Size-Based Scheduling to Improve the User Experience in Cellular Networks

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von

Magnus Christian Proebster

aus Filderstadt

Hauptberichter:	Prof. Dr.-Ing. Andreas Kirstädter
Mitberichter:	Prof. Dr.-Ing. Wolfgang Kellerer
Tag der mündlichen Prüfung:	6. Juli 2016

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2016

Abstract

Cellular access networks provide a practically universal wireless connection to the Internet. This enables manifold convenient services that attract more and more users. Consequently, the enormous traffic growth in cellular networks may increasingly lead to overload situations where traffic demand exceeds the capacity of the network. This strongly impairs user experience, especially for interactive web services and media streaming applications. Dissatisfied users may quit the operator's service and therefore diminish revenues. However, the traditional solution of infrastructure build-up will get very expensive for the predicted massive traffic growth.

Additionally, there are other approaches to mitigate the service degradation caused by overload, which influence the allocation of radio resources. Conventional service differentiation together with a subsequent prioritization of traffic classes according to their Quality of Service (QoS) requirements can improve the users' overall Quality of Experience (QoE). However, this approach requires a complex framework and computational resources for signaling, classification and prioritization. A much less complex way to achieve a graceful service degradation under overload is the application of the Shortest Remaining Processing Time first (SRPT) principle to radio resource allocation. Algorithms following this principle, called *Shortest-First* schedulers in the following, are known to minimize transmission durations and thus support an unimpaired QoE for latency-critical services. Furthermore, only the fraction of traffic which exceeds the cell capacity notices the overload at all.

On the down-side, existing proposals of such *Shortest-First* schedulers decrease the spectral efficiency compared to traditional opportunistic schedulers. This limits the positive impact and applicability to cellular networks as radio spectrum is a scarce and expensive resource. Furthermore, the known *Shortest-First* schedulers do not work reasonably for streaming services. These services would get a strong scheduling penalty due to their large object sizes. Buffered video streaming, which is an essential Internet application and contributes a major part of the total traffic volume, would thus require a dedicated handling by the scheduler.

This thesis contains two major contributions that provide a solution for both problems. First, an enhancement for *Shortest-First* schedulers – an additional parameter called length exponent – is proposed that allows to tune the schedulers towards opportunistic algorithms. Second, the introduction of a mapping function which translates the client-side buffer level of a video streaming application into an equivalent object size makes it possible to seamlessly and beneficially integrate this traffic class into the scheduling algorithm without further changes. The parameters of the proposed function allow to define the priority of video traffic and an Implicit Admission Control (IAC) which helps to improve the QoE of ongoing videos in case of overload.

To provide a background for the discussed topic, this thesis surveys literature on Internet traffic characteristics and developments. The concepts of QoE and service differentiation are presented, traffic classes are defined and the relevant application layer objects – called transactions – in combination with utility functions for QoE assessment are introduced. Concerning cellular networks, the thesis contains an introduction of the basics and gives an overview of 3GPP Long Term Evolution (LTE), which will serve as a basis for performance evaluation. Furthermore, multiple access methods, the general tasks of resource allocation as well as state-of-the-art scheduling concepts are introduced and classified, including a discussion of proposals from literature, which apply the SRPT principle in cellular networks.

Against this background, the thesis motivates the necessity to combine size-based and opportunistic scheduling principles. This combination is achieved by the length exponent γ which controls the influence of the object size on the scheduling priority of a transaction. It allows to seamlessly tune the scheduling behavior between the original *Shortest-First* scheduler and one that maximizes the instantaneous channel capacity (*Max C/I*). An additional advantage of this approach is that a configuration between these extremes exploits a synergy of both schedulers. This means that a major improvement in spectral efficiency can be achieved at a small reduction in the desired latency performance of *Shortest-First*. Therefore, the proposed solution offers both, increased flexibility and increased performance.

The second contribution allows to integrate buffered video streaming services into *Shortest-First* schedulers. For this, it incorporates information about the amount of video material in the buffer of the user equipment (i. e. the client) into the scheduling decision. Principally, a video with low buffer level is in danger of suffering an interruption in playback and should therefore be prioritized by the scheduler. On the other hand, a high buffer level allows the video player to run for some time without any network transmissions. This behavior and the aforementioned IAC are designed into the mapping function for the equivalent object size, which allows to greatly improve video QoE and to adjust video priority.

In order to confirm and quantify the benefits of the proposed solutions, an extensive performance evaluation was conducted. It is based on system-level simulations that consider the traffic within a single cell of a mobile network as well as interference form the surrounding base stations. To this end, models of the radio channel, the link layer, and user mobility were developed, which reproduce the behavior of LTE systems on an abstract level. An important part are realistic traffic scenarios coupled with the channel variations of moving users. The resulting traffic characteristics (e. g. volume, burstiness) interact with the scheduling behavior over time and have a great influence on the performance metrics.

Studies covering a range of scenarios while investigating the influence of the proposed parameters were performed and evaluated mainly with respect to cell throughput, transaction durations, and utility. This allows to compare the proposed solution against conventional schedulers and to point out the benefits and effectiveness of the contributions. The results show that the length exponent indeed extends the capacity region of the scheduler and gives the possibility to adjust the scheduler's operation point. Furthermore, the proposed mechanism shows a great improvement of the QoE for buffered streaming services while maintaining the flexibility in resource allocation and thereby mitigating the penalty for other types of traffic.

Kurzfassung

Zelluläre Zugangsnetze stellen eine praktisch allgegenwärtige drahtlose Verbindung zum Internet bereit. Dies ermöglicht vielfältige nützliche Dienste, die mehr und mehr Nutzer für sich gewinnen können. Daraus folgt ein enormes Wachstum des Datenverkehrs in zellulären Netzen, das in zunehmendem Maße zu Überlastsituationen führen kann, während welcher die Nachfrage an Datenübertragungen die Kapazität des Netzwerks übersteigt. Dies bedeutet eine starke Beeinträchtigung des Nutzererlebnisses, vor allem für interaktive Netzdienste und Multimedia-Anwendungen. Unzufriedene Nutzer könnten den Vertrag bei ihrem Netzanbieter kündigen, was zu sinkenden Erträgen führt. Die übliche Lösung eines Ausbaus der Netzinfrastruktur kann für den vorhergesagten massiven Anstieg des Datenaufkommens jedoch sehr kostspielig sein.

Es gibt andere, komplementäre Ansätze um die Beeinträchtigung von Diensten aufgrund von Überlast abzumildern, die in die Zuweisung der Funkressourcen eingreifen. Herkömmliche Dienstgütedifferenzierung (*service differentiation*) mit einer anschließenden Priorisierung nach Verkehrsklassen entsprechend der Dienstgüte-Anforderungen (*Quality of Service, QoS*) kann die Nutzererfahrung (*Quality of Experience, QoE*) insgesamt verbessern. Ein solcher Ansatz erfordert jedoch ein komplexes Framework und Rechenressourcen zur Signalisierung, Klassifizierung sowie Priorisierung. Ein sehr viel einfacherer Weg um eine graduell abnehmende Diensgüte unter Überlast zu erreichen ist die Anwendung des *Shortest Remaining Processing Time first (SRPT)* Prinzips für die Zuweisung von Funkressourcen. Algorithmen, die nach diesem Prinzip funktionieren, im Folgenden *Shortest-First* Scheduler genannt, minimieren nachweislich die Übertragungszeiten und tragen so zu einer unbeeinträchtigten QoE von latenzkritischen Diensten bei. Des Weiteren erfährt nur der Anteil des Gesamtverkehrs die Auswirkungen von Überlast, der die zur Verfügung stehende Kapazität überschreitet.

Ein Nachteil der existierenden Vorschläge solcher *Shortest-First* Scheduler ist die reduzierte spektrale Effizienz im Vergleich zu herkömmlichen opportunistischen Schedulern. Dies bedeutet eine starke Einschränkung des positiven Einflusses sowie der Anwendbarkeit in Mobilfunknetzen, da das Funkspektrum eine knappe und wertvolle Ressource darstellt. Außerdem werden Multimedia-Dienste nicht vernünftig durch *Shortest-First* Scheduler behandelt. Aufgrund ihrer großen Übertragungsobjekte würden diese Dienste vom Scheduler häufig zurückgestellt werden. Gepufferte Videoübertragungen, die eine wesentliche Internetanwendung darstellen und einen großen Teil des gesamten Verkehrsaufkommens ausmachen, würden daher eine gesonderte Behandlung erfordern.

Die vorliegende Arbeit enthält zwei wesentliche Beiträge, die eine Lösung für beide Probleme liefern. Erstens wird eine Erweiterung für *Shortest-First* Scheduler vorgeschlagen – ein zusätzlicher Parameter namens *Length Exponent* – der es erlaubt den Scheduler in Richtung opportunistischer Algorithmen zu verstellen. Zweitens ermöglicht die Einführung einer Funktion, die den nutzerseitigen Pufferfüllstand eines Videoübertragungsdienstes auf eine äquivalent Objektgröße abbildet, dass diese Verkehrsart nahtlos und vorteilhaft in den Scheduler eingebunden werden kann, ohne diesen zu verändern. Zusätzlich kann mit den Parametern der vorgeschlagenen Funktion zum einen die Priorität von Videoverkehr definiert werden sowie zum anderen eine implizite Zugangskontrolle (*Implicit Admission Control, IAC*) realisiert werden, die es erlaubt die QoE fortlaufender Videos im Überlastfall zu verbessern.

Um einen Hintergrund für das behandelte Thema zu liefern, enthält diese Arbeit einen Literaturüberblick zu den Eigenschaften und der Entwicklung von Internetverkehr. Die Begriffe QoE und Dienstgütedifferenzierung werden vorgestellt, Verkehrsklassen werden definiert und die maßgeblichen Objekte der Anwendungsschicht – Transaktionen genannt – werden zusammen mit Utility-Funktionen für die QoE-Bewertung eingeführt. Bezüglich Mobilfunknetzen enthält die Arbeit eine Einführung der Grundlagen sowie einen Überblick über 3GPP Long Term Evolution (LTE), das als Basis für die Leistungsbewertung dienen wird. Des Weiteren werden Verfahren zur Mehrfachnutzung, allgemeine Funktionen der Ressourcenzuweisung sowie der Stand der Technik bei Schedulern eingeführt und klassifiziert. Dazu gehört ebenfalls eine Diskussion von Vorschlägen aus der Literatur zur Anwendung des SRPT-Prinzips in Mobilfunknetzen.

Vor diesem Hintergrund motiviert die vorliegende Arbeit die Notwendigkeit der Kombination von größenbasierten und opportunistischen Scheduler-Prinzipien. Diese Kombination wird durch den *Length Exponent* γ erreicht, der den Einfluss der Objektgröße auf die Bedienreihenfolge der Transaktionen steuert. Er erlaubt es, das Verhalten des Schedulers nahtlos zwischen dem ursprünglichen *Shortest-First* und *Max C/I* einzustellen. Ein weiterer Vorteil dieses Ansatzes ist es, dass eine Einstellung zwischen diesen Extrempunkten eine Synergie beider Scheduler realisiert. Das heißt, dass eine große Steigerung der spektralen Effizienz bei einer geringen Verschlechterung der Latenzeigenschaft von *Shortest-First* erreicht werden kann. Daher bedeutet die vorgeschlagene Lösung sowohl eine Steigerung der Flexibilität als auch eine Steigerung der Leistungsfähigkeit.

Der zweite Beitrag erlaubt die Behandlung von gepufferten Videodiensten durch *Shortest-First* Scheduler. Zu diesem Zweck wird Information über die Menge des gepufferten Videomaterials im Endgerät des Nutzers in die Entscheidung der Bedienreihenfolge mit einbezogen. Grundsätzlich ist ein Video mit geringem Pufferfüllstand gefährdet unterbrochen zu werden und sollte daher durch den Scheduler bevorzugt werden. Auf der anderen Seite erlaubt ein großer Pufferfüllstand, dass das Video einige Zeit ohne weitere Netzwerkübertragungen abgespielt werden kann. Dieses Verhalten sowie die vorgenannte IAC wurden beim Design der Abbildungsfunktion für die äquivalente Objektgröße mit einbezogen und erlauben eine starke Verbesserung der QoE und eine Konfiguration der Priorität von Videoverkehr.

Um die Vorteile der vorgeschlagenen Lösungen zu bestätigen und zu quantifizieren, wurde eine ausführliche Leistungsbewertung durchgeführt. Sie basiert auf System-Level Simulationen, die den Verkehr in einer Zelle eines Mobilfunknetzes sowie die von umgebenden Basisstationen erzeugte Interferenz berücksichtigen. Dazu wurden Modelle des Funkkanals, der Verbindungsschicht und der Nutzermobilität entwickelt, die das Verhalten von LTE Systemen auf einer abstrakten Ebene nachbilden. Ein entscheidender Punkt hierbei sind realistische Verkehrsszenarien in Verbindung mit den Kanalvariationen beweglicher Nutzer. Die daraus resultierenden

Verkehrseigenschaften (z.B. Volumen, Burstartigkeit) interagieren über die Zeit mit dem Verhalten des Schedulers und haben dadurch großen Einfluss auf die Leistungsmetriken.

Studien, die eine Reihe verschiedener Szenarien bei gleichzeitiger Untersuchung des Einflusses der vorgeschlagenen Parameter abdecken, wurden durchgeführt und hauptsächlich hinsichtlich Zelldurchsatz, Transaktionsdauern und Utility bewertet. Dies ermöglicht es, die vorgeschlagene Lösung mit herkömmlichen Schedulern zu vergleichen und die Vorteile und Effektivität der Beiträge herauszuarbeiten. Die Ergebnisse zeigen, dass der *Length Exponent* tatsächlich die Kapazitätsregion des Schedulers erweitert und die Einstellung des Betriebspunktes ermöglicht. Des Weiteren erreicht der vorgeschlagene Mechanismus eine starke Verbesserung der QoE von gepufferten Videoübertragungsdiensten während zugleich die größtmögliche Flexibilität bei der Ressourcenzuweisung erhalten wird und dadurch die Einbußen für andere Verkehrsarten gering gehalten werden.

Contents

Abstract **i**

Kurzfassung **iii**

Contents **vii**

Figures **xi**

Tables **xiii**

Abbreviations and Symbols **xv**

1 Introduction **1**
1.1 Evolution of Cellular Networks 1
1.2 Problem Statement and Contributions 2
1.3 Structure of this Work 5

2 Mobile Network Traffic and User Experience **7**
2.1 Large Scale Traffic Observations and Forecasts 8
2.2 User Experience 11
2.2.1 Quality of Service and Quality of Experience 11
2.2.2 Service Differentiation 13
2.2.3 Impact of Resource Fairness 15
2.3 Traffic Classification 16
2.3.1 Interactive Web Traffic 16
2.3.2 Background Traffic 17
2.3.3 Video Streaming 17
2.3.4 Real-Time Traffic 19
2.4 Application Layer Objects – Transactions 19
2.4.1 Transaction Definition and Properties 20
2.4.2 Utility Functions 22

3 Scheduling in Cellular Networks **25**
3.1 Radio Propagation and the Concept of Cellular Networks 25
3.2 Spectrum Sharing Technologies 28
3.3 Tasks of Scheduling 31
3.3.1 Allocation of Radio Resources 32

3.3.2 Relevant Time Scales 33
3.4 Classification of Cellular Schedulers 33
3.4.1 Information Sources 34
3.4.2 Optimization Targets 34
3.5 Classical Scheduling Approaches in Cellular Networks 36
3.5.1 Round Robin 36
3.5.2 Maximum Channel Quality 36
3.5.3 Proportional Fair 37
3.5.4 Queue-aware schedulers 37
3.5.5 QoS and QoE-Based Schedulers 38
3.6 Applying Size-Based Scheduling to Cellular Networks 39
3.6.1 Origin of Shortest-First Scheduling 39
3.6.2 Analytical Evaluation of Shortest-First Scheduling in Cellular Networks 41
3.6.3 Schedulers Applying Shortest-First in Cellular Networks 42
3.6.4 Size-Based Schedulers for Video-Streaming 44

4 Combining Size-Based and Opportunistic Scheduling Principles 47
4.1 Motivation 47
4.2 Algorithm Design 50
4.3 Influence and Benefits of the new Scheduler 51

5 Inclusion of Buffered Video-Streaming into Shortest-First Scheduling 53
5.1 Motivation 53
5.2 Application Layer Information on Video Streaming 54
5.2.1 Required Data Rate for Video Transport 54
5.2.2 Knowledge of the Remaining Buffered Playtime 55
5.3 Including Video Streaming into Size-Based Scheduling 56
5.3.1 Handling Video-Streaming under Shortest-First 57
5.3.2 Implicit Admission Control with Buffer Knowledge 58
5.4 Influence and Benefits of the Proposed Algorithm 60

6 Simulation Model and Metrics 61
6.1 Simulation Model and Scenario 61
6.1.1 Cellular Network Layout 62
6.1.2 Channel and Mobility Model 64
6.1.3 Link Layer Model 68
6.1.4 Traffic Scenarios 71
6.1.5 Reference Scheduler Parameterization 76
6.2 Performance Metrics 76
6.2.1 Cell Throughput 76
6.2.2 Transaction Duration 77
6.2.3 User Experience and Utility Functions 78

7 Performance Evaluation 83
7.1 Evaluation of the General System Behavior 83
7.1.1 Determining the Cell Capacity 83
7.1.2 Influence of the Mobility Model 87
7.1.3 User Experience with Interactive Web Traffic 89

7.1.4 Sensitivity to the Object Size Distribution 93
7.2 Evaluation of the Length Exponent 95
7.2.1 Trade-Off between Cell Throughput and User Experience 95
7.2.2 Traffic Model Influence on the Choice of the Length Exponent 98
7.3 Scheduling including Video Streaming Traffic 99
7.3.1 General Utility Behavior with Video Streaming 99
7.3.2 Configuring Implicit Admission Control 102
7.3.3 Controlling Video Priority under Shortest-First 103
7.3.4 Choice of the Length Exponent for the Combined Traffic-Mix 105
7.4 Conclusions for the Overall System Performance 108

8 Conclusions and Outlook 111

A Influence of the Forgetting Factor 117

Bibliography 121

Acknowledgments 131

List of Figures

2.1 Predicted traffic growth in mobile access networks world-wide [Cis14]. 8
2.2 Traffic composition in Europe measured by Sandvine [San14]. 9
2.3 Illustration of an example network topology with the path between client and server.† 13
2.4 Screenshot of a video played in YouTube [You]. 18
2.5 Message-sequence chart of an exemplary transaction illustrating a client requesting a web-site and its embedded objects.† 20
2.6 Evolution of utility depending on the duration of a transaction. 23
2.7 Illustration of the timeline for the playback of buffered video streaming with buffer underruns leading to stalling. 24

3.1 Typical hexagonal cellular layout.† 26
3.2 Simplified illustration of the LTE access network.† 27
3.3 Grid of radio resources for an OFDMA system. 29
3.4 Layers 1–3 of the LTE protocol stack (simplified view). 31
3.5 Simple queuing system illustrating the task of a scheduler. 32
3.6 Capacity region of a cell showing the trade-off between fairness and throughput. 35

4.1 Accumulated volume over the fraction of transactions ordered by size for the scenario from Section 7.1.1 at an offered traffic of 40 Mbit/s. For comparison, the transmitted volume of *SF* is given. 49
4.2 Trading throughput for utility; operation points of different schedulers and flexibility introduced by the length exponent. 49
4.3 Nominator of transaction cost depending on the transaction size for different length exponents γ. 50
4.4 Scheduling cost of two exemplary transactions with different length exponents γ. 52

5.1 Illustration of the evolution of transmitted data, played data, and buffer size over time for buffered video streaming. 56
5.2 Transformation of the client-side buffer level into a size equivalent for *SF*. . . . 58
5.3 Implicit admission control by defining a target buffer and a cost penalty. 59

6.1 Comparison of different cell layouts.† 63
6.2 Cumulative distribution function of SINR for different scenario layouts. 63
6.3 Example of user trajectories.† 66
6.4 Example channel realizations with Rayleigh fading. 67

6.5 Capacity of a scheduling resource (180 kHz, 1 ms) according to the ideal spectral efficiency defined by the Shannon-Hartley theorem, clipped at 25 dB. . . . 69
6.6 Empirical CDFs of video object properties sampled from a simulation run. The video traffic model is parameterized to fit to the measurements in [FMM$^+$11]. . 76
6.7 Parameters of the s-shaped utility function. 80
6.8 QoE of video transactions; MOS model from [CSH13]. 81

7.1 Cell throughput depending on the offered traffic rate for different schedulers. . 85
7.2 SINR distribution of the resources allocated by different schedulers. Dashed lines show full-buffer behavior, straight lines represent an offered traffic of $R_{\text{arr}} = 40\,$Mbit/s. 86
7.3 Cell throughput depending on the speed of the users. 88
7.4 Average transaction utility depending on the offered traffic. 89
7.5 Distribution of transaction durations at $R_{\text{arr}} = 40\,$Mbit/s. 90
7.6 Distribution of transaction utilities at $R_{\text{arr}} = 40\,$Mbit/s. 92
7.7 Histogram of finished transactions depending on their size ($R_{\text{arr}} = 40\,$Mbit/s). . 92
7.8 Scheduler performance with a general log-normal object size distribution. Dashed lines show the performance with the standardized traffic model. 93
7.9 Histogram of finished transactions depending on their size for the general traffic model ($R_{\text{arr}} = 40\,$Mbit/s). 94
7.10 Trade-off between throughput and utility for different offered traffic levels with $\gamma \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1\}$. 95
7.11 CDF of the possible rates of dropped transactions at $R_{\text{arr}} = 40\,$Mbit/s. 96
7.12 Utility histogram over transaction size depending on the choice of the length exponent; compared against the reference schedulers ($R_{\text{arr}} = 40\,$Mbit/s). 97
7.13 Trade-off between throughput and utility with the general traffic model and $\gamma \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1\}$. 98
7.14 Application utilities for a traffic mix containing video streaming for the reference schedulers and non-optimized *SF*. 100
7.15 Video performance under different schedulers at $R_{\text{arr}} = 40\,$Mbit/s. 102
7.16 Influence of the parameterization of IAC for video transactions under *SF*. . . . 103
7.17 Relative prioritization of video transactions against the other application classes with the cost offset ($R_{\text{arr}} = 40\,$Mbit/s). 104
7.18 Influence of the cost offset on cell throughput and application-specific transmission volumes ($R_{\text{arr}} = 40\,$Mbit/s). 105
7.19 Throughput-utility trade-off for the combined traffic-mix depending on the choice of the length exponent γ and different values of c_{min}. 106
7.20 Influence of the length exponent γ at $R_{\text{arr}} = 40\,$Mbit/s and $c_{\text{min}} = 2\,$MBytes . . 107
7.21 Cell throughput for a traffic mix containing video streaming for the reference schedulers and optimized *SF*. 108
7.22 Application utilities for a traffic mix containing video streaming for the reference schedulers and optimized *SF*. 109

A.1 *PF* performance for different settings of the forgetting factor β. 118
A.2 Duration CDF for different settings of the forgetting factor β at $R_{arr} = 40\,$Mbit/s.119

[†] The indicated figures use clip-arts from https://openclipart.org for illustrational purposes.

List of Tables

4.1 Influence of the length exponent γ on cell capacity and finish times of two example transactions. 51

6.1 Parameterization of the radio channel model and cellular layout (mostly complying with [3GP06, Table A.2.1.1-3]). 68
6.2 Parameters for the standardized [NGM08] and general traffic models. 74
6.3 YouTube sending behavior [RMPGA+14] and video properties [FMM+11]. . . 75
6.4 Standard parameterization of the reference schedulers. 77
6.5 Parameterization of the utility functions. 80

Abbreviations and Symbols

Abbreviations

3G	3rd Generation
3GPP	3rd Generation Partnership Project
API	Application Programming Interface
APN	Access Point Name
ARQ	Automatic Repeat Request
BLER	Block Error Rate
BPSK	Binary Phase Shift Keying
CAGR	Compound Annual Growth Rate
CARA	Context-Aware Resource Allocation
CDF	Cumulative Distribution Function
CDM	Code-Division Multiplexing
CDMA	Code-Division Multiple-Access
CIR	Channel Impulse Response
CQI	Channel Quality Indicator
CRC	Cyclic Redundancy Check
CSI	Channel State Information
CSMA	Carrier-Sense Multiple Access
CSMA/CA	CSMA / Collision Avoidance
CSMA/CD	CSMA / Collision Detection
CSS	Cascading Style Sheets

DASH	Dynamic Adaptive Streaming over HTTP
DCF	Distributed Coordination Function
DiffServ	Differentiated Services
DNS	Domain Name System
DPI	Deep Packet Inspection
DRA	Dynamic Resource Allocation
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
EDF	Earliest Deadline First
EDGE	Enhanced Data Rates for GSM Evolution
eNB	enhanced Node B
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
FB	Foreground-Background
FDD	Frequency-Division Duplex
FDM	Frequency-Division Multiplexing
FDMA	Frequency-Division Multiple Access
FEC	Forward Error Correction
FIFO	First-In-First-Out
FTP	File Transfer Protocol
GPRS	General Packet Radio Services
GSM	Global System for Mobile Communications
HARQ	Hybrid-ARQ
HoL	Head of Line
HSDPA	High-Speed Downlink Packet Access
HSPA	High-Speed Packet Access
HSUPA	High-Speed Uplink Packet Access
HSS	Home Subscriber Server

HTML	Hyper-Text Markup Language
HTTP	Hyper-Text Transfer Protocol
IAC	Implicit Admission Control
IAT	Inter-Arrival Time
ICIC	Inter-Cell Interference Coordination
IEEE	Institute of Electrical and Electronics Engineers
IMAP	Internet Message Access Protocol
IMS	IP Multimedia Subsystem
IMT	International Mobile Telecommunications
IntServ	Integrated Services
IP	Internet Protocol
IPv4	Internet Protocol version 4
ISP	Internet Service Provider
ITU	International Telecommunication Union
LAN	Local Area Network
LAS	Least Attained Service first
LIFO	Last-In-First-Out
LoS	Line of Sight
LTE	3GPP Long Term Evolution
MA	Moving Average
MAC	Medium Access Control
MCS	Modulation and Coding Scheme
MIMO	Multiple Input and Multiple Output
M-LWDF	Modified Largest Weighted Delay First
MME	Mobility Management Entity
MOS	Mean Opinion Score
MRL	Minimum Relative Length
MUD	Multi-User Diversity

MSE	Mean Squared Error
NAS	Non-Access Stratum
NGMN	Next Generation Mobile Networks
NNTP	Network News Transfer Protocol
NSiS	Next Steps in Signaling
OFDM	Orthogonal Frequency-Division Multiplexing
OFDMA	Orthogonal Frequency-Division Multiple Access
OS	Operating System
PAPR	Peak-to-Average Power Ratio
PCF	Point Coordination Function
PDCP	Packet Data Convergence Protocol
PDF	Probability Density Function
PDU	Protocol Data Unit
PEP	Performance Enhancing Proxy
P-GW	Packet Data Network Gateway
PF	Proportional Fair
PHB	Per-Hop Behavior
PHY	Physical layer
PRB	Physical Resource Block
PS	Processor Sharing
QAM	Quadrature Amplitude Modulation
QCI	QoS Class Identifier
QoE	Quality of Experience
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RACH	Random Access Channel
RAN	Radio Access Network
RLC	Radio Link Control

RNG	(pseudo-) Random Number Generator
RoHC	Robust Header Compression
RR	Round Robin
RRC	Radio Resource Control
RRM	Radio Resource Management
RSVP	Resource reSerVation Protocol
RTT	Round-Trip Time
SAE	System Architecture Evolution
S-GW	Serving Gateway
SC-FDMA	Single-Carrier FDMA
SF	Shortest First
SFBC	Space-Frequency Block Coding
SINR	Signal to Interference-plus-Noise Ratio
SIP	Session Initiation Protocol
SISO	Single-Input Single-Output
SLA	Service Level Agreement
SRF	Shortest Remaining First
SRPT	Shortest Remaining Processing Time first
TAOS	Traffic-Aided Opportunistic Scheduling
TCP	Transmission Control Protocol
TDD	Time-Division Duplex
TDM	Time-Division Multiplexing
TDMA	Time-Division Multiple Access
TLV	Type-Length-Value
TTI	Transmission Time Interval
UE	User Equipment
UDP	User Datagram Protocol
UI	User Interface

UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
VHF	Very High Frequency
VoIP	Voice over IP
WAN	Wide Area Network
W-CDMA	Wideband-CDMA
WEDD	Weighted Earliest Due Date
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WWW	World Wide Web
XML	eXtensible Markup Language

Symbols

a	Slope factor (for *PF-QoE* scheduler)
a_{PL}	Attenuation due to path-loss
a_{SF}	Slope of the equivalent object size below $b_{t,\text{SF}}$
a_{SH}	Attenuation due to shadowing
B	Channel bandwidth
$b(t)$	Client-buffer of a streaming transaction at time t (in seconds or kBytes)
b_t	Target buffer (for *PF-QoE* scheduler, in seconds)
$b_{t,\text{SF}}$	Target buffer (for *SF* scheduler, in kBytes)
C	Channel capacity
c_{min}	Minimum size equivalent for the cost of a video transaction (for *SF* scheduler)
d	Distance between sender and receiver
F	Transaction size
$f(b(t))$	Mapping function of client-buffer to scheduling weight (for *PF-QoE* scheduler)
f_D	Maximum Doppler shift
f_{min}	Minimum value of $f(b(t))$ (for *PF-QoE* scheduler)
$F_r(t)$	Remaining transaction size at time t
$F_{\text{video}}(t)$	Equivalent object size of a video transaction at time t
$f_X(x)$	Probability density function of a random variable X
I_i	Noise power received from interferer i
$M(t)$	Number of active transactions at time t
MOS$_{\text{min}}$	Minimum possible Mean Opinion Score (MOS)
N	Noise power
$R(t)$	Instantaneous possible rate of a transaction at time t
$\overline{R}(t)$	Moving average of a transaction's throughput at time t
$\overline{R^*}(t)$	Moving average of a transaction's possible rate at time t
R_{arr}	Arrival rate of transactions
R_{drop}	Dropping rate of transactions

r_{exp}	Expected data rate
R_{fin}	Finishing rate of transactions
s	Scaling factor for the steepness of the logistic utility function
S	Signal power
s_τ	Scaling factor between expected duration τ_{exp} and inflection point τ_{infl}
t	Time
U	Utility (general)
$U'(t)$	Logistic utility shape
u_{min}	Minimum utility for a finished transaction
u'_{exp}	Value of $U'(t_{\text{exp}})$
U_{video}	Utility of a video streaming transaction
$W(t)$	Head-of-line packet delay of a waiting queue at the base station

α	Shape of the Pareto distribution
β	Forgetting factor
γ	Length exponent
λ	Ratio of accumulated video stalling time and playback time
μ	Mean (of a probability distribution)
ρ	Correlation coefficient of shadowing, between cells
σ	Standard deviation (of a probability distribution)
τ	Transaction duration
$\tau_{\text{buf,exp}}$	Expected duration of the initial video buffering phase
τ_D	Feedback delay between measurement and application of Channel State Information (CSI)
$\tau_{\text{drop,min}}$	Minimum duration before a transaction gets dropped
τ_{exp}	Expected transaction duration
$\tau_{\text{exp,min}}$	Minimum expected transaction duration
τ_{infl}	Inflection point of the logistic utility function
$\tau_{\text{stall,max}}$	Maximum duration of a video stalling event

1 Introduction

Nowadays, ubiquitous Internet access is a reality in most countries. Among other access technologies, mainly the large-scale deployment of wireless cellular networks makes this possible. However, for several years, traffic in cellular networks has shown an exponential growth and has pushed the networks to their limits. Economic and technical constraints forbid an unlimited increase in network capacity. Consequently, peak traffic situations have become more common and may impair the service quality and satisfaction of the users.

This work discusses how the scheduling of radio resources can affect the user experience and provide a graceful service degradation in high load situations. Existing resource allocation algorithms in cellular networks are introduced and evaluated. Thereby, a focus lies on the advantages and drawbacks of schedulers[1] applying the Shortest Remaining Processing Time first (SRPT) principle. These schedulers are then enhanced by this thesis to provide the desired improvement in Quality of Experience (QoE) while still delivering a good cell throughput. Furthermore, they are enabled to handle video streaming traffic beneficially. For getting into the context of cellular networks, their evolution will be shortly introduced in the following. Then, the contributions of this thesis are presented and the structure of the chapters is given.

1.1 Evolution of Cellular Networks

Wireless communication has a very long history. A major milestone of radio communications, which is the basis for today's cellular networks, was the first transmission over the Atlantic Ocean by Marconi at the beginning of the 20th century [TV05, p.2]. The concept of cellular networks was developed beginning in 1947 [STB09, p.1]. It solves two major challenges. First, it enables providing a complete wireless connection over a large area by a network of fixed base stations. Second, it allows reusing the same radio spectrum in different locations, a so-called spatial reuse. In other words, the necessity of wireless cells comes from the signal degradation with increasing distance of the communication partners, which limits the reach of a single base station. Furthermore, the limited radio spectrum, which is a shared medium among all competing transmissions, should be used efficiently. In addition to space, the time and frequency domains are also available to separate different transmissions from each other. Beyond that, cellular networks provide mobility, i. e. a seamless connectivity when traveling between the different cells of an access network. Today, even world-wide roaming between the networks of different operators is possible.

[1] The terms resource allocation and scheduling will be used synonymously throughout this thesis.

First generation cellular networks transmitted analog voice signals [TV05]. Systems accounted to this generation, were the first to see a large-scale deployment and appeared in the 1980s [STB09][2]. However, only the second generation system Global System for Mobile Communications (GSM) was the first offering world-wide roaming and today offers practically pervasive coverage [GSM]. While still being developed for voice transmission, it employs digital modulation. Packet-based data services where added later under the terms General Packet Radio Services (GPRS) and Enhanced Data Rates for GSM Evolution (EDGE). The third generation networks employ Code-Division Multiple-Access (CDMA), also called Wideband-CDMA (W-CDMA). A major standard is the Universal Mobile Telecommunications System (UMTS) which has been developed by the European Telecommunications Standards Institute (ETSI) and is now overseen by the 3rd Generation Partnership Project (3GPP). Such standardization bodies are very important for wireless communications. As the medium is shared between all users in the signal range, their devices have to stick to common protocols in order to control and mitigate the impact of interference. Consequently, the vendors of radio devices engage in standardization bodies and agree on certain medium access schemes. The 3GPP is a very important standardization body and developed the UMTS advancements High-Speed Packet Access (HSPA) and HSPA+.

The most advanced cellular networks currently deployed are denoted as the fourth generation. The *ITU-R* (Radiocommunication Sector of the International Telecommunication Union (ITU)) issued a regulatory specification called *IMT-Advanced* (International Mobile Telecommunications (IMT)) in 2008 that defines performance requirements for cellular systems [IR08]. Systems satisfying these requirements may be operated in the respective reserved spectrum and are generally attributed to be fourth generation cellular networks. The major standard of this generation is 3GPP Long Term Evolution (LTE)[3] and will also serve as the basis for performance evaluation in this thesis. Differing from earlier generations, Orthogonal Frequency-Division Multiple Access (OFDMA) is employed to separate transmissions originating from the base stations (the *downlink* direction) and Single-Carrier Frequency-Division Multiple Access (SC-FDMA) is used in the opposite direction (*uplink*). This allows an improvement in spectral efficiency compared to previous generations and provides high flexibility in the frequency domain [STB09]. LTE was designed from the start as a packet-based, all-Internet Protocol (IP) network accommodating the fact that the majority of today's traffic volume is created by data services. Another important standards family by the Institute of Electrical and Electronics Engineers (IEEE) employing OFDMA is called Worldwide Interoperability for Microwave Access (WiMAX), with the most recent standard IEEE 802.16m achieving the IMT-Advanced requirements [WiM10]. However, especially in the sector of mobile networks, it is not as widespread as LTE networks.

1.2 Problem Statement and Contributions

The possibility to use services like web browsing, video streaming and social networking virtually everywhere through cellular infrastructure is very convenient and attracts more and more

[2] The following short historical recapitulation is mainly inspired by [STB09] and [TV05].

[3] The initial LTE Release 8 does not match all IMT-Advanced criteria; this is achieved by *LTE-advanced* beginning with Release 10. Nevertheless, LTE is often referred to as fourth generation technology.

users. This leads to an ever increasing traffic demand in mobile networks [Cis14]. As a consequence, the load relative to the cells' capacities and especially the peaks of data traffic increase. During such high- or overload cases, the service quality may decrease significantly, leading to a bad QoE of the network users. Widely-used resource allocation algorithms like *Proportional Fair* and *Max C/I* (introduced in Section 3.5) do not behave beneficially in this situation. A majority of users may suffer large latency and low data rates. For many services this means a strong degradation in QoE, e. g. long web response times or interrupted video playback.

A conventional approach to improve Quality of Service (QoS) and QoE is service differentiation. This means that services are prioritized against each other according to their QoS requirements. However, there are disadvantages regarding the complexity, scalability and efficiency of this approach. All traffic needs to be classified and prioritized a-priori according to QoS requirements. This is difficult as new services arise over time and the number of QoS classes should not be too large for practical reasons. Furthermore, separating traffic into different classes reduces the diversity that can be exploited by cellular schedulers. Other approaches tackle the resource allocation itself by incorporating an optimization of service quality directly into the scheduling algorithms. However, this often results in a large computational complexity and requires additional frameworks to provide the necessary information. As in the case of service differentiation, the QoS requirements of the traffic need to be known.

Addressing the problem of latency, algorithms based on the SRPT-principle were proposed several years ago for application in cellular networks [Tsy02]. Resource allocation based on this principle is known to be optimal for minimizing job processing times since a long time [Sch68]. Schedulers applying SRPT will be denoted as *Shortest-First* schedulers. The existing *Shortest-First* schedulers for wireless networks bring a severe drawback in spectral efficiency compared to conventional opportunistic schedulers with them. By giving priority to short transmission objects instead of only considering the channel quality, the influence of the latter is reduced. Therefore, there is a smaller capacity gain compared to the one achieved by conventional opportunistic schedulers (please refer to Chapter 4 for a discussion). This limits the applicability in cellular networks as radio spectrum is a scarce and expensive resource. So far, there is no solution to trade spectral efficiency, i. e. cell capacity, for short latency. Furthermore, existing *Shortest-First* schedulers offer no solution to handle video streaming specifically. By their nature, the comparably large video objects will get a low priority leading to a bad QoE of this important service class. These factors preventing a deployment of *Shortest-First* schedulers in practical cellular networks are addressed by the contributions in this work.

Contributions

This thesis contains two main contributions. They extend *Shortest-First* scheduling and thus improve it to serve as a comprehensive scheduler for cellular networks. Their focus is on an efficient usage of the expensive bandwidth and infrastructure resources while offering a satisfactory QoE to as many users as possible.

1. Combining and Adjusting of Size-Based and Opportunistic Scheduling Principles

Users are happy when they have an interactive service experience, which means that the duration to finish a web request, for example, should be as short as possible. On the other side, cellular network operators often sell capacity and traffic volume to their customers. To achieve a profitable operation, the service provider has to serve as many users as possible with the existing bandwidth and infrastructure. So far, it is not possible to trade short transmission durations for cell capacity. It depends on the implemented scheduler to obtain either the one or the other. Furthermore, depending on the traffic-mix, which is subject to change over time, different configurations may be advantageous.

This thesis proposes a combination of the *Shortest-First* and *Maximum Channel Capacity* (*Max C/I*) scheduling principles. A parameter called *length exponent* is added to an existing *Shortest-First* scheduler that allows to continuously tune the original algorithm towards a *Max C/I* scheduler.

The thesis contains an explanation of the proposed scheduling scheme together with a motivation for the algorithm design and an extensive performance evaluation by network simulation, based on a model for state of the art LTE cellular technology. The strengths of this algorithm are pointed out by using a realistic traffic model with object sizes and random arrivals resembling current Internet traffic. The trade-off between short transmission durations and cell capacity and its adjustability is demonstrated and discussed thoroughly.

2. Inclusion of Buffered Video-Streaming into Shortest-First Scheduling

Shortest-First offers a superior QoE performance during overload situations for traditional Internet browsing and interactive web applications compared to conventional opportunistic schedulers. However, without further enhancements, it is not possible to support buffered video streaming satisfactorily. Naturally, traffic objects for video streaming are usually larger than the objects of other Internet services. This means that video objects would obtain a low priority by *Shortest-First* schedulers resulting in frequent playback interruptions and long buffering times. This would be very annoying for the video users. Recent studies of Internet traffic (e. g. [San14]) show that video traffic accounts for a third up to one half of the total Internet access traffic. A scheduler which does not support such a large fraction of the traffic is not a viable option in cellular networks.

This thesis therefore proposes a method to seamlessly handle buffered video streaming traffic by *Shortest-First* schedulers. The idea is to consider the buffered playtime at the client-side for scheduling. While the usage of client-side buffer information has been proposed in literature [WSP+12, NOALS+13], it was not applied to *Shortest-First* schedulers. By defining an equivalent object size based on the buffer level, it is possible to integrate video handling into *Shortest-First* without changing the underlying algorithm. A big advantage of this approach is that videos automatically get more scheduling weight when they are in danger of a buffer underrun which would lead to a playback interruption and impair the QoE. On the other side, a video session with plenty of buffered playtime at the client-side can be postponed by the scheduler in order to serve more urgent transmissions first. This approach can thus deliver an excellent QoE for video streaming while limiting the drawbacks for other traffic classes.

Furthermore, two parameters are developed in this thesis to adjust the size-information for scheduling. They allow to prioritize video traffic against other services and to achieve a so-called *Implicit Admission Control (IAC)*. First, by defining an offset for the equivalent object size, the maximum priority given to video transmissions can be limited. Second, IAC is achieved by reducing the priority of videos with a very low buffer-level. Thereby, in case of overload, ongoing video transmissions can be protected implicitly against an unsustainable amount of newly arriving videos and against other videos, for which the radio channel is not sufficient to support the required data rate.

As for the first contribution, the design of the algorithm and its prerequisites are discussed and the performance is evaluated thoroughly by network simulation. Emphasis is laid on the influence of the traffic mix and trading the requirements of different abstract application classes (web, download, streaming). When using buffer information, a superior performance for the total network traffic can be achieved.

1.3 Structure of this Work

This thesis is structured as follows. Chapter 2 gives an overview of Internet traffic studies with respect to traffic characteristics and composition from literature. Furthermore, the influencing factors on QoE are discussed, and the central traffic objects for traffic modeling and assessment – transactions – are introduced. Then, Chapter 3 introduces the concept of cellular networks. It first presents the principles of cellular access networks and radio communications with a focus on LTE technology. Resource allocation, including spectrum sharing techniques and a classification of state-of-the-art schedulers, is introduced. Chapter 3 concludes by giving a survey of literature on applying size-based scheduling to cellular networks.

The main contributions of the thesis are motivated and derived analytically in Chapter 4 and Chapter 5. In detail, Chapter 4 presents the *length exponent* γ which allows to combine size-based and opportunistic scheduling principles into one algorithm. Followingly, Chapter 5 considers the treatment of buffered video streaming traffic by *Shortest-First* schedulers and motivates the approach of using client-side buffer information. It is discussed, how this information can be acquired and applied for resource allocation.

As a basis for performance evaluation, Chapter 6 introduces the simulation model and evaluation metrics. This comprises primarily the models of the cellular network and the wireless channel as well as realistic traffic scenarios. Besides conventional metrics like cell throughput and transmission durations, utility functions for the assessment of QoE are introduced. Chapter 7 contains a detailed performance evaluation of the proposed algorithms and reference schedulers. The presented simulation studies cover realistic traffic scenarios with varying boundary conditions. First, the general system behavior of different schedulers with respect to the performance metrics is evaluated and compared for a range of boundary conditions. Then, the influence of the introduced parameters is highlighted for different traffic scenarios. The chapter closes with the inclusion of video traffic and a discussion of the chances offered by the proposed contributions. Finally, the thesis is concluded in Chapter 8 and an outlook is given.

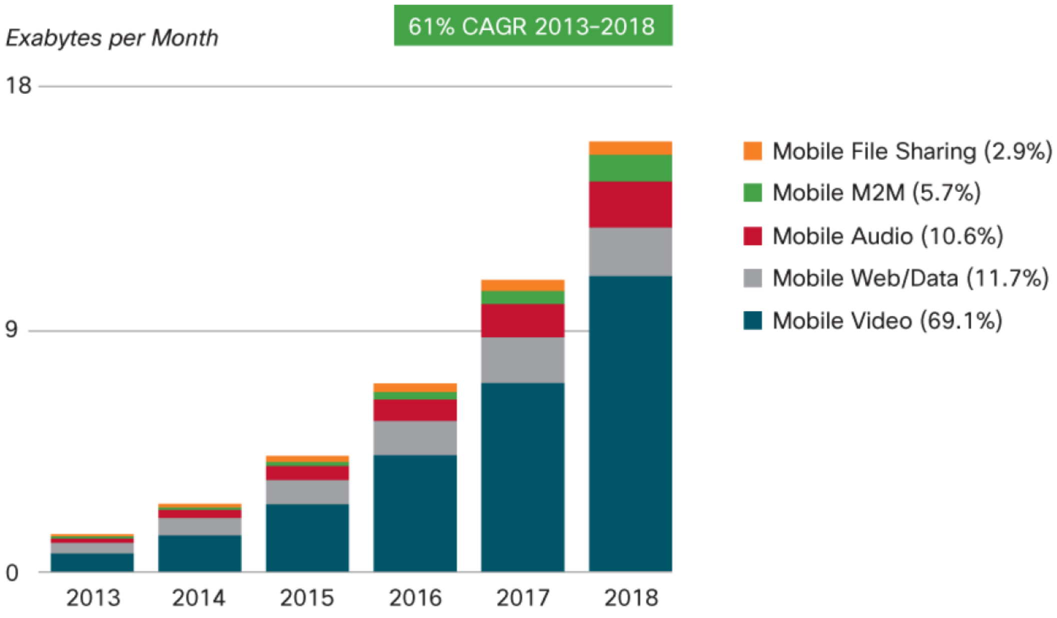
2 Mobile Network Traffic and User Experience

Communication networks serve the purpose of transmitting information from one place to another. Traditionally, cellular networks, which are the subject of this thesis, offered a mobile telephone service. However, today the fraction of such voice traffic in the total volume is very small. The transmission of data traffic, e. g. for web browsing, multimedia, social networking, and file transfers over the Internet makes up the major part of the transmitted volume. The characteristics and volume of the transported traffic have an impact on network operation and consequently on the QoE the users have with their connections.

The Internet is packet-switched with IP as the common network layer protocol. Internet traffic is known to be bursty [LTWW94]. This challenges especially the cellular access link, as it is a shared medium and often the rate bottleneck for the connections using it. The aggregated traffic of the users within a cell arrives at the base station and queues up in a buffer. This buffer grows whenever the instantaneous arrival rate exceeds the capacity of the radio resources which leads to increasing waiting times. While the functioning of cellular networks will be thoroughly discussed in Chapter 3, the resulting transmission behavior may lead to packet delay and loss. In this chapter, we will first discuss the impact of this transmission behavior on the user's service quality.

Because mobile network operators earn money by selling communication services to their customers, it is in their interest to serve as many users as possible. At the same time, the operation of networks is costly, which means that an efficient usage of the deployed resources is required. However, pushing the utilization of the available network resources to the limit leads to diminishing user experience, because the bandwidth available to individual users will decrease and waiting times will increase. This may lead to users quitting the service. To sum up, for network operators it is essential to understand the nature of the expected traffic as well as its estimated volume and the influence on user experience depending on the deployed resources.

In this chapter, we will therefore first take a look at today's Internet traffic in cellular networks by discussing recent traffic observations and forecasts from literature, see Section 2.1. This allows us to derive traffic models for performance evaluation. Section 2.2 deals with user experience and how it can be evaluated and improved. Then, the application classes on which we focus in this thesis are introduced in Section 2.3. Finally, the central traffic objects for modeling and performance evaluation – *transactions* – are presented in Section 2.4.



Figures in parentheses refer to traffic share in 2018.
Source: Cisco VNI Mobile, 2014

Figure 2.1: Predicted traffic growth in mobile access networks world-wide [Cis14].

2.1 Large Scale Traffic Observations and Forecasts

An often cited traffic forecast is the Cisco Visual Networking Index [Cis14]. It predicts an exponential traffic growth in mobile networks for the following years. Figure 2.1 shows a graph from Cisco's forecast illustrating the total traffic and its composition in mobile networks. It states a Compound Annual Growth Rate (CAGR) of 61% up to the year 2018.

Apart from Cisco Systems Inc., another company which works on improving QoE and investigating today's Internet traffic is Sandvine Inc. It publishes a bi-annual report of Internet traffic observations [San14]. These reports discuss developments of fixed and mobile network traffic. Especially traffic volume and traffic composition are investigated thoroughly with data from many network operators all over the world. Figure 2.2 shows the traffic composition for mobile networks in Europe given in [San14]. In downstream direction, real-time entertainment is the largest fraction of traffic with YouTube being the service with most volume. Except for Africa and Latin America, this is similar in the other regions. Also web browsing and social networking contribute large fractions to the total traffic. The dominance of video streaming in the mobile traffic volume is in agreement with the forecast from Cisco [Cis14].

An interesting analysis of a large-scale traffic trace in a 3rd Generation (3G) cellular network from the year 2007 is presented in [PSBD11]. Although the findings on data rate are likely superseded by the roll-out of new technology, the indicated characteristics of user behavior are expected to still be valid. [PSBD11] states that the majority of users exhibit a low mobility, i. e. the users tend to remain within a single cell and move within a radius of one mile. Furthermore, the authors point out a traffic imbalance with respect to users and base stations: *"Less than 10% of subscribers generate 90% of the load, while 10% of base stations carry 50–60% of the load."* [PSBD11]. This fits well to the findings on subscribers' traffic distribution in the recent

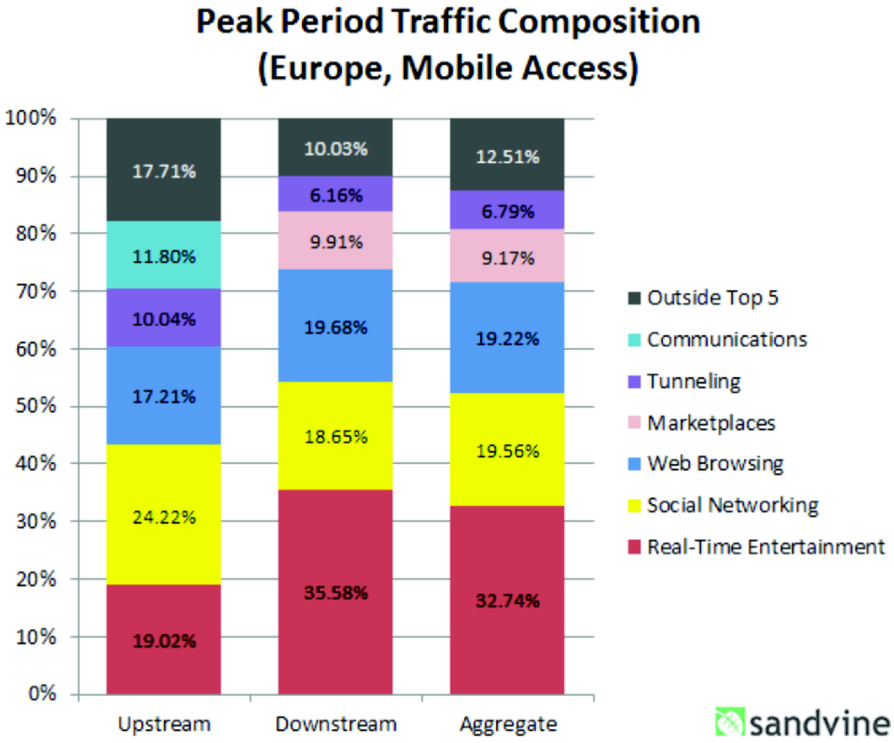


Figure 2.2: Traffic composition in Europe measured by Sandvine [San14].

Sandvine report [San14] (with variations depending on the investigated region). More recently, the authors of [HQG$^+$13] conducted a large-scale measurement study in a commercial LTE network in the U.S. They report a heavy-tailed flow size distribution. This means that most flows are very small (*"90 % of flows carry no more than 35.9 KB downlink payload"* [HQG$^+$13]) while most traffic volume is transported in a few large flows (61.7 % of downlink volume in 0.6 % of flows). Another academic work investigating fixed Internet traffic and covering five years of traffic traces (ending in the year 2010) is [IP11]. The authors use log files from a global proxy network and analyze the evolution of content type and web page characteristics by inspecting the structure of individual web pages. Important findings are the increase of video streaming and the complexity of web pages offering dynamic content and client-side interactions (i. e. the usage of JavaScript, Cascading Style Sheets (CSS), and eXtensible Markup Language (XML)).

[FLM$^+$10] investigates traces of smartphone traffic from the year 2009. It attests the dominance of web browsing with respect to traffic volume and a large fraction of objects with small transfer sizes (30% smaller than 1 kByte for one of the data sets). Comparing [FLM$^+$10] with the more recent studies in [Cis14, San14] shows that the fraction of multimedia in the traffic volume has grown dramatically. In contrast to [FLM$^+$10], which measures traffic at the client side, another study is presented in [MSF10] taking traffic traces at the network side and filtering this traffic for mobile device signatures. With this approach, [MSF10] investigates the traffic of hand-held devices that access the Internet through Wireless Local Area Networks (WLANs) which are connected via residential Digital Subscriber Line (DSL). The traffic traces are from the years 2008 and 2009. At that time, iPhones and iPods from Apple clearly dominated the traffic from mobile devices (86–97%). Most residential traffic went over Hyper-Text Transfer Protocol (HTTP), with a median object size of $\approx$ 10 kBytes.

A similar work is [FMM+11] which is based on a traffic measurement from February 2011 in wireline access networks and concentrates on YouTube video streaming. [FMM+11] thoroughly investigates the difference between mobile and PC clients as well as the user interactions when watching videos from YouTube. For example, video size distribution and abortion ratio are considered. Also focusing on video streaming towards mobile devices, [LLG+13] investigates server-side traffic traces of a video service supplying transcoded versions of a video to mobile devices. The authors point out the heterogeneity of software and hardware of mobile devices and the many versions of a video this necessitates. For a 2010 traffic trace, [LLG+13] obtains a mean file size of 2.78 MBytes per video, a median playback duration of 162 s, and encoding rates ranging from 51–423 kbit/s.

We can summarize the following central findings from the discussed literature:

- An excessive (exponential) growth of data traffic volume is predicted for the following years, especially driven by video streaming.
- Today, the largest fractions of traffic volume originat from (buffered) video streaming and web browsing (including social networking, which has similar characteristics).
- The general object size distribution is assumed to have a heavy-tailed nature.
- Due to client side interactions and caching, web browsing comprises a large number of very small objects (few kBytes).
- In relation to that, video streaming consists mostly of larger objects (several MBytes).
- A large fraction of users in cellular networks exhibit low mobility.

With such an intensive traffic growth, it is likely that peaks of traffic demand occur, during which the required data rate of the users' applications exceeds the capacity of a cell. While network operators try to extend their network bandwidth such that the network can cope with the offered traffic, an ever increasing demand still may lead to frequent high load situations. Apart from overload caused by an insufficient capacity of the access network, load peaks exist. An often observed characteristic of Internet traffic is its burstiness [LTWW94]. Such traffic bursts have the potential to temporarily overload a cell. While these situations do not last long, interactive services which require low latency can still be significantly impaired. Therefore, during overload situations, it is important to have a scheduler that allows a graceful service degradation. This means that, although it is not possible to satisfy all traffic demands anymore, most users should not notice the overload. Instead of reducing the QoE of all traffic, only some low priority or background applications get delayed. The transmissions of these applications usually are not time-critical such that there is little degradation in QoE. At the same time, the interactive services still work satisfactory.

In this thesis, we focus on scheduling to improve the user experience in a cellular system. The presented traffic observations motivate that for an ever increasing demand, an efficient network operation is essential. An intelligent scheduling algorithm allows to distribute the limited resources such that most users will not notice inevitable overload situations during peak traffic times. By replacing common schedulers like Proportional Fair (PF) with the ones presented in this thesis, operators can allow more traffic in their cells without increasing network resources or reducing QoE.

2.2 User Experience

The focus of this thesis is on improving the user experience in cellular systems. Therefore, the assessment of this user experience plays a crucial role. In this section, we first distinguish between QoS and QoE. Then, classical service differentiation mechanisms are introduced and the relationship between user experience and fairness is discussed.

2.2.1 Quality of Service and Quality of Experience

The term *Quality of Service (QoS)* usually denotes quantifiable metrics at the network layer which help to predict how well an application will behave. Important metrics for QoS evaluation are bandwidth, delay, jitter and packet loss distributions. They are usually measured at network nodes like routers and allow to evaluate traffic crossing this node. Furthermore, QoS metrics are used to define Service Level Agreements (SLAs) which describe the traffic and forwarding properties of contracted service delivery between customer and network operator.

In contrast to this classical way of measuring QoS with network level metrics, the prevalent term for approaches assessing the user experience is *Quality of Experience (QoE)*. This shall underline that not the network metrics alone but also their impact on the application behavior, which is experienced by the user, is of interest. For example, while jitter may not have any influence on the playback of a streamed video when the buffer is large enough to compensate it, it can render a voice call completely dissatisfactory, where large buffers are prohibitive due to latency requirements. Thus, the type of application and its requirements are decisive whether a certain QoS metric has an impact on the QoE or not.

The approach in this thesis for evaluating user experience complies with what is denoted by QoE. We want to observe what influences the satisfaction of a user with a certain application. The difficulty is that user experience is a very subjective measure and depends on the respective application. A way to evaluate QoE is the Mean Opinion Score (MOS). It expresses the averaged subjective opinions of users with a service. In a survey, the users can rate the service under test on a scale between 5 (excellent service) and 1 (bad service). For voice services a complete test setup to conduct a MOS survey is defined by the ITU in [IT96]. In [NUN10], a survey has been conducted, in which users were asked to rate several Internet services including web browsing, voice calls, e-mail service and file downloading. The essential parameter for all services in [NUN10] is the waiting time. As a result, the authors give functions mapping waiting times to MOS for the respective services. Also differences with respect to the mood of users, i. e. how relaxed they were, and the place of use are investigated.

A further survey on MOS for web browsing is given in [ARMNO$^+$10]. The article describes a user experience study resulting in MOS values for different service response times of web servers. It is worth noting that their derived utility functions contain the size of the web page, which agrees well with our approach described in Section 2.4.2.[1]

Nielsen [Nie10] points out three different time thresholds which are important for human perception. Waiting times below 0.1 s are perceived as an *"instantaneous response"* [Nie10]. Up to 1 s, humans feel an increasing lag, but still perceive the service as interactive. Between 1 s and 10 s, a delay is clearly noticeable and annoys the users, while for more than 10 s many users will switch to other tasks and may abandon their requests. These time thresholds have also been adopted by the ITU and are applied to a model for predicting the MOS of web browsing [IT05].

A generic model for the mapping from QoS metrics to QoE is proposed in [FHTG10]. Deteriorations in the QoS required for an application, called *QoS disturbances*, lead to a reduction in QoE. Similarly to [Nie10], the authors identify three QoE regions: Unaffected user experience, i. e. the quality only depends on the original service, QoE degradation where the user satisfaction quickly decreases, and finally unacceptable service quality, where most users abandon their request, i. e. they have no service quality at all. A central finding is that a certain amount of QoS degradation has a large impact when the QoE is high and a small impact when the QoE is already low. Accordingly, assuming a linear gradient for this observation, [FHTG10] supposes the QoE value to be an exponential function of the respective QoS metric. As [FHTG10] states, this is in contrast to [IT05, KH02] which assume a logarithmic relationship. Examples and experimental results are given for voice quality that depends on loss and jitter, and web browsing that depends on the response time. [FHTG10] bases its investigation of web browsing on [KH02], which contains a survey of canceled web requests and derives a relation to delivery bandwidth and estimated delivery time.

While for most applications the QoE mainly depends on the time it takes to finish a transmission, for video traffic an uninterrupted playback is important. We discuss this in detail in Section 2.4.2 and introduce utility functions for video streaming to assess the user experience in this case. Also resolution, image quality and other aspects of the video itself have an impact. However, such service parameters are out of scope for this thesis as they are not influenced by the network behavior (assuming no network adaptive quality changes are applied).

A study focusing on the QoE of video streaming (at the example of YouTube) is [HSH$^+$11]. The authors used crowd sourcing to let people across the Internet evaluate their satisfaction with video playback depending on how the video was transmitted. To this end, [HSH$^+$11] varies the number and length of video interruptions as influence factors. [MCC11] presents a much smaller survey with 10 users assessing video quality in a testbed scenario. In this scenario, they vary bandwidth, loss rate and delay and relate these network QoS metrics to video QoE. Both, [HSH$^+$11] and [MCC11], agree that the number of video interruptions has the largest impact on the QoE of buffered video streaming.

A good QoE leads to users being content with their service and network operator. On the other side, a bad QoE may result in customers quitting the service of the provider. For a profitable network operation, it is therefore advantageous to provide the best possible QoE with the existing resources and may offer a competitive advantage for the network operator.

[1] We will discuss the resulting scheduling algorithm from [ARMNO$^+$10] in Section 3.5.5.

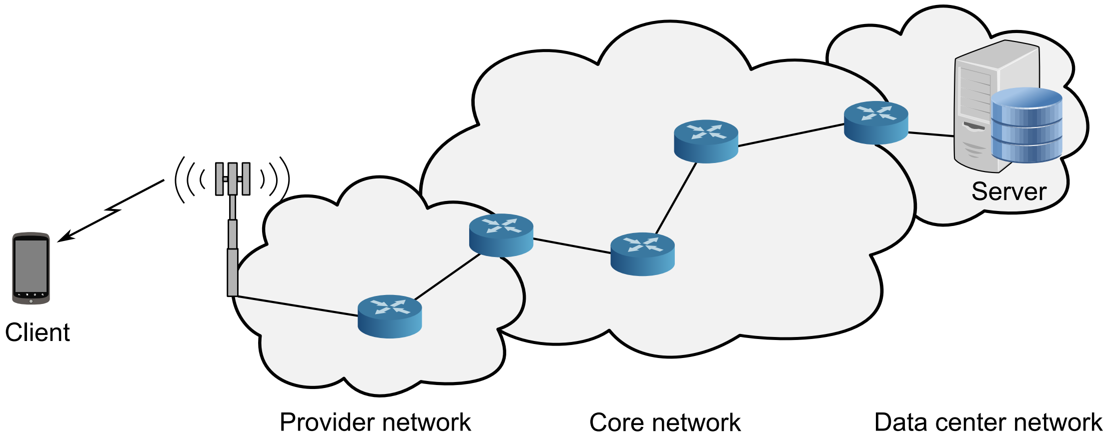


Figure 2.3: Illustration of an example network topology between client and server.

2.2.2 Service Differentiation

Classical approaches aiming to improve the QoE employ *service differentiation*. As it was pointed out above, different applications have different requirements for a satisfactory QoE. While some transmissions like voice calls have tight latency requirements, much of the traffic like web browsing should just be transported as fast as possible, denoted by *best effort*.

The basic finding that for real-time communication the QoS offered by best effort forwarding is not sufficient has led to the development of the *Integrated Services (IntServ)* model [RFC1633]. Bodamer gives an overview of it in [Bod04, pp.59–66]. IntServ relies on the Resource reSer-Vation Protocol (RSVP) [RFC2205] to reserve resources for a flow on the whole path between server and client. This allows to give absolute guarantees in terms of QoS metrics. Figure 2.3 illustrates an example of Internet topology containing the different network domains between a mobile terminal client and a server located in a data center. To reserve resources for a connection, all routers along the path need to participate in RSVP. If too few resources are available, admission control will reject new reservations. However, as [Bod04] points out, there are several drawbacks limiting the scalability of IntServ and prevented its wide-spread adoption.

An alternative approach of service differentiation in IP networks is *Differentiated Services (DiffServ)* [RFC2475]. Instead of managing absolute QoS guarantees per flow, it performs a relative prioritization between different service classes with the possibility of QoS guarantees for the traffic aggregate in a certain class. This avoids the necessity of resource reservation and flow state in routers, while packets of time-critical applications can still be treated preferentially in the routers. In contrast to end-to-end reservations, as in the case of IntServ, DiffServ only specifies the Per-Hop Behavior (PHB) when forwarding a packet in a router.

For this purpose, the classes *expedited forwarding* [RFC3246] and *assured forwarding* [RFC2597] have been defined in addition to the default best effort class. Packets of the expedited forwarding class shall experience low delay, jitter, and loss. This often means that they are strictly prioritized over the packets of other classes and that admission control is necessary to avoid over-booking. Assured forwarding comprises several classes with relative priority levels. For each class, a certain minimum bandwidth and buffer space shall be available and packets of different classes have to be treated independently. Within a class, a drop precedence of three levels is defined, which means a relative distinction of drop probabilities. In general, a low packet drop probability shall be achieved when the bandwidth consumed by the traffic of a class is below its specification, while excess traffic may be subject to higher drop probabilities.

For packet classification, DiffServ relies on the Differentiated Services Code Point (DSCP) header field defined in [RFC2474], which is a part of the IP packet header. It can be set by the end system and the edge nodes of a DiffServ domain. A DiffServ domain is *"a contiguous set of nodes which operate with a common set of service provisioning policies and PHB definitions"* [RFC2475]. Usually, the network of an individual operator represents such a domain. In the example of Figure 2.3, the provider network and the core network could be different DiffServ domains. According to the SLA between the network domains, the traffic of a class may be metered, shaped or reclassified at the edge of a domain. Accordingly, the DSCP may be changed to comply with the domain-internal PHB of the respective traffic class. This means that the end-to-end behavior cannot be guaranteed with DiffServ, because the PHB may be different in the traversed domains. The idea behind the DiffServ architecture is to move complex functions to the edge of the network and to have simple forwarding rules within the internal nodes [RFC2475].

To sum up, service differentiation allows to distinguish between traffic classes with diverse requirements. It goes beyond pure best effort forwarding in that it allows to prioritize the packets of time-critical services. Thus, a QoE deterioration in the case of network congestion can be avoided for these services. This works in particular well, if time-critical applications require few bandwidth and high volume services have weaker latency requirements as it is common for most services[2]. A challenge is to classify packets of different applications and to map these classes to a certain forwarding behavior in order to improve the overall QoS. Several QoS-aware scheduling algorithms for network nodes exist for this purpose. [Bod04] presents a survey of algorithms and proposes Weighted Earliest Due Date (WEDD) which allows the simultaneous differentiation of latency and packet loss requirements between QoS classes.

In this thesis, service differentiation is assumed for voice and other real-time traffic, which should be handled separately from best effort traffic to achieve the necessary QoS requirements. The QoS technology of IP Multimedia Subsystem (IMS), which is implemented in LTE cellular networks, is based on DiffServ. Therefore, we assume the base station's scheduler to be able to identify real-time traffic and to treat it accordingly. Beyond that, the proposed scheduling solution shall be able to improve the QoE of all other traffic within a single class. For buffered video streaming, application information will be exploited allowing to integrate this type of traffic into the best effort class without further differentiation. This combined scheduling offers advantages that will be discussed in Chapter 5.

Assuming a network topology similar to the one illustrated in Figure 2.3 means that the wireless link will often be the rate bottleneck of the path from end to end. While data center upstream connection, core network links and the links inside of Internet Service Provider (ISP) networks usually have rates in the order of Gbit/s and exhibit large capacity reserves, the average capacity of an LTE cell is below 100 Mbit/s. In this situation, data handling at the base station can have a major impact on latency and transmission times. First, data can queue up in the buffer of the

[2] For example, an exception is high-definition video conferencing, which is therefore rarely found in the public Internet.

base station when the instantaneous arrival rate is larger than the cell capacity, e. g. due to new traffic flows or rate fluctuations of the wireless channel. The resulting waiting time in the buffer adds to the end-to-end latency of the affected traffic and can be significant. Second, the radio bandwidth is a shared resource among all traffic flowing through a cell. When the rate bottleneck from end to end is the wireless link, it follows that the rate available to an individual connection depends on the amount of competing traffic and the behavior of the scheduler. In turn, the resulting rate determines the transmission time of a certain amount of data. Consequently, service differentiation and the scheduling algorithm in the base station can greatly influence the QoE of the served users.

2.2.3 Impact of Resource Fairness

Resource fairness is an important factor in scheduling tasks. It means that competing jobs have a similar chance to obtain resources. Especially in wireless networks, where the channel conditions are very unevenly distributed among users, it is important to maintain a certain minimum rate for every transmission. However, this comes at the cost of a reduced cell capacity as it is expensive in terms of radio resources to transmit a certain amount of data over an unfavorable channel (see Section 3.4.2).

Traditional fairness criteria applied in cellular networks are, Jain's fairness index [JCH84], a distribution-based fairness criterion (see [NGM08]), or comparing cell center and cell border users, for example. These fairness criteria are often applied to the rate or amount of resources the users get. However, these approaches are only suitable for full buffer traffic (also known as greedy traffic). Only when every user demands as many resources as possible, the average rate is well-defined. In contrast, the realistic traffic in a radio cell tends to be heterogeneous and bursty. When a user requests a small data object which only requires a fraction of the radio resources in a time slot, the rate relevant for fairness cannot be defined reasonably.

Furthermore, evaluating fairness is also a matter of the time-scale at which it shall be measured. It is very difficult to maintain fairness at a time-scale of milliseconds, because this would greatly reduce the flexibility of the scheduler. On the other side, for fairness at large time-scales (e. g. hours, days) the scheduler has effectively no influence, because the average radio channels of the users are very similar for such long periods. Also, the effect of fairness is only relevant at the time-scale at which it influences the application experience for the user. Transmission durations differing by some milliseconds may not be noticeable in most applications, whereas the average data rate during one hour is not relevant for the QoE of an interactive service.

In general, a lack of fairness becomes a problem when it results in a degraded user experience or a permanent deterioration in service quality for some users. This is especially significant when users have unfavorable channel conditions for a long time, e. g. when they stay at a place with bad cellular coverage. However, this cannot be expressed in terms of average data rate over some time and depends on the used applications. Therefore, fairness will be evaluated by discussing the distributions of transmission durations and user experience over all considered transmissions. This allows us to determine outage and inequalities between different transmissions and to assess how a scheduler could improve fairness and at which cost in terms of cell capacity and reduction of the average QoE.

2.3 Traffic Classification

A large number of network applications and Internet services exist. New services can emerge from one day to the other and existing ones may shrink in popularity and disappear over time. Therefore, it is not reasonable to reproduce today's Internet traffic and draw general conclusions from this. In this thesis, we will instead use a set of application classes which exhibit a common behavior and have similar requirements. The classification is defined with a focus on the suitability to assess QoE. Existing network applications can then be mapped to these classes without changing the fundamental findings qualitatively. Namely, we concentrate on interactive traffic, background traffic, and buffered video streaming. The majority of traffic volume and application requirements that are reported in the literature discussed above, e. g. in [San14], can be attributed to one of these classes. In the following sections, important properties of the respective classes will be introduced. Then, Section 2.3.4 also presents real-time traffic and argues why it will not be covered by the simulation studies.

2.3.1 Interactive Web Traffic

Interactive web traffic contains everything related to web surfing. Traditionally, this is the World Wide Web (WWW), where a user with a web browser types a Uniform Resource Locator (URL) or clicks a link and waits for the result to appear on the screen. This represents a classical client-server-architecture, in which the client (i. e. the user) requests content and the server responds and delivers this content. Until today, many other services like Internet shopping and social media emerged, sometimes also called *Web 2.0*. Compared to traditional web services, they offer increased client-side functionality by applications running inside the web browser (e. g. JavaScript) and customizable content. However, the mentioned web services have in common that a user interaction usually leads to a network transmission fetching requested content or uploading data. The user then sees the result once this content is available or the upload is complete.

Most web traffic uses HTTP. The transmission of a web-site usually consists of multiple transport layer connections comprising the transmission of the main object (main frame) and the embedded objects (e. g. images, layout, scripts, multimedia content). The *HTTP-Archive* tracks the properties of the most popular web-sites [HTT]. Although it tracks only the entry pages of the inspected web-sites, some conclusions about the traffic characteristics can be drawn. Compared to other traffic classes, the objects of web traffic are rather small. Especially transmissions for navigation within a web service, when layout and client-side scripts like JavaScript have already been transmitted, can be very small. As the user usually looks at the screen and waits for a response, it is crucial to transmit these objects fast. This is discussed in detail in Section 2.2.

Web traffic usually is *elastic traffic*. That means that the data rate adapts to the maximum possible rate on the whole path end-to-end; the higher the data rate, the shorter the transmission duration. Limits to this elasticity come from transport protocols, congestion on the transmission path and server behavior. For example, Transmission Control Protocol (TCP) requires some time to adapt to a change in the available data rate on the end-to-end path, e. g. caused by fluctuating cross-traffic or varying channel quality on the access link.

Through an extensive measurement study in a large commercial LTE network, [HQG$^+$13] shows that protocol interactions and application behavior can be a problem leading to an inefficient bandwidth usage. This is mainly due to TCP and application layer effects. [HQG$^+$13] reports undesired TCP slow starts and undersized receiver windows limiting the effective data rate. To reduce the effects from such rate fluctuations, especially the operators of cellular access networks use caching and proxy servers (so called *Performance Enhancing Proxies*, PEPs [RFC3135]) in their networks [XJF$^+$15]. Several proposals and evaluations of split-connection PEPs for 3G and LTE networks exist, e. g. [MSH03, IBL08, FHN12]. They demonstrate a significant improvement in the utilization of bandwidth resources on the wireless access link. Buffers in the access network further help to make fast channel fluctuations transparent to TCP. In future networks, router-assisted congestion control could help to improve the reaction time towards rate fluctuations [PSH09].

Such mitigation mechanisms limit the influence of the transport layer on Medium Access Control (MAC)-layer resource allocation. Furthermore, the influences from the application and transport layers are not in the focus of this thesis and are orthogonal to the performance comparison between the investigated MAC-layer schedulers. Consequently, we assume an ideal elasticity, i. e. an instantaneous adaptation to the available bandwidth of the access link for web traffic. In contrast, for the video streaming class (see Section 2.3.3), rate pacing by the content server will be considered as it limits the transmission rate on a much larger time scale.

2.3.2 Background Traffic

We attribute all traffic that is not interactive and neither belongs to the streaming nor the real-time classes to the background traffic class. That means that usually a user does not wait for the completion of a transmission without doing something else in the meantime. As background traffic, we classify for example the download of large files, application updates, e-mail synchronization, backup services etc. Often, a background transmission consists of a single or few large objects (e. g. a file download). The size of an object can be several orders of magnitude larger than the average web traffic object. Background traffic usually also is elastic, because many background applications use TCP at the transport layer.

Peer-to-peer traffic also belongs to background traffic, as it is normally used to exchange large files like software archives, videos, and music. While the transmission characteristics differ from classical web traffic, the principal behavior as well as the QoE requirements are similar. Therefore, we also assign peer-to-peer traffic to the background class.

2.3.3 Video Streaming

Today, a large fraction of mobile Internet traffic is video streaming [Cis14, San14]. There are some real-time video transmissions like live-TV and video calls, but the major part is buffered video streaming like YouTube [You]. Sandvine Inc. reports YouTube to be the top peak period application by volume for mobile access in North America and to be number two in Europe, Latin America, and Asia-Pacific [San14]. Furthermore, Alexa [Ale] reports *youtube.com* to be

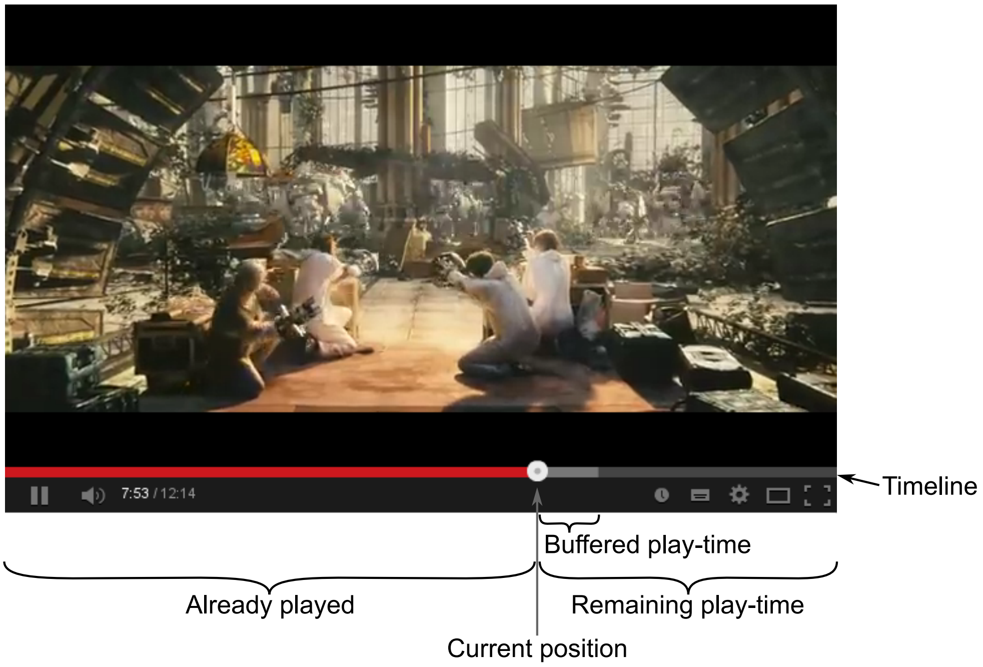


Figure 2.4: Screenshot of a video played in YouTube [You].

the third most popular web site in the world. Correspondingly, we assume its behavior to be representative for video streaming.

Buffered streaming means that the transmitted data is not played out immediately, but a buffer of usually some tens of seconds worth of playtime is maintained at the client side. The purpose of this buffer is to compensate temporary reductions of the effective data rate below the video rate, e. g. due to congestion or bad quality of the wireless channel. The advantage of saving the users from most network problems and rate fluctuations has led to the wide-spread adoption of this technique. Figure 2.4 shows a screenshot of the YouTube player (Flash-player embedded into the web-site). The indicated timeline provides information about the current position in the video and how much data is buffered ahead. When the buffer drops to zero, the video will pause and a re-buffering phase is required before playback resumes.

A user wishing to watch a video will send an initial request for it. Once the video plays, there are usually few user interactions ([FMM$^+$11] reports less than 2% of sessions with a resolution switch). In terms of scheduling, the playback duration is a very long time (the average video duration is in the order of several minutes). During this time, video data is transmitted according to the rate of the video. For a satisfactory video service, there should be no buffer under-runs as this would interrupt video playback. The median object size of such buffered video streaming is larger than for the other classes. [FMM$^+$11] reports a median of $\approx$10 MBytes. Popular servers of video streaming services throttle the data transmission, because users often abort a video after having seen a small fraction and the service providers want to save resources. This means that the elasticity of this kind of traffic is limited.

A proposed mechanism to react to changes in the available bandwidth or to comply with the capabilities of the client is adaptive bitrate streaming. A prominent example is Dynamic Adaptive

Streaming over HTTP (DASH), which has been standardized [ISO14] and has been specified for usage in 3GPP Release 10 [3GP14a]. The idea is that segments of the video are available at the server in different formats (i. e. different resolution, quality, etc.). Whenever the video rate exceeds the available bandwidth, the client is able to request the following segments with a lower encoding rate. If the transmission bandwidth increases again, the client updates video quality accordingly. By this, interruptions in playback shall be avoided and the video content shall always be delivered at the best possible quality. While DASH certainly is an interesting technique to improve the QoE of video streaming, it is out of the scope of this thesis. We concentrate on those aspects that can be controlled by scheduling radio resources at the MAC-layer. Adaptive bitrate streaming, as an application layer technique, can still be used on top of the investigated algorithms.

2.3.4 Real-Time Traffic

We specify applications with very short latency requirements per packet as *real-time traffic*. Notable examples are Voice over IP (VoIP), video-calls, and live-TV. When making a voice call, according to the ITU, the end-to-end latency of the audio signal ("mouth-to-ear") should not exceed 150 ms or the QoE will be impaired [IT03]. Depending on the codec and processing times at the end systems, several tens of milliseconds will be required for coding and decoding. Also the network latency and scheduling-independent delay of the access network add up to the delay budget. In the end, there is a limited flexibility for scheduling this kind of traffic. Consequently, many networks strictly prioritize packets transporting real-time traffic. Because of that, we will not investigate real-time traffic any further in this thesis. As discussed above, this only concerns a small fraction of today's traffic volume. The investigated schedulers could be easily extended by a hierarchical scheduler instance which strictly prioritizes real-time traffic.

2.4 Application Layer Objects – Transactions

In this work, the focus is on user experience. Therefore, it is important to model traffic such that we can evaluate effects visible to the user. Consequently, traffic objects are modeled at the application layer and the performance is evaluated by formalized QoE metrics. Such application layer objects are denoted as *transactions*. Transactions contain all network traffic that leads to a perceptible result for the user. They were developed in [PKV11] and [PKWV12]. An extensive work on the term, its meaning, definition and consequences has been done by M. Kaschub.

Modeling transactions is fundamentally different from modeling traffic at packet granularity. Packets are bound to a small object size, which means that – considering the scheduling granularity in our system – they have a similar size. In contrast, the size of transactions can differ by orders of magnitude, with some transactions containing several Gigabytes (e. g. a DVD image of a Linux distribution). More important, the transmission of single packets is not noticeable by the user. Only a finished transaction leads to an observable result. From this perspective, only the delay of the last packet in a transaction is important, while all the others are irrelevant for the user (as long as they are delivered before the last one). This will be represented in the QoE performance metrics, which consider transmission properties like duration or rate with respect to transactions.

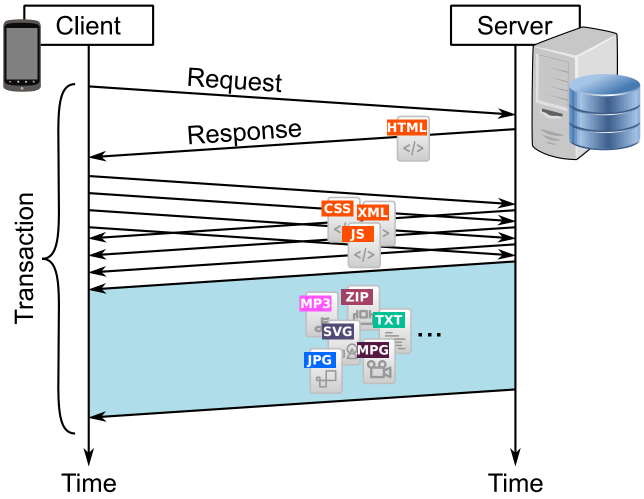


Figure 2.5: Message-sequence chart of an exemplary transaction illustrating a client requesting a web-site and its embedded objects.

In this section, we first define transactions and give examples which transmissions belong to a single transaction. An overview of the common properties is given and differences with respect to the application class, a transaction belongs to, are introduced. Then, utility functions for the evaluation of a transaction's QoE are discussed. Different application classes have different requirements for delivering a satisfactory experience to the users.

2.4.1 Transaction Definition and Properties

As introduced above, a transaction is defined from the user's perspective. Therefore, a transaction contains all transmissions that are necessary for a perceptible result. For example, in web-surfing a user clicks on a link and the browser sends a request towards a web-server as illustrated in Figure 2.5. Then, the server responds with the main Hyper-Text Markup Language (HTML) object, which possibly references to CSS, JavaScript, other scripts and content media like images, video or audio files. Requests for these embedded objects are issued by the browser and finally, when all necessary objects have been received by the client, the browser displays the web-site. For this type of interaction, the most important factor of the network transmission influencing the user experience is the delay between the initial request and the observable result, as we discuss in Section 2.4.2.

Such an approach directly considering application layer information can be denoted as cross-layer scheduling. In this thesis, the focus is on resource allocation at the MAC layer of the wireless link. Therefore, using knowledge from the application layer means a cross-layer concept. It goes beyond traditional resource allocation in that not only a single queue is kept per user device, but the scheduler has to distinguish between different transactions of the same user. This could be achieved for example by using the concept of *radio bearers* defined for LTE networks to classify different traffic streams[3]. Joshi *et al.* [JKPS00] argue already in the year 2000 for the upcoming CDMA systems the availability of job size information in the base station.

[3] We will discuss the cellular architecture in detail in Chapter 3

They refer to the *"isolation between the wired and wireless links"*[JKPS00] by using proxies and other techniques in the access network to be common practice[4]. Such proxies could deliver information on application layer objects. Furthermore, in modern cellular networks, it is possible to obtain transaction size information. For example, LTE specifications [3GP10] state that the Packet Data Network Gateway (P-GW) may use Deep Packet Inspection (DPI) techniques for packet filtering. With DPI available, for the web surfing example, one could inspect the HTTP header containing the content length as well as the HTML frame containing URLs to the embedded objects. Getting the content size of the embedded objects and combining this information then leads to the total size of the transaction. The accuracy could be extended by application layer signaling, with application programmers having an incentive to implement such a feedback channel in order to improve QoE.

Deviating from the above-mentioned example of web surfing, other applications may have different transmission patterns. We model different types of transactions corresponding to the traffic classes introduced in Section 2.3. Interactive web traffic and background traffic are modeled as elastic best effort traffic. We assume for this class that all data belongs to a single traffic object that arrives as a whole at the base station. Following the discussion in Section 2.3.1, we neglect HTTP interactions as well as transport layer effects like TCP congestion control. As the MAC layer queuing and transmission delay shall be evaluated, it is isolated by abstracting from the mentioned higher layer mechanisms. These interactions are independent from scheduling and would contribute in a similar fashion to the total latency experienced by the user. In actual systems, operators often deploy caches and proxies in their networks to avoid network and higher level effects, which supports the assumption to abstract from them.

Buffered video streaming has a different transmission behavior than best effort traffic. Instead of transmitting the whole object at once, the server only sends a fraction to ensure a certain playtime ahead of the current position in the buffer [FMM$^+$11, ARMNOLS12]. This has mainly two reasons. First, many users abort the video after some time before it has finished, which would mean a large waste of transmission resources for the content server. Second, mobile devices may not have sufficient memory to store the whole video [FMM$^+$11]. Therefore, we model video streaming traffic by slicing the video into chunks of equal size and model their arrival at the base station with an initial buffering phase and a throttling phase where the chunks are transmitted at a rate proportionally to that of the video. This means that the scheduler can only allocate radio resources corresponding to the amount of data already buffered in the base station. Furthermore, the QoE has to be evaluated differently than for best effort traffic, as we will discuss in Section 2.4.2.2.

An important property of a transaction is its size and especially the object size distribution of the accumulated traffic. The composition of transaction sizes defines traffic burstiness and influences to a great extent the behavior of the scheduler and its QoE performance. Many traffic measurements observe a heavy-tailed size distribution in the Internet [MSF10, FLM$^+$10, IP11]. Practically, this means that most of the objects are very small while most of the volume is contained in few large transactions. To model file size distributions, log-normal and Pareto distributions and combinations thereof are often used [NGM08, HCMSS04, Dow01]. We will discuss the parameterization of the object size distribution in Section 6.1.

[4] This is in agreement with the discussion on PEPs in Section 2.3.1.

2.4.2 Utility Functions

We formalize the evaluation of user experience, by using *utility functions*. The *utility* describes the QoE of a transaction, i. e. how the user experienced the transmission. Naturally, this depends on the requirements of the application and the adherence to these. We define utility for whole transactions, i. e. it can only be determined when the transaction is finished, either because it was completely transmitted or because the user aborted it. For example, in web surfing only the time when all packets have arrived and the result shows up on the screen is relevant to the user.

In this thesis, we employ scalar utility values between zero and one, where one means the best possible QoE and zero the worst, e. g. when the user aborts the transmission. Naturally, such a normalization leads to a common weight for all transactions irrespective of their size. A small transaction can achieve the same utility value as a larger one. This means that a small transaction is more "efficient" in terms of transmission resources per utility contribution. However, it is out of the scope of this thesis to model different utility weights for different types and sizes of transactions. It is prevalent in literature on QoE assessment to work with utility scores independent of the size of traffic objects. Widely accepted is the MOS, e. g. used in [IT05, FHTG10, ARMNO$^+$10, NUN10], which is the basis for the employed utility functions. The average utility metric will be complemented by observations of transaction durations and the distribution of utility over the size of transactions to get a comprehensive view on the QoE performance of a scheduler. Furthermore, we distinguish between the outcomes of a measured property for the different traffic classes. In Section 6.2, the metrics for performance evaluation will be introduced.

In the following, we discuss utility functions for the application classes evaluated in this work. Interactive elastic traffic comprises all transactions where the client requests information which can be used or displayed only after all data has been transmitted. Buffered video streaming models non-real-time services where a user can request content at any time (also known as *video on demand*). The most prominent example of such a service is YouTube [You].

2.4.2.1 Interactive Elastic Traffic

The utility functions for interactive elastic traffic were introduced in [PKWV12]. For services in this class, the duration of the complete transmission determines the utility. Therefore, the utility function for this type of transactions depends on the transmission duration. The longer the user has to wait for a result, the less satisfied he or she is. That means that the utility function must decrease strictly monotonic. We choose an S-shaped function, as illustrated in Figure 2.6, motivated by findings in [Nie10] and [NUN10]. Translating the time thresholds of [Nie10] (discussed in Section 2.2.1) to the utility functions means that utility is high for short durations, as the user cannot notice or expects a certain small delay. With increasing duration, utility steeply decreases, because the user feels interrupted and gets increasingly annoyed by the service quality. After some time, the utility cannot get worse, because the user is already very annoyed and just wants to get the requested result. Therefore, utility remains at a very low level. Eventually, the user will abort the transaction, which means that the utility drops to zero. [FHTG10] argues similarly and defines three areas depending on the respective QoS degradation (see Section 2.2.1). In [NUN10], the authors performed a survey on the MOS for

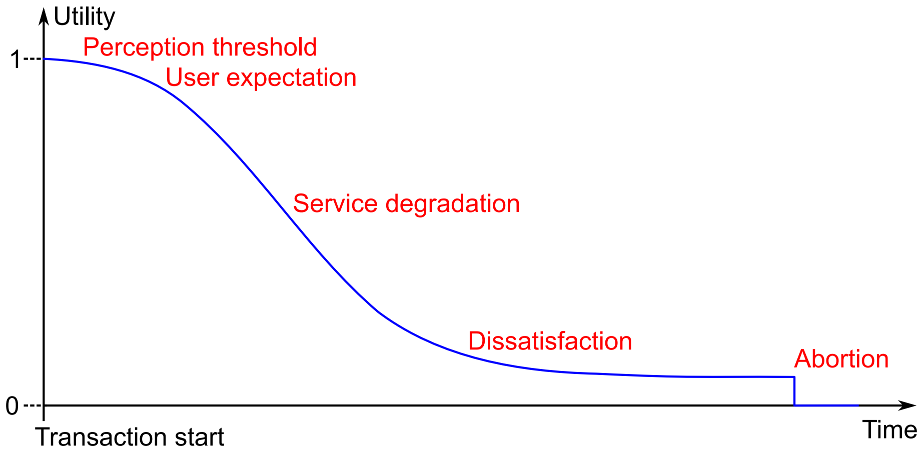


Figure 2.6: Evolution of utility depending on the duration of a transaction.

certain applications. Their findings agree on the fact that the duration is essential for the users' happiness. Their MOS values will be used to parameterize the utility functions in Section 6.2.3.

The shape of the utility function over transmission duration is influenced by the transaction size. First, interactive services which rely on a fast response usually transmit only small chunks of data. Consequently, such small transactions often have tight latency requirements. On the other hand, large transactions usually are less time-critical. For example, incrementally browsing through pictures from a web gallery versus downloading the whole archive and watching it afterwards. Second, users expect that large objects like music, video or high quality image files require longer to transmit than small objects. This expectation makes them generally more relaxed with respect to the transaction duration. The assumption that the users' expectations of response times is proportional to the object size is also supported by [MRSG99].

A metric describing a similar observation as our utility function for elastic traffic is the *stretch* metric introduced in [BCM98] (also called *slow down*). It is defined as the relation between the transmission time in an unloaded system and the actual transmission time. The model used here provides a similar performance metric because the expected duration is proportional to the transaction size. Minimizing average stretch is therefore similar to maximizing the average utility for elastic best effort traffic. In [RRSS05] strict packet-deadlines are relaxed with a z-shape towards a time-utility function which has a similar shape to our transaction utility functions. Another study observing that the expected transmission duration depends on the object size is [ERHS12]. Two MOS surveys are presented showing that users rate the same waiting time for a 10 MBytes file download much higher than a 2.5 MBytes file download.

2.4.2.2 Buffered Video Streaming

As mentioned before, the user experience with video streaming behaves differently than for interactive elastic traffic. Users already consume the video while the transmission is still ongoing. Again, the time between the user interaction and the start of video play-back has an influence on the QoE. However, as [HSH$^+$11] and [MCC11] show, the single most important influencing factor is the number of video stalling events. While video properties like resolution, compres-

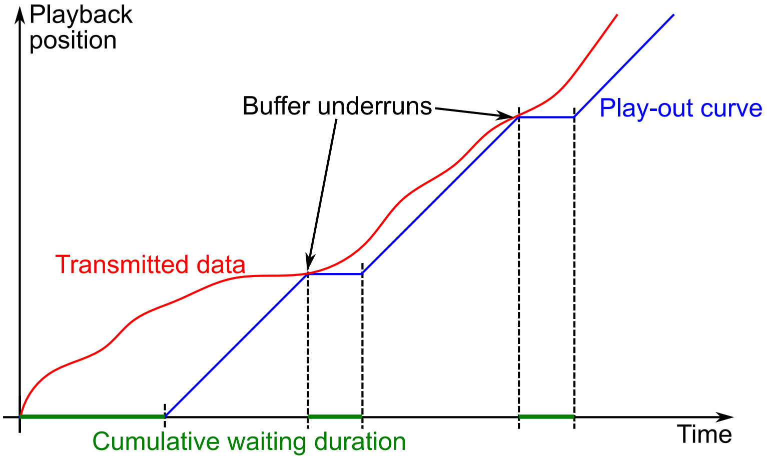


Figure 2.7: Illustration of the timeline for the playback of buffered video streaming with buffer underruns leading to stalling.

sion and quality and the type of the content may also affect QoE, we do not model them because MAC-layer scheduling has no influence on them.

Figure 2.7 illustrates an exemplary timeline of video playback. In the beginning, there is a prebuffering phase, during which the client-side buffer gets filled. When a certain buffer level is reached, the video starts playing. If during playback the transmission rate is below the video rate for some time, the buffer will shrink and may eventually run empty. In the case of such a buffer underrun, the video stops playing and a new buffering phase is required. Playback resumes when a sufficient buffer level is reached again. The resulting video interruptions, or *stalling events*, have a detrimental effect on QoE. In contrast, a short initial buffering time and playback without interruptions lead to high QoE.

Therefore, we model the utility function for buffered video streaming depending on the cumulative buffering duration and the number of stalling events. The parameterization of this model bases on the findings in [CSH13] and will be presented in detail in Section 6.2.3.2. Generally, playback without interruptions means a utility close to 1 with a slight impact of the initial buffering duration. Each stalling event, i. e. an interruption of video playback, decreases utility. The longer the interruption, the stronger is the utility degradation. The duration of an interruption depends on how fast the buffer gets refilled. The final utility of the video is determined by the number of interruptions and the relation between the accumulated stalling duration and the total video playback duration. When a single stalling event exceeds a maximum allowed threshold, the video transaction is aborted and the utility is $U = 0$.

3 Scheduling in Cellular Networks

This chapter first introduces radio propagation and cellular networks and distinguishes them from other wireless access technologies in Section 3.1. General possibilities how to divide the radio spectrum between different users are discussed in Section 3.2. Then, Section 3.3 describes the scheduling tasks, i.e. what needs to be considered when assigning radio resources to different users. In Section 3.4, existing concepts of cellular schedulers are introduced and classified according to the information sources they use and the optimization targets they have. An introduction to classical scheduling approaches in cellular networks will be given in Section 3.5 and in Section 3.6, the focus will be on algorithms applying the *Shortest-First* principle.

3.1 Radio Propagation and the Concept of Cellular Networks

The wireless transmission of information from a source to a destination can be achieved by modulating electromagnetic waves. On their way to the destination, such radio signals lose strength due to various factors. The combined effects from the propagation environment and the sender and receiver properties can be described in a model of the radio channel. First, due to expansion and depending on the medium, the signal loses strength with increasing distance between sender and receiver. This is called *path-loss*. Second, obstacles in the direct path between sender and receiver can reduce the signal power, which is often called *shadowing*. Finally, reflection and diffraction of the electromagnetic waves emitted by the sender at environmental obstacles can lead to multiple paths on which the signal reaches the receiver. The superposition of the signals from different paths can add up constructively or destructively at the receiver. In general, sender, receiver or environmental obstacles are in motion, which leads to a fluctuation of the signal attenuation on a small time-scale. This effect is often termed *fast fading*. An introduction to the effects and the modeling of wireless channels is given in [TV05].

For decoding the sent information at the receiver, it is crucial that the receiver can distinguish the useful modulated signal from thermal noise and interfering signals emitted from other radio senders. The relation between these signal strengths is expressed in the Signal to Interference-plus-Noise Ratio (SINR):

$$\text{SINR} = \frac{S}{N + \sum_i I_i} \tag{3.1}$$

Here, S and N are powers of signal and noise, respectively. I_i is the power of the interfering signal from interferer i.

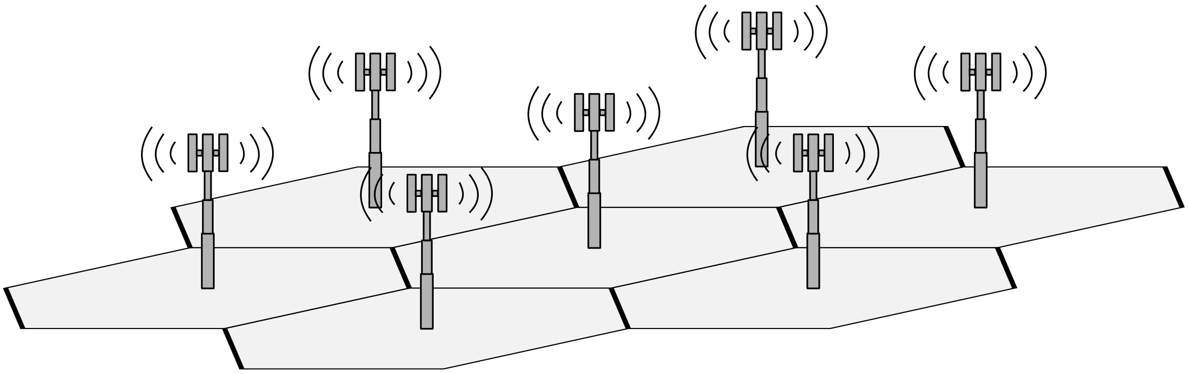


Figure 3.1: Typical hexagonal cellular layout.

The SINR, sometimes also denoted as *channel quality*, determines how much information can be transmitted within a certain bandwidth and time. The ideal bound for this *spectral efficiency* in a Single-Input Single-Output (SISO) channel is given by the Shannon-Hartley theorem [Sha49]:

$$C/B = \log_2(1 + \text{SINR}) \tag{3.2}$$

where C is the channel capacity in bits/s and B is the bandwidth in Hertz. The ratio C/B is the spectral efficiency in bits/s/Hz.

Together with the time-varying nature of the radio channel, Equation (3.2) shows an important point in wireless communications. For a given bandwidth, the channel quality between sender and receiver determines the bit rate of the transmission link. Furthermore, for a given spectral efficiency and transmit power, the maximum distance between sender and receiver and the maximum amount of allowed interference are limited. Depending on the prevailing term in the denominator of Equation (3.1), we speak of *noise-limited* or *interference-limited* wireless transmission systems.

As mentioned before, cellular networks offer one approach to exploit this design space. Base stations are deployed by an operator over the whole serving area. In contrast to WLANs, which operate in free (unregulated) spectrum and are freely placed by the end-users, cellular networks usually operate in a regulated spectrum and are deployed in a planned way in order to minimize regions of no connection (so called *coverage holes*) and maximize channel quality in the serving area. To this end, in the common case, fixed antennas are mounted to towers distributed over the whole area. In the ideal case of a flat serving area, this would lead to a hexagonal arrangement of base stations, as illustrated in Figure 3.1. In reality, network planning is conducted prior to the deployment of base stations to optimize their positions in the actual environment topology. All base stations have a connection to the operator's network, which is called *backhaul* and could be realized by optical fibers or microwave beam radio, for example. Users of the network move freely throughout the serving area and their *User Equipments (UEs)* connect to the base station with the strongest signal. The area served by an antenna or antenna array is called a *cell*.

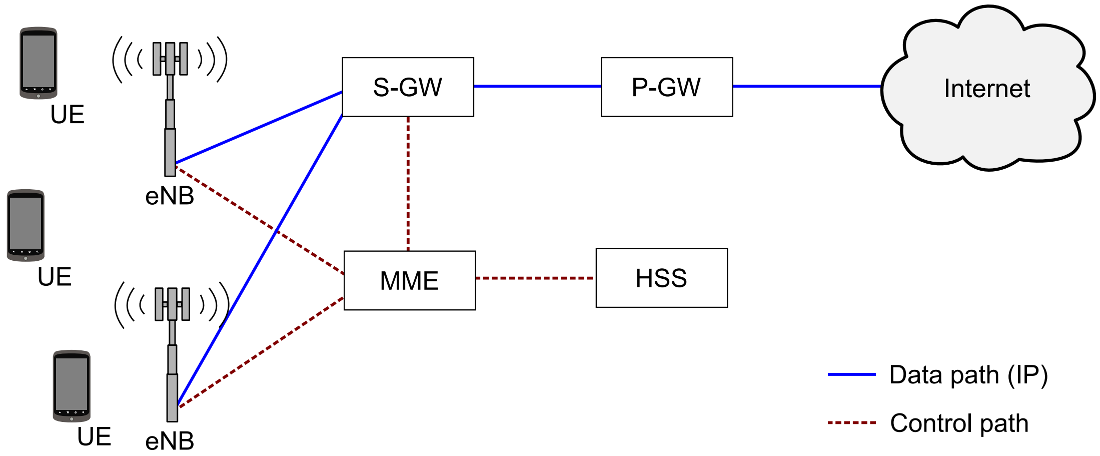


Figure 3.2: Simplified illustration of the LTE access network showing relevant components.

Cellular networks operate in a regulated spectrum, where the base stations assign radio resources to the users. This enables much more efficient spectrum sharing technologies and interference coordination compared to wireless networks using unregulated MAC algorithms like Carrier-Sense Multiple Access (CSMA), for example. Spectrum sharing will be discussed in Section 3.2. In general, base stations and UEs are produced by different manufacturers, so standardization is required especially at the physical (PHY), the Medium Access Control (MAC), the Radio Link Control (RLC) and the network layers (layer 1–3).

The major standardization body of the third and fourth generation of wireless cellular networks is the 3rd Generation Partnership Project (3GPP). The most advanced 3GPP standards currently deployed are LTE and LTE-Advanced, also called the fourth generation of cellular networks. With LTE, the underlying network architecture became an all-IP network, developed under the term System Architecture Evolution (SAE). The idea was to flatten the architecture for cost savings and increased performance [Fir].

Figure 3.2 illustrates the core components of the Evolved Packet Core (EPC) and introduces some 3GPP terminology. An overview of the LTE radio access network is given in [3GP10]. The users' data IP-packets travel along the data path indicated in Figure 3.2. The *Packet Data Network Gateway (P-GW)* is the first-hop router towards the Internet from the view of the UE. It is responsible for IP address allocation and can be used for per-packet filtering and inspection. Directly connected to the base station, or *enhanced Node B (eNB)*, is the *Serving Gateway (S-GW)*. It forwards data packets from the UE to the Internet, called *uplink*, and vice-versa, called *downlink*. Furthermore, the S-GW represents the "mobility anchor" during handovers, terminates the data path when the UE is in idle state and signals to the UE when new data packets arrive (*paging*).

For control signaling, the *Mobility Management Entity (MME)* is the most important entity, often co-located with the S-GW. All Non-Access Stratum (NAS) signaling goes through the MME. Among other things, it is responsible for authorization, access control, handover management including the selection of S-GW, P-GW and possibly the next MME and the establishment of the data link (called *radio bearer*). The *Home Subscriber Server (HSS)*, which basically contains a data base with user and subscription information, offers this information to the MMEs of the network operator.

Apart from the core network (EPC), Figure 3.2 also illustrates eNBs making up the actual Radio Access Network (RAN). In LTE, the eNBs are responsible for layer 1 and 2 (physical and data link layer) of the actual uplink and downlink transmissions towards the UEs in the form of radio signals. The procedures of the physical layer provide a transport service to the higher layers by means of so called *transport channels*. At layer 1, the eNB performs error detection (using Cyclic Redundancy Checks (CRCs)), coding (Forward Error Correction (FEC)) and matching the coded bits to the modulated symbols which are then transmitted as radio frequency waveforms [3GP09a]. The spectral efficiency is determined by the selection of the Modulation and Coding Scheme (MCS) and the resulting Block Error Rate (BLER). Layer 2 mainly includes Radio Resource Management (RRM), error correction by retransmission through Automatic Repeat Request (ARQ) and Hybrid-ARQ (HARQ), compression and encryption of data packets and forwarding them to the EPC. RRM means that the eNB dynamically allocates radio resources in uplink and downlink and performs admission control in overload situations [3GP10].

With the example of LTE, the most important properties and functions of cellular networks shall be introduced. A synonym for cellular network is *mobile network*, as it is able to serve mobile users and to provide a seamless connectivity to these traveling users. This is achieved by *handing over* a UE from one eNB to another without interrupting an ongoing session, e. g. a voice call. The higher level protocols of the core network (NAS) ensure that the UE keeps the same IP address and the session context is migrated from one eNB to another. All eNBs have a backhaul link to the core network as introduced above. So called *relay nodes*, which use the same frequency band for the backhaul and for the connection to the UEs, are specified to extend the reach of an eNB with a connection to the core network. However, they are not considered in this thesis.

Another important capability of cellular networks is QoS provisioning. Because the shared medium of the wireless link is centrally managed, service differentiation and admission control can be performed. This makes it possible to offer QoS guarantees like a low latency or minimum data rates to the users. Usually, users pay their ISP for connectivity, traffic volume, maximum data rate or other service metrics. For this, cellular networks provide the means for traffic shaping, authentication, and accounting to charge the users according to their network usage.

3.2 Spectrum Sharing Technologies

As introduced in the previous section, the base stations in cellular networks allocate radio resources for transmission. The meaning of such resources depends on the technology in use and can be defined along different dimensions. The basic physical dimensions along which resources can be separated are space, time, and frequency. Depending on the technologies, the freedoms along these dimensions are used differently, which is shortly presented in the following.

Time-Division Multiplexing (TDM) separates the channel access of transmitters in time. Resources are defined as time slots. Users competing for transmission use the whole channel

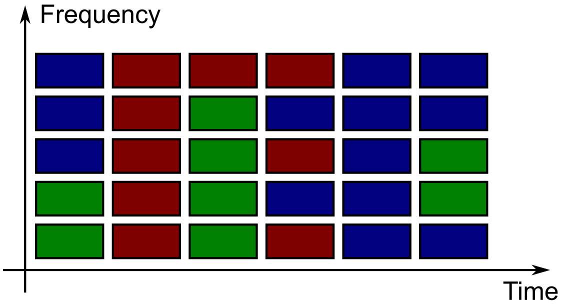


Figure 3.3: Grid of radio resources for an OFDMA system.

bandwidth for sending in the allocated time slots. An example for a cellular system using *Time-Division Multiple Access (TDMA)* is GSM, which uses time slots of $577\ \mu s$ length. Important for TDMA is the synchronization of transmitting devices and guard times between two consecutive time slots in order to avoid a collision of signals, which can lead to the receiver not being able to decode the transmitted information.

In *Frequency-Division Multiplexing (FDM)*, different frequency bands are used for competing transmissions. The whole radio spectrum is divided into bands for different services and controlled by national regulation bodies. For example, radio stations broadcast at the Very High Frequency (VHF) band and at the same time WLANs operate in the unregulated 2.4 GHz band. The multiple access technique *Frequency-Division Multiple Access (FDMA)* is employed for example in satellite communications, because it is robust with respect to timing and synchronization, which is a big issue for large ranges. Because it is not possible to create sharp edges in the spectrum (the sharper the edge, the more complex the frequency filter) and because the Doppler shift can change the frequency for moving antennas, guard bands exist to separate transmissions on neighboring frequencies and reduce their interference.

Code-Division Multiple-Access (CDMA) not directly maps to a physical dimension but rather on the correlation in time of the transmitted signal and a *code* known in advance at the receiver. It is important that different codes are orthogonal to each other such that the correlation with the wanted signal is high and interfering signals are rejected. Because all senders transmit on the whole frequency band, it is also called a spread-spectrum technology. Third generation cellular networks like *UMTS* and *CDMA2000* employ CDMA.

In LTE, *OFDMA* is used in the downlink and *SC-FDMA* in the uplink together with TDMA. That means that resources are divided in time and frequency. Accordingly, LTE defines Physical Resource Blocks (PRBs) in a two-dimensional grid in time and frequency, as illustrated in Figure 3.3. These PRBs can then be allocated by the scheduler to the users, as described in Section 3.3. In comparison to ordinary FDM, Orthogonal Frequency-Division Multiplexing (OFDM) uses orthogonal sub-carriers eliminating interference between them and allowing to place many more sub-carriers in a given bandwidth. SC-FDMA simplifies the transmission by modulating the OFDM-signal onto a single carrier. That makes it suitable for battery-limited handheld devices[1].

[1] This technique reduces the Peak-to-Average Power Ratio (PAPR) and thus allows for a reduced size and consumption of the power amplifier in the UE [STB09, pp.345–346].

Wireless networks in general and especially cellular networks inherently separate transmissions in space due to the limited reach of the signal. An emitted signal gets weaker with increasing distance from the sender, so another sender can operate in the same band at an appropriate distance. At the location of the intended receiver, the interfering signals will be much weaker than the wanted signal and the receiver is able to decode the transmitted information. Especially for CDMA networks, flexible power allocation is important to serve users at different distances from their base stations, because serving areas are designed to overlap and all transmissions are in the same spectrum.

By deploying more base stations in a certain area, i. e. densifying the access network, more users can be served in this area and frequency band. On the downside, this approach increases the cost of the infrastructure. Technologies explicitly exploiting the spatial dimension are *beamforming* and *Multiple Input and Multiple Output (MIMO)*. Beamforming uses directional antennas which have a focused signal radiation in the main direction and reduced radiation to other directions. In this way, it is possible to direct a "beam" towards a receiver and reduce the interference for other directions of departure. MIMO requires multiple, spatially separated antennas at both sides. Then, it is possible to exploit different characteristics of signal propagation paths between different antenna pairs. MIMO can be used to either send different information streams to one or several receivers (*spatial multiplexing*) or to send the same information along different paths to increase *diversity* (e. g. by the use of Space-Frequency Block Coding (SFBC)). Furthermore, comparable to beamforming, MIMO can be employed to control the signal strength at certain receiver locations via *precoding*.

Similarly to separating the transmissions between different users, the uplink and the downlink have to be separated. In this case, only time and frequency dimension are possible as the locations of the communication pair are identical for both directions. The respective methods are called *Time-Division Duplex (TDD)* and *Frequency-Division Duplex (FDD)*.

In contrast to CSMA, which is applied in WLANs[2], the multiple access techniques of cellular networks have the advantage that they are coordinated by the base station. That means that there is no need to listen on the channel whether it is idle and by design no collisions can occur[3]. This enables a much higher utilization of the shared medium.

For all of the aforementioned technologies, there are limitations with respect to the minimum granularity at which resources are separated. Standard specifications define the granularity by considering the size of data blocks, overhead, and technical limitations.

In this thesis, we demonstrate scheduling with the example of an LTE configuration. It is a wide-spread standard in today's commercial cellular networks and is devised to replace the GSM and UMTS standards[4]. We therefore use radio resource blocks defined in time and frequency as specified by the 3GPP for LTE. However, the demonstrated scheduling principles are

[2] The wide-spread IEEE 802.11, also denoted as WLAN, uses CSMA / Collision Avoidance (CSMA/CA) in case of the mostly used Distributed Coordination Function (DCF). Another method is the Point Coordination Function (PCF) where a central access point coordinates the stations, but this is rarely found in practice.

[3] Only in the uplink, a small fraction of the resources is reserved for the Random Access Channel (RACH). A newly activated UE uses the RACH to register for the first time at the serving base station.

[4] Some trial implementations of *LTE-Advanced*, which is the evolution of LTE and has many common properties, exist.

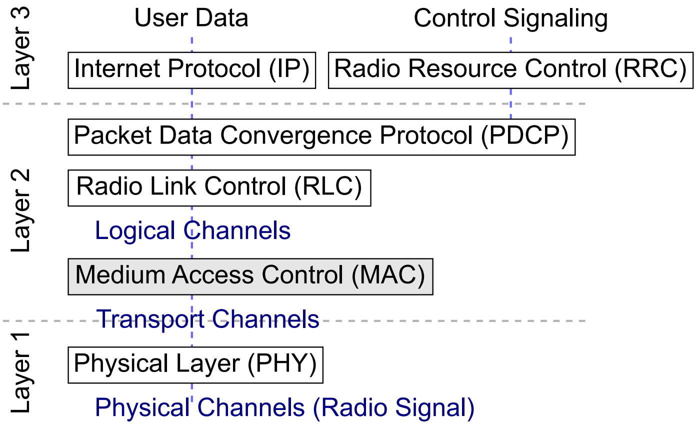


Figure 3.4: Layers 1–3 of the LTE protocol stack (simplified view).

not limited to this manner of spectrum sharing. The proposed algorithms can also be adapted to different technologies like CDMA, for example. The central question is which transmissions compete for the shared medium, a question which is relevant for all cellular network technologies. MIMO and power allocation will not be considered in this thesis. For our observations, MIMO just adds a degree of freedom (and thereby additional complexity) without changing the fundamental problem. Power allocation could be performed on top of the discussed resource allocation. Furthermore, the focus is on the scheduling in the downlink of a single cell rather than interference coordination between different cells, for example. This would have to be done on a higher-level control loop, which is out of the scope of this thesis.

3.3 Tasks of Scheduling

To classify the role of scheduling, we look at the lower layers of the LTE protocol stack, illustrated in Figure 3.4. It shows the protocol entities as specified by the 3GPP [3GP09a, 3GP10]. For all protocols at layers 1–3, there is a logical one-to-one connection between the UE and the eNB.

Scheduling, called Dynamic Resource Allocation (DRA) by the 3GPP, is a task inside the MAC-sublayer of layer 2. It will be detailed in Section 3.3.1. The MAC-sublayer is also responsible for mapping and multiplexing logical channels from the higher layers to transport channels which will then be processed by the PHY. Furthermore, retransmissions for HARQ are scheduled at the MAC layer and the transport format, i. e. the Modulation and Coding Scheme (MCS), is selected.

The PHY is responsible for emitting the data bits from the transport channels as a radio signal in so called *physical channels*. This mainly comprises error detection, channel coding and decoding, modulation and synchronization. LTE uses turbo codes for FEC and HARQ-soft combining. The code rate determines, how much redundancy is added to the data bits, i. e. how robust the transmission is against bit errors. Different modulation schemes – Quadrature Phase Shift Keying (QPSK), 16-Quadrature Amplitude Modulation (QAM), and 64-QAM – are used in LTE for data transmission [3GP10]. They define how many bits are modulated in one

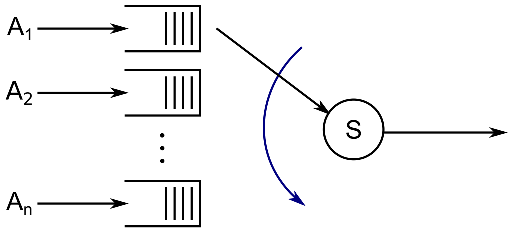


Figure 3.5: Simple queuing system illustrating the task of a scheduler.

OFDM-symbol. Together, code rate and modulation define the number of bits per PRB, which have to be matched to the number of bits to be sent (*rate matching*).

Directly above the MAC-sublayer is the RLC. It is mainly responsible for segmentation and aggregation of higher layer Protocol Data Units (PDUs) to transport blocks and for error recovery by retransmission (ARQ). The topmost sublayer of layer 2 is the Packet Data Convergence Protocol (PDCP) which takes care of header compression and ciphering.

On layer 3, user data is transported via IP and the Radio Resource Control (RRC) handles control procedures. The latter comprise, among others, mobility management functions like paging, cell selection, and handover.

3.3.1 Allocation of Radio Resources

In the general context of cellular networks, scheduling means the task of assigning radio resources to waiting transmissions. Figure 3.5 shows an abstract view on this task illustrated as a queuing system. Data objects to be transmitted arrive according to a random traffic process. They are buffered in different queues, distinguishable by the scheduler. For example the traffic from different users, from different radio bearers, or even from different applications can be classified upon arrival and will be placed in the respective queue. The service unit represents the shared medium of the competing transmissions. In cellular networks, the service time is highly variable, as it not only depends on the amount of data to be transmitted but also on the quality of the radio signal between sender and receiver. The scheduler decides from which queue data will be transmitted next.

Scheduling is sometimes also denoted as resource allocation, because it means mapping data bits to the physical radio resources. In the LTE case, radio resources are structured into PRBs, as illustrated in Figure 3.3. Theoretically, the scheduler could assign each PRB to a different UE. However, the standards specification restricts the number of UEs served simultaneously, because it would mean a large signaling overhead and the benefits would be limited. We will not consider this restriction, because it has a small influence on the investigated scheduling ideas. At a given point in time, the scheduler has to decide which UEs to serve and how to distribute the PRBs among them. For this decision, it considers several information sources and constraints like channel quality, fairness constraints, and service requirements. This will be detailed in Section 3.5. According to the scheduling decision, data bits will be put into a transport block of appropriate size and processed by the PHY.

3.3.2 Relevant Time Scales

When discussing scheduling, an important question is at which time scale it reacts and how it influences the transmission behavior at larger time scales. In LTE, the scheduler typically operates at Transmission Time Intervals (TTIs) of 1 ms, which corresponds to one LTE *sub-frame* [STB09, p. 285]. Directly related to the scheduler are the feedback loops for HARQ and Channel Quality Indicator (CQI). For HARQ operation, the receiver signals if it was able to decode the transport block or if additional information is required. In the latter case a retransmission is scheduled after 8 ms [STB09, p. 234].

The UE measures the received signal strength and reports this by means of the CQI to the eNB. Different operation modes are specified for CQI measurement and feedback leading to varying latencies. An average age of 5 ms for the CQI is a common assumption. This latency is especially important with respect to the channel coherence time, which determines how fast the CQI becomes obsolete. The coherence time depends on the Doppler shift and with it on the speed of the user. As a rule of thumb, the coherence time is about 100 ms for a pedestrian user and about 10 ms for a fast vehicular user. Together with the latency of CQI reporting, this makes it hard to adapt scheduling to the channel of a vehicular user.

Another important interaction is that between scheduling and traffic characteristics. In a typical LTE system, an average transaction may require from less than a second up to some minutes to be transmitted. This is the time scale relevant to the user. Users have a good experience when transmissions complete quickly and may get annoyed when they have to wait. According to [Nie10], about 100 ms is the threshold for feeling an instantaneous response (we also discussed this topic in Section 2.4.2.1).

The focus of this thesis is on the influence of scheduling on user experience. This means application-dependent time requirements recognized by the user are of interest. We therefore need to model all effects contributing at a time scale of less than a second up to several minutes. We will not rely on a *time scale separation* assumption as sometimes found in the literature (see Section 3.6.2). Instead, it will be shown how the user-relevant behavior emerges from the slot-level decisions of the scheduler at a millisecond basis. This means that we need to capture fast and slow channel fluctuations and the transmission of application layer traffic objects over time.

The behavior of the transport layer, especially TCP congestion control, can have an influence on the available data as well. TCP reacts on a time-scale of Round-Trip Times (RTTs) between the end systems, usually in the order of 100 ms. However, as discussed in Section 2.3.1, PEPs and other mechanisms exist to mitigate the influence of TCP on scheduling. We thus do not consider TCP in our model.

3.4 Classification of Cellular Schedulers

A large variety of schedulers for cellular networks exists. They can be classified according to the amount and type of information they consider. Furthermore, we revisit the different optimization targets behind common scheduling strategies in this section.

3.4.1 Information Sources

The minimal information every scheduler requires to be a *work conserving* scheduler is the amount of data in the queues. Work conserving means that all resources are used for transmission when data is buffered in any queue. With information about the amount of data in a queue, the scheduler only serves non-empty queues and reserves at most the necessary resources to transmit the backlogged data. Still, padding may occur due to the granularity of resource units.

An important information in wireless communications is Channel State Information (CSI). Many widely used cellular schedulers use this channel knowledge to improve the capacity of the system. This is called *opportunistic scheduling* and exploits the effect of *multi-user diversity* [VTL02]. This means that a UE is served preferentially when it has a relatively good channel quality because the spectral efficiency is higher than for a bad channel quality, i. e. the same amount of resources can transmit more data.

There is a range of different forms of application knowledge, which can be used for scheduling. Such cross-layer information can have different degrees of complexity. The simplest approach is to use different traffic classes for class-based prioritization. This information can be transported for example via the network layer. The IP header contains a 6 bit field called DSCP (see Section 2.2.2), which is employed by IMS in LTE-networks to classify the QoS level of a radio bearer. Besides classification, the scheduler also needs to know the requirements of the respective traffic classes. In the simplest form, this means a static prioritization between the classes. More advanced is the specification in terms of QoS requirements, e. g. the average bandwidth or packet delay, a certain class should obtain. Usually, this information is statically configured by mapping requirements to QoS-classes. To this end, 3GPP defines the QoS Class Identifier (QCI) in LTE, which encodes standardized requirements (e. g. for voice or video conversations) and contains a mapping to DSCP.

Going further, a scheduler can rely on DPI to get information about a traffic flow directly from the application layer. A wide-spread use case is the interpretation of the HTTP header, which contains, among other things, information about content type and length. Also for other applications known to the inspecting element, individual requirements could be derived, e. g. the average rate of a video stream.

Another idea of advanced application knowledge is support from the client-side. In contrast to the class-based approach discussed above, no predefined QoS classes are used for this. The client could inform the scheduler about the exact requirements and current state of its network transmissions. This idea of Context-Aware Resource Allocation (CARA) was followed and thoroughly discussed in [PKWV12]. Assuming ideal knowledge of application requirements, an intelligent scheduling algorithm could plan resource allocation into the future to improve the overall QoE of the users.

3.4.2 Optimization Targets

When a cell is not fully loaded, i. e. when not all radio resources are required to transmit the buffered data, the scheduling algorithm has practically no influence. Scheduling becomes relevant when all resources are occupied and data gets buffered. Then, the scheduler decides which

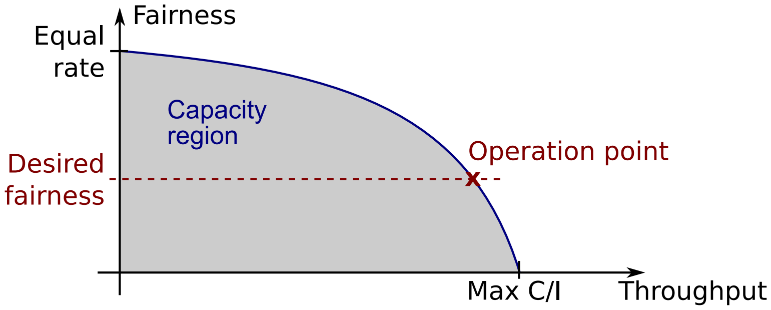


Figure 3.6: Capacity region of a cell showing the trade-off between fairness and throughput.

of the waiting transmissions to serve and which ones to defer. If such an overload situation lasts for some time, it may not be possible to satisfy all requirements. According to the optimization targets the scheduler was designed for, it will decide which requirements to violate first.

One optimization target in cellular networks is maximizing the capacity of the system. Because the spectral efficiency depends on the SINR according to Equation (3.2), there is a direct connection between the cell capacity and the scheduling decision. Opportunistic schedulers aim to maximize capacity by scheduling users with good channel conditions in order to improve the overall spectral efficiency.

However, if this would be the only scheduling criterion, situations can occur, where a user with a low SINR gets very few resources. This would lead to a very bad service quality for this user and operators want to avoid this situation. Therefore, another optimization criterion is fairness. Different definitions of fairness exist. Well-known for cellular networks are Jain's fairness index [JCH84] and the Cumulative Distribution Function (CDF) based fairness criterion, see [NGM08]. Another frequently used metric is the relation between the average user throughput and the 5%-ile. This translates to comparing the performance of a user located at the border of a cell with the average performance. These fairness metrics can be applied either to radio resources or to data rates.

Optimizing fairness always reduces cell capacity, because it inherently means taking resources from a high-SINR user and giving them to a low-SINR user. This means that all schedulers have to trade fairness for capacity. The area below the Pareto optimum of both metrics, as illustrated in Figure 3.6, is called *capacity region*. In [PMB10] this trade-off was investigated with respect to dynamically adjusting the scheduler to the current traffic situation in order to optimize throughput while meeting the desired fairness requirement. This means to maintain an operation point as shown in Figure 3.6.

Rate fairness restricts the scheduling decision more than resource fairness, because users with a low SINR require more resources than users with a high SINR. This limits the multi-user diversity that can be exploited by opportunistic scheduling. Furthermore, it is important on which time-scale fairness shall be enforced, because it leaves more or less flexibility for the scheduling decision. If fairness is to be guaranteed on a very short time-scale, there is a low degree of freedom for other decision criteria. A proposal for a fairness definition at different time-scales is presented in [AAR12].

Another important optimization criterion is delay. For some applications, like telephony, network latency is the major factor influencing the service quality. In a heavily loaded cell, packets get buffered and the waiting time in the queue increases. Some schedulers aim to minimize the packet delay in order to mitigate this problem. However, in most applications other than voice telephony, not the delay of the individual packets but the delay of whole transactions is relevant. In literature this is often denoted as flow time optimization. In [BCM98], the authors introduce the *stretch metric* which is the relation between the actual finish duration of a flow and the duration in an unloaded system. When discussing size-based schedulers in Section 3.6, we will further elaborate on transaction durations as they are essential for the QoE we aim to optimize.

3.5 Classical Scheduling Approaches in Cellular Networks

This section introduces the state-of-the-art for cellular scheduling without considering size-based schedulers, which will be covered in detail in the next section.

3.5.1 Round Robin

The simplest scheduler is *Round Robin (RR)*. As the name suggests, it implements a taking-turns principle, i. e. one resource unit is given to a transaction, the next resource unit is given to the next transaction, and so on. This means that *RR* is agnostic to the channel situation and cannot exploit opportunistic gains. It can be seen as an implementation of the Processor Sharing (PS) principle with limited granularity. *RR* therefore offers a perfect resource fairness, but has no advantages with respect to spectral efficiency or QoS. A variant is *Weighted Round-Robin* which allows to give different weights to competing transactions. The weights determine the fraction of resources, a transaction gets during one round.

3.5.2 Maximum Channel Quality

The most straight-forward way to maximize the capacity of the cell is to always assign resources to the UE with the best SINR. This maximizes the multi-user diversity gain under a full-buffer assumption[5]. An early work stating that only the user with the best channel quality should transmit is [KH95] which discusses uplink channel access and power allocation. This central finding also applies to the downlink situation of an OFDMA system as considered in this thesis. Adapted to fit here, it means that the scheduler assigns each PRB to the UE with the best SINR at the respective frequency and time. The widely-used name for this scheduler is *Max C/I*, with C/I meaning the ratio between Channel and Interference. This corresponds to the notation SINR used in this thesis.

[5] When assuming a realistic traffic behavior, this strategy may not always lead to the maximum average throughput due to an interdependence with the user and channel distributions.

3.5.3 Proportional Fair

A very common scheduling principle for cellular networks is *proportional fairness*. It combines the idea of opportunistic scheduling with fairness and was introduced in [Kel97]. The optimization target is to

$$\text{maximize} \sum_{x} \log(R_x), \tag{3.3}$$

where R_x are the average rates of the active users x. It relaxes the strictness of *max-min fairness*[6] in that a large increase of one user's rate can compensate a small decrease of another user's rate [Kel97]. [JPP00] presents an application of this principle to cellular networks in a simple algorithm. Today, this *Proportional Fair (PF)* algorithm is applied in many different forms and notations in cellular networks. The *PF* scheduler assigns a resource unit to the transaction with the largest weight w_{PF}, with

$$w_{PF} = \frac{R(t)}{\overline{R}(t)} \tag{3.4}$$

where $R(t)$ is the instantaneously possible data rate of a user and $\overline{R}(t)$ is the exponential moving average of the data rate. It is updated as follows:

$$\overline{R}(t+1) = \begin{cases} \beta \cdot R(t) + (1-\beta) \cdot \overline{R}(t) & \text{if scheduled} \\ (1-\beta) \cdot \overline{R}(t) & \text{else} \end{cases} \tag{3.5}$$

Here, β is the so-called forgetting factor controlling the decay rate of earlier values. A small value of β means that the moving average changes slowly. With Equation (3.5), the moving average follows the average throughput a transaction obtains. When a transaction does not get resources for some time, the moving average decreases and makes it more likely that this transaction will be served in the next slot. Usually, the moving average is initialized with a value close to zero for new transactions, so that they get served preferentially until their moving average converges to the average throughput.

When the rate fluctuations of the users are statistically identical, a *PF* scheduler will assign in average the same amount of resources to all users [Bor05] (i. e. it is *resource fair*). Furthermore, [Bor05] shows that with some simplifying assumptions, *PF* can be modeled as a PS system.

Variations of *PF* exist, which make it more robust against different channel properties or allow to specify minimum and maximum rate requirements for the individual users. To address the first issue, [Bon04] proposes the *score-based scheduler* which serves a user when the current rate is large compared to the recent possible rates. [BMZ08] goes into the same direction by looking at the quantile of the current rate compared to previous rates. An extension to *PF* with maximum and minimum rate constraints is given in [AQS05]. A token counter is applied to track the average rates of the users and integrated into the scheduling algorithm.

3.5.4 Queue-aware schedulers

The classical scheduling approaches were developed with a *full-buffer* assumption in mind. Fairness is defined on the mean rate, assuming users always want to transmit something. How-

[6] *Max-min fairness* means that the minimum rate shall be maximized.

ever, in real situations, users do not always have data to transmit. For performance evaluation, this is modeled by random traffic arrivals and per-user queues. With such fluctuating traffic conditions, much research has been done on the *stability region* of a cellular system. In this context, *stable* means that no queue would grow without limits for the theoretical case without packet losses. So-called *throughput optimal* schedulers that offer the maximum stability region have been proposed in literature. They are either aware of the packet delay or the queue-length.

A well-known case is the Modified Largest Weighted Delay First (M-LWDF) algorithm proposed in [AKR+01], which was designed to serve the maximum possible number of real-time users without violating their delay requirements. It schedules the user with the maximum weight w_{MLWDF}:

$$w_{MLWDF} = aW(t)\frac{R(t)}{\overline{R}(t)}, \tag{3.6}$$

where a is a parameter derived from the delay threshold and its violation probability, and $W(t)$ is the Head of Line (HoL) packet delay. [AKR+01] states that $W(t)$ could also be replaced by the queue length.

Other throughput optimal schedulers are for example exponential rule [SS02] and log rule [SBdV11]. The log rule improves over M-LWDF and the exponential rule in that it, besides being throughput optimal, additionally tries to minimize the average packet delay.

3.5.5 QoS and QoE-Based Schedulers

Many schedulers aim to optimize QoS parameters of the users' applications. A common way is to apply service class differentiation to distinguish traffic with respect to the applications' requirements. [Nec06] gives an overview of schedulers applying service class differentiation. The central point is the separation of real-time and non-real-time traffic, i. e. traffic with strict per-packet latency requirements and best effort traffic. For the real-time traffic, deadlines may be introduced and queues prioritized, in which the HoL packet-delay is large or where deadlines are in danger of being violated. For the non-real-time traffic, the overall throughput is more important. For this, the schedulers usually exploit multi-user diversity by scheduling queues with a good instantaneous channel quality.

Other scheduling approaches define utility functions for the different service classes [SL05, RRSS05]. In [SL05], a utility scheduler is proposed that considers average packet waiting times and the instantaneous rate of the users. The shape of the utility functions is chosen differently for real-time and best effort traffic in order to offer good delay performance to real-time traffic while still offering a good throughput to best effort traffic, when possible. [RRSS05] proposes a similar approach named *urgency and efficiency based packet scheduling*, which uses a combination of time-utility functions and channel quality. It schedules packets from the queue with the largest product of utility decline and relative channel quality. The utility functions depend on the type of the traffic. For real-time traffic, they have a steep decline shortly before the packet deadlines and for non-real-time traffic they are slowly declining over the waiting time.

More recently, the focus of scheduler development has shifted to directly improving QoE. [ARMNO+10] provides a mapping of MOS to rate utility functions for web browsing and incorporates these into the scheduler of Song and Li [SL05]. An interesting aspect is that [ARMNO$^+$10] also incorporates the size of a web site in the utility functions, so to some extend it comes close to size-based schedulers, which will be covered in the next section. In [PKWV12], a similar concept was demonstrated, where the scheduler is assumed to have an exact knowledge about the transactions' requirements and incorporates this knowledge in optimizing the resource allocation. Utility functions for transactions were introduced and a scheduling heuristic was presented that optimizes the total utility by planning the resource allocation of the active transactions for their complete transmission. A similar approach is presented in [KDSK07] which proposes a MOS-based scheduler. Functions relating data rate and packet error probability to MOS for different applications (voice, video and file downloads) are integrated into a greedy scheduling heuristic which aims to optimize the weighted average MOS. This concept is extended in [TKSK09] and adapted to High-Speed Downlink Packet Access (HSDPA) systems, where packet loss is not relevant any more, as it is mitigated by layer 2 and 3 functions as it is the case in LTE. The authors of [TLFA13] design time utility functions for MOS values and develop a model of a Markov decision process which optimizes the overall utility. As a practical solution, they propose a heuristic which ranks the flows based on the achievable QoE.

3.6 Applying Size-Based Scheduling to Cellular Networks

The classical cellular schedulers presented so far do not use size information for their scheduling decision. This section first motivates size-based scheduling by looking at its origins in job scheduling and operations research. Then, approaches from literature which evaluate the performance in wireless systems analytically will be presented and the trade-off between the opportunistic capacity gain and size-based delay improvement will be discussed. Finally, actual scheduling algorithms taking slot-by-slot decisions on which transaction to serve next are introduced.

3.6.1 Origin of Shortest-First Scheduling

The roots of SRPT go back to 1966, when Schrage and Miller proposed the algorithm for an M/G/1-system [SM66] and shortly later Schrage published the proof of optimality which states that SRPT minimizes the number of jobs in the system [Sch68]. In conjunction with Little's law, this means that also the average waiting time of the jobs in the system is minimal. These strong advantages make it the algorithm of choice for a scheduling task whenever the remaining processing time can be reasonably estimated. A survey summarizing research on the statistical properties of SRPT in an M/G/1-system is [Sch93]. It details on mean delay, delay variance, the CDF and correlation properties of delay and attests a superior performance compared to other classical job schedulers like Last-In-First-Out (LIFO), First-In-First-Out (FIFO), and PS.

Schmickler and Görg propose a first application to packet networks for Ethernet Local Area Networks (LANs) [SG89]. They apply SRPT to CSMA / Collision Detection (CSMA/CD) and use protocol information to estimate the remaining message size[7](the term *message* roughly

[7] In [Goe90], Görg proposed to introduce a header field containing reverse packet numbering, which means that the remaining number of packets in a message is known for all packets.

corresponds to what is denoted as a transaction here). The station with the smallest message gets a higher channel access priority, which reduces collisions and mean transfer delay.

An interesting metric introduced in [BCM98] is the *stretch metric* (see also Section 2.4). It is defined as the relation between the *flow time* (*here:* transaction duration) and the *processing time* (the time a transaction would need in an unloaded system) and is sometimes also called *slowdown*. [BCM98] focuses on maximum stretch, where SRPT performs poorly because the largest jobs have to wait for smaller ones and therefore may take quite long. By contrast, [MRSG99] states that the average stretch is a better metric for the system performance. In [MRSG99], it is proven that SRPT is *2-competitive* with respect to average stretch for a single-server system. This means that the difference between the optimal solution and SRPT is at most a factor of 2.

The often stated objection against SRPT that it penalizes the large jobs in favor of the small ones is addressed in [BHB01]. The authors investigate the stretch for general and heavy-tailed size distributions and show that with respect to the mean stretch SRPT always outperforms PS, especially when the load is high. Furthermore, in many cases *all* jobs finish faster under SRPT in comparison to PS, especially for heavy-tailed size distributions. In the high load case (load >0.8), the largest jobs last longer than with PS by moderate factors. However, 99% of the jobs can profit from shorter transmission durations. [BHB01] analytically proves specific bounds for this behavior. Another advantage of SRPT is that in an overloaded system the processing time of the shorter jobs remains finite, whereas for PS all job sizes have an infinite mean response time.[8] Motivated by these results, [HBSBA03] proposes to use SRPT for web servers. The authors built an experimental test-setup with servers playing out static HTML web-sites and show the performance improvements in LAN and Wide Area Network (WAN) scenarios.

Another direction of research considering the *Shortest-First* principle is the *Foreground-Background (FB)* scheduler (also known as *Least Attained Service first (LAS)*). Its application to wireless networks is proposed in [SM02] and works without knowing the actual size of the requests. Assuming a heavy-tailed object size distribution, a request that has been there for a long time is likely to be a large request that will have a relatively large remaining size to transmit. By scheduling the request which has received the fewest service so far, the probability of scheduling a short request is increased. In [NW08] a survey of this kind of scheduling is given. Among schedulers without size knowledge, FB minimizes the mean response time for size distributions with *"decreasing failure rate"*[9]. That means distributions, where *"larger jobs have a smaller failure rate, and thus are less likely to be complete when they are served"* [NW08]. This is true for heavy-tailed size distributions, e. g. the log-normal distribution that is used here for transaction size modeling (see Section 6.1.4). Apart from the influence of the job size distribution, [NW08] discusses the drawbacks of FB in comparison to SRPT, the influence of the offered load, the distributions of the response-time and queue length, and the performance of large jobs under FB.

[8] The average processing time remains finite for the size-quantile up to which the offered traffic adds up to a load <1 [BHB01].

[9] The *failure rate* in [NW08] can be seen intuitively as the probability that a job finishes when it gets served.

[YS06] investigates the asymptotic delay properties of SRPT. For this, the *"many flows regime"*[10] is evaluated to derive the delay distribution of a job of fixed size asymptotically with the number of traffic sources. The so-called *decay rate function* characterizes the decay of the delay distribution with increasing delay. [YS06] gives an expression for this decay rate function under SRPT and compares it to a classical FIFO scheduler. The authors state that the larger the size of the largest objects, the smaller is the penalty for these jobs in comparison to FIFO. At the same time, the delay performance for the smaller jobs is significantly better under SRPT.

In [WHBO05], the authors generalize the class of schedulers that bias towards servicing small objects preemptively and introduce the term "SMART" scheduling for them. They prove upper and lower bounds for the mean response times relative to PS and show that the performance is within a factor of 2 of the optimal policy SRPT. Furthermore, it is shown that the variability of the size distribution of traffic objects has only a small influence on the mean response time. [YWSHB12] further generalizes the notion of SMART scheduling schemes by removing boundary conditions and calling this class SMART-LD (for *Large Deviations*). In succession of [YS06], [YWSHB12] concentrates on the *"many flows asymptotics"* and extends the investigation to the SMART-LD class, proving that all schedulers in this class have the same delay decay rate as SRPT. Furthermore, the decay rate of FB is derived, which has a similar shape but is uniformly worse than that of SMART-LD. With numerical experiments, the authors demonstrate that the asymptotic bounds of the decay rate functions are already reached for about 20 flows. This means that, at least for the example in [YWSHB12], the delay distribution of transactions of a certain size can be practically estimated by using the asymptotic delay decay rate function. Finally, [YWSHB12] compares SMART-LD, FB, and PS numerically, indicating that SMART-LD is the best option especially for traffic size distributions with high variability. This fits to web traffic characteristics which are usually described by heavy-tailed distributions, as we will discuss in Section 6.1.4.

3.6.2 Analytical Evaluation of Shortest-First Scheduling in Cellular Networks

The theoretical work discussed in the previous section is good to provide boundaries for the behavior of SRPT and characteristics of the delay distribution. However, a key issue when applying SRPT to wireless networks is the variability of the service rate in time. The central findings still hold when abstracting from the rate fluctuations by looking at the average rate on a larger time scale. Then, the question is how the instantaneous rate fluctuations of the radio channel contribute to the so-called *capacity region* which determines rate vectors to the individual users at this larger time scale. The capacity region increases with the ability of the scheduler to consider the instantaneous rate of the users in order to exploit multi-user diversity for an opportunistic gain (see Section 3.4). However, this collides with the *Shortest-First* principle stating that the shortest transaction should be served.

[SDV10] investigates the trade-off between opportunistic gain and *Shortest-First* scheduling analytically. A queuing model for a symmetric and stationary channel model and corresponding

[10] This method investigates the asymptotic behavior of the scheduler when the capacity scales with the number of flows. Thus, probabilistic expressions for rare events (like the waiting time of a very large transaction) can be given, which are reached asymptotically when the number of flows tends to infinity [YS06].

capacity region is derived and used to define lower bounds for the sojourn time of flows[11]. The idea is a separation of time scales, assuming that flows last long in comparison to the channel fluctuations. With this assumption, a capacity region is derived that models the effect of multi-user diversity, i.e. the total throughput increases with the number of concurrent users. As upper bound for the capacity region, [SDV10] proposes a *polymatroid* depending on the number of concurrently served users. This means that all rate vectors[12] on the border of this region have the same magnitude g_n, where n is the number of users. In other words, there is no trade-off between opportunistic gain and prioritizing short jobs in this case [SDV10]. The algorithm *SRPT-Highest Possible Rate (HPR)* gives the highest rate g_1 to the shortest flow (i. e. the maximum rate a single user can get), the second highest rate $g_2 - g_1$ to the second shortest flow (i. e. the additional capacity for a two-user region) and so on. This algorithm is proven to be optimal in this capacity region and serves as a lower bound for the average waiting time. For a realistic capacity region, the authors propose a scheduler combining opportunistic and size-based scheduling by equally sharing the radio resources (according to a *PF* variant) among a fixed number of the smallest transactions to transmit. By looking at the difference in mean sojourn time between pure opportunistic scheduling and SRPT prioritization, [SDV10] concludes that when the opportunistic gain is limited, SRPT can offer a significant improvement.

Another interesting work considering the trade-off between SRPT and opportunistic scheduling is [APLO11]. It takes a similar approach by using the time scale separation argument to derive a capacity region which grows with the number of active users in the system. However, compared to [SDV10], the authors define the capacity region more general as they only require it to be compact and symmetric. With a recursive derivation, the optimal scheduling policy is derived, including an expression for the mean delay. For the case of a polymatroid capacity region, they come to the same results as [SDV10]. Apart from the lower bound, [APLO11] also gives an upper bound for the minimum mean delay.

These analytical evaluations give valuable information on the optimal scheduling policies as well as the minimum delay characteristics. However, the results cannot be directly applied to practical systems. The symmetry of the capacity region (i. e. the rate fluctuations being identically distributed for all users) as well as the fact that no further arrivals of traffic objects were considered after the initial state are not given in reality. This makes it hard to prove exact boundaries for real systems. Furthermore, working directly on the capacity region at a flow-level time scale leaves the question open how to design an actual slot-level scheduler which converges to the desired behavior (especially when considering that the average achievable rate is not known in advance and that the radio channel of a user is in general not stationary). Apart from that, the time-scale separation argument itself not always holds in high data rate systems like LTE with realistic traffic, because many transactions only require very few slots to transmit.

3.6.3 Schedulers Applying Shortest-First in Cellular Networks

In this section, we will discuss actual slot-level schedulers for cellular networks, which apply the *Shortest-First* principle. An early work mentioning a *Shortest-First* scheduler is [Tsy02]. It

[11] The term *flow sojourn time* from [SDV10] corresponds to transaction durations.

[12] The elements of a rate vector are the average rates assigned to the individual users at the time scale of the capacity region.

proposes a straight-forward implementation of SRPT in wireless networks, which will be called *Shortest Remaining First (SRF)* in the following ([Tsy02] denotes this algorithm by *Minimum Relative Length (MRL)*). SRF schedules the transaction with the smallest cost c_{SRF}, with

$$c_{SRF} = \frac{F_r(t)}{R(t)} \tag{3.7}$$

where $F_r(t)$ is the remaining size of the transaction at time t. The cost c_{SRF} represents the remaining transmission duration estimated with the current data rate, when the transaction would be scheduled all the time. Thus, Equation (3.7) represents the SRPT principle based on the current information.

Apart from some limited numerical considerations with SRF, [Tsy02] focuses on finding an optimal scheduling sequence recursively by dynamic programming. However, it requires a system without transaction arrivals after the initial state and knowing the probability distributions of all radio channels. [JKPS00] investigates and compares various scheduling algorithms for CDMA networks and also evaluates SRPT without providing implementation details. The authors state that it uses rate and remaining size information, so it can be assumed that it is equal or similar to Equation (3.7). The main part of [JKPS00] consists of extensive simulation studies investigating throughput and delay measures with realistic traffic traces for a CDMA network. These studies confirm the superior performance of SRPT predicted by theoretical analyses with respect to mean response time and stretch.

Hu *et al.* [HZS04] investigate schedulers applying the SRPT-principle in cellular networks under the term *Traffic-Aided Opportunistic Scheduling (TAOS)*. The SRF scheduler according to Equation (3.7) is called *TAOS 1b* in their terminology. Furthermore, [HZS04] gives a scheduler *TAOS 1a* which calculates the costs with the initial transaction sizes. In this work, this scheduler will be called *Shortest First (SF)*, with

$$c_{SF} = \frac{F}{R(t)} \tag{3.8}$$

where F is the total size of a transaction. The main variant proposed by [HZS04] is *TAOS 2* which aims to find the locally optimal solution with respect to the sum completion time of the currently active transactions[13]. The algorithm *TAOS 2* works as follows:

1. Enumerate active transactions in ascending order of $\frac{F_r(t)}{\overline{R^*}(t)}$ with rank i. [HZS04] defines $\overline{R^*}(t)$ as the expectation of the data rate. However, in order to use the same information basis for *PF* and *TAOS 2*, the moving average is employed here instead. It is determined by always inserting the current channel rate in Equation (3.5) (the upper case).

2. Compute the costs

$$c_{TAOS2} = (i-1) - (M(t) - i + 1)\left(\frac{R(t)}{\overline{R^*}(t)} - 1\right) \tag{3.9}$$

where $M(t)$ denotes the number of currently active transactions.

[13] [HZS04] proposes a fourth variant denoted as *TAOS 1* which we do not consider any further, because it performed inferior to the other variants in our evaluations.

3. Schedule the transaction with the smallest cost c_{TAOS2}.

According to [HZS04], *TAOS 2* tries to balance the advantage of scheduling a user with a channel above average for the higher ranked transactions and the delay for the lower ranked transactions (i. e. those with shorter remaining size).

The authors of [LA08] compare *TAOS 2* and different SRPT variants (including FB) with the classical opportunistic scheduler PF and variants thereof. Assuming a set of discrete rates, [LA08] proposes so-called *priority rules* which always serve flows with the maximum instantaneous rate and use SRPT (or other criteria) as a tie-breaking rule when multiple such flows exist. Ordinary opportunistic schedulers are called *index policies* in [LA08]. As given in [LA08], Equation (3.9) can be simplified to scheduling the user with maximum weight

$$w_{TAOS2} = (M(t) - i + 1)\frac{R(t)}{\overline{R^*}(t)} \tag{3.10}$$

In a simple scenario with two discrete rates and two flows, [LA08] derives the optimal scheduling (with dynamic programming like in [Tsy02]) and compares it with the heuristics. For the case with asymmetric rate probabilities, low variability and equal sizes of the flows, *TAOS 2* deviates from the optimal mean delay but is very close to it in the other considered cases. Furthermore, [LA08] presents simulation studies with continuous traffic arrivals of varying load with HSDPA rates (11 discrete rates). For the investigated settings, *TAOS 2* performs worse than PF and the priority rules with respect to mean delay. The authors do not give reasons for this, but the large discrepancy between the current rate and the rate average could cause the rank indicator of *TAOS 2* to not work as expected.

A set of discrete rates may not be a good model for OFDMA systems, especially for LTE, because the resource granularity is much finer than in HSDPA and resembles more to a continuous distribution of rates (allocation of individual PRBs, many MCSs). This limits the applicability of the priority rules from [LA08] in these systems. They would likely behave similar to a *Max C/I* scheduler in this case.

3.6.4 Size-Based Schedulers for Video-Streaming

Ordinary size-based scheduling like SRPT is not beneficial for video streaming and similar applications. The reason is that even short videos are relatively large files compared to an average web-site. Consequently, video transactions would obtain a low priority from any *Shortest-First* scheduler, which leads to buffer underruns and bad QoE under high load.

Different proposals exist in literature to apply SRPT to video streaming such that the QoE is improved. [MDRR03] proposes a frame scheduling algorithm called *Fair-SRPT*. It schedules video flows in increasing order of their instantaneous data rate (based on the current mean frame size). However, this does not work for buffered video streaming, which is prevalent for today's video services (see Section 2.3). The scheduler is not able to distinguish between the individual frames, because they are embedded in a HTTP session, for example. Furthermore, the instantaneous frame rate is not relevant in this case, as data is buffered at the client-side anyway.

Recent publications investigating the usage of buffer sizes for video streaming are [WSP+12] and [NOALS+13]. While the schedulers proposed therein do not work according to the *Shortest-First* principle, they are still size-based with respect to buffer size. The information about the buffered playback-time of video streams at the client is considered for the scheduling decision. The concept in [WSP+12] defines two thresholds of the buffered time triggering a prioritization of the respective video stream. When the buffered play-time falls below a first threshold, the stream is prioritized until it exceeds a second threshold, after which it will be handled normally again.

The approach in [NOALS+13] is more comprehensive in that it defines a continuous weighting function depending on the buffer level. The value of this function, which is proposed to be s-shaped, is then multiplied with an ordinary PF scheduling weight. The authors call the scheduler QoE-aware, we therefore denote it by *PF-QoE* in this work. The scheduler function is[14]:

$$w_{PF-QoE} = f\left(b(t)\right) \cdot \frac{R(t)}{\overline{R}(t)}, \tag{3.11}$$

where $b(t)$ is the buffer level in time units. The function $f(\cdot)$ maps the buffer level to a scheduling weight as follows

$$f(b(t)) = f_{min} + \frac{2 \cdot f_t}{1 + e^{-a(b_t - b(t))}}. \tag{3.12}$$

Here, f_{min} and f_t are the minimum and target priority factors, respectively, and b_t is the target buffer level. The factor a scales the inclination around b_t. To avoid stalling problems when new video transactions (with initially empty buffers) arrive, [NOALS+13] proposes a factor ≤ 1, denoted f_{init} here, which scales w_{PF-QoE} until the buffer reaches b_t for the first time.

Principally, $f(\cdot)$ is large when the buffered playtime is below b_t and small when it is above. The advantage of this approach is that many competing video streams can be gradually weighted with respect to the probability of a buffer underrun.

Both scheduler proposals employing the buffered playtime at the client greatly reduce the probability of video interruptions and are therefore able to improve the user experience for this kind of traffic compared to schedulers without knowledge of the buffer level.

[14] The original notation from [NOALS+13] is adapted to fit here.

4 Combining Size-Based and Opportunistic Scheduling Principles

During peak load situations, the bandwidth of a cell is not sufficient to satisfy all transmission demands. An important consequence are prolonged durations for these transmissions, which leads to an unsatisfactory service behavior of many applications. Therefore, scheduling algorithms which achieve minimal transmission durations generally lead to an improved user experience. In classical scheduling tasks like job scheduling on a processor, SRPT has been proven to minimize the average waiting times. This corresponds to small transmission durations in the case of a base station scheduler. Apart from that, for a shared wireless medium, opportunistic scheduling can reduce the backlogged data by increasing the capacity compared to non-opportunistic schedulers.

In this chapter, desirable features of a scheduler and why it is advantageous to combine size-based and opportunistic schedulers will be motivated in Section 4.1. Then, Section 4.2 discusses how this combination can be implemented and what is the control range. Finally, we peek at the influence of the new scheduler on a cell's transmission behavior in Section 4.3. This impact will be thoroughly investigated in Chapter 7, where the scheduling behavior is evaluated by simulation studies.

4.1 Motivation

A widely-used scheduler in cellular networks is *Proportional Fair (PF)*. It combines the ideas of opportunistic scheduling and fairness. While opportunistic scheduling means to give radio resources to UEs in good channel conditions, fairness shall ensure that no UE is cut off from the medium over a longer time span (called *starvation*). *PF* achieves this fairness by normalizing all UEs' channels to their respective time averages. Consequently, resources are distributed equally among them in average[1].

The rationale behind this is that it shall not be possible that UEs in the center of the cell get all resources, while those at the border starve. However, such a fairness paradigm incorporates the assumption that users take all resources they can get – so called full buffer traffic. Nowadays, as most of the traffic volume is made up of packet-based data traffic which is bursty (see Chapter 2) and with the high data rate of LTE further contributing to this burstiness, the purpose of resource

[1] At least when assuming identical distributions for the channel fluctuations.

fairness is less clear, and in some cases fairness is even difficult to define. UEs often require just a fraction of the resources they could get, e. g. small transactions that can be transmitted in a fraction of a time slot. In this case, the rate for fairness evaluation cannot be defined reasonably. Furthermore, as traffic demand is limited, requests from the center of the cell can be served faster than new requests appear. This leaves sufficient resources for users with less favorable channel conditions. Except for severe overload conditions, this practically reduces starvation solely by the bursty nature of the offered traffic.

A drawback of the fairness property of *PF* is that it leads to PS-like scheduling during overload situations. This means that all users get low data rates. With TCP congestion control, which adapts the rates to the available bandwidth, this leads to long durations for all transactions. Consequently, instead of the MAC scheduler, a superordinate control-loop at a larger time scale would be suited better to perform fairness enforcement with less restrictions on radio resource allocation and less deterioration of latency.

In general, opportunistic scheduling increases capacity and therefore makes overload more seldom. Furthermore, very large transactions finish comparably fast under *Max C/I* due to the high capacity it offers. On the downside, conventional opportunistic schedulers do not exhibit graceful service degradation. This is not possible, as they are not designed to consider application layer aspects and can therefore not distinguish between transactions or evaluate QoS requirements. Similarly to *PF*, most transactions will suffer from longer transmission durations under *Max C/I* in case of overload. According to the channel fluctuations, the scheduler chooses between UEs with similar channel conditions so that they get comparable resource shares on average, leading to the same PS-like scheduling scheme.

In such a situation, cross-layer information can greatly improve the average service quality of the whole system. By identifying and postponing transactions which have uncritical latency requirements, the interactive ones can be transmitted much faster. A very simple and effective scheme, requiring only the knowledge of transaction sizes, are *Shortest-First* schedulers. Because transactions only compete with those of smaller and similar size, as discussed in Section 3.6, many transactions do not notice overload at all. To illustrate the implications from this, Figure 4.1 shows the accumulated volume of transactions ordered by their sizes at a load of 40 Mbit/s (a slight overload situation for the simulated cell). For comparison, Figure 4.1 gives the volume transmitted with a *SF* scheduler in the same scenario. Especially for Internet object size distributions as assumed here, most transactions ($\approx 95\%$ at the given load) belong to the fraction of traffic which could be transmitted with the available cell capacity. However, for conventional schedulers which are unaware of object sizes, all transactions would suffer from overload by greatly reduced data rates (the load is ≈ 1.4 in Figure 4.1). On the other hand, *Shortest-First* schedulers exhibit the desired property of graceful service degradation, thereby greatly improving system QoE, because only the fraction of traffic that exceeds the cell capacity gets delayed significantly[2].

While the large advantage of SRPT with respect to latency is very desirable for cellular networks, it comes at the cost of reduced cell capacity. This is sketched in Figure 4.2. Choosing transactions according to their object sizes limits the degree of freedom available to exploit

[2] In the case of permanent overload, the largest transactions would never be transmitted. However, under PS, this is true for all objects [BHB01].

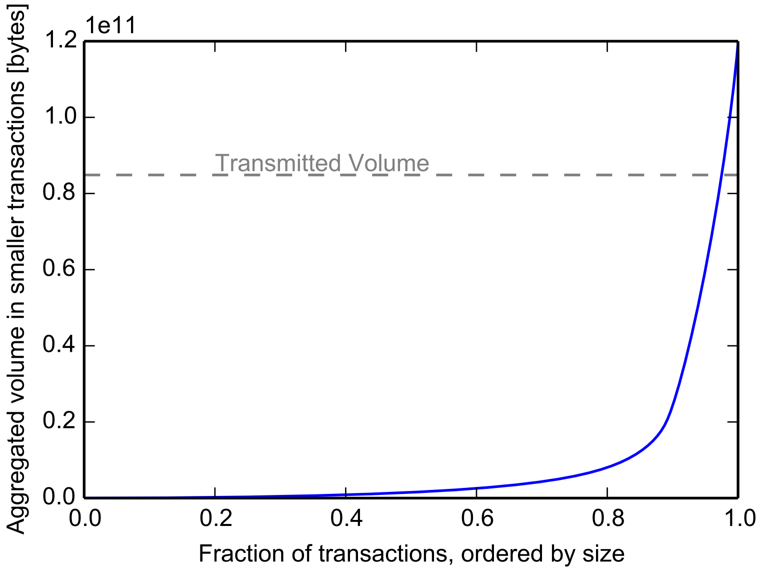


Figure 4.1: Accumulated volume over the fraction of transactions ordered by size for the scenario from Section 7.1.1 at an offered traffic of 40 Mbit/s. For comparison, the transmitted volume of *SF* is given.

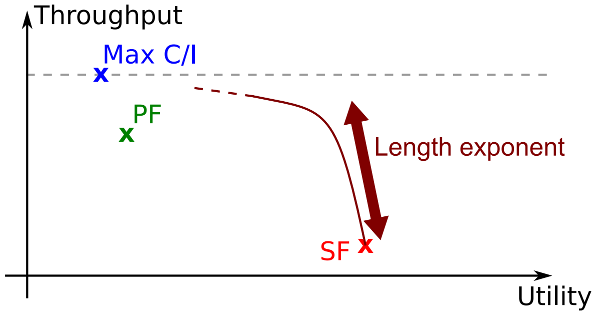


Figure 4.2: Trading throughput for utility; operation points of different schedulers and flexibility introduced by the length exponent.

multi-user diversity. Naturally, *SF* includes CSI in its scheduling decision, because it is integrated into the estimation of the transmission duration. Still, CSI has less weight than under *Max C/I*.

Therefore, a new parameter applicable to different *Shortest-First* schedulers is proposed in this thesis. It offers the choice to exploit object size information or multi-user diversity. According to its function, the parameter is called *length exponent* γ and will be introduced in Section 4.2. The new schedulers including the length exponent allow to mitigate the drawback of reduced capacity and allow to trade it for utility. Figure 4.2 illustrates this flexibility. While increasing the throughput in general comes at the cost of a utility decrease (depending on the traffic scenario), large throughput gains can be achieved at a negligible utility drop. Furthermore, by increasing the effective capacity of the cell, under realistically fluctuating load, overload situations appear less frequently and last shorter than for the original *Shortest-First* schedulers. Operators may therefore use this opportunity to allow more customers in their networks without building up additional infrastructure.

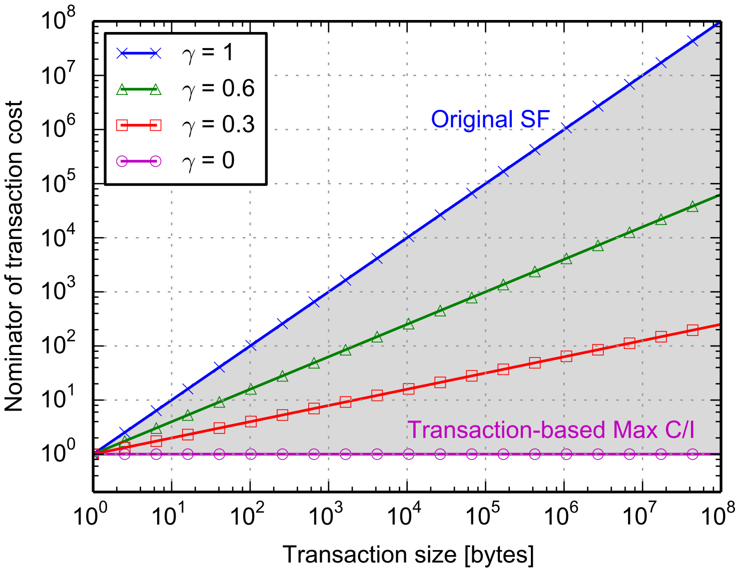


Figure 4.3: Nominator of transaction cost depending on the transaction size for different length exponents γ.

4.2 Algorithm Design

The implementation of the motivated parameter is straight forward. It shall enable the control of the influence of object sizes versus the influence of CSI. For both, *SF* and *SRF*, the cost attributed to a transaction is the fraction of the size-based quantity and the instantaneously possible data rate. Therefore, we introduce the *length exponent* γ into the original cost function Equation (3.8) as follows:

$$c_{SF} = \frac{F^\gamma}{R(t)} \tag{4.1}$$

Analogously, this also works for *SRF* by adding a parameter to Equation (3.7):

$$c_{SRF} = \frac{F_r(t)^\gamma}{R(t)} \tag{4.2}$$

The length exponent now controls the variability of the size metric and therefore the influence on the cost value. Because the scheduler always gives resources to the transaction with the smallest cost, γ defines how much a bad channel quality can be compensated with a small size of a certain transaction. By choosing $\gamma \in [0, 1]$, we can gradually vary the behavior between that of *SF* ($\gamma = 1$) and that of *Max C/I*[3] ($\gamma = 0$). Figure 4.3 illustrates this property. For $\gamma = 0$, the nominator of the cost function is constantly one and the cost is solely defined by the instantaneous data rate, as it is the case for *Max C/I* (minimizing $1/R(t)$ leads to the same allocation as maximizing $R(t)$). For $\gamma = 1$, Equation (4.1) and Equation (4.2) are identical to the original cost functions, i. e. the cost of a transaction grows linearly with its size. The variant with *SRF* together with some earlier simulation studies was published in [Pro14].

[3] More precisely, it is a variant of *Max C/I*, because it allocates transactions instead of resource blocks (see Section 6.1.5).

Table 4.1: Influence of the length exponent γ on cell capacity and finish times of two example transactions.

γ	Full buffer capacity	$\tau(T_1)$	$\tau(T_2)$	Sum duration
0	24.2 Mbit/s	928 ms	673 ms	1601 ms
0.5	17.5 Mbit/s	254 ms	857 ms	1111 ms
1	8.1 Mbit/s	181 ms	894 ms	1075 ms

With this, it is possible to choose how much emphasis should be laid on cell throughput versus trying to achieve short transmission durations. For $\gamma > 0$ the scheduler is principally able to consider boundaries of application layer objects and to use this information for a reduction of transmission durations. Therefore, already values of γ close to 0 lead to a significant improvement in utility, because the scheduler is able to distinguish between transactions of similar channel quality. In Figure 4.2, this is illustrated by the steep utility increase at practically no cost in throughput close to *Max C/I*.

Theoretically, it is possible to choose $\gamma > 1$. However, some experiments have shown that this further increase of the influence of object size is not beneficial. It makes the scheduling decision more fragile when new transactions arrive and increases the susceptibility to estimation errors in size and channel quality. We therefore do not consider this possibility in the following.

4.3 Influence and Benefits of the new Scheduler

We now discuss the influence the new schedulers have on the general system behavior and particularly the influence on transmission durations. The discussion complements the comprehensive performance evaluation that will be presented in Chapter 7.

A simple example to illustrate the influence of the length exponent on the scheduling decision is shown in Figure 4.4. The figure shows the evolution of scheduling cost under *SF* for two exemplary transactions over time. The subfigures contain different settings of $\gamma \in \{0, 0.5, 1\}$. T_1 has a size 200 kBytes and T_2 has a size of 2 MBytes. For $\gamma = 0$, the scheduling cost is anti-proportional to the channel quality. T_1 has a better channel, which leads to a smaller cost and means that it will always be scheduled. On the other side, $\gamma = 1$ means that the size has a relatively strong influence on the scheduling cost and leads to a preference of T_2. Accordingly, for $\gamma = 0.5$ the transactions get served alternatingly in this example. The cost advantage of either transaction depends on the instantaneous channel qualities.

Table 4.1 contains properties on which the length exponent γ has an influence. The full buffer capacity is the total data rate that would be achieved in the cell, when always the UE with the smallest cost would be scheduled (irrespective of the actual size of T_1 and T_2). $\tau(T_1)$ and $\tau(T_2)$ are the transmission durations of the respective transaction. The last column contains the sum duration of both transactions.

Although the example only illustrates a single situation, some general observations can be made. Increasing size influence, i. e. increasing γ, reduces the cell capacity because small transactions get served although they may have sub-optimal channel quality. Without size influence ($\gamma = 0$),

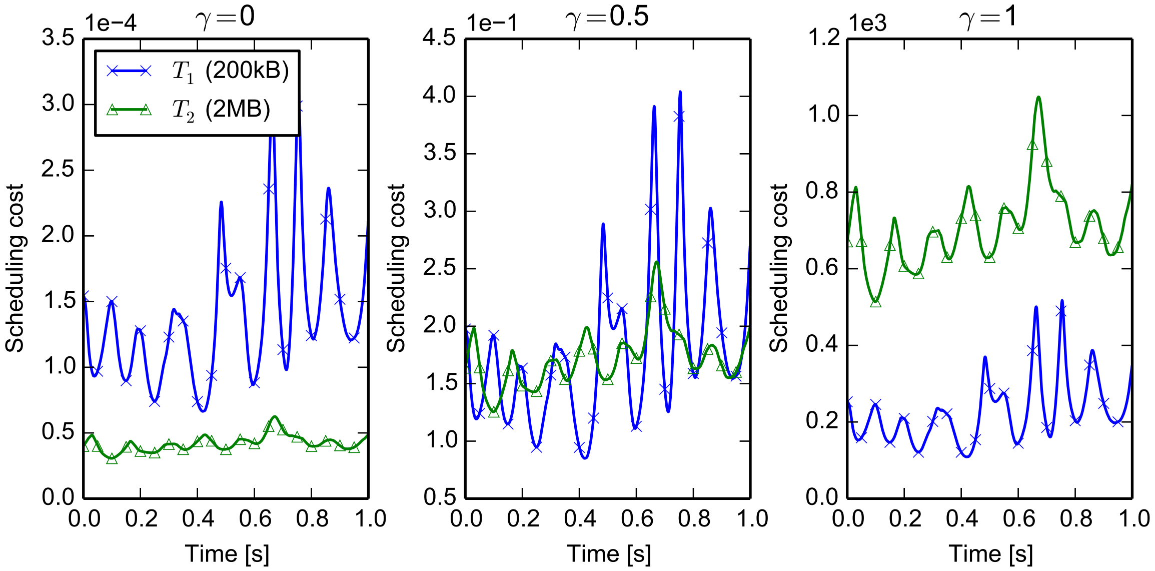


Figure 4.4: Scheduling cost of two exemplary transactions with different length exponents γ.

a large transaction like T_2 may block a small transaction like T_1. This leads to a larger aggregated transmission duration and therefore impairs the average user experience. Furthermore, the relative change in duration is much greater for small transactions than for large ones. In the above example, the duration of T_2 only increases by $\approx 30\%$ between $\gamma = 0$ and $\gamma = 1$. On the other side, T_1 decreases by about 80%[4]. An interesting point is that for $\gamma = 0.5$ the sum duration is almost as small as for $\gamma = 1$, while the full buffer capacity is closer to the one with $\gamma = 0$. Consequently, a balanced setting of γ is able to combine advantages from both worlds, *Max C/I* and *SF*. These principal observations and the influence and benefits of the length exponent γ will be quantified by simulation studies in Chapter 7.

Principally, as indicated in Figure 4.2, the length exponent offers the choice between a high cell capacity and short transmission durations, which are both desirable properties. A high cell capacity means that a larger data volume is transmitted in a certain time, i. e. a network operator could sell more volume options to the customers. On the other side, short transmission durations have the potential to offer a good QoE. The operator could determine the relative importance between both properties. Then, by measuring the curve from Figure 4.2 in the actual network, the optimal choice of γ can be derived. Such an approach and the possible improvement by the length exponent will be evaluated in Section 7.2 and Section 7.3.4.

[4] Naturally, the chosen example is quite clear in this respect. Nonetheless, a small transaction with bad channel conditions may profit even more from *Shortest-First* scheduling in realistic situations.

5 Inclusion of Buffered Video-Streaming into Shortest-First Scheduling

Different applications have different requirements for their network transmissions, e. g. with respect to latency or bandwidth. While it is always an option to classify data packets according to their requirements and handle them with different schedulers in a hierarchical structure, this comes at the cost of increased complexity and reduced flexibility for the sub-schedulers. Therefore, Section 5.1 motivates what the challenge in handling video streaming with a *Shortest-First* scheduler is and what the benefits of an inclusion of video streaming are. Then, in Section 5.2, we discuss the application layer information that is important for the scheduling of video traffic. Section 5.3 presents the proposed approach for the new schedulers and finally Section 5.4 discusses its impact.

5.1 Motivation

A large fraction of Internet traffic is video streaming, with most of it being buffered video streaming (see [San14] and the discussion in Section 2.3.3). Therefore, it makes up a very important traffic class. Ordinary *Shortest-First* scheduling would strongly penalize video streaming objects, as they are very large compared to web browsing traffic ([FMM$^+$11] reports a median video size of $\approx$10 MBytes). This would make video traffic very unsatisfactory as soon as it has to compete against bursty web traffic. Additionally, video streaming requires a different definition of *utility* and therefore also of *object size*, because the user already watches the content while the transmission is still ongoing.

Different possibilities are thinkable to resolve this problem. A simple solution would be to separate streaming traffic and web browsing traffic by means of packet classification and separate buffers. Then, static prioritization or resource reservations could control the video QoE. However, that would mean a reduced flexibility in scheduling video streaming packets and would penalize the other traffic more than necessary, e. g. during a load peak when all videos already have full buffers at the client-side. Furthermore, shifting transmissions in time for improved multi-user diversity would be constrained by separating both traffic classes and would consequently lead to a degraded overall capacity.

In this thesis, a different approach for handling video traffic is proposed, which aims at integrating it into *Shortest-First* scheduling. This requires defining a size equivalent for streaming traffic. It offers the possibility of exploiting application layer information and gives the full flexibility to the opportunistic size-based scheduler. A very useful information is the duration of buffered play-time at the client, because it allows to prioritize video transactions precisely according to their requirements for an uninterrupted playback. It is the information that allows to anticipate interruptions in video playback and to know if a transmission can be postponed in the order of tens of seconds without affecting the video's QoE. This increases the flexibility of the scheduler and can lead to an overall improvement in utility and throughput. Thus, using the client-side buffer level of streaming transactions for defining an *equivalent object size* enables us to provide all QoE-relevant information and to have a common weight quantity for all considered transactions managed by a *Shortest-First* scheduler.

Regarding MAC layer scheduling of video streaming, there are two recent publications proposing the usage of the client-side buffer level [NOALS$^+$13, WSP$^+$12]. In the following, it will be shown how such an approach integrates with *Shortest-First* scheduling and which advantages this offers over the cited proposals. Furthermore, with this approach the proposed length exponent extends seamlessly to video streaming and thus provides the desired all-in-one scheduler solution which exhibits a superior QoE, graceful service degradation, and an adjustable cell capacity.

5.2 Application Layer Information on Video Streaming

As stated before, some application layer information is required to determine the QoS requirements of buffered video streaming. The question is how the scheduler in the base station can obtain information about the current amount of buffered playtime on the UE, which is the only QoE indicator relevant for resource allocation[1]. In the following, we first discuss the effect of the video's data rate and then how the base station could obtain knowledge about the state of the client buffer.

5.2.1 Required Data Rate for Video Transport

The transported streams for video and audio are usually compressed. A so-called *codec* (coder-decoder) compresses the content according to a coding format. Subsequently, this compressed data is transported to the video player on the UE, where another instance of the codec decompresses it for playback. There is a large variety of coding formats for audio and video streams. For example, [FMM$^+$11] reports a number of different video and audio codecs used together by YouTube. Within YouTube, these combinations along with other properties like container format[2] and video resolution make up different video formats specified by the so-called *itag*.

Usually, the video stream has a much higher bit rate than the audio stream and therefore is mainly responsible for the total bit rate. In the following, the sum encoding bit rate is meant when discussing the rate of a video transaction. Many codecs, e. g. the typical configuration of

[1] Proposals like DASH exist, where the video resolution and quality are adapted to the bandwidth of the transmission path (see Section 2.3.3). However, here we assume that the scheduler has no influence on the video data. Consequently, the content's influence on QoE is out of the scope of this thesis.

[2] A *container* is a file format which allows to include audio and video streams and possibly additional meta-data in a single file. Video player applications can interpret such containers and extract the content from them.

the widely used H.264 [IT14], employ a lossy[3] compression which leads to a large reduction in bit rate compared to the uncompressed data stream. Such compression algorithms may lead to a bit rate varying with the target image quality and the nature of the content, e. g. a still scene requires a lower encoded bit rate than a panning shot. However, for simplification, we assume a constant encoding bit rate for a video. For the video traffic model, we therefore only attribute the average code rate to a streaming transaction (see Section 6.1.4.3). While the instantaneous encoding rate may vary with realistic coding formats, this gives a reasonable model of the required data rate when averaged over several seconds. See [WSP+12] for an example of the evolution of the encoding rate over playback time. As discussed in Section 2.3.3, adaptive bit rate streaming techniques are also not covered here.

5.2.2 Knowledge of the Remaining Buffered Playtime

Two recent publications introduce the consideration of buffered playtime on the client for MAC layer scheduling at the base station [WSP+12, NOALS+13]. Whereas the approach proposed in [WSP+12] only defines a threshold for buffered play-time below which a video transmission gets prioritized statically, the one in [NOALS+13] is more fine-grained in that it employs an *s*-shaped mapping function between the buffer level and the scheduling weight of a video transmission (see Section 3.6.4).

An important prerequisite for such an approach is the availability of knowledge about the remaining buffered playtime. [WSP+12] assumes a feedback loop between the YouTube client application and the base station. The client application (e. g. the video player embedded on the YouTube web-page) triggers a signaling event to the base station, whenever a buffer threshold is reached. In [NOALS+13], the authors propose to take the encoding rate from meta-information in the container file, which could be obtained by DPI. Furthermore, they note that buffer reports as specified for the 3GPP DASH framework [3GP14a] could help to improve the accuracy of the buffer estimation.

More generally, knowledge about the buffer state could be either estimated at the base station from inspecting application layer information (e. g. *itag* or container meta-data) or, leading to more accurate results, with feedback signaled from the client application. While the latter is not widely available today, the former could be implemented with standard techniques in the access network. In any case, it requires state-keeping and flow classification at the base station to make the information about the buffer level of a video transaction available for the scheduler.

For illustration, Figure 5.1 gives an example of the evolution of transmitted data, played data and buffer size over time. The top graph shows the accumulated amounts of transmitted and played data. Data is transmitted in bursts (e. g. YouTube transmits bursts (or *chunks*) of 64 kBytes [ARMNOLS12]), which explains the steps in the curve for transmitted data. As discussed above, a constant encoding rate of the video is assumed. This leads to a constant slope for the amount of played data over time, once the video is playing. The size of the client-buffer, which is indicated in the lower graph, can be derived from the difference between the transmitted and the played data volume. In this example, the transmission rate is often lower than the encoding

[3] Lossy means that it is not possible to restore the original data when decompressing. However, lossy video coding formats try to only drop information that is not important for the impression of a human observer.

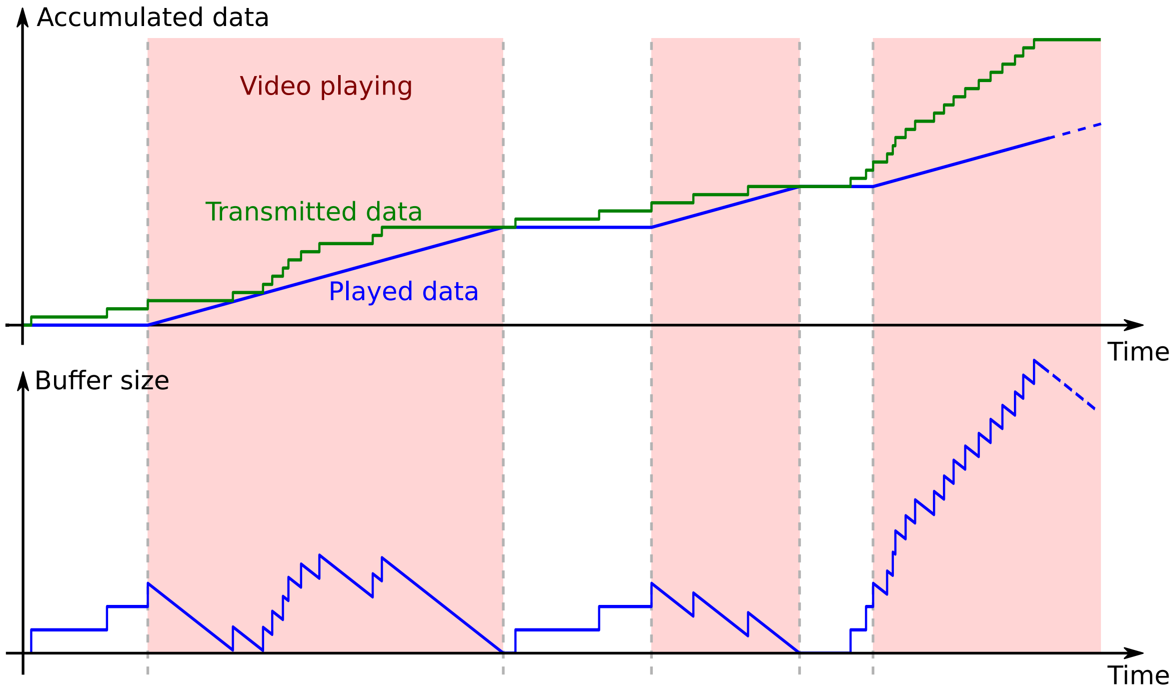


Figure 5.1: Illustration of the evolution of transmitted data, played data, and buffer size over time for buffered video streaming.

rate of the video. This leads to a depleting amount of buffered data and an interruption of video playback as soon as the buffer is empty. Playback stops until a certain amount of buffered data is reached again. During a buffering phase, the buffer size grows with each incoming burst. In addition to that, it constantly shrinks according to the encoding rate when the video is playing.

5.3 Including Video Streaming into Size-Based Scheduling

As motivated in the beginning of this chapter, there are challenges in scheduling video streaming transactions with *Shortest-First* algorithms. Using the complete object size would lead to a strong scheduling penalty with the SRPT principle, as video transactions are usually very large compared to interactive transactions. We discussed above that using the buffer size for the derivation of an equivalent object size can solve this problem. The buffer size exhibits the desired behavior that the urgency of transmitting data for a streaming transaction decreases with increasing size. Challenges remain with respect to the parameterization of the equivalent object size. Under overload, video traffic should not take all available resources and a differentiation between videos should be possible. This means that instead of all videos suffering from insufficient resources, only those where a degradation of QoE is unavoidable shall be penalized by the scheduler.

5.3.1 Handling Video-Streaming under Shortest-First

The most straightforward and simple approach to include video streaming in *SF* (or *SRF*) would be to use the buffer level at the client-side itself as the equivalent object size. Then, a transaction with much buffered playtime would have a high scheduling cost whereas another one with a lower buffer level would have a small cost and would more likely be scheduled. However, this approach has two major drawbacks. When the load in a cell is high, a single transaction only gets a small fraction of the radio resources. For video transactions, this means that the transmission rates often are smaller than the respective encoding rates. Thus, the buffer level of all active videos successively shrinks during an overload situation in a cell. First, this means that videos get more and more priority, as their buffer levels decrease to the order of short interactive transactions. Second, an equivalent object size of zero in case of an empty buffer means that the channel condition has no influence on the scheduling cost in Equation (3.7) and Equation (3.8) anymore. Thus, it is not possible to distinguish between video transactions with empty buffers and schedule opportunistically.

Another option would be to employ an s-shaped function – inverted in comparison to the one in [NOALS$^+$13] – as equivalent object size. As the authors state, this shape limits the weight of a transaction being in danger of a buffer under-run or already interrupted. On the other side, the function never falls below the offset value for large buffer levels. This means that under *PF-QoE*, videos having considerably more buffered data than the target value are treated like the traffic from other applications. For *SF*, instead of an upper and lower limit, just a minimum is required for the equivalent object size. It is no problem that the mapping function grows with the buffer size indefinitely. In contrast, this is a desired behavior, because the urgency of a transmission also shrinks continuously.

Therefore, we define the following simple linear mapping function between the buffer size $b(t)$ (in kBytes) at the client and the equivalent transaction size $F_{\text{video}}(t)$ assumed by the scheduler:

$$F_{\text{video}}(t) = b(t) + c_{\text{min}} \tag{5.1}$$

$F_{\text{video}}(t)$ replaces F and $F_r(t)$ in Equation (4.1) and Equation (4.2), respectively. Figure 5.2 illustrates this mapping between client buffer and equivalent object size. It means that a video with a large playback buffer always has a large cost relative to the transaction's channel quality. The offset c_{min} prevents the drawbacks of a linear function discussed above. Any value $c_{\text{min}} > 0$ will avoid a multiplication by zero which would erase the influence of the channel conditions in case of an empty buffer. Furthermore, with c_{min} we can define the object size up to which an interactive transaction never gets delayed by video transactions. It is therefore not possible that videos monopolize the radio resources in case of an enduring overload situation.

While the equivalent object size would work with both, buffered data and buffered playback time, it was a design choice to use the size of the client buffer. When using time units in Equation (5.1), a mapping factor to object sizes would be required. A constant mapping factor would mean that a video with a large encoding rate has the same equivalent object size as another video with lower encoding rate and the same amount of buffered playback time. In terms of radio resources, this would mean a preference of videos with large encoding rates. In contrast, the buffer size directly refers to the metric relevant for resource allocation – a

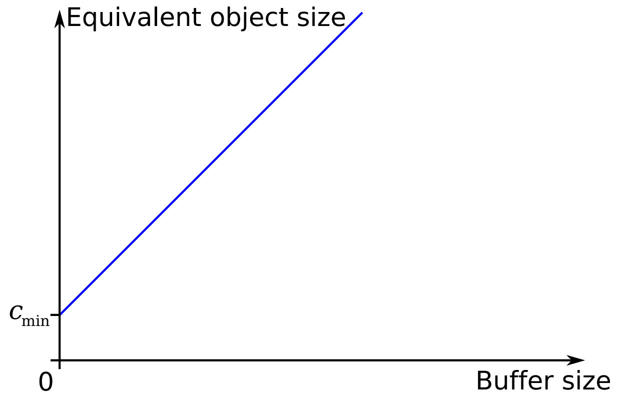


Figure 5.2: Transformation of the client-side buffer level into a size equivalent for *SF*.

data volume. Accounting for the amount of buffered data as equivalent object size implicitly prefers those videos with a small encoding rate compared to their channel conditions. This is a desired behavior in overload situations. The videos with an unsustainable required data rate (either due to the encoding rate of the video or bad channel conditions) run out of buffered data faster compared to the competing videos with more favorable conditions. Thus, during ongoing overload, especially together with the mechanism proposed in the next section, it is likely that the transactions in unfavorable conditions are dropped relatively early, which leads to an improved QoE for the remaining video transactions.

5.3.2 Implicit Admission Control with Buffer Knowledge

One challenge in scheduling video streaming is the behavior during an overload situation. It is easy to avoid interruptions when there are sufficient resources to satisfy all demands. However, when the required data rate exceeds the cell capacity, it is important how the scheduler behaves. Without further measures, *SF* with video integrated according to Equation (5.1) would privilege video transactions with empty buffers in such a situation. This means that videos which still have buffered data get fewer resources and are likely to deplete their buffer, too. Thus, also video transactions which would principally have adequate channel conditions are affected by the interrupted ones. Consequently, the overall QoE degradation is worse than it could be. To improve this situation, we extend Equation (5.1) as follows:

$$F_{\text{video}}(t) = \begin{cases} a_{\text{SF}} \cdot (b(t) - b_{t,\text{SF}}) + c_{\text{min}} & \text{for } b(t) < b_{t,\text{SF}} \\ b(t) - b_{t,\text{SF}} + c_{\text{min}} & \text{for } b(t) \geq b_{t,\text{SF}} \end{cases} \tag{5.2}$$

Here, $b_{t,\text{SF}}$ is a target buffer size that can be chosen arbitrarily. The inclination a_{SF} for $b(t) < b_{t,\text{SF}}$ should be chosen negative, which means that a transaction gets a penalty when its buffer size falls below the target buffer[4]. This also applies to newly arriving transactions that start with an empty buffer.

[4] A further generalization would be the parameterization of the slope for $b(t) > b_{t,\text{SF}}$. However, a preliminary evaluation showed a small influence of this property, which is why it will not be discussed in this work.

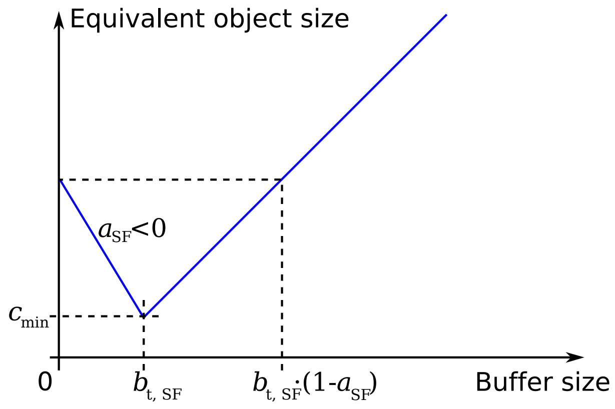


Figure 5.3: Implicit admission control by defining a target buffer and a cost penalty.

Figure 5.3 shows $F_{\text{video}}(t)$ over the buffer size $b(t)$. Assuming constant channel quality, the cost of a streaming transaction gets minimal at $b_{t,\text{SF}}$. Larger buffers mean that it gets less urgent to schedule the transaction; the cost increases proportionally with the buffer size. On the other side, $b(t) < b_{t,\text{SF}}$ for a running video means that it is likely that the encoding rate exceeds the possible transmission rate at the current cell load. Consequently, we give a scheduling penalty to this transaction in order to prevent it from impairing other videos. For a newly starting transaction either the channel quality should be above average or the overload should be gone in order to get scheduled. Thereby, an empty buffer is attributed the same cost as with $b(t) = b_{t,\text{SF}} \cdot (1 - a_{\text{SF}})$.

The extended equivalent object size according to Equation (5.2) introduces an *Implicit Admission Control (IAC)*. The desired effect of admission control is to restrict the access to a medium in case an additional request would lead to overload, which would mean an impaired QoS for all traffic on the medium. For example, in the IntServ approach this is achieved by only accepting a flow when all elements on the path have sufficient resources. This is what we accomplish with the proposed mechanism. In case of overload, the ongoing video transactions are preferred to newcomers. Therefore, their QoE remains favorable while otherwise many more transaction would be unsatisfactory.

In contrast to IntServ, the approach here is not binary but instead gradually adapts to the load situation and the level of buffered data of all active video transactions in the cell. Implicitly, the most resource inefficient videos – i. e. the transactions requiring most radio resources per second playback time – will be penalized. New transactions are still able to "jump" over the target buffer level, when their channel conditions overcompensate the initial penalty.

Thanks to its implicit nature, the proposed IAC is easy to implement and merely adds algorithmic complexity. No additional tracking of flow state or cell load is required besides the knowledge of buffer level, which we require for the integration of video streaming. One implementation detail that can further improve the effectiveness of IAC is that we can introduce negative buffer sizes. This means a linear extrapolation of the equivalent object size when a video has stopped playing. With increasing buffering duration, the penalty increases, as it becomes less likely that the video will be able to resume. Therefore, no further radio resources are wasted but instead are used to save other video transactions from being interrupted. The effectiveness of IAC will be investigated in detail in Section 7.3.2.

5.4 Influence and Benefits of the Proposed Algorithm

In this chapter, an approach to include the scheduling of video streaming into *Shortest-First* schedulers was presented. This allows for a seamless handling of different service requirements within a single scheduler instance and achieves a good QoE performance for interactive and for streaming traffic. The proposed algorithm allows to control the priority of video streaming relative to interactive traffic. Furthermore, the configuration of IAC allows to improve the average video QoE in overload situations.

In the low load case, tracking the client-side buffer level helps to avoid interruptions in video playback for all transactions where this is possible[5]. For high load, the proposed algorithm preserves the QoE of as many videos as possible. Effectively, videos that require many radio resources per second playback time, either because they exhibit a comparably high encoding rate or bad channel conditions, will be restricted by IAC in this case. As long as the average buffer level of the active video transactions is well above the target buffer size $b_{t,\mathrm{SF}}$, transactions with relatively small buffers obtain cost advantages which help to avoid playback interruptions. However, when the average buffer level drops near $b_{t,\mathrm{SF}}$ due to overload, the penalty of those videos that fall below $b_{t,\mathrm{SF}}$ becomes effective for the relative cost comparison and preserves the more resource-efficient video transactions. The usage of buffered data volume versus buffered playback time is favorable for this notion of resource-efficiency, as a video with low encoding rate will play longer with a certain amount of buffered data.

Summarizing the effects of the proposed algorithm, it can be stated that a further service differentiation is not required anymore. All transactions can be processed in the same allocation loop, which gives maximum flexibility to the scheduler for exploiting multi-user diversity and considering QoE requirements. A prerequisite is information about the instantaneous client buffer state at the base station. This can be achieved either by network-side inspection and estimation techniques or by application feedback from the UE. Apart from that, the proposed solution comes at a very low implementation effort and requires only the evaluation of a mapping function between buffer size and equivalent object size. IAC on its own is very flexible compared to traditional admission control and gradually becomes effective when ongoing video transmissions are in danger of QoE impairment from newly arriving traffic. The performance of the proposed algorithm in scheduling a traffic mix of streaming and interactive applications will be evaluated in Section 7.3.

[5] Even for a single video transaction, uninterrupted playback may not be possible when the encoding rate exceeds the cell capacity, e. g. when the respective UE has a very bad channel quality.

6 Simulation Model and Metrics

The proposed algorithms will be evaluated by system level simulations. To obtain valid results, it is important to model all relevant effects with respect to the desired performance metrics. Furthermore, input parameters like scenario layout, channel and traffic properties have an essential influence on the outcome. Therefore, the model and scenario will be introduced in Section 6.1. This includes the parameter set and layout of the cellular network, the channel and traffic models. Then, in Section 6.2, we discuss the performance metrics on which we focus in this thesis and how they are measured from the simulation results.

6.1 Simulation Model and Scenario

For the performance evaluation, a discrete event-based simulation is employed. This means that time only passes between events, discrete points in time at which the state of the system is evaluated. The simulation framework is based on the IKR SimLib [Sim] and IKR RadioLib. These simulation libraries provide the basic components and software architecture to model general communication networks (IKR SimLib) and cellular networks (IKR RadioLib).

The evaluations are aimed at the assessment of the latency introduced by MAC-layer queuing delays in a single cell. Therefore, neither the effects at other layers nor other MAC-layer mechanisms like retransmissions are modeled. The reason is that these effects are only barely influenced by resource allocation and are similar for all investigated scheduling algorithms. We focus on comparing the differences in queuing and transmission times between schedulers, so the other effects are out of the scope of this thesis. Even more, the LTE model we use is just a prominent technology example to cover the relevant behavior. However, scheduling algorithms are needed for all kinds of cellular networks, no matter how the resources are defined. So our findings are not limited to LTE but may also be applied to other OFDMA systems or, with some adaptations, to other multiple access methods like CDMA. Furthermore, the effects from which we abstract, most notably MIMO, power allocation, and Inter-Cell Interference Coordination (ICIC), are to a certain degree independent from the choice of time and frequency resources[1].

As a basis for the cellular network model, we use a general LTE Release 8 macro-cell deployment scenario. We will not completely stick to 3GPP standards but make some simplifications,

[1] Many scheduling heuristics from literature which consider the mentioned functions also separate the decisions of time-frequency resources and power allocation, for example [SL05]. Naturally, such an approach may not find the global optimum, but it reduces the computational complexity to a feasible level.

e. g. with respect to the flexibility of assigning PRBs and signaling overhead. As stated before, we are interested in the downlink behavior of a single cell. According to LTE specifications, OFDMA and FDD as well as the respective time and frequency resolution will be used. The modeled network has a carrier frequency of 2 GHz and a bandwidth of 10 MHz.

6.1.1 Cellular Network Layout

For evaluation, macro cells in an urban environment are considered. This means that the base station antennas are assumed to be mounted above rooftop with a base station distance of 500 m (3GPP case 1 [3GP06]). The regarded scenario is interference-limited, which means that the major influence limiting SINR is interference[2]. The cellular layout defines the SINR distribution (also known as SINR *geometry*).

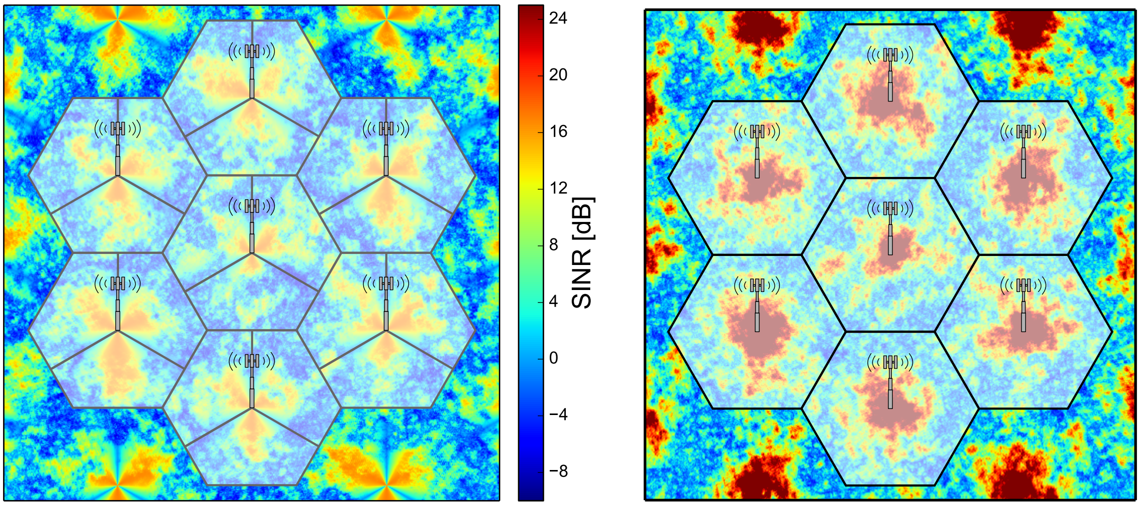
In real networks, base station locations are chosen with the help of network planning tools to optimize the coverage in a certain area with a given number of locations. For simulation, most studies abstract from the landscape topology and assume a deployment on a flat surface. This leads to a hexagonal arrangement of the base stations. The number of simulated base station locations usually represents a number of hexagonal rings around the center, e. g. 7 locations for one ring (as illustrated in Figure 6.1) or 19 locations for two rings. To avoid border effects, the scenario is wrapped around at the edges. That means that there are six duplicates of the simulated base stations arranged around the original version. A receiver thus always seems to be in the center cell with the other cells around it. Throughout this work, it is assumed that interfering cells always transmit at their full power in the whole spectrum.

Real cellular network deployments often have sector antennas at the base stations with typically three sectors to allow for a higher capacity per area compared to isotropic antennas. However, the simulation of a tri-sectorized scenario requires much more computational resources because a higher number of transmitting antennas have to be evaluated for each resource allocation. Due to the directional antenna pattern, an interferer at two times the inter-site distance still may have a significant influence on SINR. Therefore, at least two rings of interferers, which makes 57 transmitting antennas, have to be simulated in this case. To assess the influence of the cellular layout on the SINR, we discuss a comparison of different scenarios in the following.

Figure 6.1 shows two examples of a cellular network with an area distribution of SINR including path-loss and shadowing. On the left side, it shows a sectorized scenario with three sectors per location. The three sector antenna pattern from [NGM08], which has a backward attenuation of 20 dB, is in use. On the right side, a simple scenario with isotropic antennas and a single cell per location is shown.

The resulting geometries are compared in Figure 6.2. We can see that the isotropic cell layout leads to an optimistic SINR distribution compared to the three sector case. This is mainly due to the generally higher number of interfering cells with some interfering antennas pointing directly towards the evaluated cell. Especially at high SINR, the degradation is determined by the maximum attenuation of the antenna pattern. The backward attenuation limits the SINR to

[2] By contrast, rural areas may have much larger cells, where the signal at the cell border is comparable to the noise level. This is then called a noise-limited scenario.



(a) Tri-sectorized cell layout.

(b) Isotropic cell layout.

Figure 6.1: Comparison of different cell layouts.

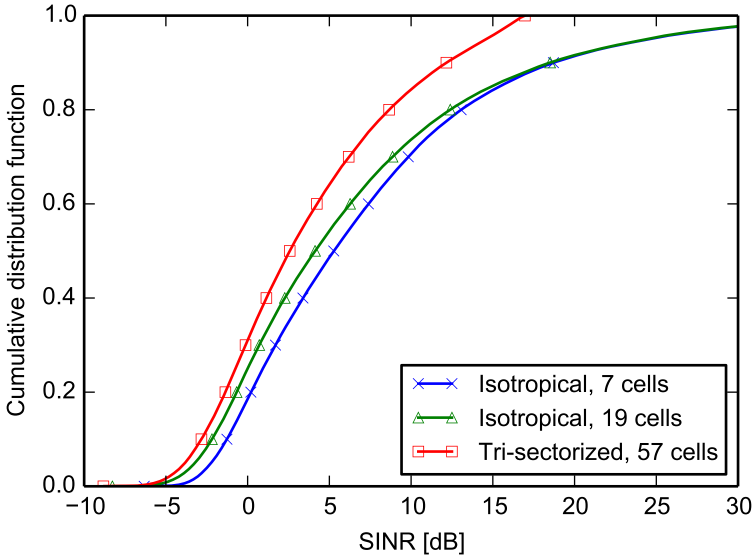


Figure 6.2: Cumulative distribution function of SINR for different scenario layouts.

17 dB, as there are two interfering antennas at the location of the transmitting antenna, which experience the same channel attenuation (when only considering path-loss and shadowing). For the isotropic case, we investigate one ring (7 cells) and two rings (19 cells) of interferers. Figure 6.2 shows a slight degradation in SINR when considering more interferers, but the general geometry is very similar.

For the performance evaluation, frequency-selective channels are simulated. To reach a feasible amount of computational resources, a single-sector scenario with seven base stations will be employed. The shape of the geometry in Figure 6.2 is, apart from few dB difference, very similar for all scenarios at low SINR. At high SINR, the large difference is mitigated by two effects in the final simulation model. First, the 17 dB upper bound only holds for the case without fast fading. However, we consider fast fading in our final channel model. Second, the capacity of the channel is clipped at 25 dB to account for the maximum transport format. As we are interested in the relative differences between the schedulers, the shape of the geometry is not essential for the results.

6.1.2 Channel and Mobility Model

The concept of the radio channel was introduced in Section 3.1. The channel model used in the simulation relies on recommendations by the Next Generation Mobile Networks (NGMN) alliance and 3GPP specifications, mainly [NGM08] and [3GP06]. In detail, three components make up for the attenuation of a single wireless link between transmitter and receiver: Path-loss, shadowing, and fast fading. We discuss these components in the following sections.

Deviating from many other studies published in literature, user mobility is also a part of the simulation model. The reason for this is the time-scale of interest. As discussed in Section 3.3.2, we want to capture the evolution of the radio channel over a time of several tens of seconds. During this time, even for the walking speeds of pedestrian users, the attenuation level of the radio channel can change significantly. Then again, for short flows, the time-scale separation argument sometimes found in literature [SDV10, APLO11] (see Section 3.6.2 for a discussion) is not applicable to LTE networks as they can be completely transmitted within few slots. For Internet traffic, such short flows contribute a major part to the total utility of the users' applications. Even more, the short flows are especially time-critical as they often belong to interactive tasks (e. g. browsing through the catalog of a web shop or an image gallery). Consequently, we model at the TTI-level and include user mobility, which is presented in Section 6.1.2.4.

6.1.2.1 Path-Loss

Path-loss means the signal attenuation over the distance between transmitter and receiver. It strongly depends on the environment. In free space with isotropic antennas, the power of the signal would decay proportionally to the distance squared [TV05]. However, with reflection and diffraction of the electromagnetic waves at environmental obstacles, the exponent in relation to distance changes. As proposed for a macro-cellular system in an urban area in [3GP06], distance-dependent path-loss a_{PL} in dB is determined as

$$a_{\mathrm{PL}} = 128.1 + 37.6 \log_{10}(d) \tag{6.1}$$

where d is the distance between sender and receiver in kilometers. This model is defined for a distance $d \geq 35\,m$ between transmitter and receiver. However, as we limit the SINR at a value of 25 dB anyway, we do not impose any restrictions on the receiver placement.

6.1.2.2 Shadowing

Obstacles in the path between transmitter and receiver further attenuate the signal power beyond pure path-loss. This effect is called shadowing. Gudmundson [Gud91] proposes a statistical model to capture the fading effects from shadowing, namely an exponential function as the spatial auto-correlation. From [3GP06], we use a correlation distance of 50 m, a standard deviation of $\sigma = 8$ dB, and a correlation of $\rho = 0.5$ between cells. The latter means that the shadowing attenuation of a UE towards two different base station is correlated. This respects the fact that the environment in the vicinity of the UE is the same in both cases.

As we include user mobility into the simulation model, we are able to employ a model with a realistic spatial correlation. This means that each receiver crossing a certain location perceives the same shadowing attenuation. To this end, a stochastic sum-of-sinusoids model is used, which is implemented in IKR RadioLib following the proposal in [CG03]. For each transmitter (i. e. the base stations) a random shadowing plane is created and an additional plane common to all transmitters. The common plane is used to create the correlation between different senders. This models the environment in the vicinity of the UEs, which is the same for all transmitters. We obtain a correlation coefficient ρ for the shadowing between transmitters with

$$a_{SH} = \sqrt{\rho\sigma^2} a_{SH,1} + \sqrt{(1-\rho)\sigma^2} a_{SH,2}, \tag{6.2}$$

where a_{SH} is the shadowing attenuation in dB, σ is the standard deviation and $a_{SH,1}$ and $a_{SH,2}$ are the values from the transmitter-specific and the common shadowing planes, respectively.

6.1.2.3 Fast Fading

The transmitting antenna emits the radio signal into multiple directions (as given by the antenna pattern, e. g. equal strength in all directions for isotropic antennas). This signal gets reflected at obstacles in the environment. Thus, it may not only reach the receiver along the direct Line of Sight (LoS) but along multiple paths. Each path can have different a propagation delay and attenuation experienced by the signal along this path. The different signal versions add up at the receiver side and can interfere constructively or destructively, depending on their relative phase shifts. This leads to fast fluctuations of the received signal strength at the receiving antenna.

Such fast fading from multi-path propagation is modeled by Rayleigh fading[3] when there is no dominant path (e. g. from a strong LoS component, see [TV05, Ch 2.4.2]). We assume this case of non-LoS in our studies, which is common for urban environment models. Due to the different path propagation delays, a single transmitted impulse will be spread in time (called *delay spread*), which can be described by a tapped Channel Impulse Response (CIR). We use the *typical urban* channel model for which the CIR is given in [3GP13] and which is also recommended by other simulation scenario descriptions for the SISO case [NGM08, 3GP06].

[DBC93] describes a way to generate pseudo-random Rayleigh fading with a sum-of-sinusoids model with random phases[4]. As input, we need to define the carrier frequency and the maximum frequency offset of the fading process. The reason is that the relative movement between transmitter and receiver creates a Doppler shift depending on the speed of this movement, i. e. the user's speed determines the spectrum of the fast fading.

[3] More precisely, the amplitudes of the individual paths are modeled as Rayleigh random variables [TV05].

[4] A modification of the Jakes model which improves the statistical properties of the original model [DBC93].

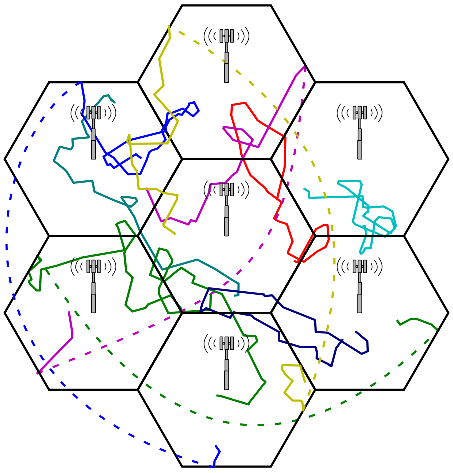


Figure 6.3: Example of user trajectories with the mobility model.

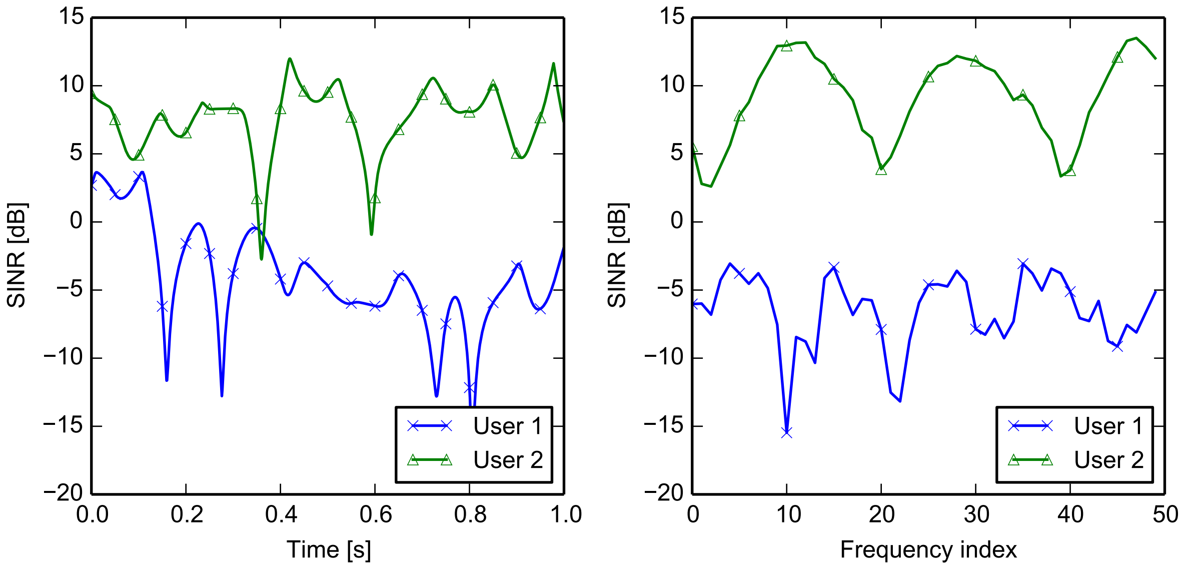
For the frequency-selective fast fading model, we assume flat fading within a single PRB. That means that we have one fading value for a PRB and abstract from the individual sub-carriers within a PRB.

6.1.2.4 Mobility Model

Because we look at a time-scale of some tens of seconds, we need to include user mobility into the channel model. To this end, a random walk is implemented. Users walk on a straight line for a random time. After this time, they turn randomly and continue walking into the new direction. Figure 6.3 illustrates an example of some user trajectories. It also shows the effect of wrap-around. A user leaving the scenario at the bottom enters it again on the top. In the point of view of the UE, it always senses to be in the central cell with 6 interfering cells around it. Once the UE moves from one cell to another, the remote interferers will appear as neighbors to the currently visited cell. As suggested by the simulation recommendations [3GP06, NGM08], all users have the same speed of 3 km/h.

It is important that the users move at all to avoid that the path-loss and shadowing components of their channel remain static. This would be unrealistic for pedestrian or vehicular users at the observed time-scale. To capture this variability, the simple model described above is employed. While modeling basic mobility is essential for the temporal evolution of the signal strength, handovers are not a part of our observations. We are interested in the behavior of a scheduling algorithm performing resource allocation in a single cell. To this end, users entering or leaving the cell are not essential for the base station as long as the average number of users remains constant. For a stable operation point, the number of receivers in the cell (i. e. the number of UEs) is kept constant irrespective of the mobility model. We therefore model a *virtual* cell, which serves all UEs no matter to which base station antenna they are currently connected to. This means that the channel conditions behave as if the users would move all over the scenario[5], whereas the resources are allocated as if all users were placed in a single cell. From the user's perspective, it is not important in which cell the UE is registered as long as the number of

[5] For SINR determination, the UE is always assumed to be attached to the base station with the strongest signal.



(a) SINR evolution over time (single frequency).

(b) Frequency selectivity of SINR.

Figure 6.4: Example channel realizations with Rayleigh fading.

competing transactions from other users results in the same traffic intensity. In contrast to the mobility model, the traffic model has an influence on the number of active receivers because not all UEs may have something to transmit[6].

6.1.2.5 Summary of the Channel and Mobility Model

Summarizing, the channel and mobility model represents an outdoor scenario with macro-cells and slowly moving users. The channel model gives the attenuation of a single link, i. e. the link between a pair consisting of UE and base station. For the correct reception of a transport block, as discussed in Section 3.1, the SINR of the signal is relevant. To calculate the SINR according to Equation (3.1), the channels from all base stations to a UE have to be evaluated. For simplicity, because we are interested in the effects from scheduling within a single cell, all interfering base stations are assumed to transmit permanently at their full transmit power.

Figure 6.4 shows two example realizations of the described channel model. On the left side, the SINR for a single frequency is plotted over time. The characteristic *deep fades* of the signal can be seen. The fluctuation rate of these fades depends on the speed of the UEs. The right side of Figure 6.4 shows the SINR depending on the subcarriers at a single point in time. We can see that the signal also varies over frequency. A summary of the parameterization of the whole radio model is given in Table 6.1.

[6] We will see in Section 7.1.1 that the interaction between scheduler and traffic model creates a feedback-loop with respect to the spatial distribution of receivers, as UEs with good channel are likely to complete their requests quickly.

Table 6.1: Parameterization of the radio channel model and cellular layout (mostly complying with [3GP06, Table A.2.1.1-3]).

Property	Value
Cellular layout	Hexagonal, 7 sites
Antenna pattern	Isotropic
UEs per cell	50
Inter BS distance	500 m
BS/UE height	32 m / 1.5 m
Carrier frequency	2 GHz
System bandwidth	10 MHz
BS TX power	46 dBm
Path-loss	$128.1 + 37.6 \log_{10}(d)$, distance d in km
Shadowing	$\sigma = 8$ dB; log-normal; correlation distance 50 m
Shadowing correlation	$\rho = 0.5$ between cells
Multipath propagation	Rayleigh fading with Jakes-like temporal correlation [DBC93], frequency-selective fading with typical urban channel taps [3GP13]
UE velocity	3 km/h
Mobility model	Random walk; mean walk duration 30 s
Frame duration	1 ms
Link adaptation	Ideal (Shannon-Hartley; SINR clipped at 25 dB)
Inter-cell interference	Interfering cells transmit at full power

6.1.3 Link Layer Model

The channel model provides the SINR for the radio transmissions. However, to assess the influence of the scheduler on the application level performance, we need the capacity in bits that can be transported in the respective resource allocations.

A real LTE systems employs FEC through turbo codes. Together with different modulation levels (i. e. QPSK, 16-QAM, and 64-QAM for the data channel), this makes up the so-called MCS. Each coded transport block has a certain probability to be decoded successfully at the receiver. This probability increases with the SINR. The higher the order of the MCS[7], the higher the SINR needs to be for a successful decoding. Therefore, a trade-off exists between using a high MCS to transport as many bits as possible per OFDM symbol and reducing the BLER. This is controlled by defining a *Target-BLER* and matching the MCS to the CQI at the transmitter. Transport blocks that cannot be decoded successfully at the receiver need to be retransmitted. LTE defines HARQ (e. g. soft bits with incremental redundancy) and ARQ (transmitting a new transport block containing the missing data) for this purpose.

6.1.3.1 Ideal Link Layer Adaptation

For the link layer model, we use the information theoretical upper bound of the channel capacity, which is a common assumption to abstract from link layer adaptation. The Shannon-Hartley theorem (see Equation (3.2) and [Sha49]) defines the amount of mutual information

[7] Higher MCS means more bits per symbol and less code redundancy.

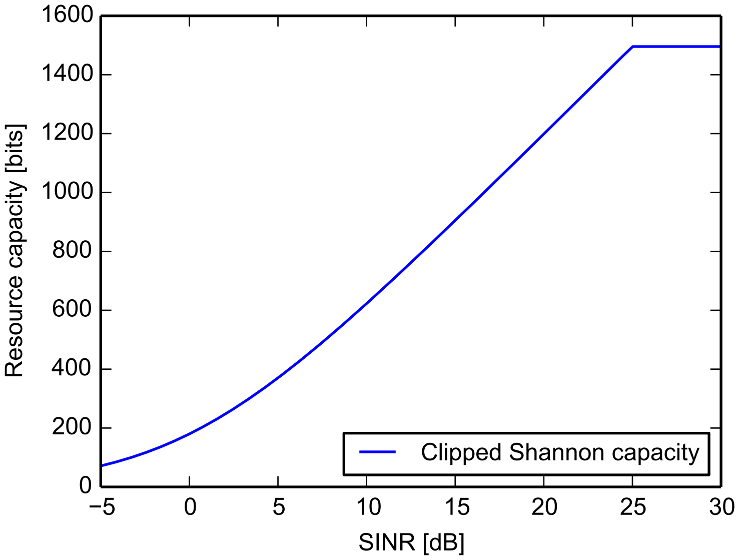


Figure 6.5: Capacity of a scheduling resource (180 kHz, 1 ms) according to the ideal spectral efficiency defined by the Shannon-Hartley theorem, clipped at 25 dB.

that a sender can transmit ideally towards a receiver at a certain SINR[8]. By using the Shannon bound, we assume ideal CSI at the sender, a perfect channel adaptation (i. e. no discrete MCSs) and abstract from transmission errors and retransmissions. We discuss the influence of realistic channel impairments in the next section. To model a maximum number of bits that can be transported per resource (i. e. the maximum *spectral efficiency*), we limit the capacity to the value at SINR = 25 dB. A resource unit is equivalent to two LTE PRBs, because we adapt it to the scheduling interval. LTE specifications define slots of 0.5 ms, which means 7 OFDM symbols in time that make up one PRB [3GP09b]. However, two consecutive slots make up a sub-frame, which is the scheduling granularity in time (i. e. the TTI), and is therefore suitable for our resource definition[9]. This means that a single resource, with a bandwidth of 180 kHz and 1 ms duration, neglecting signaling overhead and pilots, can transport at most 1495 bits. Figure 6.5 shows the number of bits such a single resource can transport depending on the SINR.

6.1.3.2 Influence of Realistic Channel Impairments

An ideal adaptation to the radio channel is not realistic for several reasons. First, the CSI can never be ideal due to limited signaling capacity and measurement errors. Second, due to processing and signaling times, the CSI always exhibits a delay between measurement and its availability at the sender. Third, channel adaptation with modulation and coding is not seamless. There are discrete steps in the MCSs and in the granularity of the adaptation in frequency and time. Consequently, the sender is not able to choose a modulation format and code rate that ideally fit to the current channel state.

[8] This information theoretical upper bound holds for a SISO system and can be overtaken with MIMO.

[9] Restrictions apply to the resource allocation between the first and second slot of a sub-frame, like the maximum number of concurrent users and frequency patterns. We abstract from this by freely allocating in frequency once in a TTI.

Delay, measurement errors and quantization lead to a statistical error distribution for the CSI, which is used by the sender to determine the MCS of a transmission. If the CQI (i. e. the quantized version of CSI) is lower than the actual channel quality, the sender will use an MCS not exploiting the available capacity. On the other hand, if the CQI is too high, the sender will use an MCS with too little protection such that the receiver cannot decode the respective transport block due to transmission errors. Then, a retransmission is required. To avoid a large wastage of capacity or excessive retransmissions, senders are configured to approach a certain defined Target-BLER[10]. A decode error leads to a HARQ retransmission in LTE, which commonly adds a delay of 8 ms to the transmission[11][STB09, p.234].

Effectively, realistic channel impairments lead to a reduced spectral efficiency and the probability of additional transmission delays in case of retransmissions. For our evaluation, it is important how these effects would influence the performance metrics for different schedulers.

The reduction in spectral efficiency affects cell throughput and, consequently, transaction durations and QoE. While it is not the focus of this thesis, there are many sources in literature covering the effect of imperfect CSI. For example, an early work investigating the impact of imperfect CSI on a single OFDM connection with adaptive modulation is [YBC02]. More recent publications that evaluate the capacity degradation for the downlink of a multi-user OFDMA system are [KK08, VW10]. Furthermore, many works exist that propose how to mitigate the drawbacks from imperfect CSI, e. g. by channel prediction or adapted resource allocation algorithms. The cited works [YBC02, KK08, VW10] are in agreement that the delay of the CSI is the essential factor determining the reduction in spectral efficiency, more accurately, the relation between the coherence time of the channel and the delay. For our standard scenario with a user speed of 3 km/h, assuming an average feedback delay of $\tau_D = 5$ ms, the relevant metric $f_D \tau_D \approx 0.028$ (f_D is the Doppler shift) is very small. For this value [YBC02] and [KK08] attest a very small impact of imperfect CSI, while [VW10] is more pessimistic. Here, the different impact of imperfect CSI on the evaluated schedulers is important. It is likely that *Max C/I*, which profits most from frequency-selective multi-user diversity, is affected most. However, also the other schedulers (with the exception of *RR*) are opportunistic and consider CSI for their scheduling decisions. We therefore do not expect large relative differences in throughput performance from imperfect CSI.

Retransmissions affect the metrics transaction duration and user experience. The delay from a retransmission only adds to the total transaction duration if it happens at the end of a transaction. In this case, it only has a significant influence if the transaction could have been transmitted within a single or few TTIs and would have a very small waiting time without a transmission error. The impact of retransmission on the overall results therefore would be very small, because it would affect only few transactions and the order of the additional delay around 10 ms is very small on the time-scale relevant for QoE (see Section 6.2.3).

Concluding, realistic channel impairments would primarily influence the spectral efficiency of our cellular system. However, we expect only minor relative differences of this effect between the evaluated schedulers. Furthermore, the potential influence of retransmissions on the evalu-

[10] For example Target-BLERs between 0.1 and 0.02 for downlink transmissions in LTE are defined in [3GP14b].

[11] There also is a small probability that a decode error cannot be recovered by multiple HARQ retransmissions. Then, ARQ (at the RLC-layer) initiates a new transmission for the MAC-layer, which leads to additional delay.

ated metrics is limited. We therefore do not consider such channel impairments in the simulation model.

6.1.4 Traffic Scenarios

The nature of offered traffic plays a crucial role in the overall system performance. First, it is central to the user experience that shall be assessed, how the application layer objects (i. e. transactions) are transmitted and which requirements the respective application has, especially with respect to latency. Second, the object size distribution defines the burstiness of the traffic aggregate and how long a certain transaction remains in the system.

A negative exponential distribution will be employed for the transaction arrival process[12]. This allows us to simply vary the amount of offered traffic by adjusting the mean Inter-Arrival Time (IAT). Together with the object size distribution, covered in more detail in the next sections, the resulting traffic intensity is controlled by this IAT rather than the number of UEs. The number of simulated UE is always fixed to 50. This guarantees that we have a sufficient diversity of receiver locations and channel realizations. For a single simulation point, the IAT is kept constant. To obtain the dependency between the observed metric and offered traffic, the simulation is conducted for a range of IAT settings.

As introduced in Section 2.4, we model three application classes: foreground and background interactive traffic and buffered video streaming. Whereas the arrival process is assumed to be identical for all transactions, the object size distributions and server behavior are different. The traffic mix, i. e. the share of the individual application class in the total traffic will be varied in the evaluation studies.

When simulating realistic traffic consisting of actual traffic objects, it is important to reach a stationary state. This is especially challenging in cellular networks as not only the offered traffic varies on a short time scale, but also the cell capacity is variable. A stationary load situation is reached by assuming that users abandon their transactions after a certain time. A dropped transaction means that it is immediately removed from the buffer in the base station. In the equilibrium the following holds:

$$R_{\text{arr}} = R_{\text{fin}} + R_{\text{drop}} \tag{6.3}$$

With R_{arr} being the rate of arriving traffic (or *offered* traffic), R_{fin} being the traffic rate of finished transactions and R_{drop} being the rate of the dropped traffic. The relation between the rate of traffic arrivals and the average capacity of the cell influences the number of dropped transactions. In a low load situation, almost all transactions get finished and the dropping rate is close to zero. However, when the cell is in overload[13], it is not possible to serve all transactions anymore so the dropping rate will initially increase with the simulation time until it matches the difference between finish rate and arrival rate. We are interested in the properties of this steady

[12] In teletraffic theory, the corresponding Poisson arrival process is assumed when there is a large number of independent traffic sources. This approximation is also used here.

[13] Please note that the average arrival rate is constant during a simulation run and therefore the overload is permanent.

state. Therefore, the traffic process and buffer levels define the duration of the initial transient phase. During this phase, no samples are collected.

6.1.4.1 Standardized Web Model

The base line traffic scenario applies the web model recommended in [NGM08] and in 3GPP specifications. [NGM08] provides models for File Transfer Protocol (FTP), web browsing (HTTP), video streaming, VoIP, and gaming. We will use the FTP and web browsing models for interactive elastic traffic. For video streaming, a new model will be derived from recent publications, see Section 6.1.4.3.

The object size distributions are based on truncated log-normal distributions. We use the FTP model for the background traffic class. It simply draws the object sizes from a single truncated log-normal distribution with the Probability Density Function (PDF)

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{\frac{-(\ln x - \mu)^2}{2\sigma^2}} \tag{6.4}$$

and the parameters given in Table 6.2.

For interactive web browsing traffic, [NGM08] defines a more complex object model consisting of multiple distributions. It bases on the structure of web sites consisting of a main object and embedded objects. To this end, three distributions are defined:

- The size of the main object as a truncated log-normal distribution.
- The number of embedded objects as a truncated Pareto distribution.
- The sizes of embedded objects as a truncated log-normal distribution.

Table 6.2 gives the parameters for the object size distributions. The number of embedded objects is distributed as [3GP04][14]

$$f_X(x) = \begin{cases} \frac{\alpha_k^\alpha}{x^{\alpha+1}} & \text{for } k \leq x < m \\ \left(\frac{k}{m}\right)^\alpha & \text{for } x = m \end{cases} \tag{6.5}$$

with $\alpha = 1.1, k = 2$, and $m = 55$[15]. To get the number of embedded objects N_d, k shall be subtracted from the outcome of the (pseudo-) Random Number Generator (RNG) with the indicated PDF, which gives a maximum number of embedded objects of $N_d = 53$.

For the web traffic model, we assume that the main and embedded objects of a transaction arrive as a whole at the base station, i. e. in a single transmission. This simplification compared

[14] The description in [NGM08] contains an error, which is why the original source [3GP04] is given.

[15] The special probability at the upper bound m indicates that no resampling shall take place when the RNG of the underlying unbounded Pareto distribution returns a value $> m$, but instead m shall be returned.

to the original behavior can be justified for example by content caches in the operator's network which reduce the RTT to the server, or by approaches like in the propositions for HTTP/2.0 (or *SPDY*) [BELSHE2] which introduce a server push mechanism, meaning that the server may be allowed to send embedded objects without an explicit request. Furthermore, as mentioned before, this thesis focuses on the delays introduced by scheduling and abstracts from application layer effects.

6.1.4.2 General Traffic Model

The web model recommended in [NGM08] relies on the description in [3GP04] which in turn uses findings from publications on traffic traces measured in the years 1998–2000 [CL99, SCJO01]. A recent study in an LTE network in the year 2012 supports the assumption that most transmissions are very short [HQG+13]. Also [FLM+10] indicates that traffic contains a large number of very small objects.

However, other recent measurements of Internet traffic [HTT, IP11] suggest that object sizes, at least in the fixed Internet, are much larger than the ones from the model in [NGM08]. An example scenario for this could be notebooks equipped with LTE dongles, which produce traffic similar to clients connected through fixed access networks (e. g. residential broadband). Therefore, we complement the studies by a more generic traffic model with an adjustable object size distribution. For this general traffic model, we only use a single traffic class for foreground and background applications.

According to analytic interpretations of large scale traffic measurements in [HCMSS04] and references therein (e. g. [Dow01]), Internet traffic object sizes may be modeled as a mixture of multiple Pareto and/or log-normal distributions. For simplicity, we take a single truncated log-normal distribution for the general traffic model (comparable to the FTP model with a different parameterization). We do not aim to capture all details of Internet traffic but want to have a tractable distribution which can be controlled easily and contains larger objects than the web model introduced above. Nevertheless, the model should contain a large number of small objects, which many measurement studies attest. This abstraction is backed by [GLMT01] which argues that the tail of the size distribution has little practical influence on algorithm design and is very sensible to changes in the underlying assumptions. Furthermore, truncation is required to simulate a stationary system as arbitrarily large objects would accumulate over the simulation time otherwise.

For parameterization, [IP11] is used as a basis and an offset is introduced to the mean object size to come closer to the average total transfer size of web sites reported in [HTT] for today's traffic. Importantly, the claim is not to model realistic traffic of today's cellular networks but the objective rather is to investigate scheduler performance in the face of traffic using characteristics from the mentioned measurements. Section 4.3 of [IP11] contains a discussion of entire page characteristics. The average median for small, medium and large pages is ≈150 kBytes[16]. The largest pages have a size of up to 370 MBytes. By taking a median of 150 kBytes and an average object size of 1800 kBytes (approximately the average in the year 2014 on [HTT]), we obtain the parameters in Table 6.2. Truncating at 100 MBytes is above the 0.998 quantile and is necessary

[16] To assume a median for the general traffic model, this simple rule of thumb estimation shall suffice.

Table 6.2: Parameters for the standardized [NGM08] and general traffic models.

Object type	μ	σ	minimum	maximum
FTP	14.45	0.35	0	5 MBytes
HTTP main obj.	8.37	1.37	100 Bytes	2 MBytes
HTTP embedded obj.	6.17	2.36	50 Bytes	2 MBytes
General	$\ln(1.5 \cdot 10^5)$	≈ 2.229	0	100 MBytes

to avoid that outliers distort the simulation. Both, the standardized web model and the general traffic model, pose different challenges on the schedulers that could occur in realistic traffic situations and may strongly affect the user experience.

6.1.4.3 Video Traffic Model

The video streaming model in [NGM08] represents life-streaming and defines a traffic pattern for packets transporting individual frames. Therefore, it is not suited for the case of buffered streaming. For this, we rely on measurements in [ARMNOLS12, FMM$^+$11, RMPGA$^+$14] instead.

[ARMNOLS12] provides a very detailed model of the sending behavior of YouTube application servers. The sending rate scales with the encoding rate V_r of the transmitted video. First, an initial burst containing 40 s of playtime is sent out as fast as possible. Then, the server throttles the sending rate relative to V_r. During this throttling phase, the server emits bursts of packets with an IAT to obtain the desired rate. In [RMPGA$^+$14] the measurement methodology from [ARMNOLS12] is applied in mobile networks. Whereas the general behavior is similar, the parameters are slightly different. Table 6.3 gives the parameters from [RMPGA$^+$14], which will be used in the model.

When the video buffer at the client-side is empty, it will not start playing. [SHW$^+$10] details on the buffering behavior of the YouTube client application. The authors could not exactly determine how much buffered playtime is required before a paused video starts playing. However, in their stalling experiments, videos resumed playing with 2 s worth of playtime in the buffer. We therefore use this value in the video model. Furthermore, for simplification, we assume unlimited buffer space at the client.

A video object has three properties – size, encoding rate, and duration – of which two define the third one[17]. We are interested in MAC layer scheduling and the interaction of traffic objects and application classes and therefore model the decisive properties for this focus: size and encoding rate. The video duration then follows from them.

For the video object properties, we rely on the findings in [FMM$^+$11]. Although [ARMNOLS12] and [RMPGA$^+$14] also derive representative distributions of encoding rate and duration for the video files on YouTube, we require the properties of the resulting traffic seen in the network. For example, some videos are very popular and requested by many users or users may not watch a video completely. This leads to differences between the video properties

[17] We do not focus on modeling the exact video behavior and assume a constant encoding rate for a transaction.

Table 6.3: YouTube sending behavior [RMPGA$^+$14] and video properties [FMM$^+$11].

	Property	Value
Sending behavior	Initial burst time	34 s
	Rate throttling	$2 \cdot V_r$
	Burst size	64 kBytes
Video properties	Size distribution	Truncated log-normal
	Size parameters	$\mu = 16.4$; $\sigma = 1.2$
	Size limits	min = 1 MByte; max = 100 MBytes
	Encoding rate distribution	Uniform
	Encoding rate limits	min = 200 kbit/s; max = 900 kbit/s

and the actual traffic. The distribution of video sizes is given in Figure 4 of [FMM$^+$11]. This is approximated by a log-normal distribution with the parameters in Table 6.3.

The default resolution measured in [FMM$^+$11] is 360p for both, PCs and mobile devices, accounting for around 80% of videos in both cases (*itag* 34 and 18, respectively). The empirical CDF of the corresponding encoding rate roughly lies between 200 kbit/s and 900 kbit/s, which is in agreement with [ARMNOLS12] for *itag* 34. For the video model, we assume a distribution of encoding rates approximating this common case. Consequently, the encoding rates are randomly selected such that a uniform distribution between 200 kbit/s and 900 kbit/s is obtained (see [ARMNOLS12, Figure 4a]). [RMPGA$^+$14] reports smaller resolutions and encoding rates (*itag* 17 and 36) from measurements in 3G mobile networks. However, due to the higher available bandwidth, it is more likely that the default video encoding bit rate in LTE networks is similar to that of videos accessed by mobile devices with a WiFi connection, as in the measurement of [FMM$^+$11].

[FMM$^+$11] does not provide information on the correlation between size and encoding rate. Therefore, by independently choosing values from the discussed size and encoding rate distributions, we get video durations bounded between 9 s and 4200 s. While the lower bound is reasonable and in agreement with the sources, it is not feasible to cover 4200 s in the simulation. The duration CDF in [FMM$^+$11] stops at 1000 s with $\approx 99\%$ of videos being shorter. Therefore, the duration is limited to $1000\ s$ by choosing a new sample when the combination of size and encoding rate leads to a longer duration. Figure 6.6 visually indicates that the video model fits well to [FMM$^+$11] with the lower duration quantiles being a little smaller.

The publications [FMM$^+$11] and [RMPGA$^+$14] observe a different transmission behavior between YouTube servers and mobile clients compared to requests originating from PC clients. We do not model such special mobile client behavior because [RMPGA$^+$14] states that two out of three mobile devices exhibited a behavior very similar to a PC client. Furthermore, for simplicity, an unlimited buffer at the client is assumed.

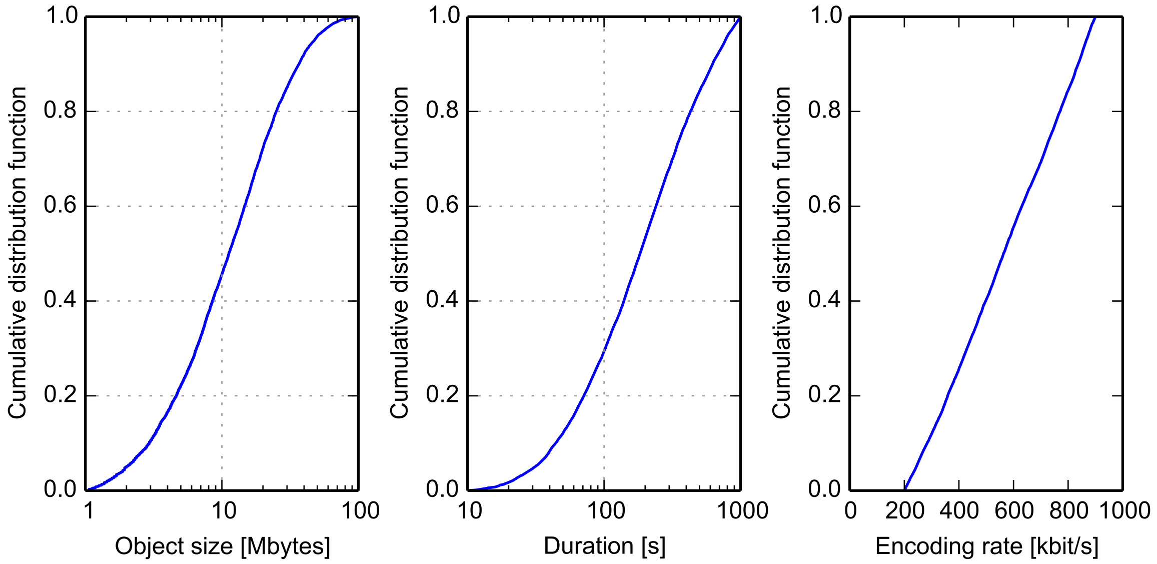


Figure 6.6: Empirical CDFs of video object properties sampled from a simulation run. The video traffic model is parameterized to fit to the measurements in [FMM+11].

6.1.5 Reference Scheduler Parameterization

A standard set of parameters is used for the reference schedulers. These parameters are summarized in Table 6.4 and are taken from specifications commonly used in standardization (e. g. 3GPP and 3GPP2). The parameterization of *PF-QoE* is taken from the original source.

There is a slight difference in the scheduling approach between the reference and the *Shortest-First* schedulers. The reference schedulers *Max C/I* and *PF* are frequency-selective, which means that they decide for each resource unit to which transaction to allocate it. On the other hand, the *Shortest-First* schedulers decide in each TTI on the preferred transaction and allocate as many resources as required to it. When there are resources left, the scheduler selects the next transaction. This behavior is logical for transaction-aware schedulers, because they aim to finish the prioritized transaction as fast as possible.

6.2 Performance Metrics

With the model and scenario described in the previous section, it is possible to simulate the transmission of application layer transactions within a cell of a mobile network. This section introduces the relevant metrics for performance assessment. In the evaluations, not only the average behavior of a certain metric, but also the distribution or the correlation with other parameters is of importance. This will be detailed in the discussions of the respective results.

6.2.1 Cell Throughput

A central metric in any network is the ability to transport data from one point to another. In our studies, we investigate the downlink transmission rate of a single cell, also called cell through-

Table 6.4: Standard parameterization of the reference schedulers.

	Description	Parameter	Value
PF, *TAOS 2* & *PF-QoE*	Forgetting factor[18]	β	$1/1500 (\approx 0.00067)$
	Initial moving average[18]		10^{-15}
PF-QoE [NOALS+13]	Offset	f_{min}	1
	Target buffer	b_t	15 s
	Priority at target	f_t	20
	Slope factor	a	1
	Initialization factor	f_{init}	0.25

put. Apart from the cell throughput, also the distribution of data rates to individual users is important as it determines the QoE of these users.

The possible cell throughput not only depends on the technology and the available bandwidth of the base station but also on the traffic and user models. Network assessments are often conducted with a full-buffer assumption, i. e. all users transmit at the maximum possible rate. For example, in [NGM08] most throughput metrics require this assumption. However, in a practical situation this assumption is not common. In most cases, an application will transmit a certain amount of data (here: one transaction) and then wait for the next user interaction. This property creates a feedback-loop between resource allocation and the user distribution in the cell. We therefore calibrate our system scenario in Section 7.1.1 with respect to cell throughput in order to put throughput figures into relation with what is achievable with full-buffer traffic and with more realistic traffic scenarios.

Naturally, cell throughput has an impact on the per-transaction metrics duration and utility. A higher cell throughput enables shorter transaction durations and better utility. However, also the throughput on its own is an interesting metric, because it states how much offered traffic a cell can handle. A scheduler providing a good throughput performance is therefore less likely to be overloaded and is able to transport more data volume than another scheduler with worse cell throughput.

6.2.2 Transaction Duration

A very important metric for the human perception of interactive applications is how fast a reaction to an interaction occurs. The faster the response of a network application arrives, the better is the user experience. While we discuss the modeling of this QoE assessment in the next section, the underlying metric is *transaction duration*.

We account as transaction duration t the time between the arrival at the base station and the transmission of the last bit to the UE. As introduced in Section 6.1, only the queuing time is observed in the simulation. This is the metric directly influenced by the scheduler and with a major impact on user QoE.

[18] In literature, time constants for the exponential moving average ranging between 0.1 s [Bon04] and 1.67 s [JPP00] (sometimes even larger) are proposed, attributing to the trade-off between system throughput and worst-case latency. We use the model specified in [3GP04] with a time constant of 1.5 s and initialization close to zero. A test study for the parameterization of β is presented in Appendix A.

Followingly, a transaction could be transmitted in zero time, when it fits into the resources of a single TTI and arrives shortly before the scheduling time. In real systems, there is an offset to this for the total transmission time. Among others, it comprises network-latency, modulation and coding, and propagation delay. In our scenario, with non-real time applications, an abstraction from higher layer effects and the assumption that the radio access link is the bottleneck, this offset is usually below the human perception threshold[19]. Under heavy load, buffering time represents the major part of transmission delay and can range from hundreds of milliseconds to several minutes, depending on the size of the transaction.

In our studies, the transaction duration is relevant for interactive transactions. Transaction durations are not investigated for video streaming, because the duration heavily depends on the video properties and the server behavior in this case. Furthermore, the latency alone is not sufficient to assess the QoE of video playback, as we discuss in the next section.

6.2.3 User Experience and Utility Functions

In this thesis, the objective is to compare different schedulers with respect to their influence on user experience of network applications. We therefore need a model to assess this QoE quantitatively. Besides being influenced by subjective impressions of the users, the QoE behavior also depends on the application. A prominent use case in the Internet is interactive web surfing. Users like a fast response when they click on a link, whereas they get annoyed when this takes too long ([Nie10], see Section 2.2.1). To capture such effects quantitatively, utility functions are employed.

According to the description in Section 2.4.2, we discuss the parameterization of utility functions for the cases of interactive elastic traffic and buffered video streaming. The utility is normalized to the interval [0, 1], where 1 is the best utility (instantaneous transmission) and 0 the worst (aborted transmission).

This choice of utility functions allows it to compare the QoE performance of different schedulers. The findings in this thesis do not depend on the exact choice of these parameters. Instead, the relative difference in the behavior of the schedulers is of importance.

6.2.3.1 Interactive Elastic Traffic

The approach of utility functions for interactive elastic traffic was presented in [PKV11] and [PKWV12], the offset was introduced in [Pro14]. Following from the discussion in Section 2.4.2.1, S-shaped functions are used. To obtain such a shape, we choose the logistic function

$$U'(\tau) = \frac{1}{1 + e^{s(\tau - \tau_{\text{infl}})}} \tag{6.6}$$

where τ is the transmission duration of the transaction, τ_{infl} is the inflection point, and s scales the steepness of the curve.

[19] In some cases, network-latency is significant (e. g. for long distance links: RTT between UK and NZ ≈300 ms [Ver]). However, for most popular web-sites, content delivery networks choose servers near the receiver, which eliminates this problem.

The inflection point is modeled in relation to the user expectation. It is assumed that a user expects a certain data rate r_{exp} for the service to be satisfactory. Together with the size of the transaction F, this bandwidth translates to an expected duration. There is a lower bound $\tau_{\text{exp,min}}$ corresponding to the human perception (as discussed in Section 2.2.1) so that we get

$$\tau_{\text{exp}} = \max\left(\frac{F}{r_{\text{exp}}}, \tau_{\text{exp,min}}\right). \tag{6.7}$$

The expected bandwidth itself depends on the application. For example, users accept longer durations for background tasks than for interactive ones. Depending on the application class, we scale the inflection point with factor s_τ:

$$\tau_{\text{infl}} = s_\tau \cdot \tau_{\text{exp}} \tag{6.8}$$

A further degree of freedom is the steepness of $U'(\tau)$ parameterized by s. This is defined by choosing a value u'_{exp} at the expected duration. Evaluating Equation (6.6) at τ_{exp}, inserting Equation (6.8) and solving for s yields

$$s = \frac{1}{\tau_{\text{exp}}(1 - s_\tau)} \cdot \ln\left(\frac{1}{u'_{\text{exp}}} - 1\right). \tag{6.9}$$

Because the utility only slightly increases when the transaction finishes earlier than expected, u'_{exp} is chosen close to one.

It is assumed that users abort their transmissions when they take too long. Therefore, a threshold τ_{drop} is defined, which scales with the expected duration and has the minimum $\tau_{\text{drop,min}}$. This results in

$$\tau_{\text{drop}} = \max(3s_\tau \cdot \tau_{\text{exp}}, \tau_{\text{drop,min}}). \tag{6.10}$$

The scaling by $3s_\tau$ means that $U'(\tau)$ has practically dropped to zero. A completed transaction is accounted for with a minimum utility u_{min}, whereas a dropped transaction has a utility of 0. Adding this utility offset and normalizing to the interval [0, 1], we get the utility $U(\tau)$ for a transaction with transmission duration τ as follows:

$$U(\tau) = \begin{cases} u_{\text{min}} + \frac{1-u_{\text{min}}}{U'(0)} \cdot U'(\tau) & \text{for } \tau < \tau_{\text{drop}} \\ 0 & \text{for } \tau \geq \tau_{\text{drop}} \end{cases} \tag{6.11}$$

Figure 6.7 illustrates the utility function and shows the defining parameters. Table 6.5 gives the parameters for the HTTP and FTP traffic classes. We parameterize s_τ according to the MOS for different waiting durations in [NUN10]. The authors studied the satisfaction of users when they have to wait for different applications. We approximately map MOS 4 of satisfied users to $U'(\tau_{\text{exp}})$ and MOS 2 of dissatisfied users to $U'(\tau_{\text{infl}})$ and then determine the relation between both durations (τ_{exp} and τ_{infl}). Using the application classes “Web Site” for HTTP and “Download” for FTP, we get the values in Table 6.5. The minimum expected and dropping durations comply with the thresholds in [Nie10]. The parameters r_{exp}, u'_{exp} and u_{min} are reasonable estimations from everyday usage experience.

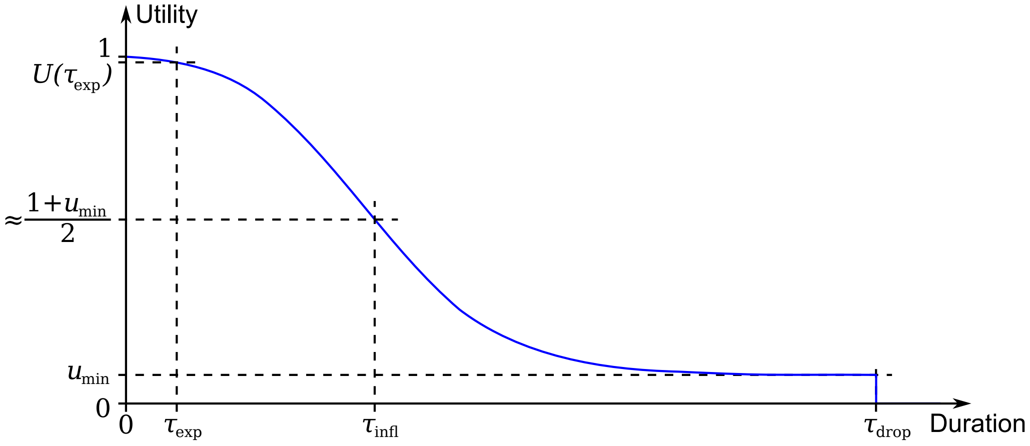


Figure 6.7: Parameters of the s-shaped utility function.

Table 6.5: Parameterization of the utility functions.

Parameter	Symbol	HTTP	FTP
Expected data rate	r_{exp}	3 Mbit/s	1.5 Mbit/s
$U'(\tau)$ with expected bandwidth	u'_{exp}	0.95	
Turning point factor	s_τ	5.13	5.75
Minimum expected duration	$\tau_{\text{exp,min}}$	0.1 s	
Minimum drop duration	$\tau_{\text{drop,min}}$	10 s	
Minimum utility of finished transaction	u_{min}	0.1	

6.2.3.2 *Buffered Video Streaming*

Similarly to interactive traffic, a MOS model from literature is used to define the utility functions for buffered video streaming. In [CSH13], the authors extract the findings from several field studies of perceived video QoE to define a general MOS model for YouTube streaming traffic. The resulting model is illustrated in Figure 6.8, which is taken from [CSH13, Fig. 5]. The parameter λ is the fraction of accumulated stalling time in relation to the total time. This makes the model suitable for video transactions of different durations. Depending on λ, different parameters are chosen for the exponential functions in Figure 6.8. The selected exponential function gives the MOS depending on the number of stalling events. When there are more than 6 stalling events, a video always has the worst MOS [CSH13].

The MOS is mapped to utility values as follows. MOS = 5 is the best possible score and therefore translates to a utility of one. At the other end of the scale, the minimum possible MOS (MOS_{min}) is mapped to u_{min}. With this, we get:

$$U_{\text{video}} = u_{\text{min}} + (\text{MOS} - \text{MOS}_{\text{min}}) \cdot \frac{1 - u_{\text{min}}}{5 - \text{MOS}_{\text{min}}} \tag{6.12}$$

In addition to [CSH13], a maximum stalling duration of a single stalling event $\tau_{\text{stall, max}} = 15\,\text{s}$[20] is defined, after which a user aborts the video and the utility is zero. Furthermore, an initial

[20] We choose $\tau_{\text{stall,max}} > \tau_{\text{drop,min}}$ to reflect the usually longer duration of videos compared to interactive transactions.

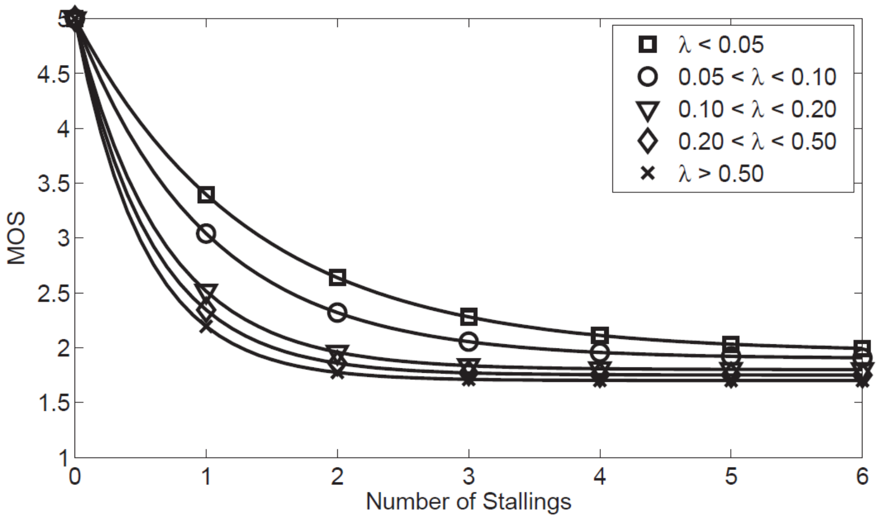


Figure 6.8: QoE of video transactions; MOS model from [CSH13].

expected buffering duration of less than $\tau_{\text{buf,exp}} = 3\,\text{s}$ before the video starts playing is not attributed to the total stalling duration. Again, these values represent an estimation from everyday usage experience.

7 Performance Evaluation

In this chapter, the benefits of the proposed algorithms will be evaluated by simulation with respect to the model and metrics discussed in Chapter 6. We first study the general system performance in Section 7.1 for a calibration of the model and the metrics in use. This study extensively investigates the relation between the behavior of the original schedulers and the observed outcome. Also, we look at the influence of boundary conditions like user mobility and traffic scenario. Then, in Section 7.2, the enhanced *Shortest-First* scheduler *SF* with length exponent is considered. We discuss the impact of the new parameter and which control opportunity it provides as well as the boundary conditions which influence the preferred parameterization. Section 7.3 focuses on video streaming. It shows how the proposed buffer-aware *SF* scheduler improves video QoE while still maintaining a good experience for other traffic classes compared to the reference schedulers. The influence of the additional parameters to integrate video streaming is evaluated as well as the control opportunities these parameters provide for network operators. Section 7.4 then discusses, how an optimized configuration can lead to the overall desired behavior and concludes this chapter.

7.1 Evaluation of the General System Behavior

Before studying the influence of the proposed algorithms, we first evaluate the general system behavior of the reference and *Shortest-First* schedulers from literature. Because the network performance strongly depends on the boundary conditions like channel, traffic and user models, this is done to put the metrics into relation. This allows us to discuss more general findings in the following.

In this section, we first look at cell capacity, i. e. how much cell throughput the different schedulers achieve. Then, we investigate the mobility influence, user experience and transaction durations for interactive traffic, and finally the sensitivity to the traffic model by changing the object size distribution.

7.1.1 Determining the Cell Capacity

In contrast to fixed line links, the rate of the wireless link between UE and eNB varies over time. Consequently, the capacity of a cell not only fluctuates, but it also depends on the scheduler assigning resources and on the variable traffic. In the following, the capacity behavior of the simulation model is investigated and the dependence on the mentioned influencing factors is shown.

Here, we use the base-line scenario applying the standardized web model with 80% FTP and 20% HTTP traffic volume. Due to the different object size distributions, this translates to a relation in the number of transactions of 10% FTP and 90% HTTP transactions. Such a large disparity is also found in measurements. For example, in the measurements of [HQG+13] the mismatch between the majority of flows and their fraction of the traffic volume is even larger. We will investigate the influence of the traffic model in Section 7.1.4.

The maximum theoretical capacity of the cell is

$$\frac{50\,\text{resource units} \cdot 1495\,\text{bits}}{1\,\text{ms}} = 74.75\,\text{Mbit/s}. \tag{7.1}$$

This means that all resources transport the maximum number of bits, i. e. the SINR for the respective resource is $\geq$25 dB. In practice, this situation occurs rarely, because the channel is usually worse and UEs with a good channel often do not require all resources to complete their transmissions. However, the scheduler can achieve a certain SINR gain by giving resources preferably to UEs with a relatively good channel compared to the others or compared to their own history. This is called *opportunistic scheduling* (see Section 3.4). Furthermore, a UE can be served preferentially on those sub-carriers with relatively good SINR (called *frequency-selective scheduling*).

Figure 7.1 shows the cell throughput depending on the offered traffic rate for the schedulers

- *Round Robin* (RR)
- *Max C/I* (MaxCI)
- *Proportional Fair* (PF)
- *Shortest First* (SF, original version)
- *Shortest Remaining First* (SRF, original version)
- *Traffic Aided Opportunistic Scheduler 2* (TAOS2).

Furthermore, it illustrates the maximum cell capacity and the horizontal dashed lines show the full-buffer throughput of the different schedulers. There is no full-buffer performance for the *Shortest-First* schedulers, because they do not work reasonably without object size definitions.

We can observe in Figure 7.1 that *Max C/I* almost reaches the upper bound of cell capacity for full-buffer traffic. This is because we have 50 receiver antennas and it is very likely that one of them has an SINR around 25 dB. However, with realistic traffic, *Max C/I* throughput is very far from this bound. In this case, UEs with good channel conditions finish their transactions fast and therefore require relatively few resources. Consequently, the remaining resources go to UEs with lower SINR. We verify this behavior below, when discussing the SINR distribution.

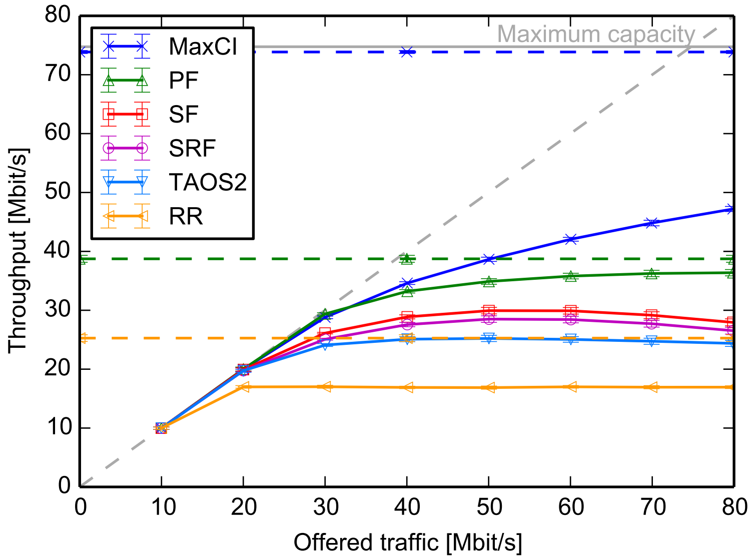


Figure 7.1: Cell throughput depending on the offered traffic rate for different schedulers.

The difference between the bisecting line and the cell throughput in Figure 7.1 corresponds to the rate of dropped traffic R_{drop} according to Equation (6.3)[1]. For low load, all transactions get finished and the cell throughput is identical to the offered traffic. As soon as the offered traffic exceeds the sustainable cell capacity of a certain scheduler, the throughput deviates from the bisecting line and eventually saturates. Theoretically, for unlimited offered traffic, cell throughput under *Max C/I* would reach the full-buffer bound.

The throughput of *RR* represents scheduling without channel awareness. With realistic traffic, throughput is worse than with a full-buffer assumption due to the different modeling approach. When modeling full-buffer traffic, each UE has a single transaction transmitting an unlimited amount of data. For realistic traffic, transactions accumulate in UEs with bad channel, as explained below. *RR* therefore saturates at $R_{fin} \approx 17$ Mbit/s.

PF is the only reference scheduler achieving nearly the full-buffer performance at reasonable load levels (see Figure 7.1). The accumulation effect of transactions in UEs with bad channel exists but is less pronounced than for *RR*. This is due to the channel-awareness and fairness properties. In average, *PF* assigns the same amount of resources in a certain time interval to all transactions[2]. However, it prefers UEs with a relatively good instantaneous channel state and thus exploits temporal fluctuations and multi-user diversity.

Finally, the size-based schedulers *SF*, *SRF* and *TAOS 2* exhibit a smaller opportunistic gain compared to *Max C/I* and *PF*, but are much better than *RR* in this respect. An interesting behavior is the slight decline in cell throughput at higher offered traffic. The higher the traffic level, the more active transactions exist from which the scheduler can select. Thus, the degree of channel awareness decreases when there are many tiny objects, where scaling the cost with the CQI has not much influence.

[1] It is not identical to R_{drop}, because transactions may get dropped after some data has been transmitted.

[2] This time interval is determined by the time constant of the moving average depending on the forgetting factor β, see Table 6.4.

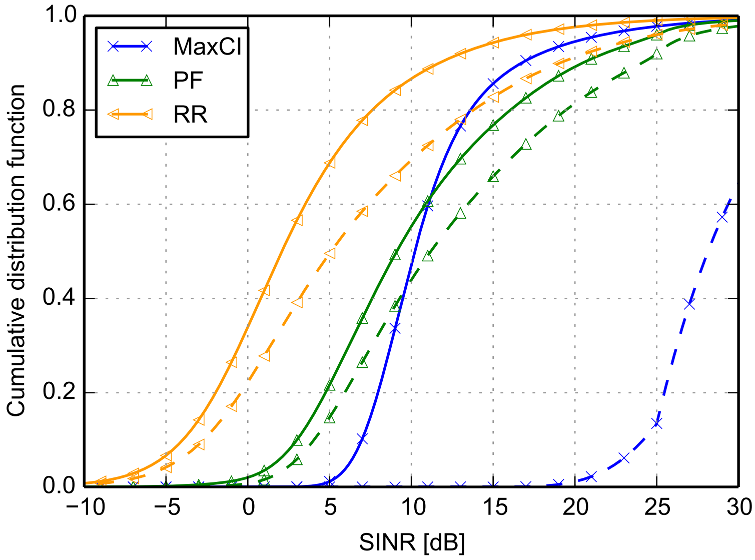


Figure 7.2: SINR distribution of the resources allocated by different schedulers. Dashed lines show full-buffer behavior, straight lines represent an offered traffic of $R_{\mathrm{arr}} = 40\,\mathrm{Mbit/s}$.

Figure 7.2 shows the SINR distribution of the transmitted resources for the reference schedulers. The dashed lines refer to the full-buffer case, where all UEs wish to transmit all the time, whereas the straight lines correspond to an offered traffic rate of $R_{\mathrm{arr}} = 40\,\mathrm{Mbit/s}$. In our studies, we will use this load level of $R_{\mathrm{arr}} = 40\,\mathrm{Mbit/s}$ as reference operation point. At this level, the cell is slightly overloaded for all schedulers, but still a good user experience can be reached. For our evaluation, load levels below the cell capacity are less interesting, because all requirements could be simply reached by service differentiation. Therefore, the scheduling algorithm would have less impact on the system performance. On the other side, higher load is less relevant, because the performance is bad for all schedulers and network operators would likely avoid this situation by building up their infrastructure.

For the full-buffer case shown in Figure 7.2, *Max C/I* allocates $\approx 85\%$ of resources to channels with SINR > 25 dB and most of the remaining resources to channels with SINR > 20 dB. With realistic traffic, the offered traffic rate is distributed evenly among all UEs. Therefore, the UEs with good channel conditions are served quickly and those with relatively bad SINR accumulate in the stationary state, because their transactions require longer to finish than for the others. Consequently, *Max C/I* allocates many resources at lower SINR and the CDF moves to the left. The SINR distribution of *RR* represents scheduling without channel awareness. This means that the resulting SINR distribution for the full-buffer case corresponds to the geometry of the cellular layout including fast fading. Comparing Figure 6.2 and Figure 7.2 shows that the lower quantiles have significantly lower SINR due to fast fading. *PF*, as shown in Figure 7.2, achieves a superior SINR compared to *RR* over the whole range. This confirms the exploitation of temporal channel fluctuations and multi-user diversity. At the observed traffic load of $R_{\mathrm{arr}} = 40\,\mathrm{Mbit/s}$, it even performs better than *Max C/I* for the top 40% of resources, because *Max C/I* does not consider temporal channel fluctuations of the UEs.

As stated initially, Figure 7.1 and Figure 7.2 confirm that the capacity of a cellular network not only depends on the wireless technology and the available spectrum, but also to a large extent on resource allocation and traffic properties. In this thesis, we therefore are not interested in the absolute figures of cell throughput, but rather on the relative performance differences between the evaluated schedulers. To this end, *Max C/I* reaches the maximum opportunistic gain without exploiting temporal channel fluctuations. *PF* is state of the art in many cellular systems with respect to fairness and a reasonable opportunistic capacity gain. In the further studies, *RR* will not be considered. It only served as a base-line for capacity without any optimizations.

7.1.2 Influence of the Mobility Model

An important property in all studies is the temporal evolution of the users' channels. It interacts with the arrival process through the volume of buffered transmission requests for a certain UE. A UE with bad channel conditions may accumulate waiting transactions, because the arrival rate exceeds the channel capacity. However, when the user moves to a place with more favorable channel quality, this backlog can be transmitted. Both, the mobility and traffic processes thereby together determine the channel quality distribution seen by the scheduler.

The SINR fluctuations in time and frequency depend on the speed of the users. For path-loss and shadowing, the movement of the user causes a change of the attenuation. The speed together with the direction of movement relative to the serving base station[3] and the gradient of the shadowing at the position of the user define how fast this change is. Furthermore, the speed defines the Doppler shift f_D relevant for fast fading. The larger the speed and correspondingly the Doppler shift gets, the faster are the signal fluctuations from fast fading.

In Figure 7.3, the scheduler throughput at different speeds is compared. Here, we still use the assumption of an ideal CSI. As discussed in Section 6.1.3.2, this leads to optimistic results for larger speeds (especially at 30 km/h), because the channel changes so fast that it is difficult for the transmitter to adapt to the current channel conditions with outdated CSI.

As a general trend, Figure 7.3 shows that cell throughput increases with higher speed for all schedulers. This is especially visible for *Max C/I*, for which the capacity does not saturate for the evaluated load at 30 km/h. The reason for this is the increased diversity in the channel conditions that can be exploited by opportunistic schedulers. However, as discussed, the results for 30 km/h are very optimistic, because they only reflect the increased diversity and not the impact of outdated CSI.

In comparison to *Max C/I*, the maximum cell capacity of *PF* only slightly grows with an increasing mobility of the users. This is due to the fairness constraint, which limits the exploitation of multi-user diversity. In a TTI, *PF* gives a PRB to a user having a favorable channel relative to its average on this resource. Therefore, it will always allocate some resources with sub-optimal SINR. At higher speeds, this cell throughput boundary is already reached at less offered traffic.

In a similar fashion as *Max C/I*, the *Shortest-First* schedulers *SF* and *SRF* profit from increased channel diversity. This is intuitive, because, identically to *Max C/I*, the cost of a transaction

[3] For example the user being on a tangential or a radial trajectory to the base station.

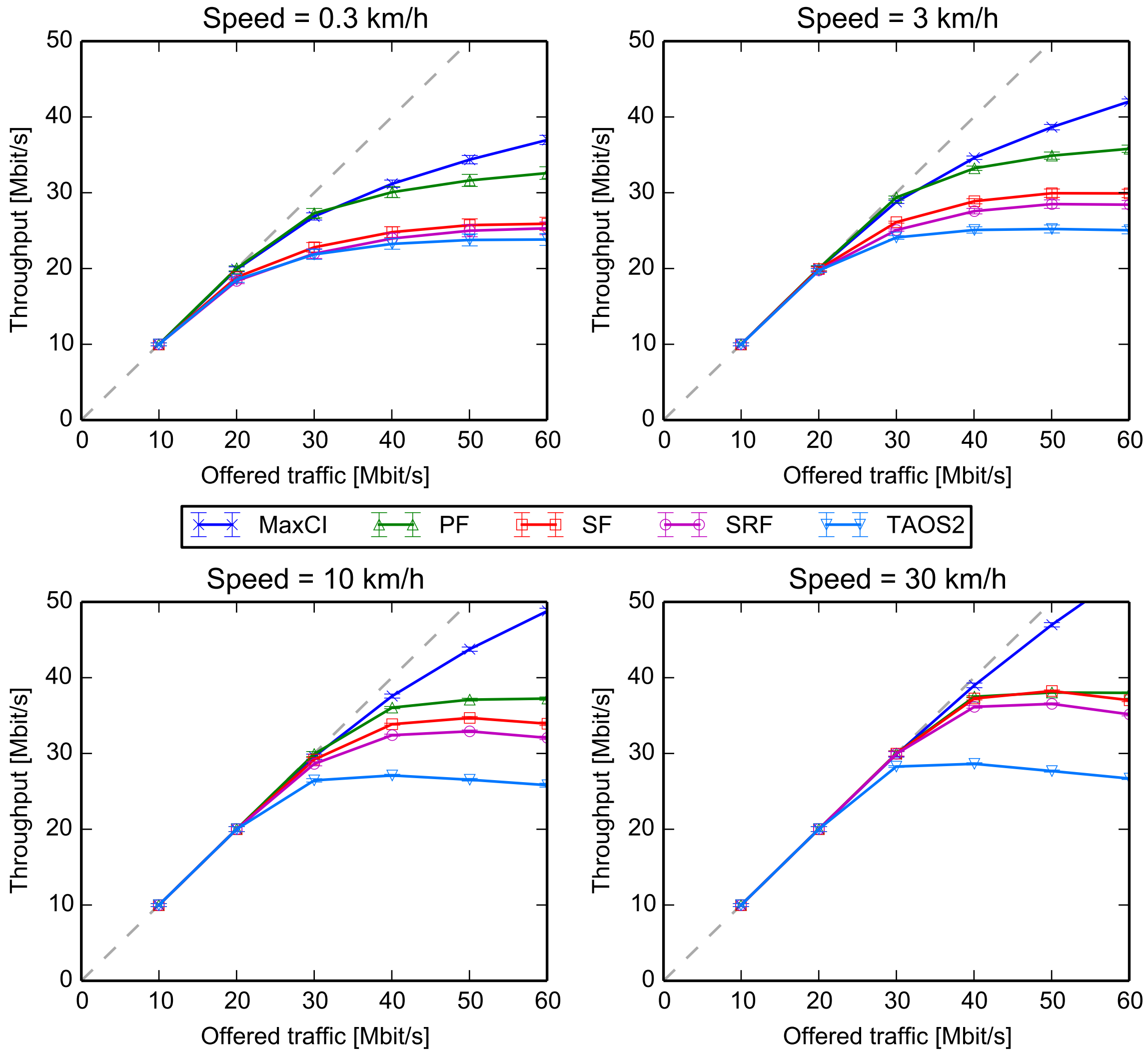


Figure 7.3: Cell throughput depending on the speed of the users.

scales with CSI and is not put into relation with the time average of the channel. *TAOS 2* performance is relatively insensitive to changes in the mobility model. In the ranking process, it puts object sizes in relation to the average channel quality and therefore is not able to fully exploit fluctuations faster than the time parameter in the moving average. Only the compensation term in Equation (3.9) considers the instantaneous CSI and has a limited influence. The throughput decline of all size-based schedulers at high loads (discussed in Section 7.1.1) is more pronounced at higher speeds. Especially at 30 km/h, it is clearly visible.

After having investigated the effects from varying the mobility parameters, we stick to our baseline configuration with a user speed of 3 km/h. From the observation it can be stated that the relative differences in throughput performance between the schedulers remain comparable, with a penalty for *SF* and *SRF* in a stationary situation and a penalty for *PF* and *TAOS 2* at higher speeds.

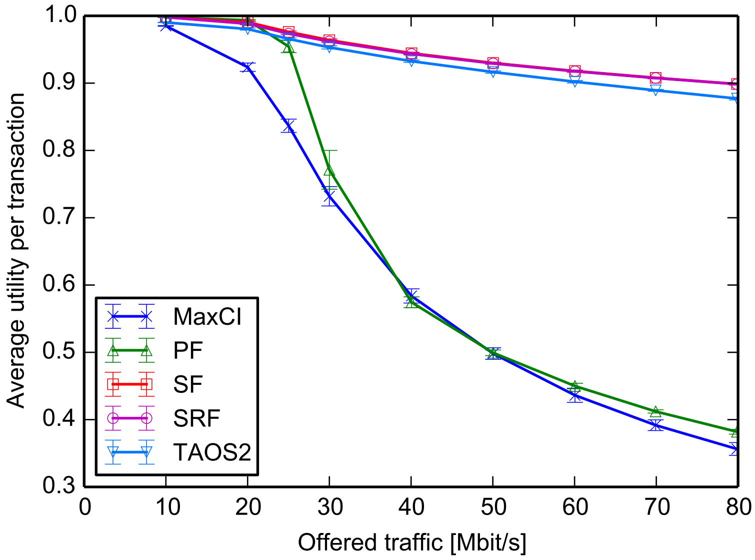


Figure 7.4: Average transaction utility depending on the offered traffic.

7.1.3 User Experience with Interactive Web Traffic

In this thesis, the main objective is to improve the user experience. We now investigate the basic behavior of the reference schedulers and the original *Shortest-First* schedulers. Therefore, we will again use the traffic scenario containing 80% FTP and 20% HTTP traffic volume, corresponding to $\approx 10\%$ and 90% of transactions, respectively. As utility is normalized, this means that the average utility is mostly defined by HTTP performance. We use the parameterization of utility functions from Section 6.2.3 to assess the user experience. This means that there is foreground traffic with interactive requirements and small object sizes and background traffic with relaxed latency requirements and comparably large object sizes.

Figure 7.4 shows the average utility performance for the considered schedulers depending on the offered traffic R_{arr}. At low loads, the cell capacity exceeds the offered traffic for all schedulers. In this situation, an incoming transaction often is the only transmission and gets served immediately. Thus, the scheduler has very limited influence. With increasing offered traffic, the average utility of the size-oblivious schedulers *Max C/I* and *PF* drops sharply. Despite its high cell throughput, the average utility under *Max C/I* quickly deteriorates with increasing load. As *Max C/I* does not consider transactions or users, many transactions having sub-optimal channel conditions are starved in the presence of others with favorable conditions. *PF* offers a very good utility up to $R_{\text{arr}} = 20\,$Mbit/s, but also performs badly at higher loads due to its fairness constraint. When there are more transactions than can be served, the scheduling weight of a transaction becomes smaller as soon as it gets resources. This leads to an interleaved service between active transactions and results in long durations and, consequently, in a bad utility for all of them. Figure 7.1 shows that at $R_{\text{arr}} = 30\,$Mbit/s we only have a very slight overload situation, where almost all the offered traffic gets served. Still, the average utility drops to $U \approx 0.77$ because all transactions have to wait longer[4].

[4] This behavior is also known from the original processor sharing, where the mean response time of jobs is infinite in constant overload conditions [BHB01] (we avoid infinite durations by dropping late transactions).

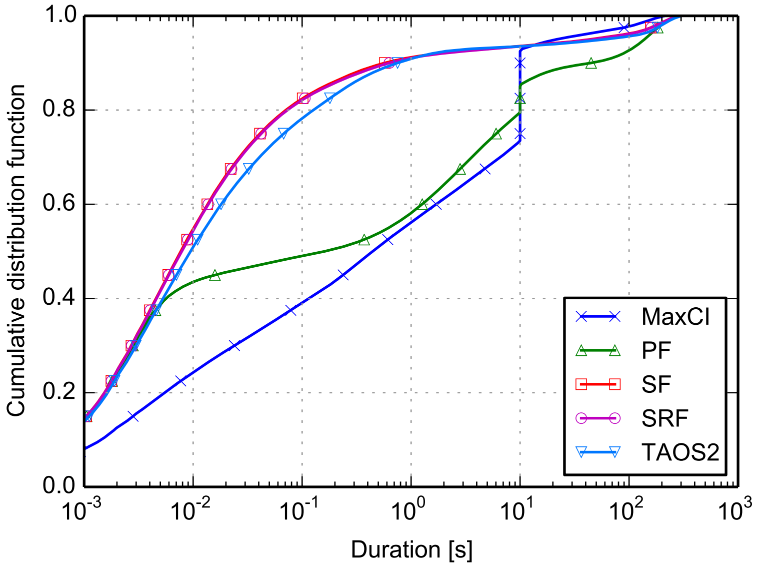


Figure 7.5: Distribution of transaction durations at $R_{\text{arr}} = 40$ Mbit/s.

In contrast, the *Shortest-First* schedulers *SF*, *SRF* and *TAOS 2* offer a good utility over the whole range. We can also observe the property of *graceful service degradation* in Figure 7.4. This means that only some transactions suffer from overload instead of all transactions. With the SRPT principle, short transactions do not notice the existence of larger ones as they are always prioritized to them. Especially in the currently observed traffic scenario, but also for more general heavy-tailed object size distributions, where a significant fraction of traffic volume is concentrated in few large transactions, this explains the excellent utility performance. Most transactions are very small and make up a small sum traffic rate compared to the cell capacity. These transactions finish fast no matter what the total offered rate is.

Comparing between *SF* and *TAOS 2* shows the superiority of the former. As stated in [HZS04], *TAOS 2* was optimized for the transient setting without new traffic arrivals. However, in the scenario here *SF* exhibits a more robust behavior, because it is less likely that it has to revise its decision and serve a different transaction than the one currently served. Interestingly, *SF* delivers practically the same utility as *SRF*, although it has a significantly better throughput (see Figure 7.1).

To get more insight into the utility behavior, we now look into the distribution of transaction durations. As stated before, we only consider the waiting time in the buffer of the base station. Figure 7.5 shows the duration CDF at $R_{\text{arr}} = 40$ Mbit/s over a logarithmic x-scale. The figure contains the durations of finished and dropped transactions. This explains the step in the curves of *Max C/I* and *PF* at $t = 10$ s, which corresponds to the minimum dropping duration of short transactions.

Except for *Max C/I*, the curves start at 15%. These are the transactions finishing within one TTI after they arrive at the base station. The *Shortest-First* schedulers serve transactions which are able to finish within one TTI immediately (unless there are more of them than would fit into the bandwidth). Also *PF* implicitly exhibits this behavior, because the moving average of the throughput is initialized close to zero for a new transaction. This gives a "jump start" to the respective transaction, which is then able to finish immediately if the instantaneous channel capacity is sufficient.

Figure 7.5 clearly explains the superior utility of the *Shortest-First* schedulers. They offer smaller transmission durations to almost all transactions. For example, around the 70%-ile the difference to the reference schedulers is around two orders of magnitude. Only the top 5% of the longest lasting transactions are better off when using *Max C/I*. In this region we find the longest (mostly FTP) transactions which profit from *Max C/I*'s superior cell throughput. However, this comes at the cost of much longer durations for all the shorter transactions[5], which is especially bad as this includes mainly the interactive services with strict latency requirements. Also the utility behavior of *PF* at $R_{\text{arr}} = 40$ Mbit/s is backed by Figure 7.5. Above the shortest 40% of transactions[6], *PF* requires much longer transmission times, comparable to *Max C/I* below the 10 s threshold. Above 10 s, i. e. mainly for the FTP transactions, *PF* has the longest transmission durations. Among the *Shortest-First* schedulers, *TAOS 2*, as previously observed, has the longest transaction durations whereas the difference between *SF* and *SRF* is merely visible.

To inspect the influence of the duration distribution on utility, Figure 7.6 shows the CDF of utilities at $R_{\text{arr}} = 40$ Mbit/s. The step at utility $U = 0$ represents the dropped transactions and the step at $U = 0.1$ corresponds to the minimum utility u_{min} for a finished transaction. All *Shortest-First* schedulers exhibit a similar utility behavior. A small number of transactions get dropped ($\approx 5\%$), while practically all others receive a utility of one. *Max C/I* and *PF* only offer this optimal utility to roughly 50% of transactions. Furthermore, for both schedulers, around 10% of transactions have utilities between the extreme values. With *Max C/I*, around 15% of transactions get u_{min}, whereas the remaining 25% get dropped. Under *PF*, these values are approximately 25% and 15%, respectively. To explain the reason for this utility distribution, we now look at the scheduling behavior with respect to different object sizes.

In Figure 7.7, we can see a histogram of transactions ordered by their sizes on a logarithmic x-scale. The bins are chosen such that they have an equal width on this logarithmic scale. The transaction numbers are normalized to a sum of 1. The gray bars in the background comprise all simulated transactions, whereas the curves for the individual schedulers represent the number of finished transactions. The shape of the histogram nicely illustrates the traffic composition in the current traffic scenario. Namely, the log-normal distributions of the HTTP transactions – the larger bell curve on the left – and the FTP transactions – the smaller one on the right – are clearly visible. Please note that 80% of the traffic volume is represented in the smaller FTP peak.

Figure 7.7 therefore allows to observe which transactions get dropped by the different schedulers. As observed before, *Max C/I* has a poor performance for small transactions. The number of finished transactions of small size is considerably less than for the other schedulers (this also explains the large step in the duration CDF in Figure 7.5). Many of them require longer than the user acceptance threshold and consequently get dropped. On the other side, *Max C/I* finishes most large transactions due to its good cell capacity.

[5] The CDF is not ordered according to transaction sizes, but, in average, longer transactions last longer. This is especially true for the higher quantiles where positive and negative channel fluctuations cancel each other out.

[6] The transactions profiting from the initialization bias, also see Appendix A.

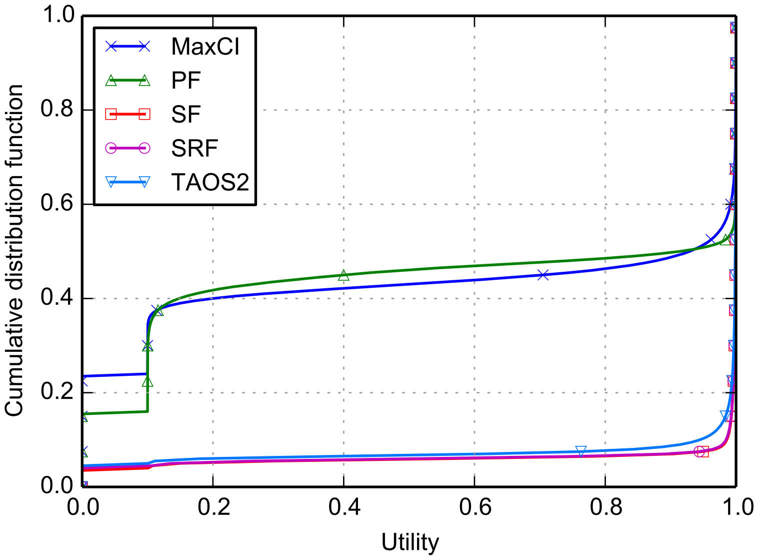


Figure 7.6: Distribution of transaction utilities at $R_{\text{arr}} = 40\,\text{Mbit/s}$.

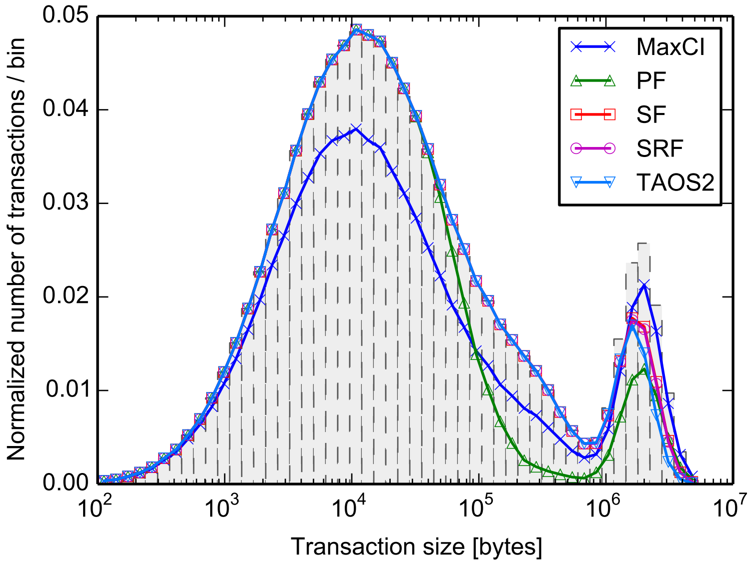
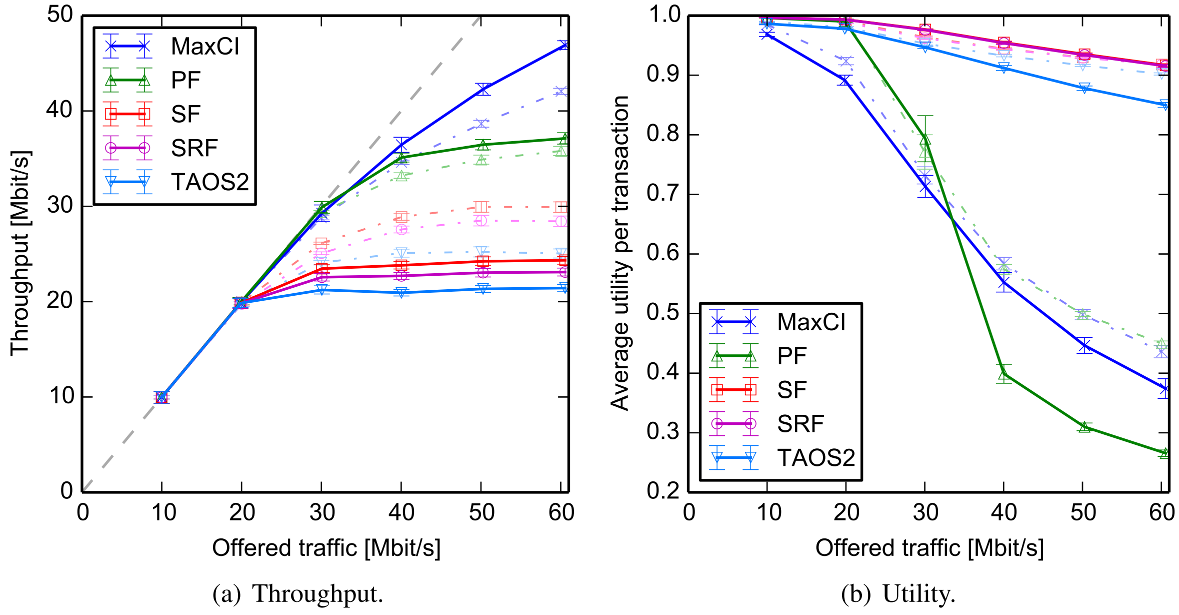


Figure 7.7: Histogram of finished transactions depending on their size ($R_{\text{arr}} = 40\,\text{Mbit/s}$).

Up to a size of about 50 kBytes, all schedulers but *Max C/I* transmit all arriving transactions within their maximum latency requirement. For medium transaction sizes, *PF* performs significantly worse than the *Shortest-First* schedulers, which all finish almost the same number of transactions. *TAOS 2* is slightly worse than the other two for the 5 largest bins and *PF* catches up for the 4 largest bins.

Having in mind the cell capacity of the different schedulers (Figure 7.1), it seems exceptional that the *Shortest-First* schedulers finish more transactions than the reference schedulers, although they achieve less throughput. This can be resolved by two explanations. First, the *Shortest-First* schedulers penalize the large transactions, which make up a large part of the traffic volume. Second, the transactions receiving service under the size-based schedulers are very



(a) Throughput.

(b) Utility.

Figure 7.8: Scheduler performance with a general log-normal object size distribution. Dashed lines show the performance with the standardized traffic model.

likely to get finished, whereas the transaction-oblivious reference schedulers allocate many resources to transactions which get dropped later on. This reduces the resource efficiency with respect to utility improvement. Especially *PF*, which is resource-fair in average, equally distributes the resources among all active transactions, of which many are aborted by their users subsequently.

7.1.4 Sensitivity to the Object Size Distribution

Because the traffic properties have a major influence on scheduling behavior and user experience, we now study the system with a different traffic model. We use the general traffic model from Section 6.1.4.2, which employs a single log-normal object size distribution. The transactions are generally larger than for the HTTP/FTP traffic mix and have a peak at 150 kBytes. Again, we investigate cell throughput and average transaction utility. Figure 7.8 shows the results. The lighter dashed lines represent the previous results with the standardized traffic model.

Comparing the cell throughput between both traffic models in Figure 7.8(a), we can observe that for *Max C/I* and *PF* throughput is larger with the general traffic model, while it is lower for the *Shortest-First* schedulers. This observation is backed by the SINR distribution (not drawn here), which shifts accordingly for the different schedulers. For *PF*, this can be explained with a better functioning of the moving average for larger traffic objects (less distortion by newly arriving small transactions). *Max C/I* profits from less burstiness caused by the larger transactions[7], which makes it more likely that a UE with a good channel has something to transmit. On the other hand, the larger variation in transaction sizes increases their weight on the

[7] Especially the maximum object size of 100 MBytes compared to 5 MBytes for the standardized traffic model brings the general traffic model closer to the behavior of the full-buffer model.

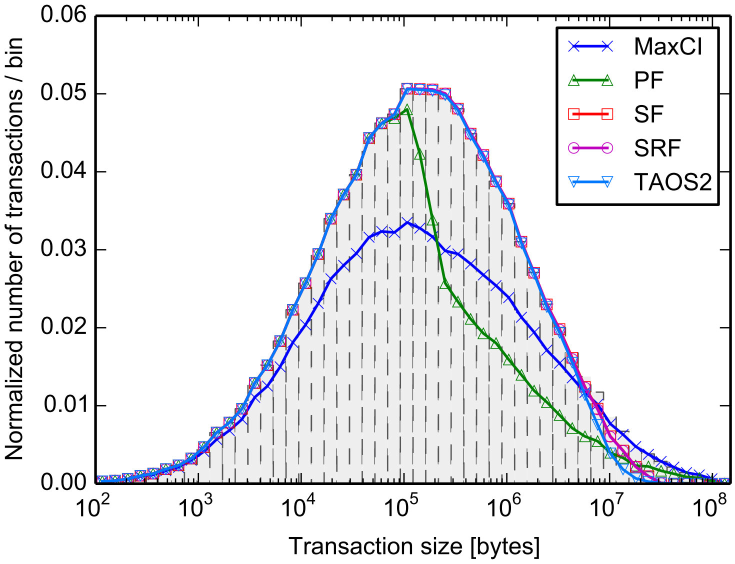


Figure 7.9: Histogram of finished transactions depending on their size for the general traffic model ($R_{\text{arr}} = 40\,\text{Mbit/s}$).

scheduling decision for *Shortest-First* schedulers. Consequently, the *Shortest-First* schedulers are less opportunistic and have lower throughput.

The average utility performance of the *Shortest-First* schedulers is very similar for both traffic models. Only *TAOS 2* shows a slight deterioration at high load with the general traffic model. Also *Max C/I* and *PF* perform worse in this case. Especially *PF* has a drawback for $R_{\text{arr}} > 30\,\text{Mbit/s}$. This is because the fraction of dropped transactions increases and the fraction of instantaneously transmitted transactions decreases. For example, at $R_{\text{arr}} = 40\,\text{Mbit/s}$, the fraction of dropped transactions raises from 15% to 26% and the fraction of instantaneously transmitted transactions drops from 15% to 5%.

Figure 7.9 again shows the completion of transactions over the size histogram. Compared to Figure 7.7, we now only have one Gaussian shaped peak originating from the single log-normal distribution. With respect to their treatment of transactions of different sizes, the schedulers behave as expected. The relative performance differences between the schedulers are comparable for the two traffic scenarios.

We can conclude that the general scheduling behavior, regarding the treatment of transactions of different sizes, is not affected by the traffic model. However, some differences exist. While the throughput gap between the *Shortest-First* and the reference schedulers increases, the latter are not able to translate this into a better utility in comparison to the *Shortest-First* schedulers. At high load the opposite is true with the reference schedulers achieving a worse average utility than in the standardized traffic model. The ordering between the schedulers with respect to the measured performance metrics remains unaffected besides the average utility of *PF* being worse than that of *Max C/I* at high load.

By evaluating two very different scenarios, one with a traffic mix of very small foreground and relatively larger background transactions and the other with object sizes ranging between less

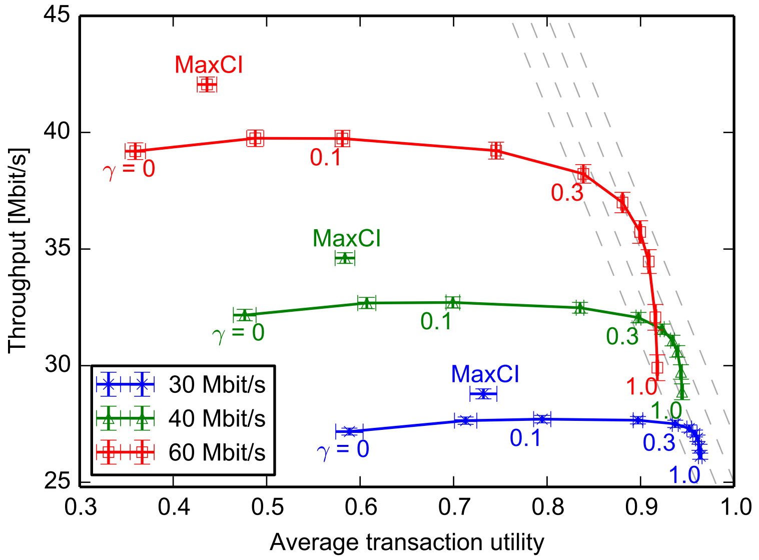


Figure 7.10: Trade-off between throughput and utility for different offered traffic levels with $\gamma \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1\}$.

than one kByte and 100 MBytes, it is demonstrated that the findings are valid for diverse traffic situations that can occur in the access network.

7.2 Evaluation of the Length Exponent

The main drawback of schedulers following the SRPT principle is their limited throughput performance. Although cell throughput is not crucial for small transactions which make up the largest fraction of transactions, it has a strong impact on larger transactions. Therefore, a reduced cell throughput, as with *SF* and *SRF*, is unfair towards large transactions which will exhibit a poor user experience (also see Chapter 4).

The length exponent controls how opportunistic the scheduler behaves with respect to the channel quality. It thus offers the possibility to tune the operating point with respect to trading cell throughput for average user experience. We study this impact in the following sections. *SRF* performance is always slightly worse than that of *SF*, but behaves almost identically apart from that. *SRF* will therefore be omitted in the following studies.

7.2.1 Trade-Off between Cell Throughput and User Experience

Figure 7.10 shows the impact of different length exponents γ on cell throughput and average utility for *SF* at different traffic levels. Furthermore, it presents the results of *Max C/I* for comparison.

As expected, $\gamma = 1$ offers the worst cell throughput. The significance of channel variations for the scheduling decision is minimal with this setting. By reducing the length exponent, a large throughput improvement is possible. It is noteworthy that even for small values of γ,

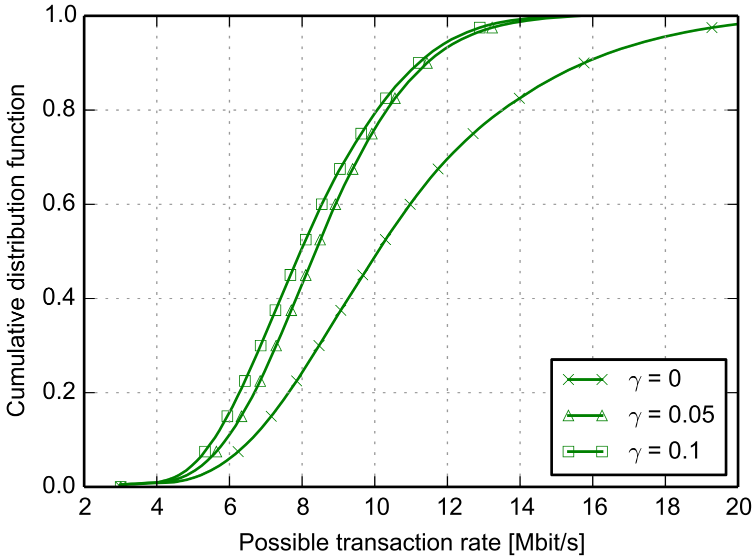


Figure 7.11: CDF of the possible rates of dropped transactions at $R_{\text{arr}} = 40$ Mbit/s.

SF does not reach the throughput of *Max C/I*. This is due to the different manner of resource allocation. The *Shortest-First* schedulers schedule transactions, whereas *Max C/I* schedules at the granularity of resource units (see Section 6.1.5). Consequently, *Max C/I* is able to exploit multi-user diversity better by being frequency-selective.

For $\gamma \to 0$, *SF* throughput slightly degrades again after the maximum at $\gamma \approx 0.1$. This behavior seems unintuitive, because one would think that always scheduling the transaction with maximum CQI maximizes cell throughput. However, it can be explained by the interrelation between resource allocation and user population in this scenario. Even a length exponent slightly larger than zero leads to a preference of small transactions. This means that among the transactions with optimal channel conditions, the scheduler chooses the smaller ones. With $\gamma = 0$, this preference does not emerge anymore. However, shorter transactions get dropped earlier (according to Equation (6.7) and Equation (6.10)). This means that for $\gamma \to 0$ more transactions with good channel conditions get dropped compared to a configuration with $\gamma > 0$. Consequently, the scheduler can only allocate resources to the remaining transactions with slightly worse channel conditions. Figure 7.11 backs this explanation. It shows the CDF of the possible rates of dropped transactions[8], i. e. at which rate a transaction could have been served, if it had received all available resources at $R_{\text{arr}} = 40$ Mbit/s. Clearly, for $\gamma = 0$, the dropped transactions could have been transmitted at a higher rate than for the other configurations.

With regard to utility, increasing γ improves the performance. The original *SF* scheduler (i. e. $\gamma = 1$) offers the best average transaction utility at all traffic levels. This setting puts most emphasis on object sizes and therefore minimizes transaction durations. Especially the small transactions, which make up the largest fraction, profit from this.

[8] Please note that, although the CDFs contain a different number of samples because the number of dropped transactions depends on γ, they are comparable with respect to the influence on the distribution of channel qualities. As the offered traffic rate is identical and cell throughput is similar for all configurations, the traffic volume contained in dropped transactions is also similar according to Equation (6.3).

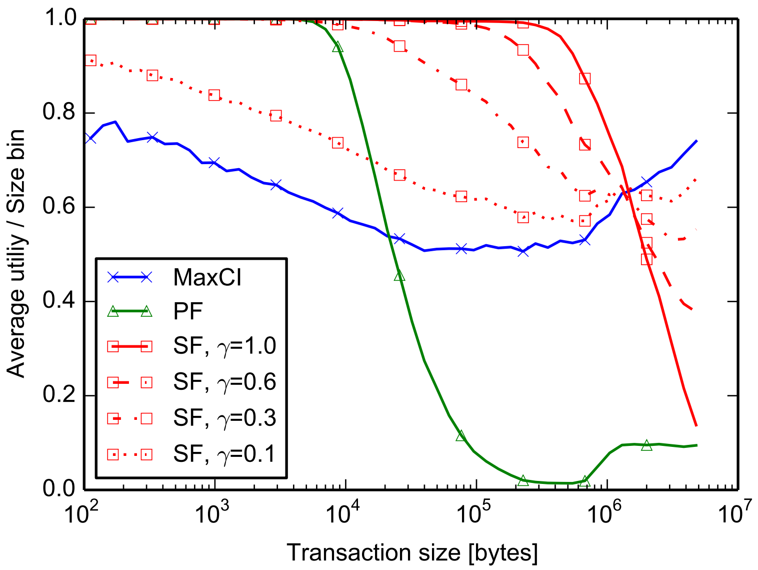


Figure 7.12: Utility histogram over transaction size depending on the choice of the length exponent; compared against the reference schedulers ($R_{\text{arr}} = 40$ Mbit/s).

Figure 7.10 illustrates the opportunity the length exponent offers to an operator. Setting $\gamma < 0.3$ is not reasonable in this scenario, as it brings a very limited throughput gain at large utility reductions. On the other hand, utility saturates at $\gamma \approx 0.6$. For larger values of γ, utility only slightly increases while cell throughput decreases considerably. For a desired relation between cell throughput and user experience, the optimal value of γ can be determined. As an example, the slope of the dashed grey lines in Figure 7.10 represents a relation of *10 Mbit/s : 0.1 utility*. For this relation, $\gamma \in [0.5, 0.6]$ is optimal for the considered traffic.

We now investigate the aspect of utility with respect to the transaction sizes in Figure 7.12. Taking the bins from Figure 7.7, Figure 7.12 shows the average utility of transactions within the respective bin. Again, the offered traffic is $R_{\text{arr}} = 40$ Mbit/s, which means that the cell is in overload. Consequently, no scheduling algorithm is able to satisfy all demands. The original *SF* clearly penalizes the largest transactions, but ideally serves the small ones (up to a size of approximately 200 kBytes).

In contrast, *Max C/I* offers a reduced utility to transactions of all sizes, but it has the best performance for the largest transactions. Under *PF*, very short transactions, up to a size of about 5 kBytes, profit from the initialization bias of the moving average[9]. Above that size, utility quickly degrades with transaction size, which reflects the long transmission durations. The effect of relaxed latency requirements for FTP transactions is visible in the slight utility improvement at a size of 1 MByte.

Decreasing the length exponent γ improves the utility of large transactions at the cost of the medium sized ones and therefore approaches *Max C/I* behavior step by step. This also explains why the average transaction utility decreases for larger values of γ, as most transactions are located in the smaller size bins.

[9] See Chapter A for a discussion of this.

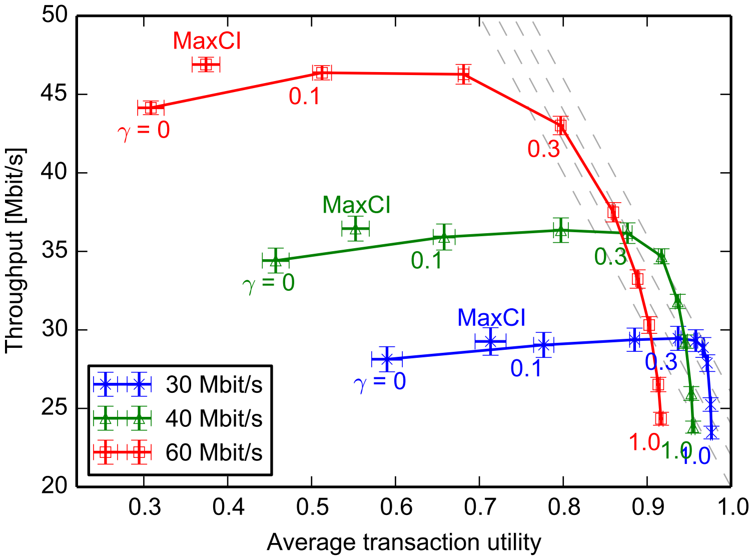


Figure 7.13: Trade-off between throughput and utility with the general traffic model and $\gamma \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.8, 1\}$.

Figure 7.12 nicely illustrates how large transactions are "starved" in favor of smaller ones, sometimes called "unfairness" (addressed in [BHB01]). The proposed length exponent directly controls this behavior and allows to tune the prioritization (or penalty) of large transactions. Notably, even with the original *SF* (i. e. $\gamma = 1$), the average utility in bins of all sizes is equal or better than for *PF*. The overall utility performance of *Max C/I*, which is best for large transactions, is poor for all other sizes, which represent the majority of traffic and users.

7.2.2 Traffic Model Influence on the Choice of the Length Exponent

To evaluate the influence of the traffic model on the choice of the length exponent, we now revisit the trade-off between cell throughput and average utility for the general traffic model in Figure 7.13. While the general shape is similar to Figure 7.10, the curves have moved significantly. This was expected due to the deviating throughput performance observed in Figure 7.8(a). As a consequence, also the optimal choice of γ changes.

Again, each of the dashed grey lines has a slope equal to the relation of *10 Mbit/s : 0.1 utility*. By trend, the favorable values of γ are a little smaller than in the standardized traffic scenario. We can read $\gamma \approx 0.4$ for $R_{\text{arr}} \in \{40, 60\}$ Mbit/s and $\gamma \approx 0.5$ for $R_{\text{arr}} = 30$ Mbit/s from Figure 7.13. This reflects the circumstance that for the generally larger transactions cell capacity is more important to achieve a good utility than with the standardized traffic model.

The decline in throughput for $\gamma \to 0$ is more pronounced than in Figure 7.10, which supports the finding that it is caused by an interaction between resource allocation and the channel quality distribution linked through the traffic model. The gap between the CDFs of possible rates (analogous to Figure 7.11, not drawn here) gets larger compared to the standard traffic scenario. Furthermore, in this scenario a throughput-optimal setting of γ leads to almost the same cell throughput of *SF* and *Max C/I*.

7.3 Scheduling including Video Streaming Traffic

In the previous sections, we evaluated traffic scenarios with interactive traffic only. Now, video streaming is included into the traffic mix and video QoE and the impact of video transactions on the interactive traffic will be investigated. As discussed in Section 2.3 and Section 6.1.4.3, the video transactions represent a buffered streaming service. Real-time streaming will not be considered. Buffered video streaming offers a limited degree of traffic elasticity. Although the data transmission of the video server is throttled, there is the data volume from the initial burst and the rate overhead, which can be scheduled flexibly in time; to a certain degree without affecting user experience. With respect to latency requirements, it is important to avoid buffer underruns which lead to an interruption of video playback and degrade the user's experience.

The advantages of a single cost/weight metric for the scheduling decision were discussed in Chapter 5. In this section, we evaluate the performance of *Max C/I*, *PF*, *PF-QoE* for reference and *SF* with and without buffer knowledge and the proposed parameters. We start these studies by comparing the application individual utilities for the reference schedulers and an unoptimized version of *SF*. Then, we investigate the configuration of IAC, the prioritization of video with the cost offset c_{min}, and the influence of the length exponent γ on the combined traffic mix. Finally, we revisit the comparison of the utility behavior between the reference schedulers and a fine-tuned configuration of *SF*.

7.3.1 General Utility Behavior with Video Streaming

As traffic mix, the standardized traffic model is used as a base-line with half of the volume of FTP traffic being replaced by video streaming. This results in 20% HTTP, 40% FTP and 40% video volume, respectively. The fraction of video lies between that of the Cisco forecast ([Cis14], Figure 2.1) and that of the "Real-time Entertainment" class from the Sandvine report ([San14], Figure 2.2). Thus, it represents a reasonable configuration for performance evaluation.

Figure 7.14 shows the utility behavior under the different schedulers depending on the offered traffic load. The sub-figures show the average utility of total traffic and separated by application class; HTTP, FTP, and video streaming, respectively. Compared to Figure 7.4, the total traffic utility in the top left figure is different for the reference schedulers, because of the changed traffic scenario. The relative utility advantage of *PF* over *Max C/I* is larger in this scenario. Apart from that, the utility level of *SF* is superior to the reference schedulers like in the previous studies.

We start discussing the application individual performance with *PF-QoE*. *PF-QoE* gives priority to video transactions which are in danger of a buffer underrun. As discussed in [NOALS$^+$13], we also observe a significant improvement of video QoE that this scheduler achieves compared to the original *PF*. However, *PF-QoE* accomplishes this at the cost of a severe degradation in utility for both other traffic classes. Already at a slight overload of $R_{\text{arr}} = 30\,\text{Mbit/s}$, *PF-QoE* offers the worst average utility for HTTP and FTP transactions. Nevertheless, video QoE quickly degrades with increasing load. This can be explained as follows. As soon as multiple videos are running low on buffered data, the relative prioritization advantage between the respective

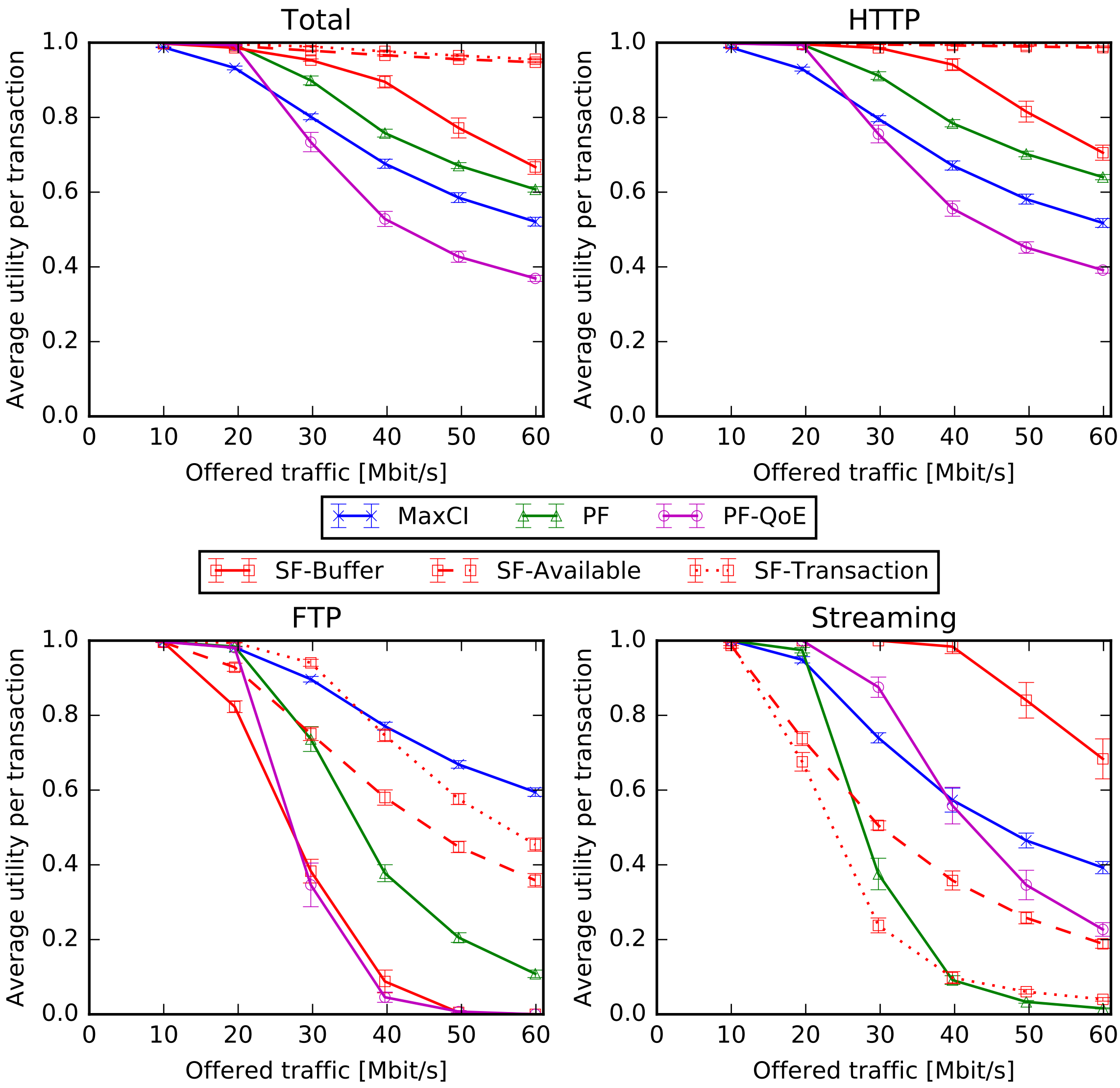


Figure 7.14: Application utilities for a traffic mix containing video streaming for the reference schedulers and non-optimized *SF*.

transactions is equalized. Therefore, resources are distributed among these transactions, which leads to *PF-QoE* not being able to prevent stalling events anymore at high load.

The reference schedulers *Max C/I* and *PF* do not treat video streaming transactions in a special way. This means that all transactions compete equally for radio resources. *Max C/I* offers a better QoE to video transactions, because it has a higher cell throughput and interleaves less between transactions than *PF*. Below, this behavior will be further elaborated when discussing the video properties.

Figure 7.14 shows three different configurations of *SF*. One with buffer knowledge (*SF-Buffer*), one using the data volume of a transaction that is available in the base station as equivalent object size (*SF-Available*) and a third configuration using the total size of the streaming object (*SF-Transaction*). The configurations without buffer knowledge, *SF-Available* and *SF-Transaction*, exhibit a bad QoE for video. However, FTP and especially HTTP traffic profit from this disadvantage and show a very favorable behavior over the load range. *SF-Transaction* assigns the largest size values to streaming transactions, which results in the worst video QoE. *SF-Available* is better for video QoE but still worse than all other schedulers except for *PF* at high load[10]. One can say that both variants are not suitable to handle streaming traffic. In contrast, *SF-Buffer* achieves the best average utility for streaming traffic. In this study, a naive parameterization for buffer information has been used. This means $\gamma = 1$ and a cost of video transactions that equals to the amount of data buffered at the client, apart from an offset of 1 Byte to avoid a multiplication with zero in case of an empty buffer[11]. This setting leads to a prioritization of videos with low buffer compared to all other transactions with similar channel quality. Consequently, HTTP and especially FTP utilities decrease significantly with increasing offered traffic.

Additionally, we can also interpret the different prioritization of traffic classes among the schedulers from Figure 7.14. *PF-QoE* provides a strong advantage for streaming transactions. Especially under heavy load, this comes close to a static prioritization of video over the interactive traffic. Comparing with the original *PF* shows that this significantly improves video QoE at the cost of a strong degradation for FTP and HTTP. On the other hand, *SF-Buffer* achieves a superior video streaming QoE that is unmatched by the other schedulers, while HTTP still performs better than for all reference schedulers. Only FTP suffers comparably to *PF-QoE*.

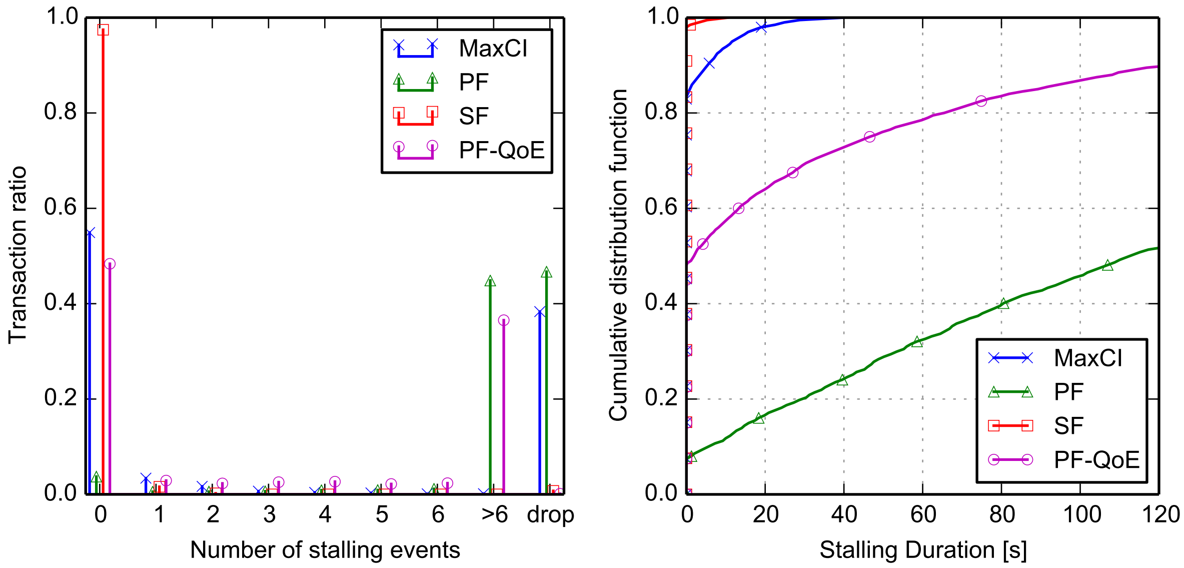
In Figure 7.15, we take a closer look at the key performance indicators for video QoE, namely the number of stalling events and the total stalling duration during video playback for an offered traffic rate of $R_{\text{arr}} = 40$ Mbit/s. To not overpopulate the graphs, we only consider *SF-Buffer* as *SF* in the following. Figure 7.15(a) gives the ratio of video transactions that suffered from a certain number of stallings or were dropped (comparable to an empirical probability mass function).

SF offers the best performance for video transactions. Almost all videos play uninterrupted at $R_{\text{arr}} = 40$ Mbit/s. There is a small number of videos which experience a single stalling event, while even less transactions are dropped. As stated before, this is due to the strong prioritization of video transactions in this configuration. Videos only contribute 40% of the traffic volume and are preferred to other transactions when they have a low buffer level, which avoids stalling in most of the cases. Under *Max C/I* a video transaction either is able to maintain a high buffer level when the respective UE has good channel conditions, or it does not get sufficient resources otherwise. The latter often leads to a prolonged stalling duration and a subsequent video abortion. Consequently, around 55% of video transactions play uninterrupted, whereas nearly 40% get dropped.

PF-QoE exhibits an interesting behavior. Although it achieves slightly less video transactions without stalling compared to *Max C/I*, practically no videos get dropped. However, this is paid by almost 40% of video transactions experiencing more than 6 stalling events and the bad performance for interactive traffic, as discussed before. Compared to the other schedulers, ordinary *PF* is the worst case for streaming videos. Only a few percent of videos are uninterrupted, while about one half of the transactions suffer more than 6 stalling events and the other half is aborted.

[10] Instead of preventing playback interruptions, *SF-Available* rather prefers videos which are not in danger of a buffer under-run, because they tend to have a smaller amount of buffered data at the base station.

[11] With a multiplication by zero, the channel quality would not influence the scheduling decision any more for stalling video transactions (see Section 5.3.1).



(a) Ratio of streaming transactions over the number of stalling events and dropped transactions.

(b) CDF of the stalling duration (finished videos).

Figure 7.15: Video performance under different schedulers at $R_{\text{arr}} = 40\,\text{Mbit/s}$.

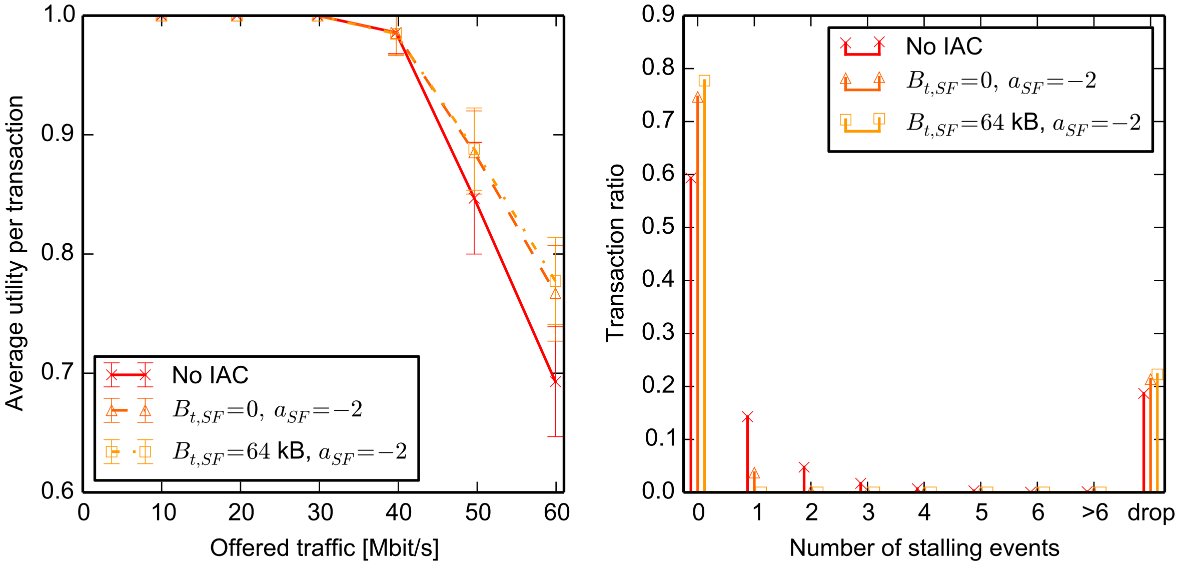
Figure 7.15(b) which shows the CDF of videos' total buffering duration (excluding the expected initial buffering duration of up to 3 s) backs these findings. Only finished videos are considered. *SF* and *Max C/I* practically avoid stalling for these transactions, approximately 95% and 80% of them have 0 s stalling duration, respectively. On the other hand, the stalling durations of *PF-QoE* and *PF* are significant, often several minutes per video are reached.

7.3.2 Configuring Implicit Admission Control

We now evaluate the proposed mechanism for Implicit Admission Control (IAC), see Section 5.3.2. The drawback for video playback with the unoptimized configuration of *SF* in the previous section is that many video transactions are impaired at high load. The scheduler is not able to distinguish between videos suffering mainly from bad channel conditions and others which could run uninterrupted if they got a little more bandwidth.

In Figure 7.16, we can observe the influence of IAC. Figure 7.16(a) shows the evolution of utility over load and Figure 7.16(b) contains the stalling events at high overload ($R_{\text{arr}} = 60\,\text{Mbit/s}$). Utility strongly decreases with additional traffic for the original setting *"No IAC"*. Already setting $a_{SF} = -2$ without target buffer ($b_{t,SF} = 0$) improves the situation[12]. It means that videos where the buffer ran empty and no new bursts arrived get a penalty with increasing waiting time. This leads to the abortion of the transactions with the worst channel quality compared to the rate of the video codec. The freed resources are then available to improve the QoE of the re-

[12] Unfortunately, due to the large variance of transactions' utilities, the confidence intervals are large compared to the relative utility differences between the configurations. However, the simulation software generates identical traffic for all cases, which means that we can determine an improvement for this traffic sample.



(a) Video utility over offered traffic.

(b) Ratio of uninterrupted, stalled and dropped video transactions at $R_{\text{arr}} = 60$ Mbit/s.

Figure 7.16: Influence of the parameterization of IAC for video transactions under *SF*.

maining videos. Setting the target buffer to $b_{t,SF} = 64$ kBytes[13] further improves the situation. Then, new transactions and those in danger of a buffer underrun are penalized additionally. The penalty means that the respective transaction requires a channel quality relatively better than that of the competing video transactions.

Figure 7.16(b) further details the improvement by IAC. The fraction of uninterrupted videos can be increased by approximately 20 percentage points in this scenario, while the fraction of dropped transactions barely increases. Stalling events for finished videos are practically completely avoided with IAC and $b_{t,SF} > 0$. Different settings for the slope a_{SF} and the target buffer $b_{t,SF}$ were also evaluated, but had only minor impact and shall not be discussed further.

To summarize, IAC practically completely eliminates stalling at a slightly increased dropping ratio. Instead of serving many video transactions at a bad QoE, the purpose of admission control is to drop those transactions that would have a bad service quality anyway and restrict newly arriving transactions in order to deliver a good QoE to the remaining active transactions. The proposed IAC seamlessly integrates into the *SF* scheduler at practically no additional computation effort. It dynamically adapts to the load situation, as it is implemented through a relative cost scaling. This means that it is only effective during overload situations and as many videos as possible are served with the available resources.

7.3.3 Controlling Video Priority under Shortest-First

Video streaming accounts for a large fraction of the traffic volume. Therefore, it is important to control the division of bandwidth between video streaming and interactive traffic. An

[13] Using the YouTube burst size of 64 kBytes as target buffer is advantageous, because it means that a transaction that received a single burst will not be penalized unless the playout buffer drops below 64 kBytes.

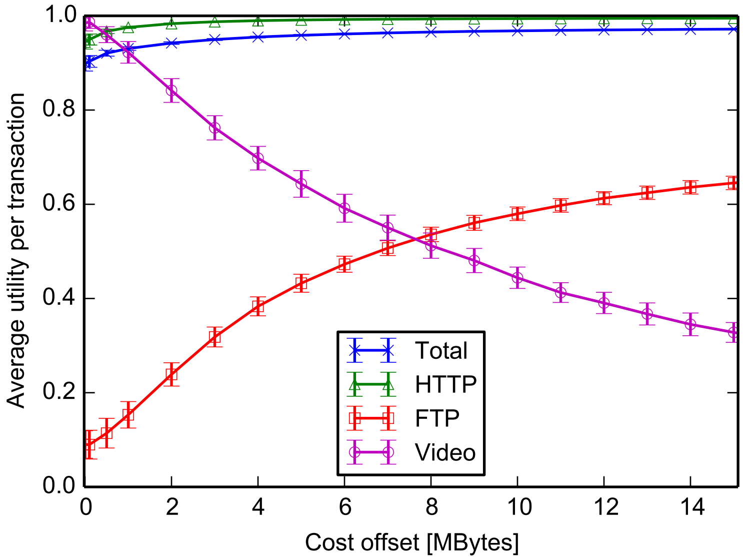


Figure 7.17: Relative prioritization of video transactions against the other application classes with the cost offset ($R_{\text{arr}} = 40\,\text{Mbit/s}$).

important question is how to "protect" both traffic classes from each other. For example, classical approaches employ service differentiation and define strict priorities or relative weight factors for the different service classes. However, we want to show that with the proposed integrated scheduling of video transactions with *SF*, it is possible to control video priority with a scheduler-internal parameter and completely relinquish additional service differentiation.

This parameter is the cost offset c_{min}, defining the minimum equivalent object size attributed to a streaming transaction. Figure 7.17 shows the average transaction utilities per application class depending on c_{min} at $R_{\text{arr}} = 40\,\text{Mbit/s}$ (including IAC with target buffer as defined in the previous section). The original setting from Section 7.3.1 is $c_{\text{min}} = 1\,\text{Byte}$. As stated before, this means that video transactions can suppress all other traffic, when they exhibit a low buffer level. Consequently, as the offered traffic is larger than the cell capacity and thereby the active videos are likely to have a low buffer level, the average utility of HTTP and FTP traffic is worst at this setting.

By increasing the cost offset, the relative priority of video traffic decreases. We can observe the resulting reduction in the average utility of video traffic in Figure 7.17. At small values of c_{min}, an increment of c_{min} is most effective to improve the utility of interactive traffic, because the relative priority change is large. When c_{min} is larger than most HTTP transactions their utility improvement for further increases of c_{min} saturates. The same is true for FTP traffic. The slope of the utility improvement of FTP is largest for $c_{\text{min}} \in (1\,\text{MByte}, 3\,\text{MBytes})$, which is the size of the majority of FTP transactions. For $c_{\text{min}} > 5\,\text{MBytes}$ (the maximum size of FTP objects), there still is an increase of FTP utility and decrease of video utility with smaller gradients. More and more, FTP transactions with relatively bad channel quality are preferred to videos with better channel quality when increasing c_{min}. The saturation of this effect is also visible in Figure 7.17.

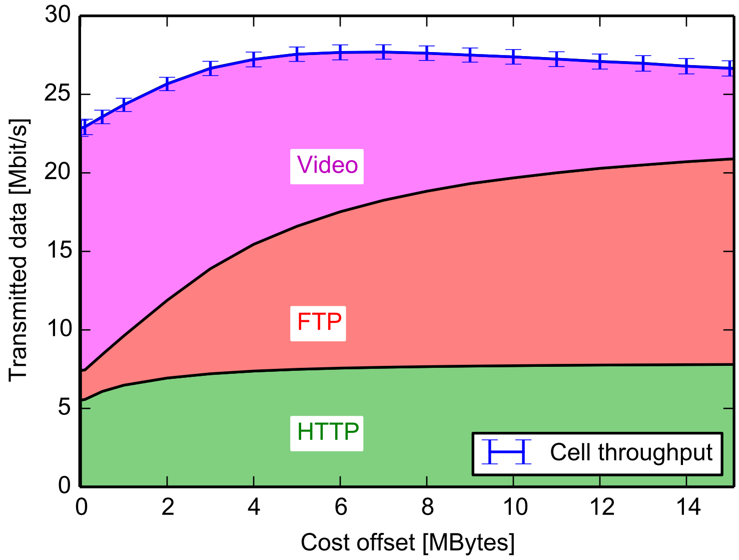


Figure 7.18: Influence of the cost offset on cell throughput and application-specific transmission volumes ($R_{\text{arr}} = 40\,\text{Mbit/s}$).

An interesting aspect of the cost offset is its influence on the transmitted data volume of the individual application classes, as well as the total cell throughput. This is illustrated in Figure 7.18. The figure shows a stacked area plot with the areas representing the different traffic classes. The accumulated data volume represents the total cell throughput.

The division of volume among the applications is straightforward. For a cost offset $c_{\text{min}} = 1\,\text{Byte}$, video traffic contributes two thirds to the transmitted volume, while the major part of the remaining resources goes to HTTP transactions. The latter are preferred to FTP transactions, because they are smaller in average. By increasing c_{min}, the relative transmitted volume of streaming traffic shrinks and initially moves mostly to HTTP (for $0 < c_{\text{min}} < 2\,\text{MBytes}$) and then to FTP, once the HTTP transactions are saturated.

The evolution of total cell throughput can be explained by two mechanisms. At small cost offsets ($c_{\text{min}} < 5\,\text{MBytes}$), cell throughput is low, because the video transactions exhibiting bad channel conditions tend to have a low buffer and therefore get many resources, which results in a bad spectral efficiency. Cell throughput then increases quite steeply with c_{min}, as the mentioned videos get replaced by interactive transactions with good channel conditions. With c_{min} between 6 and 7 MBytes, cell throughput is maximal, which also coincides with the break-even in transmitted volume of FTP and video streaming traffic. For larger values of c_{min}, throughput decreases again. This time, interactive transactions with unfavorable channel conditions more and more replace streaming transactions with good channel quality.

7.3.4 Choice of the Length Exponent for the Combined Traffic-Mix

We now come back to configuring the length exponent for the traffic-mix with video streaming. For this study, again, IAC from Section 7.3.2 is in use and $c_{\text{min}} \in \{1, 2, 6\}\,\text{MBytes}$. Figure 7.19 shows the trade-off between throughput and utility at a load of $R_{\text{arr}} = 40\,\text{Mbit/s}$. As discussed

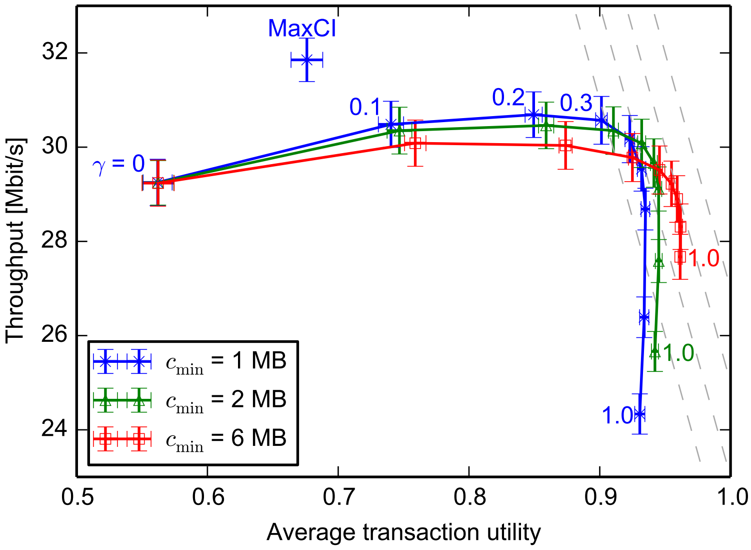
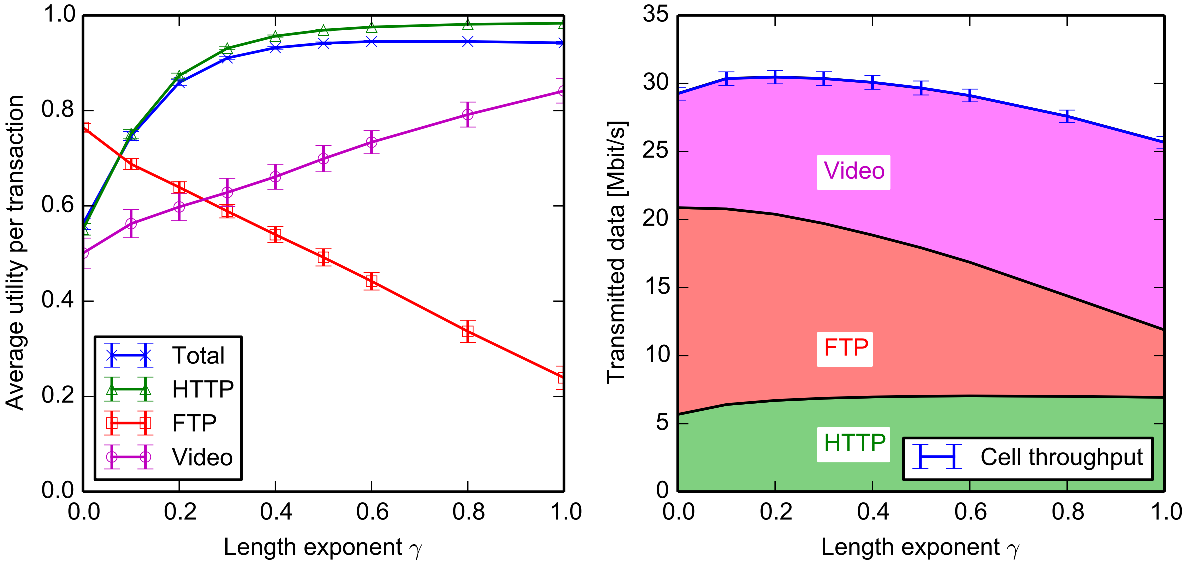


Figure 7.19: Throughput-utility trade-off for the combined traffic-mix depending on the choice of the length exponent γ and different values of c_{min}.

before, $c_{\text{min}} = 6\,$MBytes provides ideal cell throughput at $\gamma = 0.1$. However, the maximum achievable throughput is larger for smaller values of c_{min}. This is because penalizing video traffic too much reduces the freedom of the scheduler for small γ. In Figure 7.19, we can observe that the curves cross each other. That means that the configuration of c_{min} and γ should be jointly optimized according to the desired behavior in the access network. Of course, the curves join at $\gamma = 0$, because the object size has no influence in this case.

With the current traffic scenario, $\gamma = 1$ no longer provides the ideal overall utility under all circumstances. As we will discuss below, smaller values of γ are advantageous for FTP. Therefore, for $c_{\text{min}} \in \{1, 2\}$ MBytes, FTP utility increases so much for $\gamma < 1$ that the total average improves. Comparable to previous studies, the preferable configuration of the length exponent is $\gamma \in (0.5, 0.6)$. In this range, utility is optimal and cell throughput is much higher than for $\gamma = 1$. Larger values of γ considerably decrease utility.

Again, the influence of the length exponent on the treatment of the different application classes is of interest. Figure 7.20 contains the evolution of application utilities and the split-up of transmitted data volumes for the applications at $R_{\text{arr}} = 40\,$Mbit/s and $c_{\text{min}} = 2\,$MBytes. As we can see in Figure 7.20(a), FTP profits from decreasing the length exponent, while for HTTP and video streaming $\gamma = 1$ would be the optimal choice. For the relation between HTTP and FTP, we discussed this extensively in Section 7.2. Video streaming traffic loses average utility with decreasing γ, because the channel properties get more influence on the scheduling decision compared to the buffer level. This leads to worse utility (and subsequent abortions) for video transactions that experience unfavorable channel conditions. The behavior with respect to transmitted data volume in Figure 7.20(b) is as expected. Decreasing the length exponent generally improves throughput. Only for small values of $\gamma < 0.1$, this trend is reversed (see Section 7.2.1 for explanation). Most of the throughput gain goes to FTP transactions, which supports the explanation of the utility behavior.



(a) Total and per-application average utilities.

(b) Throughput and division of transmission volumes.

Figure 7.20: Influence of the length exponent γ at $R_{\text{arr}} = 40\,\text{Mbit/s}$ and $c_{\text{min}} = 2\,\text{MBytes}$

The total average utility in Figure 7.17 and Figure 7.20(a) mostly follows the evolution of HTTP utility, which contributes most transactions. The decline of video streaming with increasing cost offset c_{min} or decreasing length exponent γ barely influences the total utility, because it consists of comparably few transactions[14]. Therefore, the arithmetic mean of transaction utilities of all classes is not a good indicator of whether the network delivers the desired QoE across all applications. For the configuration of c_{min} and γ, a network operator could optimize the weighted sum of application-individual average utilities, for example.

In the end, both parameters – c_{min} and γ – essentially prioritize between FTP and video streaming transactions in the way described above. In a reasonable configuration, the small HTTP transactions should get preferred, as they usually have the strictest latency requirements and contribute fewest to the transmitted data volume. Compared to that, the other two compete for the major fraction of radio resources. For overload situations that only persist for a short time (e. g. some tens of seconds), it may be advantageous to ensure uninterrupted streaming services, as QoE degrades otherwise. On the other hand, the applications represented by our FTP model like software updates, data or e-mail synchronization are not impaired, when their transmissions are delayed for some time. However, if a cell faces constant overload, the trade-off between FTP and streaming traffic could be defined by the fraction of transactions that get finished for these traffic classes. Thanks to IAC, the average utility of streaming correlates to the fraction of "accepted" video transactions. The wanted target video utility depends on the network operator's preferences and business model. Therefore, this will not be studied further here.

[14] The mean size of video transactions is $\approx 16\,\text{MBytes}$. This is 8 times the average size of FTP objects. Because the share in traffic volume is the same, this means that video consists of 8 times less transactions than FTP.

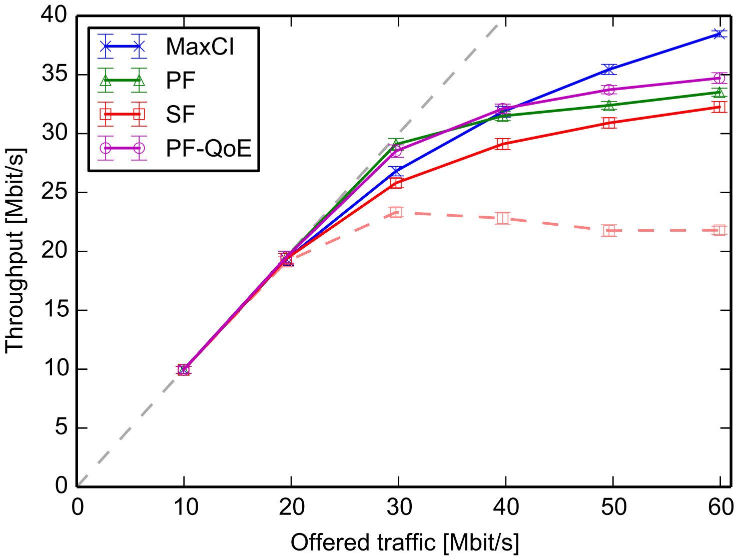


Figure 7.21: Cell throughput for a traffic mix containing video streaming for the reference schedulers and optimized *SF*.

7.4 Conclusions for the Overall System Performance

Now that the length exponent γ and the parameters to include video streaming were introduced and evaluated in the previous sections, we revisit cell throughput and average utility depending on the amount of offered traffic including video. To this end, an example configuration of *SF* with $\gamma = 0.6$, $c_{\text{min}} = 2\,$MBytes, $b_{t,SF} = 64\,$kBytes, and $a_{SF} = -2$ was chosen. Figure 7.21 and Figure 7.22 show the results for throughput and utility, respectively. The figures contain *SF* and the reference schedulers, including *PF-QoE*. The dashed line represents an unoptimized version of *SF* with buffer knowledge, but with neither length exponent nor IAC.

As we can see in Figure 7.21, the cell throughput of *SF* becomes competitive compared to the reference schedulers, when the proposed parameters are configured appropriately. In contrast, the original *SF* (using client buffer size as equivalent object size) offers poor capacity. Interestingly, both *PF*-variants achieve a higher cell throughput than *Max C/I* at $R_{\text{arr}} = 30\,$Mbit/s for the current traffic mix[15]. The reason mainly is the behavior of streaming traffic. Under *Max C/I*, it is likely that a user aborts a video when having worse channel conditions compared to other active users. This negatively impacts multi-user diversity, because video objects require a considerable transmission volume, which no longer needs to be served when they are dropped[16]. At higher traffic rates, this drawback of *Max C/I* diminishes and *Max C/I*, as expected, achieves the largest throughput. The difference between the *PF* variants with respect to cell throughput performance is small.

Switching over to the utility behavior, Figure 7.22 shows the total and per-application average utilities depending on the offered traffic. As discussed in Section 7.3.1, we can observe that

[15] Also in the other traffic scenarios, *PF* had the best throughput value at $R_{\text{arr}} = 30\,$Mbit/s, but the difference to *Max C/I* was not significant (see Figure 7.1 and Figure 7.8(a)).

[16] This explanation is backed by the mean number of active UEs, which is 18 for *Max C/I* and 34 for *PF* at the discussed operation point.

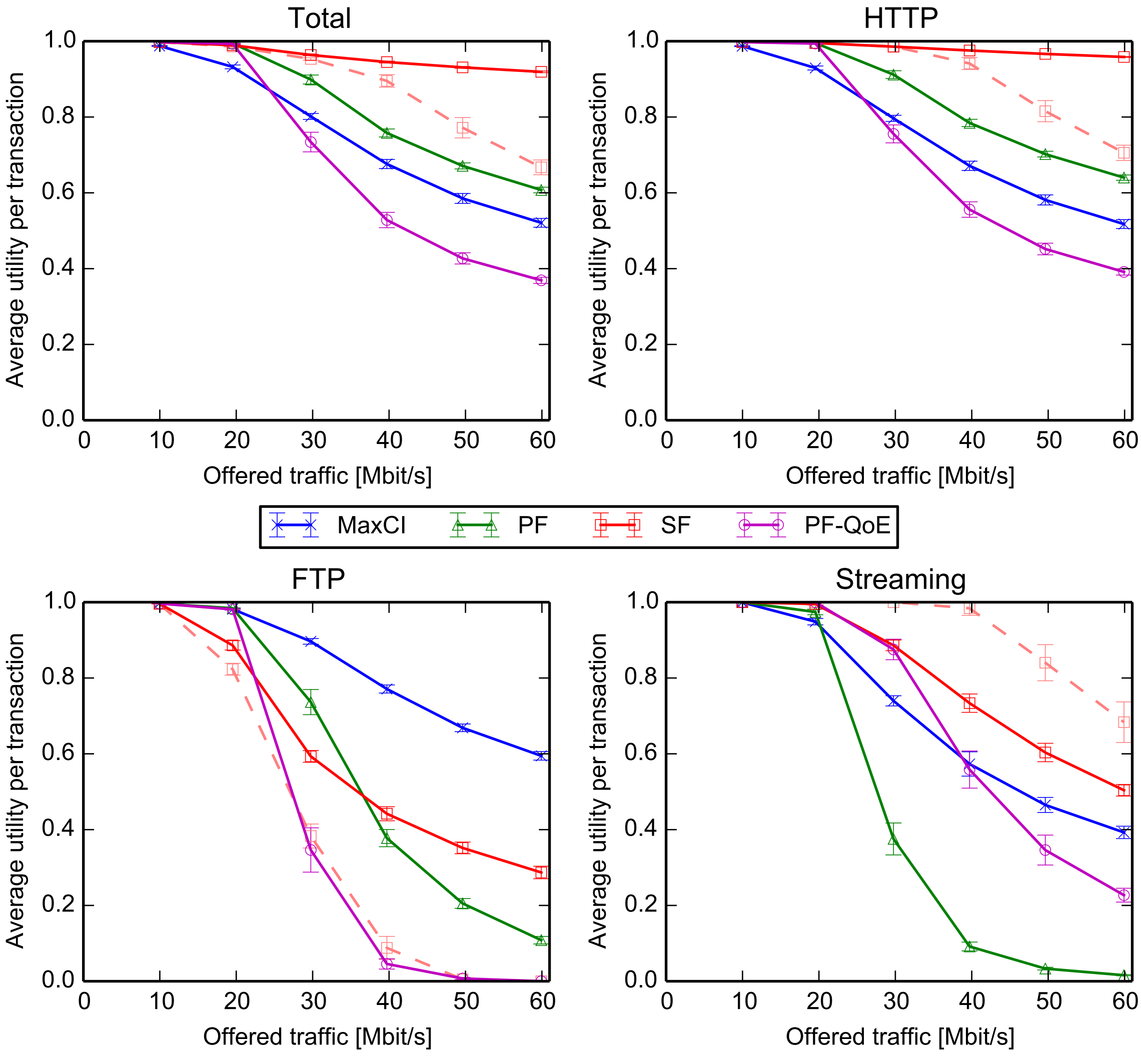


Figure 7.22: Application utilities for a traffic mix containing video streaming for the reference schedulers and optimized *SF*.

the total utility closely follows the behavior of the HTTP traffic class, which is due to the large fraction of transactions belonging to this class. The extended *SF* scheduler delivers optimal total, HTTP, and streaming utility performance among all schedulers. Due to the prioritization of video streaming, FTP gets a sub-optimal QoE, especially at low load. However, the deterioration of FTP utility with increasing offered traffic is smaller than for *PF*.

When we compare the example configuration of *SF* including the extensions proposed in this thesis with the base-line *SF* variant, the advantages become obvious. Instead of giving too much priority to video transactions, when the equivalent object size equals the client buffer size, or strongly penalizing video traffic in the case without buffer knowledge, the proposed solution offers the possibility to adjust application priorities to the operators' needs. Especially the graceful service degradation for time-critical interactive HTTP transactions can be obtained even under considerable overload. At the same time, we have a superior video QoE behavior and the desired property that videos either run uninterrupted or do not get resources at all (implemented by IAC).

We can summarize the parameters introduced in this thesis as follows:

- The length exponent γ trades the exploitation of multi-user diversity through opportunistic scheduling – i. e. cell capacity – for the minimization of transmission durations.
- The cost offset c_{min} controls up to which object size an interactive transaction will be treated preferentially compared to streaming transactions exhibiting similar channel conditions.
- The target buffer $b_{t,SF}$ determines the buffer size for which the scheduling cost of a streaming transaction gets minimal.
- The slope $a_{SF} < 0$, together with $b_{t,SF}$, defines the impact of IAC. Namely, the buffer level a newly arriving or stalling video transaction competes with. It therefore allows to control the preference of running videos with a low buffer level over paused videos.

In this chapter, the model assumptions were put into relation with respect to the considered performance metrics for the evaluated schedulers. We identified the desired properties of *Shortest-First* schedulers like short transaction durations and graceful service degradation as well as the drawbacks, which can be mainly attributed to the limited cell throughput. The improvement of these drawbacks by the proposed extensions and parameters were evaluated. Furthermore, we discussed the influence of different configurations. The sensitivity of the schedulers with respect to changing boundary conditions like traffic load, channel fluctuations and application mix were examined.

A nice property of the length exponent is that the preferable range showed to be stable at $\gamma \in [0.5, 0.6]$. This means that the system is near its optimal operation point for a static setting of γ, no matter which traffic mixture currently prevails. With respect to the configuration of c_{min}, it is obvious that it should allow short interactive transactions to overtake video streaming transactions. Beyond that, c_{min} controls the division of resources between large interactive transactions and video streaming transactions. Therefore, it should be usually set to several MBytes, according to the network operator's preferences. The configuration of IAC is straightforward and works well with the proposed setting of $b_{t,SF} = 64$ kBytes and $a_{SF} = -2$, which is derived from YouTube burst sizes. There is no necessity (at least for the investigated scenarios) to further tune these parameters, because the desired impact is already achieved.

8 Conclusions and Outlook

Cellular networks provide a pervasive network coverage that can be used for a large number of Internet services. However, the ever-growing traffic demands may exceed the available spectrum capacity. The consequence are high load situations which may strongly impair the QoE of the users. Especially prolonged transmission durations in a loaded cell can significantly deteriorate the user experience. The specific deterioration depends on the requirements of the respective service, e. g. interactive web surfing suffers from long transmission durations, while the QoE of video services is mainly impaired by playback interruptions.

Such QoE impairments can be greatly mitigated by schedulers exhibiting a *graceful service degradation*. This means that the QoE of most of the traffic remains favorable, while high load only affects a relatively small number of transactions. Without building out the network infrastructure, which is very costly, the general service quality delivered by the mobile network in times of high traffic load is much better than for conventional schedulers like *Proportional Fair*. SRPT is an algorithm with this property, which delivers optimal durations for job scheduling tasks [SM66]. Several approaches are known from literature, which apply this algorithm to cellular networks (see Section 3.6.3). However, so far these schedulers have two major drawbacks that limit their applicability in an actual cellular network:

- Compared to wide-spread opportunistic schedulers like *Proportional Fair* or *Max C/I*, the size-based variants offer significantly lower cell throughput. This reduces the sum rate a base station can transmit and therefore may lead to more frequent overload situations.

- Size-based schedulers so far are not able to serve streaming services satisfactorily. As such services usually transmit rather large transactions, the SRPT-algorithm assigns a low priority to them. Frequent buffering events and long loading times would be the consequence. Buffered video streaming contributes a major part to today's Internet traffic. This means that a size-based scheduler cannot serve as a stand-alone solution handling all traffic in a cell.

This thesis contributes solutions for both limitations by enhancing the algorithm of existing size-based schedulers like *Shortest First* and *Shortest Remaining First*. These enhancements provide new parameters for the trade-off between cell capacity and transaction durations as well as the prioritization of buffered video streaming traffic.

To this end, the following approach was taken in this thesis. Chapter 2 provided a background on traffic characteristics and user experience. Especially the burstiness of realistic traffic as well as its composition have a great influence on resource allocation and the users' QoE. In general, the object size distribution is assumed to have a heavy-tailed nature which means that the largest volume belongs to few transactions while most transactions are very small. Furthermore, the concept of transactions was introduced and influence factors on user experience were discussed. Different services have different requirements to provide a good QoE, which is translated to individual transaction utility functions in this thesis. In Chapter 3, the focus was shifted to cellular networks. Namely, the basics of mobile communications, resource allocation and different scheduling concepts known from literature were presented, with a focus on LTE technology. Important are the properties of the radio link which are exploited by opportunistic schedulers to increase the capacity of a cell. Apart from that, application awareness, especially with respect to the knowledge of transactions and their sizes, can be used together with the SRPT-principle to improve QoE.

Chapter 4 presented the first major contribution of this thesis, which is the flexible combination and parameterization of size-based and opportunistic scheduling. This is motivated by the tradeoff between cell capacity on the one side and QoE improvement from reduced transmission durations on the other side. The length exponent γ was introduced and its effects were explained. With γ, it is possible to adjust the relative importance of channel quality and transaction sizes for the scheduling decision. An advantage is that – compared to the original schedulers *Max C/I* and *Shortest First* – large gains in throughput or utility can be achieved at only a small loss in the respective other metric.

In Chapter 5, the second main topic of this thesis was addressed. This is the handling of buffered video streaming traffic by size-based schedulers. As video streaming traffic is a major part of the Internet traffic (see Chapter 2), a new scheduler has to be able to provide a good QoE for this service. Without modifications, large video transactions would obtain a low scheduling priority leading to poor QoE. This thesis proposes the usage of client buffer size information which can be integrated seamlessly into size-based schedulers. Furthermore, the concept of IAC is introduced, which allows to protect ongoing videos from overload. These contributions significantly improve video quality by reducing the duration and number of playback interruptions compared to existing wireless schedulers.

The methodical foundation for performance evaluation was laid in Chapter 6. The chapter introduces the simulation model, traffic scenarios and the considered performance metrics. Apart from the wireless channel properties, system-level aspects like interfering base stations and user distribution and mobility have an influence on the scheduler performance and are therefore a part of the model. For the evaluation of the central metrics in this thesis, namely cell throughput, transaction durations and QoE, it is important to assess realistic traffic situations. To this end, different scenarios with respect to the traffic characteristics were introduced. Finally, in Chapter 7 the performance of the developed schedulers was thoroughly investigated by simulation studies covering a range of different realistic boundary conditions in mobile networks. First, a study putting the performance of the investigated schedulers into relation for the standard scenario and with respect to the influence of speed and object sizes was presented. Then, the impact and benefits of the length exponent γ were evaluated. Especially the great improvement in utility that can be achieved compared to *Max C/I* at a relatively small reduction in cell capacity was demonstrated. Furthermore, Chapter 7 highlighted the inclusion of video streaming into size-based schedulers. The approach proposed in this thesis exhibits the desired properties. A major improvement of video QoE over the whole range of cell load conditions achieved by a reduction in the number of playback interruptions. Thereby, the parameters showed to effectively control the priority and volume, video traffic obtains from the scheduler. It is thus possible to handle interactive web traffic and buffered video streaming traffic without further service differentiation mechanisms.

Concluding, it can be stated that the identified principal drawbacks of SRPT-based scheduling in cellular networks are solved by the contributions of this thesis. Assuming that an overload situation usually does not last very long, which is a consequence of traffic burstiness, it is reasonable to postpone large transactions predominantly belonging to background tasks and instead to prefer short interactive and video streaming transactions. In this way, the majority of users will not even notice the overload conditions, as videos play uninterrupted and web browsing interactions quickly show a result. This was the reason to choose SRPT schedulers as the main subject of this thesis and to empower them to serve as a stand-alone scheduling solution in cellular networks.

The methods applied in this thesis comprise algorithm design, modeling and performance evaluation. Existing resource allocation algorithms were enhanced by functional parameters which allow to adjust their behavior with respect to a desired property. Namely, the length exponent γ as well as the offset c_{min} and the v-shape of the equivalent object size function for the integration of video streaming are important here.

The introduced length exponent γ allows to improve the throughput performance of the investigated size-based resource allocation algorithms. A large part of this improvement can be achieved without significantly sacrificing the desired property of short transaction durations. This is demonstrated with the example of the schedulers *Shortest First* and *Shortest Remaining First*, which were extended and evaluated in a simulation model of an LTE system.

With the proposed usage of client-side buffer information for buffered video streaming services, the developed schedulers are able to improve video QoE compared to conventional schedulers. This is achieved by defining an equivalent object size based on the buffer level that determines the scheduling cost for video transactions. Application-unaware schedulers are not able to recognize when there is too much video traffic in a cell. The consequence is that all videos show a bad QoE when the bandwidth is not sufficient to support their aggregated encoding rate. This behavior is improved by introducing IAC into the equivalent object size function in Section 5.3.2. Furthermore, by defining the parameters of the proposed function, the priority of video streaming relative to other traffic can be adjusted. The proposed solution therefore allows to relinquish further service differentiation. This means that all traffic is handled within the same resource allocation loop, which provides the maximum flexibility for exploiting channel and traffic diversity. Performance evaluation showed that this concept leads to superior QoE while still delivering a cell capacity comparable to the reference schedulers.

The developed algorithms exhibit a low computational complexity which is comparable to a conventional scheduler like *Proportional Fair*. Therefore, they can easily run in today's base station hardware. An important point is that the proposed mechanisms are not limited to the investigated OFDMA network like LTE but could also be adapted easily to other cellular standards like UMTS, which employs W-CDMA.

Modeling aspects include the system-level model for an LTE-like cellular network comprising network layout and channel, mobility, and link layer models. Furthermore, the development of realistic traffic scenarios is crucial to be able to measure application-dependent transmission properties that are important for the respective QoE requirements. The concept of transactions that was presented earlier in [PKWV12] lays the foundation for this traffic modeling.

For performance evaluation, this thesis contributes the definition of metrics that support an assessment of user experience for the mentioned realistic traffic model. Compared to often cited fairness properties, which are defined for full-buffer simulations, it was pointed out that for realistic traffic situations, besides spectral efficiency (determining cell throughput), the distribution of transaction durations is much more meaningful. To this end, utility functions were applied which allow to quantify and compare the QoE of different services. Finally, simulation studies basing on IKR SimLib and IKR RadioLib were conducted that demonstrate the performance of the investigated schedulers and the influence of a range of boundary conditions.

Beyond the scope of this work, there are some points that are of interest for the investigated subject. In the direction of further developing the proposed algorithms, adding frequency-selectivity for the handling of video streaming traffic could be interesting. In contrast to interactive transactions, where the transmission duration is essential, it could be advantageous to split a frame for the exploitation of multi-user diversity. Here the challenge is to identify the potential improvement and not to increase the complexity too much.

For implementing the investigated size-based schedulers, it is necessary to provide them with size information. Thereby, the relevant size is that of transactions at the application layer. While some approaches from literature which show that this is principally doable were presented in Section 2.4.1, the actual deployment of a framework providing the required information remains an open challenge. The same is true for the knowledge of the client-side buffer level in the scheduler. As introduced in Section 5.2.2, there are ways to make this possible, yet an implementation for the access network is to be done.

Another interesting question is the quantification of the influence of imprecise or incomplete size information. For example, in the case of encrypted traffic, it is not possible without application support to obtain size information or to differentiate between transactions within the same connection. In this case, an estimation from traffic characteristics could give some hints about transaction properties but no exact knowledge. As literature on the *Foreground-Background* scheduler has shown (see Section 3.6.1), a method employing very rough size estimates can already deliver a part of the desired functionality as well. Therefore, a study investigating the influence of reduced or inaccurate size information could be used to assess and possibly increase the robustness of the algorithm and would complement the studies presented here.

Going further into detail with respect to the simulation model, one could include realistic influences that have been neglected in the applied model for the sake of identifying the principal mode of action. Influential aspects could be imperfect CSI together with realistic transport format selection, block error and retransmission models (the expected impact has been discussed in Section 6.1.3.2). Furthermore, modern cellular techniques include MIMO transmission, interference coordination, and heterogeneous networks. While there should be no fundamental differences or obstacles for the proposed algorithms, some adaptations may be required. From the higher layers, especially the influence of transport layer congestion control could interact with MAC layer resource allocation and would therefore provide an interesting direction of research.

One way to investigate the ideas given in the outlook above could be the realization of the presented algorithms in a test-bed implementation of an LTE system. Traffic could be produced by traffic generators imitating real applications. Furthermore, the lower and higher layer effects that were not represented in the simulation model could be easily included. Challenges could arise in covering the system level aspects like having many users and interference from other base stations. For this, the deployment as an overlay network in an operative cellular network could be suitable.

A Influence of the Forgetting Factor

PF employs an exponential moving average for the scheduling decision. According to Equation (3.4), the scheduling weight of a transaction is the fraction of the instantaneous rate and the moving average. Also *PF-QoE*, as a derivative scheduler of *PF*, and *TAOS 2* use the moving average to normalize the current rate of a user. The difference for *TAOS 2* is that it is always updated with the current CQI instead of the received data rate in order to represent the channel conditions instead of the average throughput.

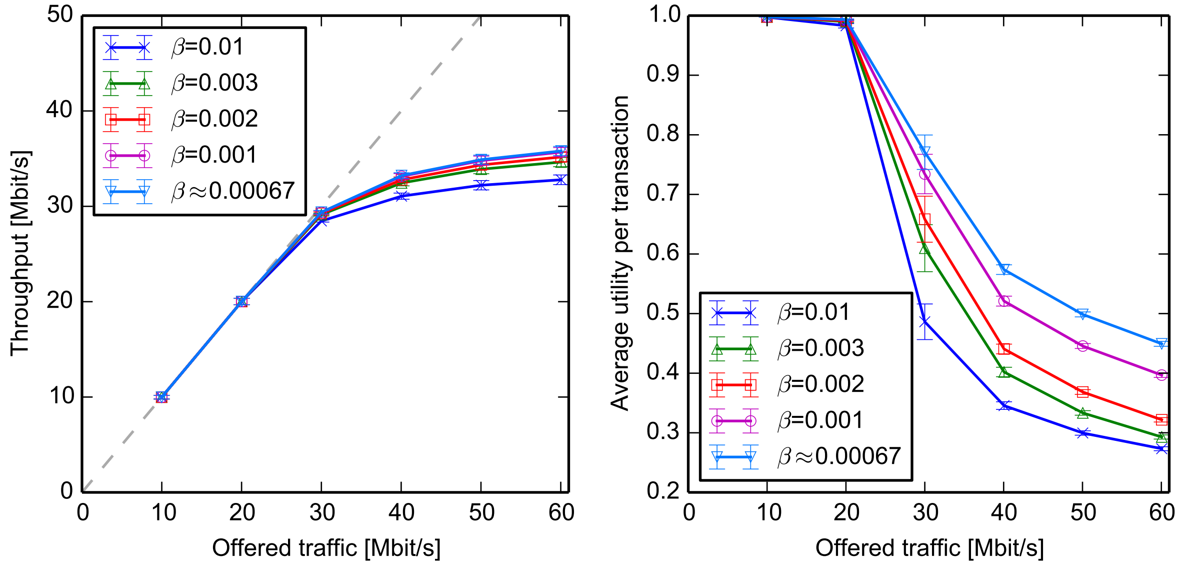
The exponential moving average $\overline{R}(t)$, given in Equation (3.5), is a first order infinite-impulse response filter (or autoregressive model) acting as a low-pass filter and thereby gives a smoothed representation of the transaction's past throughput. For convenience, both equations are printed here again:

$$w_{PF} = \frac{R(t)}{\overline{R}(t)} \tag{A.1}$$

$$\overline{R}(t+1) = \begin{cases} \beta \cdot R(t) + (1-\beta) \cdot \overline{R}(t) & \text{if scheduled} \\ (1-\beta) \cdot \overline{R}(t) & \text{else} \end{cases} \tag{A.2}$$

The forgetting factor β controls the time behavior of the moving average. A large value of β means a fast adaptation to the channel, whereas a small value leads to $\overline{R}(t)$ changing only slowly. As discussed in literature (e. g. [JPP00, VTL02, Kol03]), the effect of β is a trade-off between the maximum allowed latency and the cell throughput. When the moving average changes slowly, it converges to the average of the channel quality at this time-scale. The scheduler then is able to exploit fluctuations at shorter time-scales, because the moving average is stable and w_{PF} is largest when the channel is at its peak. On the downside, this may lead to long waiting times for a transactions when the UE moves to a place with worse channel conditions. A fast changing moving average limits the worst-case starvation time but also the recognition of peaks in CQI, because $\overline{R}(t)$ follows the channel rather quickly.

In our scenario, the traffic model also has a significant influence. We model bursty web traffic, which consists of many comparably small transactions. Some of them can be served within a single TTI. This burstiness, together with the properties of the *PF* scheduling algorithm, influences the cell capacity and average transaction utility. Figure A.1 contains both metrics depending on the offered traffic R_{arr} for the standardized traffic model (see Section 6.1.4.1)



(a) Throughput.

(b) Utility.

Figure A.1: *PF* performance for different settings of the forgetting factor β.

with 80% FTP and 20% HTTP traffic volume. This is the same traffic composition that was used in Chapter 7 for this traffic model.

In Figure A.1(a), we can indeed observe a greater throughput for smaller values of β (i. e. a more stable moving average). However, the throughput improvement is rather small for two reasons. First, the stable moving average is only reached for large transactions lasting long enough with respect to the chosen forgetting factor. Second, the initialization bias, which reduces the influence of the channel conditions on the scheduling of small transactions, gets larger for smaller forgetting factors. We consider each transaction as a single user. Consequently, we track an individual moving average per transaction and initialize $\overline{R}(t) = 10^{-15}$ for newly arriving transactions (according to [3GP04][1]). By this, transactions get prioritized initially as their CQI is much larger than $\overline{R}(t)$, which leads to a high scheduling weight w_{PF}.

Figure A.1(b) shows that $\beta \approx 0.00067$ leads to significantly better utility at high load than the smaller values. The larger time constant of the moving average (reciprocal of the forgetting factor β) means that a newly arriving transaction is served longer initially, until the moving average reaches the long-term average of the throughput of the respective UE. Implicitly, a longer time constant therefore means a behavior like the *Foreground-Background (FB)* scheduler discussed in Section 3.6.1 (plus channel-awareness). Under *FB*, the shortest active transaction is served until another transaction has received less service. This scheduler therefore is a blind heuristic approximating SRPT when the object size is unknown [SM02].

We can observe this behavior in Figure A.2 which gives the transactions' duration CDF at a load of $R_{\text{arr}} = 40\,\text{Mbit/s}$. The steep increase of the CDF for $\beta \leq 0.003$ for short durations ($\tau < 2\,\text{ms}$ up to $\tau < 5\,\text{ms}$) is due to the initialization bias. New transactions get the channel practically exclusively until their weight w_{PF} has approached the level of the other active transactions.

[1] Other sources from literature propose initialization to the current CQI and/or adaptive initialization schemes. However, as discussed in the following, this would have adverse effects on *PF*'s performance in our scenario.

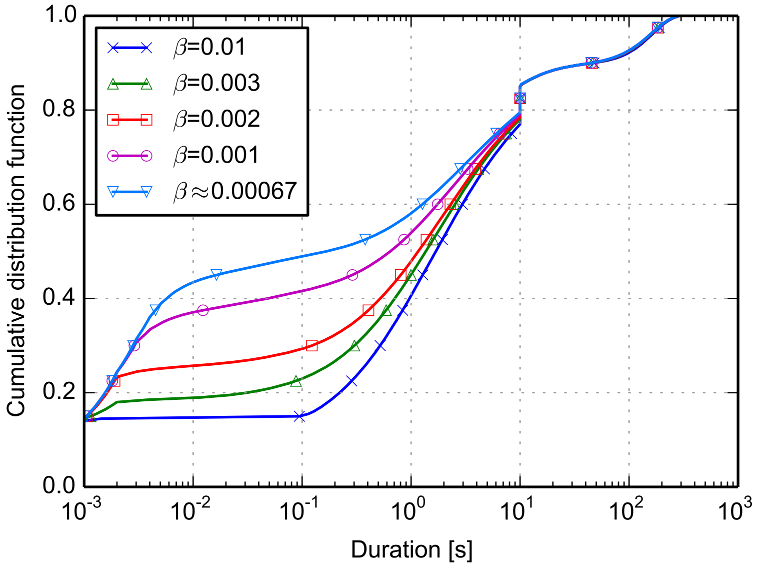


Figure A.2: Duration CDF for different settings of the forgetting factor β at $R_{arr} = 40$ Mbit/s.

For $\beta = 0.01$ this effect is not visible, as the variance of w_{PF} for existing transactions is larger and the initialization bias lasts shorter than for the other configurations. Only the step at $\tau = 1$ ms is identical, which represents those transactions that are able to finish within one TTI (approximately 15% of transactions).

Following the specifications in [3GP04], we set $\beta = 1/1500 \approx 0.0006667$. Furthermore, we apply the recommendation that the moving average shall be *"initialized to a small value greater than zero"*[3GP04]. This parameterization gives a good *PF* performance in our system and therefore provides a fair comparison to the size-based schedulers.

This preliminary study suggests that a forgetting factor larger than the one proposed in literature is beneficial for the utility and the duration of short objects. However, this would mean that we practically would implement an *FB* scheduler, which is not suitable as a reference case. Furthermore, the discussed issue of worst-case waiting time stands against larger values of β. We therefore stick to a parameterization specified by standardization and commonly used.

Bibliography

[3GP04] 3GPP2. cdma2000 Evaluation Methodology. (C.R1002-0), December 2004. V 1.0.

[3GP06] 3GPP. Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA). (TR 25.814), October 2006. V 7.1.0.

[3GP09a] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Physical Layer - General Description. (TS 36.201), March 2009. V 8.3.0.

[3GP09b] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation. (TS 36.211), December 2009. V 8.9.0.

[3GP10] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRAN); Overall description; Stage 2. (TS 36.300), April 2010. V 8.12.0.

[3GP13] 3GPP. Radio transmission and reception. (TS 45.005), December 2013. V 8.17.0.

[3GP14a] 3GPP. Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH). (TS 26.247), December 2014. V 10.9.0.

[3GP14b] 3GPP. User Equipment (UE) radio transmission and reception. (TS 36.101), December 2014. V 8.26.0.

[AAR12] E. Altman, K. Avrachenkov, and S. Ramanath. Multiscale fairness and its application to resource allocation in wireless networks. *Computer Communications*, 35(7):820–828, 2012.

[AKR$^+$01] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, P. Whiting, and R. Vijayakumar. Providing quality of service over a shared wireless link. *IEEE Communications Magazine*, 39(2):150–154, February 2001.

[Ale] Alexa top 500 global sites. http://www.alexa.com/topsites. Link verified on 2015-06-19.

[APLO11] S. Aalto, A. Penttinen, P. Lassila, and P. Osti. On the optimal trade-off between SRPT and opportunistic scheduling. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 185–196, San Jose, CA, USA, June 2011.

[AQS05] M. Andrews, L. Qian, and A. Stolyar. Optimal utility based multi-user throughput allocation subject to throughput constraints. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2415–2424, Miami, FL, USA, March 2005.

[ARMNO$^+$10] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, P. Mogensen, and J. M. Lopez-Soler. QoE oriented cross-layer design of a resource allocation algorithm in beyond 3G systems. *Computer Communications*, 33(5):571–582, 2010.

[ARMNOLS12] P. Ameigeiras, J. J. Ramos-Munoz, J. Navarro-Ortiz, and J. Lopez-Soler. Analysis and modelling of YouTube traffic. *Transactions on Emerging Telecommunications Technologies*, 23(4):360–377, 2012.

[BCM98] M. A. Bender, S. Chakrabarti, and S. Muthukrishnan. Flow and Stretch Metrics for Scheduling Continuous Job Streams. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 270–279, San Francisco, CA, USA, January 1998.

[BELSHE2] M. Belshe, R. Peon, and M.Thomson. Hypertext Transfer Protocol version 2. Internet-Draft, Internet Engineering Task Force, July 2014.

[BHB01] N. Bansal and M. Harchol-Balter. Analysis of SRPT scheduling: investigating unfairness. *SIGMETRICS Performance Evaluation Review*, 29(1):279–290, June 2001.

[BMZ08] S. Bali, S. Machiraju, and H. Zang. PAQ: A Starvation-Resistant Alternative to Proportional Fair. In *Proceedings of IEEE International Conference on Communications*, pages 3012–3016, Beijing, China, May 2008.

[Bod04] S. Bodamer. *Verfahren zur relativen Dienstgütedifferenzierung in IP-Netzknoten - 88. Bericht über verkehrstheoretische Arbeiten*. PhD thesis, Universität Stuttgart, 2004.

[Bon04] T. Bonald. A Score-Based Opportunistic Scheduler for Fading Radio Channels. In *Proceedings of European Wireless*, 2004.

[Bor05] S. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking*, 13(3):636–647, June 2005.

[CG03] X. Cai and G. Giannakis. A two-dimensional channel simulation model for shadowing processes. *IEEE Transactions on Vehicular Technology*, 52(6):1558–1567, November 2003.

[Cis14] Cisco Systems Inc. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018. Technical report, February 2014.

[CL99] H.-K. Choi and J. O. Limb. A behavioral model of web traffic. In *Proceedings of the 7th International Conference on Network Protocols*, pages 327–334, Toronto, Canada, October 1999.

[CSH13] P. Casas, R. Schatz, and T. Hossfeld. Monitoring YouTube QoE: Is Your Mobile Network Delivering the Right Experience to your Customers? In *Proceedings of IEEE Wireless Communications and Networking Conference*, pages 1609–1614, Shanghai, China, April 2013.

[DBC93] P. Dent, G. Bottomley, and T. Croft. Jakes fading model revisited. *Electronics Letters*, 29(13):1162–1163, June 1993.

[Dow01] A. Downey. The structural cause of file size distributions. In *Proceedings of the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 361–370, Cincinnati, OH, USA, August 2001.

[ERHS12] S. Egger, P. Reichl, T. Hossfeld, and R. Schatz. "Time is bandwidth"? Narrowing the gap between subjective time perception and Quality of Experience. In *Proceedings of IEEE International Conference on Communications*, pages 1325–1330, Ottawa, Canada, June 2012.

[FHN12] V. Farkas, B. Héder, and S. Nováczki. A Split Connection TCP Proxy in LTE Networks. In *Proceedings of the 18th International Conference EUNICE/ IFIP WG 6.2, 6.6*, Information and Communication Technologies, pages 263–274, Budapest, Hungary, August 2012.

[FHTG10] M. Fiedler, T. Hossfeld, and P. Tran-Gia. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2):36–41, March 2010.

[Fir] F. Firmin. http://www.3gpp.org/technologies/keywords-acronyms/100-the-evolved-packet-core, 3GPP MCC. Link verified: 2015-06-19.

[FLM$^+$10] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A First Look at Traffic on Smartphones. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pages 281–287, Melbourne, Australia, November 2010.

[FMM$^+$11] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao. YouTube Everywhere: Impact of Device and Infrastructure Synergies on User Experience. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement Conference*, pages 345–360, Berlin, Germany, November 2011.

[GLMT01] W. Gong, Y. Liu, V. Misra, and D. Towsley. On the tails of web file size distributions. In *Proceedings of the Annual Allerton Conference on Communication Control and Computing*, volume 39, pages 192–201, Urbana, IL, USA, September 2001.

[Goe90] C. Goerg. Further results on a new combined strategy based on the SRPT-principle. *IEEE Transactions on Communications*, 38(5):568–570, May 1990.

[GSM] GSM World Coverage Map. http://www.worldtimezone.com/gsm.html. Link verified on 2015-06-19.

[Gud91] M. Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electronics Letters*, 27(23):2145–2146, November 1991.

[HBSBA03] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal. Size-based scheduling to improve web performance. *ACM Transactions on Computer Systems*, 21(2):207–233, May 2003.

[HCMSS04] F. Hernandez-Campos, J. Marron, G. Samorodnitsky, and F. Smith. Variable heavy tails in internet traffic. *Performance Evaluation*, 58:261–284, 2004. Distributed Systems Performance.

[HQG$^+$13] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck. An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. *SIGCOMM Computer Communication Review*, 43(4):363–374, August 2013.

[HSH$^+$11] T. Hossfeld, M. Seufert, M. Hirth, T. Zinner, P. Tran-Gia, and R. Schatz. Quantification of YouTube QoE via Crowdsourcing. In *Proceedings of IEEE International Symposium on Multimedia*, pages 494–499, Dana Point, CA, USA, December 2011.

[HTT] HTTP Archive. http://httparchive.org/. Link verified on 2015-06-19.

[HZS04] M. Hu, J. Zhang, and J. Sadowsky. Traffic aided opportunistic scheduling for wireless networks: algorithms and performance bounds. *Computer Networks*, 46(4):505–518, 2004.

[IBL08] M. Ivanovich, P. Bickerdike, and J. Li. On TCP performance enhancing proxies in a wireless environment. *IEEE Communications Magazine*, 46(9):76–83, September 2008.

[IP11] S. Ihm and V. S. Pai. Towards Understanding Modern Web Traffic. In *Proceedings of ACM SIGCOMM Conference on Internet Measurement Conference*, pages 295–312, Berlin, Germany, November 2011.

[IR08] ITU-R. Requirements related to technical performance for IMT-advanced radio interface(s). (M.2134), November 2008.

[ISO14] ISO/IEC. Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats. (23009-1:2014), May 2014.

[IT96] ITU-T. Methods for objective and subjective assessment of quality. (P.800), August 1996.

[IT03] ITU-T. One-way transmission time. (G.114), May 2003.

[IT05] ITU-T. Estimating end-to-end performance in IP networks for data applications. (G.1030), November 2005.

[IT14] ITU-T. Advanced video coding for generic audiovisual services. (H.264), February 2014.

[JCH84] R. Jain, D. Chiu, and W. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. 1984.

[JKPS00] N. Joshi, S. R. Kadaba, S. Patel, and G. S. Sundaram. Downlink Scheduling in CDMA Data Networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 179–190, Boston, MA, USA, August 2000.

[JPP00] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In *Proceedings of IEEE 51st Vehicular Technology Conference Spring*, volume 3, pages 1854–1858, Tokyo, Japan, May 2000.

[KDSK07] S. Khan, S. Duhovnikov, E. Steinbach, and W. Kellerer. MOS-Based Multiuser Multiapplication Cross-Layer Optimization for Mobile Multimedia Communication. *Advances in Multimedia*, pages 1–11, 2007.

[Kel97] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 1997.

[KH95] R. Knopp and P. Humblet. Information capacity and power control in single-cell multiuser communications. In *Proceedings of IEEE International Conference on Communications*, volume 1, pages 331–335, Seattle, WA, USA, June 1995.

[KH02] S. Khirman and P. Henriksen. Relationship between quality-of-service and quality-of-experience for public internet service. In *Proceedings of the 3rd Workshop on Passive and Active Measurement*, pages 1–6, Fort Collins, CO, USA, March 2002.

[KK08] A. Kuehne and A. Klein. Throughput analysis of multi-user OFDMA-systems using imperfect CQI feedback and diversity techniques. *IEEE Journal on Selected Areas in Communications*, 26(8):1440–1450, October 2008.

[Kol03] T. Kolding. Link and system performance aspects of proportional fair scheduling in WCDMA/HSDPA. In *Proceedings of IEEE 58th Vehicular Technology Conference Fall*, volume 3, pages 1717–1722, Orlando, FL, USA, October 2003.

[LA08] P. Lassila and S. Aalto. Combining opportunistic and size-based scheduling in wireless systems. In *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 323–332, Vancouver, Canada, October 2008.

[LLG$^+$13] Y. Liu, F. Li, L. Guo, B. Shen, S. Chen, and Y. Lan. Measurement and Analysis of an Internet Streaming Service to Mobile Devices. *IEEE Transactions on Parallel and Distributed Systems*, 24(11):2240–2250, November 2013.

[LTWW94] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2(1):1–15, February 1994.

[MCC11] R. Mok, E. Chan, and R. Chang. Measuring the quality of experience of HTTP video streaming. In *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management*, pages 485–492, Dublin, Ireland, May 2011.

[MDRR03] R. Mangharam, M. Demirhan, R. Rajkumar, and D. Raychaudhuri. Size matters: Size-based scheduling for MPEG-4 over wireless channels. In *SPIE & ACM Proceedings in Multimedia Computing and Networking*, pages 110–122, 2003.

[MRSG99] S. Muthukrishnan, R. Rajaraman, A. Shaheen, and J. Gehrke. Online scheduling to minimize average stretch. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 433–443, New York, NY, USA, October 1999.

[MSF10] G. Maier, F. Schneider, and A. Feldmann. A first look at mobile hand-held device traffic. In *Proceedings of the 11th International Conference on Passive and Active Measurement*, pages 161–170, Zurich, Switzerland, April 2010.

[MSH03] M. Meyer, J. Sachs, and M. Holzke. Performance evaluation of a TCP proxy in WCDMA networks. *IEEE Wireless Communications*, 10(5):70–79, October 2003.

[Nec06] M. C. Necker. A comparison of scheduling mechanisms for service class differentiation in HSDPA networks. *AEU - International Journal of Electronics and Communications*, 60(2):136–141, 2006.

[NGM08] NGMN. Radio Access Performance Evaluation Methodology. January 2008. R. Irmer (ed.).

[Nie10] J. Nielsen. Website response times. `http://www.nngroup.com/articles/website-response-times/`, June 2010. Link verified on 2015-06-19.

[NOALS$^+$13] J. Navarro-Ortiz, P. Ameigeiras, J. Lopez-Soler, J. Lorca-Hernando, Q. Perez-Tarrero, and R. Garcia-Perez. A QoE-Aware Scheduler for HTTP Progressive Video in OFDMA Systems. *IEEE Communications Letters*, 17(4):677–680, April 2013.

[NUN10] S. Niida, S. Uemura, and H. Nakamura. Mobile services. *IEEE Vehicular Technology Magazine*, 5(3):61–67, September 2010.

[NW08] M. Nuyens and A. Wierman. The foreground–background queue: a survey. *Performance evaluation*, 65(3):286–307, 2008.

[PKV11] M. Proebster, M. Kaschub, and S. Valentin. Context-Aware Resource Allocation to Improve the Quality of Service of Heterogeneous Traffic. In *Proceedings of IEEE International Conference on Communications*, pages 1–6, Kyoto, Japan, June 2011.

[PKWV12] M. Proebster, M. Kaschub, T. Werthmann, and S. Valentin. Context-aware resource allocation for cellular wireless networks. *EURASIP Journal on Wireless Communications and Networking*, (216):1–19, 2012.

[PMB10] M. Proebster, C. Mueller, and H. Bakker. Adaptive fairness control for a proportional fair LTE scheduler. In *Proceedings of IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications*, pages 1504–1509, Istanbul, Turkey, September 2010.

[Pro14] M. Proebster. Improving the Quality of Experience with Size-Based and Opportunistic Scheduling. In *Proceedings of the 11th International Symposium on Wireless Communication Systems*, pages 443–448, Barcelona, Spain, August 2014.

[PSBD11] U. Paul, A. Subramanian, M. Buddhikot, and S. Das. Understanding traffic dynamics in cellular data networks. In *Proceedings of IEEE International Conference on Computer Communications*, pages 882–890, Shanghai, China, April 2011.

[PSH09] M. Proebster, M. Scharf, and S. Hauger. Performance Comparison of Router Assisted Congestion Control Protocols: XCP vs. RCP. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, number 88, pages 1–8, Rome, Italy, March 2009.

[RFC1633] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. RFC 1633 (Informational), Internet Engineering Task Force, June 1994.

[RFC2205] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification. RFC 2205 (Proposed Standard), Internet Engineering Task Force, September 1997.

[RFC2474] K. Nichols, S. Blake, F. Baker, and D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474 (Proposed Standard), Internet Engineering Task Force, December 1998.

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC 2475 (Informational), Internet Engineering Task Force, December 1998.

[RFC2597] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597 (Proposed Standard), Internet Engineering Task Force, June 1999.

[RFC3135] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby. Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations. RFC 3135 (Informational), Internet Engineering Task Force, June 2001.

[RFC3246] B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis. An Expedited Forwarding PHB (Per-Hop Behavior). RFC 3246 (Proposed Standard), Internet Engineering Task Force, March 2002.

[RMPGA$^+$14] J. Ramos-Munoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. Lopez-Soler. Characteristics of Mobile YouTube Traffic. *IEEE Wireless Communications*, 21(1):18–25, February 2014.

[RRSS05] S. Ryu, B. Ryu, H. Seo, and M. Shin. Urgency and Efficiency based Packet Scheduling Algorithm for OFDMA wireless system. In *Proceedings of IEEE International Conference on Communications*, volume 4, pages 2779–2785, Seoul, Korea, May 2005.

[San14] Sandvine Inc. Global Internet Phenomina Report, 1H 2014. Technical report, 2014.

[SBdV11] B. Sadiq, S. J. Baek, and G. de Veciana. Delay-Optimal Opportunistic Scheduling and Approximations: The Log Rule. *IEEE/ACM Transactions on Networking*, 19:405–418, 2011.

[Sch68] L. Schrage. A Proof of the Optimality of the Shortest Remaining Processing Time Discipline. *Operations Research*, 16(3):687–690, 1968.

[Sch93] F. Schreiber. Properties and Applications of the Optimal Queueing Strategy SRPT – A Survey. *AEU - International Journal of Electronics and Communications*, 47:372–378, 1993.

[SCJO01] F. D. Smith, F. H. Campos, K. Jeffay, and D. Ott. What TCP/IP Protocol Headers Can Tell Us About the Web. *SIGMETRICS Performance Evaluation Review*, 29(1):245–256, June 2001.

[SDV10] B. Sadiq and G. De Veciana. Balancing SRPT prioritization vs opportunistic gain in wireless systems with flow dynamics. In *Proceedings of the 22nd International Teletraffic Congress*, pages 1–8, Amsterdam, Netherlands, September 2010.

[SG89] L. Schmickler and C. Goerg. Performance evaluation of a new CSMA/CD protocol based on the SRPT principle. In *Proceedings of IEEE Global Telecommunications Conference and Exhibition*, pages 924–929, Dallas, TX, USA, November 1989.

[Sha49] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[SHW$^+$10] B. Staehle, M. Hirth, F. Wamser, R. Pries, and D. Staehle. YoMo: A YouTube Application Comfort Monitoring Tool. Technical Report 467, University of Würzburg, March 2010.

[Sim] IKR SimLib. http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/. Link verified on 2015-06-19.

[SL05] G. Song and Y. Li. Utility-based resource allocation and scheduling in OFDM-based wireless broadband networks. *IEEE Communications Magazine*, 43(12):127–134, December 2005.

[SM66] L. E. Schrage and L. W. Miller. The Queue M/G/1 with the Shortest Remaining Processing Time Discipline. *Operations Research*, 14(4):670–684, 1966.

[SM02] Z. Shao and U. Madhow. A QoS framework for heavy-tailed traffic over the wireless Internet. In *Proceedings of IEEE Military Communications Conference*, volume 2, pages 1201–1205, Anaheim, CA, USA, October 2002.

[SS02] S. Shakkottai and A. L. Stolyar. Scheduling for multiple flows sharing a time-varying channel: The exponential rule. *American Mathematical Society Translations, Series 2*, 207:185–202, December 2002.

[STB09] S. Sesia, I. Toufik, and M. Baker, editors. *LTE – The UMTS Long Term Evolution: From Theory to Practice*. John Wiley & Sons, Ltd, 2009.

[TKSK09] S. Thakolsri, S. Khan, E. Steinbach, and W. Kellerer. QoE-Driven Cross-Layer Optimization for High Speed Downlink Packet Access. *Journal of Communications*, 4(9):669–680, 2009.

[TLFA13] I. Taboada, F. Liberal, J. O. Fajardo, and U. Ayesta. QoE-aware optimization of multimedia flow scheduling. *Computer Communications*, 36(15–16):1629–1638, 2013.

[Tsy02] B. Tsybakov. File transmission over wireless fast fading downlink. *IEEE Transactions on Information Theory*, 48(8):2323–2337, 2002.

[TV05] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[Ver] Verizon IP Latency Statistics. `http://www.verizonenterprise.com/about/network/latency/`. Link verified on 2015-06-19.

[VTL02] P. Viswanath, D. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *IEEE Transactions on Information Theory*, 48(6):1277–1294, June 2002.

[VW10] S. Valentin and T. Wild. Studying the Sum Capacity of Mobile Multiuser Diversity Systems with Feedback Errors and Delay. In *Proceedings of IEEE 72nd Vehicular Technology Conference Fall*, pages 1–5, Ottawa, Canada, September 2010.

[WHBO05] A. Wierman, M. Harchol-Balter, and T. Osogami. Nearly Insensitive Bounds on SMART Scheduling. *SIGMETRICS Performance Evaluation Review*, 33(1):205–216, June 2005.

[WiM10] WiMAX Forum. WiMAX and the IEEE 802.16m Air Interface Standard - April 2010. Technical report, April 2010.

[WSP$^+$12] F. Wamser, D. Staehle, J. Prokopec, A. Maeder, and P. Tran-Gia. Utilizing Buffered YouTube Playtime for QoE-oriented Scheduling in OFDMA Networks. In *Proceedings of the 24th International Teletraffic Congress*, number 15, pages 1–8, Krakow, Poland, September 2012.

[XJF$^+$15] X. Xu, Y. Jiang, T. Flach, E. Katz-Bassett, D. Choffnes, and R. Govindan. Investigating transparent web proxies in cellular networks. In *Proceedings of the 16th International Conference on Passive and Active Measurement*, pages 262–276, New York, NY, USA, March 2015.

[YBC02] S. Ye, R. Blum, and L. Cimini. Adaptive modulation for variable-rate OFDM systems with imperfect channel information. In *Proceedings of IEEE 55th Vehicular Technology Conference Spring*, volume 2, pages 767–771, Birmingham, AL, USA, May 2002.

[You] YouTube. https://www.youtube.com/. Link verified on 2015-06-19.

[YS06] C.-W. Yang and S. Shakkottai. Asymptotic Evaluation of Delay in the SRPT Scheduler. *IEEE Transactions on Automatic Control*, 51(11):1848–1854, November 2006.

[YWSHB12] C. Yang, A. Wierman, S. Shakkottai, and M. Harchol-Balter. Many Flows Asymptotics for SMART Scheduling Policies. *IEEE Transactions on Automatic Control*, 57(2):376–391, February 2012.

Acknowledgments

This dissertation was developed during my time at the Institute of Communication Networks and Computer Engineering (IKR) of the University of Stuttgart. I spent in total six years at the IKR which contained interesting research projects, teaching assistance, the supervision of Diploma, Master and Bachelor theses, and finally the writing of this dissertation. I am very grateful for the rich experiences I was allowed to make during that time.

First of all, I would like to thank my supervisor and head of the institute Prof. Andreas Kirstädter for the many fruitful discussions and comments, which helped me to further develop and improve my thesis as well as for the freedom and creative atmosphere prevailing at the IKR. I am also very grateful to Prof. Wolfgang Kellerer for the evaluation of this thesis and the good teamwork to organize workshops for the ITG 5.2.4 during several years.

The atmosphere in the IKR team was always very friendly and inspiring. I had a very good relationship to all colleagues and made many friends during that time. My special thanks go to Frank Feller with whom I shared the office for a very long time and Thomas Werthmann with whom I worked closely together in the mobility group. Furthermore, I want to thank Matthias Kaschub for his contributions and the common work in the CARA project, where the notion of transactions was developed and Christian Müller who mentored me at the start of my life at the institute and during my first research projects. For reviewing parts of this thesis, I am grateful to Sebastian Scholz, Kristian Ulshöfer, David Wagner and again Thomas Werthmann. I am also very thankful to Ulrich Gemkow for his tireless commitment to the IKR administration and infrastructure as well as the many helpful guidelines and sometimes critical but always constructive discussions.

Many thanks also go to my beloved wife Maximiliane for always supporting me and bringing so much effort in caring for our children, which enabled me to successfully complete my graduation. Furthermore, I want to thank my parents for giving me so much love and always supporting my curiosity in trying to find out how things work and backing me in all the steps I have taken in my life. Finally, although we did not discuss directly about my research, I would like to thank my grandfather Prof. Walter E. Proebster whose career in research and development is a great role model for me and further motivated me to finish my graduation.