQ (2 Klassen: Voice - LLQ,
Data - WRED; Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
1 call	–		52	52	0	0	0	0	1	1	0,3	100
	UDP_STREAM		52	54	0	0	0	0	1	1	0,3	100
			57		0		0		1			
			54		0		0		0			
	TCP_STREAM		52	56	0	0	12	4	0	0	1,0	100
			58		0		0		0			
			58		0		0		0			
	TCP_CRR		53	53	0	0	0	0	1	0	0	100
			52		0		0		0			
			53		0		0		0			
1 call + 1 fax	UDP_STREAM		53	53	0	0	0	0	1	1	0,3	100

Tabelle B.21: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
2 calls	–	1	52	52	0	0	0	0	0	0	0	100
		2	55	55	0	0	0	0	1	1	0,3	100
	UDP_STREAM	1	52	52	0	0	0	0	0	0	0	100
			52		0		0		0			
			52		0		0		0			
		2	52	53	0	0	0	0	0	0	0	100
			54		0		0		0			
			53		0		0		1			
	TCP_STREAM	1	53	59	0	0	46	38	0	1	9,8	100
			52		[60699]		40		2			
			72		0		28		0			
		2	53	55	0	0	15	21	0	0	5,3	100
			58		0		0		0			
			53		0		49		0			

Tabelle B.21: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms		
2 calls	TCP_CRR	1	52		0		0		0			
			52	52	0	0	0	0	0	0	0	100
			53		0		0		0			
		2	53		0		0		1			
			53	53	0	0	0	0	0	0	0	100
			53		0		0		0			
2 calls + 1 fax	UDP_STREAM	1	53	53	0	0	1	1	0	0	0,3	100
		2	53	53	[6530]	– (0)	9	9	0	0	2,3	100

Tabelle B.22: Meßwerte Netperf - PQ

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\varnothing $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\varnothing $\frac{\text{Trans.}}{\text{s}}$
—		144,30		120,06		9,40		297,56		246,72		18,81
1 call	46,74		38,78		3,50		199,10		164,88		13,31	
	46,74	47,01	38,71	38,67	3,50	3,50	200,02	199,69	165,66	165,36	12,71	12,84
	47,55		38,52		3,50		199,94		165,53		12,50	
1 call + 1 fax	32,78	32,78					146,66	146,66				
2 calls							100,83		83,17		7,41	
							100,80	100,80	83,47	83,31	7,00	7,14
							100,77		83,29		7,00	
2 calls + 1 fax							59,27	59,27				

Tabelle B.23: Meßwerte Call-Analyse - PQ (Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
1 call	–	53	53	0	0	0	0	1	1	0,3	100
	UDP_STREAM	46	70	0	0	5	5	1	0	1,3	100
		88		0		3		0			
		76		0		7		0			
	TCP_STREAM	106	112	0	0	3	3	0	0	0,8	100
		114		0		2		0			
		116		0		4		0			
	TCP_CRR	53	53	0	0	0	0	1	1	0,3	100
		53		0		0		1			
		53		0		0		1			
1 call + 1 fax	UDP_STREAM	46	46	425	425	31	31	79	79	240,0	60

Tabelle B.24: Meßwerte Call-Analyse - PQ (Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
1 call	–		53	53	0	0	0	0	0	0	0	100
	UDP_STREAM		53	54	0	0	0	0	0	1	0,3	100
			55		0		0		0			
			54		[6264]		0		2			
	TCP_STREAM		52	58	0	0	37	12	0	0	3,0	100
			64		0		0		0			
			58		0		0		0			
	TCP_CRR		53	52	0	0	0	0	0	0	0	100
			52		0		0		0			
			52		0		0		0			
1 call + 1 fax	UDP_STREAM		53	53	0	0	0	0	1	1	0,3	100

Tabelle B.24: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
2 calls	–	1	52	52	0	0	0	0	0	0	0	100
		2	53	53	0	0	0	0	1	1	0,3	100
	UDP_STREAM	1	52	52	0	0	0	4	0	0	1,0	100
			52		0		0		0			
			53		0		12		1			
		2	52	54	0	0	0	0	0	0	0	100
			57		0		0		0			
			53		0		0		0			
	TCP_STREAM	1	53	60	0	0	4	62	0	0	15,5	100
			74		0		116		1			
			53		0		66		0			
		2	52	59	0	0	21	35	0	0	8,8	100
			72		0		32		0			
			52		0		53		0			

Tabelle B.24: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	ms	ms	ms	ms	ms	ms	ms	ms	%
2 calls	TCP_CRR	1	52		0		0		0		0,3	100
			53	53	0	0	0	0	1	1		
			53		0		0		1			
		2	52		0		0		0		0	100
			53	52	0	0	0	0	0	0		
			52		0		0		1			
2 calls + 1 fax	UDP_STREAM	1	52	52	0	0	0	0	0	0	0	100
		2	53	53	0	0	0	0	1	1	0,3	100

Tabelle B.25: Meßwerte Netperf - PQ + CAR

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{\text{bit}}{\text{s}}$	\emptyset $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\emptyset $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\emptyset $\frac{\text{Trans.}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\emptyset $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\emptyset $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\emptyset $\frac{\text{Trans.}}{\text{s}}$
—		144,30		120,06		9,40		297,56		246,72		18,81
1 call	46,73		38,41		3,50		199,09		164,97		13,00	
	47,53	46,99	38,49	38,57	3,50	3,53	199,10	199,13	164,88	164,90	13,11	12,97
	46,70		38,81		3,60		199,19		164,84		12,81	
1 call + 1 fax	24,58	24,58					160,57	160,57				
2 calls							182,00		151,00		11,91	
							181,91	181,94	151,01	150,70	11,71	11,77
							181,91		150,09		11,70	
2 calls + 1 fax							147,87	147,87				

Tabelle B.26: Meßwerte Call-Analyse - PQ + CAR (Clock Rate: 125000 bps)

Test	Netperf-Macro	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
		ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	∅ ms	ms	%
1 call	–	53	53	0	0	0	0	0	0	0	100
	UDP_STREAM	87	81	0	0	7	18	0	0	4,5	100
		85		0		3		0			
		71		0		45		0			
	TCP_STREAM	124	118	0	0	7	4	1	1	1,3	100
		109		0		3		0			
		121		0		3		1			
	TCP_CRR	54	54	0	0	0	0	1	1	0,3	100
		55		0		0		1			
		53		0		0		0			
1 call + 1 fax	UDP_STREAM	64	64	476	476	10	10	23	23	246,3	70

Tabelle B.27: Meßwerte Call-Analyse - PQ + CAR (Clock Rate: 256000 bps)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
1 call	–		52	52	0	0	0	0	0	0	0	100
	UDP_STREAM		53		0		0		0			100
			52	52	0	0	0	0	0	0	0	
			52		0		0		0			
	TCP_STREAM		53		0		17		1			100
			53	53	0	0	0	12	0	0	3,0	
			52		0		18		0			
	TCP_CRR		52		0		0		0			100
			53	52	0	0	0	0	1	0	0	
			52		0		0		0			
1 call + 1 fax	UDP_STREAM		45	45	362	0	0	0	1	1	181,3	80

Tabelle B.27: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
2 calls	–	1	45	45	[1087]	– (579)	0	0	0	0	289,5	50
		2	57	57	[70]	– (578)	0	0	0	0	289,0	50
	UDP_STREAM	1	45	49	411	420	0	0	1	1	210,3	50
			45		429		0		0			
			56		[1022]		0		1			
		2	48	50	440	488	0	0	0	0	244,0	50
			47		536		0		0			
			54		[38]		0		0			
	TCP_STREAM	1	72	68	321	340	0	1	1	0	170,3	50
			67		[839]		1		0			
			64		358		1		0			
		2	61	62	528	445	0	0	0	0	222,5	50
			58		[69]		0		0			
			68		362		0		0			

Tabelle B.27: (Fortsetzung)

Test	Netperf-Macro	Call	ReceiveDelay		LostPackets		EarlyPackets		LatePackets		gewichteter Mittelwert	Verständlichkeit
			ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	⌀ ms	ms	%
2 calls	TCP_CRR	1	45		[658]		0		0		162,3	50
			45	45	349	324	0	0	1	1		
			45		299		0		1			
		2	55		[58]		0		1		193,3	50
			45	48	452	386	0	0	0	1		
			45		320		0		1			
2 calls + 1 fax	UDP_STREAM	1	45	45	537	537	157	157	112	112	335,8	40
		2	56	56	482	482	134	134	87	87	296,3	40

Tabelle B.28: Meßwerte Netperf - PQ + CRTP

Test	Clock Rate 125000 bps						Clock Rate 256000 bps					
	UDP_STREAM		TCP_STREAM		TCP_CRR		UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\emptyset $\frac{Trans.}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\emptyset $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\emptyset $\frac{Trans.}{s}$
—		144,30		120,06		9,40		297,56		246,72		18,81
1 call	64,75		53,33		4,90		217,29		180,45		14,11	
	64,73	65,01	53,70	53,55	4,60	4,70	217,23	217,51	180,51	180,36	14,01	13,78
	65,55		53,61		4,60		218,00		180,12		13,21	
1 call + 1 fax	53,80	53,80					178,89	178,89				
2 calls							136,88		113,49		9,91	
							136,85	136,89	113,48	113,43	9,80	9,84
							136,94		113,32		9,80	
2 calls + 1 fax							108,68	108,68				

Test	UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\bigcirc $\frac{Trans.}{s}$
unbelastet		297,56		246,72		18,81
1 call	217,29	217,51	180,45	180,36	14,11	13,78
	217,23		180,51		14,01	
	218,00		180,12		13,21	
2 calls	136,88	136,89	113,49	113,43	9,91	9,84
	136,85		113,48		9,80	
	136,94		113,32		9,80	
1 fax	279,44	279,44				

Tabelle B.29: Meßwerte Netperf - Codec G.711 A law

Netperf-Macro	ReceiveDelay		Verständlichkeit
	ms	\bigcirc ms	
UDP_STREAM	53	53	100
	53		
	54		
TCP_STREAM	52	56	100
	60		
	56		
TCP_CRR	53	53	100
	53		
	52		

Tabelle B.30: Meßwerte Call-Analyse - Codec G.711 A law

Test	UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\varnothing $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\varnothing $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\varnothing $\frac{Trans.}{s}$
unbelastet		297,56		246,72		18,81
1 call	286,84		237,76		18,51	
	286,90	286,58	237,57	237,65	18,51	18,51
	286,01		237,61		18,51	
2 call	275,36		228,72		17,91	
	275,39	275,37	228,74	228,65	17,91	17,91
	275,35		228,48		17,91	

Tabelle B.31: Meßwerte Netperf - Codec G.723.1 5,3k

Netperf-Macro	ReceiveDelay		Verständlichkeit
	ms	\varnothing ms	
UDP_STREAM	130		
	130	130	70
	130		
TCP_STREAM	130		
	131	130	70
	130		
TCP_CRR	129		
	130	130	70
	130		

Tabelle B.32: Meßwerte Call-Analyse - Codec G.723.1 5,3k

Test	UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{\text{bit}}{\text{s}}$	\bigcirc $10^3 \frac{\text{bit}}{\text{s}}$	$10^3 \frac{\text{bit}}{\text{s}}$	\bigcirc $10^3 \frac{\text{bit}}{\text{s}}$	$\frac{\text{Trans.}}{\text{s}}$	\bigcirc $\frac{\text{Trans.}}{\text{s}}$
unbelastet		297,56		246,72		18,81
1 call	286,90		237,39		18,51	
	286,84	286,58	237,71	237,61	18,51	18,51
	286,00		237,74		18,51	
2 call	275,36		227,91		17,91	
	276,18	275,64	228,73	228,44	17,91	17,91
	275,38		228,68		17,91	

Tabelle B.33: Meßwerte Netperf - Codec G.723.1 6,3k

Netperf-Macro	ReceiveDelay		Verständlichkeit
	ms	\bigcirc ms	
UDP_STREAM	131		
	131	131	70
	130		
TCP_STREAM	130		
	130	130	70
	130		
TCP_CRR	128		
	130	129	70
	130		

Tabelle B.34: Meßwerte Call-Analyse - Codec G.723.1 6,3k

Test	UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\bigcirc $\frac{Trans.}{s}$
unbelastet		297,56		246,72		18,81
1 call	273,77		226,56		17,81	
	272,91	273,46	226,82	226,72	17,81	17,81
	273,71		226,77		17,81	
2 calls	249,17		206,87		16,61	
	249,14	249,15	206,79	206,84	16,61	16,61
	249,14		206,85		16,61	

Tabelle B.35: Meßwerte Netperf - Codec G.726

Netperf-Macro	ReceiveDelay		Verständlichkeit
	ms	\bigcirc ms	
UDP_STREAM	52		
	52	52	90
	53		
TCP_STREAM	52		
	56	54	90
	54		
TCP_CRR	53		
	53	53	90
	52		

Tabelle B.36: Meßwerte Call-Analyse - Codec G.726

Test	UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\bigcirc $\frac{Trans.}{s}$
unbelastet		297,56		246,72		18,81
1 call	273,85		226,92		17,81	
	273,71	273,76	227,16	227,13	17,81	17,81
	273,72		227,31		17,81	
2 calls	249,96		207,49		16,51	
	250,02	249,99	207,53	207,42	16,61	16,58
	249,99		207,24		16,61	

Tabelle B.37: Meßwerte Netperf - Codec G.728

Netperf-Macro	ReceiveDelay		Verständlichkeit
	ms	\bigcirc ms	
UDP_STREAM	53		
	53	53	100
	53		
TCP_STREAM	52		
	53	53	100
	55		
TCP_CRR	52		
	53	52	100
	52		

Tabelle B.38: Meßwerte Call-Analyse - Codec G.728

Test	UDP_STREAM		TCP_STREAM		TCP_CRR	
	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$10^3 \frac{bit}{s}$	\bigcirc $10^3 \frac{bit}{s}$	$\frac{Trans.}{s}$	\bigcirc $\frac{Trans.}{s}$
unbelastet		297,56		246,72		18,81
1 call	282,86		234,94		18,31	
	283,55	283,05	234,87	234,87	18,31	18,31
	282,73		234,81		18,31	
2 calls	268,82		223,12		17,31	
	268,81	268,81	223,13	223,00	17,31	17,31
	268,80		222,75		17,31	

Tabelle B.39: Meßwerte Netperf - Codec G.729

Netperf-Macro	ReceiveDelay		Verständlichkeit
	ms	\bigcirc ms	
UDP_STREAM	55		
	53	54	100
	53		
TCP_STREAM	54		
	56	55	100
	55		
TCP_CRR	53		
	52	52	100
	52		

Tabelle B.40: Meßwerte Call-Analyse - Codec G.729

Tabelle B.41: Übersicht Faxübertragungen

Versuch	Test	Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Fax- nummer	Übertragungs- dauer/ Dauer bis Abbruch s	Bemerkung
FIFO	1 fax	125	1	77	kein Netperf
	1 call + 1 fax	125	Abbruch	61	
		256	2	78	
	2 calls + 1 fax	256	Abbruch	73	
WFQ	1 call + 1 fax	125	Abbruch	85	
		256	3	77	\Rightarrow ¹
		256	4	77	\Leftarrow ²
	2 calls + 1 fax	256	Abbruch	107	\Rightarrow
		256	5	77	\Leftarrow
WFQ + IP RTP P.	1 call + 1 fax	125	Abbruch	61	
CBWFQ Voice: TD	1 call + 1 fax	125	Abbruch	61	
		256	6	77	
	2 calls + 1 fax	256	7	77	
CBWFQ Voice: TD Data: TD	1 call + 1 fax	125	Abbruch	72	
		256	8	132	erneute Anforderung
	2 calls + 1 fax	256	Abbruch	65	
CBWFQ Voice: TD Data: WRED	1 call + 1 fax	125	Abbruch	61	
		256	9	122	erneute Anforderung
	2 calls + 1 fax	256	Abbruch	94	
CBWFQ Voice: LLQ Data: WRED	1 call + 1 fax	125	Abbruch	60	
		256	10	77	
	2 calls + 1 fax	256	11	77	

¹Fax-Call und Netperf (unidirektional) in selbe Richtung \Rightarrow verfügbare Bandbreite zwischen beiden Flows geteilt

²Fax-Call und Netperf (unidirektional) in entgegengesetzter Richtung \Leftarrow verfügbare Bandbreite steht beiden Flows exklusiv zur Verfügung

Tabelle B.41: (Fortsetzung)

Versuch	Test	Clock Rate $10^3 \frac{\text{bit}}{\text{s}}$	Fax- nummer	Übertragungs- dauer/ Dauer bis Abbruch s	Bemerkung
PQ	1 call + 1 fax	125	Abbruch	61	
		256	12	77	
	2 calls + 1 fax	256	13	77	
PQ + CAR	1 call + 1 fax	125	Abbruch	94	
		256	14	78	
	2 calls + 1 fax	256	Abbruch	58	
PQ + CRTP	1 call + 1 fax	125	Abbruch	60	
		256	15	77	
	2 calls + 1 fax	256	16	77	
Codec G.711 A law	1 fax	256	17	77	@ 9,6 kbaud
Codec G.723.1 5,3k	1 fax	256	18	116	@ 4,8 kbaud
Codec G.723.1 6,3k	1 fax	256	19	116	@ 4,8 kbaud
Codec G.726 16k	1 fax	256	20	77	@ 9,6 kbaud
Codec G.728	1 fax	256	21	77	@ 9,6 kbaud
Codec G.729	1 fax	256	22	94	@ 7,2 kbaud

Literaturverzeichnis

- [Cis00] CISCO SYSTEMS INC.: *Cisco IOS Quality of Service Solutions Configuration Guide*, 2000.
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm
- [Hew96] HEWLETT-PACKARD COMPANY, INFORMATION NETWORKS DIVISION: *Netperf: A Network Performance Benchmark*, 1996.
<http://www.netperf.org>
- [Höy99] HÖYNCK, MICHAEL: *Testsignaldesign und Parametersatzbestimmung zum Zweck der maximalen Unterscheidung ausgewählter Sprachcodec-Kaskaden*. Studienarbeit, Technische Universität Berlin, Institut für Nachrichtentechnik und Theoretische Elektrotechnik, 1999.
<http://www-ft.ee.tu-berlin.de/~liebchen/sada.html>
- [ITG97] ITG FACHGRUPPE 5.2.1: *Begriffe der Nachrichtenverkehrstheorie (Empfehlung 5.2-03)*, 1997.
http://www.comnets.rwth-aachen.de/~itg/FG521/home_ger.html
- [MEH00] MATHY, LAURENT, CHRISTOPHER EDWARDS UND DAVID HUTCHINSON: *The Internet: A Global Telecommunications Solution*. IEEE Network, July/ August, 2000.
- [MM98] MINOLI, DANIEL UND EMMA MINOLI: *Delivering Voice over IP Networks*, 2000.
- [PF94] PAXSON, VERN UND SALLY FLOYD: *Wide-Area Traffic: The Failure of Poisson Modeling*. ACM Computer Communication Review, 24(4): 257-268, October 1994.

[RFC 0768] POSTEL, J.: *User Datagram Protocol*. Request for Comments: 0768, STD: 6, August 1980.

<http://www.ietf.org/rfc/rfc0768.txt>

[RFC 0791] POSTEL, J.: *Internet Protocol*. Request for Comments: 0791, STD: 5, September 1981.

<http://www.ietf.org/rfc/rfc0791.txt>

[RFC 0793] POSTEL, J.: *Transmission Control Protocol*. Request for Comments: 0793, STD: 7, September 1981.

<http://www.ietf.org/rfc/rfc0793.txt>

[RFC 1104] BRAUN, HANS-WERNER: *Models of Policy Based Routing*. Request for Comments: 1104, Juni 1989.

<http://www.ietf.org/rfc/rfc1104.txt>

[RFC 1144] JACOBSON, VAN: *Compressing TCP/IP Headers for Low-Speed Serial Links*. Request for Comments: 1144, Februar 1990.

<http://www.ietf.org/rfc/rfc1144.txt>

[RFC 1242] BRADNER, SCOTT: *Benchmarking Terminology for Network Interconnection Devices*. Request for Comments: 1242, Juli 1991.

<http://www.ietf.org/rfc/rfc1242.txt>

[RFC 1633] BRADEN, ROBERT, DAVID CLARK, SCOTT SHENKER: *Integrated Services in the Internet Architecture: an Overview*. Request for Comments: 1633, Juni 1994.

<http://www.ietf.org/rfc/rfc1633.txt>

[RFC 1661] SIMPSON, WILLIAM A.: *The Point-to-Point Protocol (PPP)*. Request for Comments: 1661, STD: 51, Juli 1994.

<http://www.ietf.org/rfc/rfc1661.txt>

[RFC 1717] SKLOWER, KEITH, BRIAN LLOYD, GLENN MCGREGOR, DAVE CARR: *The PPP Multilink Protocol (MP)*. Request for Comments: 1717, November 1994.

<http://www.ietf.org/rfc/rfc1717.txt>

- [RFC 1889] SCHULZRINNE, HENNING, STEPHEN L. CASNER, RON FREDERICK, VAN JACOBSON: *RTP: A Transport Protocol for Real-Time Applications*. Request for Comments: 1889, Januar 1996.

<http://www.ietf.org/rfc/rfc1889.txt>

- [RFC 1990] SKLOWER, KEITH, BRIAN LLOYD, GLENN MCGREGOR, DAVE CARR, TOM CORADETTI: *The PPP Multilink Protocol (MP)*. Request for Comments: 1990, August 1996.

<http://www.ietf.org/rfc/rfc1990.txt>

- [RFC 2205] BRADEN, ROBERT, LIXIA ZHANG, STEVE BERSON, SHAI HERZOG, SUGIH JAMIN: *Resource ReSerVation Protocol (RSVP)*. Request for Comments: 2205, September 1997.

<http://www.ietf.org/rfc/rfc2205.txt>

- [RFC 2309] BRADEN, ROBERT, DAVID D. CLARK, JON CROWCROFT, BRUCE DAVIE, STEVE DEERING, DEBORAH ESTRIN, SALLY FLOYD, VAN JACOBSON, GREG MINSHALL, CRAIG PARTRIDGE, LARRY PETERSON, K. K. RAMAKRISHNAN, SCOTT SHENKER, JOHN WROCLAWSKI, LIXIA ZHANG: *Recommendations on Queue Management and Congestion Avoidance in the Internet*. Request for Comments: 2309, April 1998.

<http://www.ietf.org/rfc/rfc2309.txt>

- [RFC 2330] PAXSON, VERN, GUY ALMES, JAMSHID MAHDAVI, MATT MATHIS: *Framework for IP Performance Metrics*. Request for Comments: 2330, Mai 1998.

<http://www.ietf.org/rfc/rfc2330.txt>

- [RFC 2460] DEERING, STEPHEN E., ROBERT M. HINDEN: *Internet Protocol, Version 6 (IPv6) Specification*. Request for Comments: 2460, Dezember 1998.

<http://www.ietf.org/rfc/rfc2460.txt>

- [RFC 2474] NICHOLS, KATHLEEN, STEVEN BLAKE, FRED BAKER, DAVID L. BLACK: *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. Request for Comments: 2474, Dezember 1998.

<http://www.ietf.org/rfc/rfc2474.txt>

- [RFC 2475] BLAKE, STEVEN, DAVID L. BLACK, MARK A. CARLSON, ELWYN DAVIES, ZHENG WANG, WALTER WEISS: *An Architecture for Differentiated Services*. Request for Comments: 2475, Dezember 1998.
<http://www.ietf.org/rfc/rfc2475.txt>
- [RFC 2507] DEGERMARK, MIKAEL, BJORN NORDGREN, STEPHEN PINK: *IP Header Compression*. Request for Comments: 2507, Februar 1999.
<http://www.ietf.org/rfc/rfc2507.txt>
- [RFC 2508] CASNER, STEPHEN L., VAN JACOBSON: *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links*. Request for Comments: 2508, Februar 1999.
<http://www.ietf.org/rfc/rfc2508.txt>
- [RFC 2509] ENGAN, MATHIAS, STEPHEN L. CASNER, CARSTEN BORMANN: *IP Header Compression over PPP*. Request for Comments: 2509, Februar 1999.
<http://www.ietf.org/rfc/rfc2509.txt>
- [RFC 2544] BRADNER, SCOTT, J. MCQUAID: *Benchmarking Methodology for Network Interconnect Devices*. Request for Comments: 2544, März 1999.
<http://www.ietf.org/rfc/rfc2544.txt>
- [RFC 2581] ALLMAN, MARK, VERN PAXSON, W. RICHARD STEVENS: *TCP Congestion Control*. Request for Comments: 2581, April 1999.
<http://www.ietf.org/rfc/rfc2581.txt>
- [RFC 2686] BORMANN, CARSTEN: *The Multi-Class Extension to Multi-Link PPP*. Request for Comments: 2686, September 1999.
<http://www.ietf.org/rfc/rfc2686.txt>
- [RFC 2700] REYNOLDS, JOYCE K., ROBERT BRADEN: *Internet Official Protocol Standards*. Request for Comments: 2700, STD: 1, August 2000.
<http://www.ietf.org/rfc/rfc2700.txt>
- [Rob94] ROBERTAZZI, THOMAS G.: *Computer Networks and Systems*. Queueing Theory and Performance Evaluation, Zweite Ausgabe, 1994.

- [Roc00] ROCKMANN, RAINER: *Nachrichtenverkehrstheorie*. Vorlesungsskript, Universität Rostock, Institut für Nachrichtentechnik und Informationselektronik, 2000.
- [Wal00] WALLSTABE, JÖRN: *Synchrone Dienste in IP-Netzen*. Diplomarbeit, Universität Rostock, Institut für Nachrichtentechnik und Informationselektronik, 2000.