QUALITY OF EXPERIENCE ON SMARTPHONES

NETWORK, APPLICATION, AND ENERGY PERSPECTIVES

Selim Ickin

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2015:06

Department of Communication Systems

Quality of Experience on Smartphones

Network, Application, and Energy Perspectives

Selim Ickin

Blekinge Institute of Technology Doctoral Dissertation Series
No 2015:06

Quality of Experience on Smartphones

Network, Application, and Energy Perspectives

Selim Ickin

Doctoral Dissertation in
Telecommunication Systems

Department of Communication Systems
Blekinge Institute of Technology
SWEDEN

Department of Communication Systems
Publisher: Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
Printed by Lenanders Grafiska, Kalmar, 2015
ISBN: 978-91-7295-303-1
ISSN: 1653-2090
urn:nbn:se:bth-00615

A book should serve as the ax for the frozen sea within us.
–Franz Kafka

Abstract

Smartphones have become crucial enablers for users to exploit online services such as learning, leisure, communicating, and socializing. The user-perceived quality of applications and services is an important factor to consider, in order to achieve lean resource management, to prevent user churn and revenue depletion of service or network providers. This is often studied within the scope of Quality of Experience (QoE), which has attracted researchers both in academia and industry.

The objective of this thesis is to study the most important factors influencing QoE on smartphones and synthesize solutions for intervention. The temporal impairments during a real-time energy-hungry video streaming are studied. The aim is to quantify the influence of temporal impairments on the user-perceived video QoE at the network and application level together with energy measurements, and also to propose solutions to reduce smartphone energy consumption without degrading the user's QoE on the smartphone for both user-interactive, *e.g.*, video, and non-interactive cases.

QoE measurements on smartphones are performed throughout in-the-wild user studies. A set of quantitative Quality of Experience (QoE) assessment tools are implemented and deployed for automatic data logging at the network- and application-level. Online momentary survey, Experience Sampling Method (ESM) software, and Day Reconstruction Method (DRM) along weekly face-to-face user interviews are employed. The subjective QoE is obtained through qualitative feedback including Mean Opinion Score (MOS) as well as in-situ indications of poor experiences by users. Additionally, energy measurements on smartphones are conducted in controlled-lab environment with the Monsoon device.

The QoE of smartphone applications and services perceived by users depends on many factors including anomalies in the network, application, and also the energy consumption. At the network-level, high packet delay variation causes long video freezes that eventually impact negatively the end-user perceived quality. The freezes can be quantified as large time gaps in-between the displayed pictures during a video stream at the application-level. We show that the inter-picture time in cellular-based video stream can be represented via two-state exponential ON/OFF models. We show models representing the non-linear relationship between the QoE and the mean inter-picture time. It is shown that energy measurements help to reveal the temporal impairments in video stream enabling energy consumption as a QoE indicator. Next, energy waste and saving during temporal impairments are identified. Additionally, other video streaming use cases, *e.g.*, "download first and watch later", are studied and appropriate energy-saving download scheduling mechanisms are recommended. The possibility for decreasing energy consumption when the smartphone screen is OFF, while maintaining QoE, is presented. We first show exponential models to represent user's interaction

with smartphone, then propose a NyxEnergySaver software, to control the cellular-network interface in a personalized manner to save smartphone energy. According to our findings, more than 30 % smartphone energy can be saved without impacting the user-perceived QoE.

Acknowledgements

First of all, I would like to thank and extend my sincere gratitude to Prof. Dr. -Ing. Markus Fiedler for accepting me as a Ph.D. student, enabling tremendous outstanding opportunities all over the world, encouraging me in my research in good and bad times. This thesis would not be realized without his support, motivation, and guidance.

My co-supervisor Dr. Katarzyna Wac from University of Copenhagen, thanks to her for providing full-time support, sharing deep experiences, inspiring, and encouraging me to research towards intelligent future Internet mechanisms with the "quantified-self" and the "Quality of Life" vision.

Thanks to Dr. Patrik Arlos from BTH for enlightening me with his comprehensive knowledge on 3G-network traffic characteristics, and accuracy on measurements. I have learnt a lot from him on criticizing a scientific work along with constructive feedback.

Thanks to Dr. Lucjan Janowski from AGH University of Technology in Poland for his indispensible support and for pushing me towards systematic methodology in my research. I have benefited a lot from his great experience and knowledge on the video QoE.

Special thanks to Dr. Anind K. Dey and Dr. Jin-Hyuk Hong from Carnegie Mellon University in the U.S.; they deserve great appreciation for supervising and supporting me during my stay at Carnegie Mellon University.

I've learnt a lot during the discussions with Prof. Dr. Adrian Popescu particularly on Green Networking and future Internet (*e.g.*, 5G), thanks to him.

Thanks to Dr. Thomas Zinner from University of Würzburg in Germany, for the discussion and the collaboration on future research topics concerning energy consumption and Quality of Experience (QoE) on mobile terminals.

Thanks to Luis Guillermo Martinez Ballesteros from Royal Institute of Technology (KTH) in Sweden, for extensive discussions and the collaboration on video QoE and energy consumption on mobile terminals.

To all my colleagues, especially Dr. Alexandru Popescu, Dr. Yong Yao, Dr. Karel De Vogeleer, and Dr. Dragos Illie. Thanks to them for sharing valuable experiences and thoughts during occasional formal and informal discussions.

And last but not the least, big thanks to my family for their tremendous support until now.

Full List of Publications[1]

Journal Articles [Peer-reviewed]

1. S. Ickin, K. Wac, M. Fiedler, L. Janowski, J. H. Hong, and A. K. Dey, Factors influencing Quality of Experience of commonly-used mobile applications, *IEEE Communications Magazine*, vol. 50, no. 4, pp. 48-56, April 2012.*

2. S. Ickin, M. Fiedler, K. Wac, P. Arlos, C. Temiz, K. Mkocha. VLQoE: Video Quality of Experience Instrumentation on the Smartphone, *Multimedia Tools and Applications Journal. Special Issue on Advances in Tools, Techniques and Practices for Multimedia QoE*. Springer US, vol.74/2, pp.381-411, 2015.*

Conference Papers [Peer-reviewed]

3. S.Ickin, K. Wac, M. Fiedler, "Quality of Experience-Based Energy Reduction by Controlling the 3G Cellular Data Traffic on the Smartphone", *The 22nd International Teletraffic Congress Specialist Seminar on Energy Efficient and Green Networking (ITC SSEEGN)*, New Zealand, 2013.*

4. S.Ickin, T. Zinner, K. Wac, M. Fiedler, "Catching the Download Train: Energy-efficient File Downloading On Smartphones", *26th IEEE International Teletraffic Congress (ITC)*, Sweden, 2014.*

5. S. Ickin, M. Fiedler, K. Wac, Energy-based anomaly detection in quality of experience, *16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, vol., no., pp.1-6, USA, 2013.*

6. M. Gustarini, S.Ickin, K. Wac. Evaluation of Challenges in Human Subject Studies "In the Wild" Using Subjects' Personal Smartphones. *ACM conference on Pervasive and ubiquitous computing adjunct publication (UbiComp '13 Adjunct)*, ACM. New York, USA, 1447-1456, 2013.

7. S. Ickin, L. Janowski, K. Wac, and M. Fiedler, "Studying the challenges in assessing the perceived quality of mobile-phone based video", *Fourth International Workshop on Quality of Multimedia Experience (QoMEX)*, Australia, 2012.

8. S. Ickin, K. D. Vogeleer, M. Fiedler, and D. Erman, "The effects of packet delay variation on the perceptual quality of video", *4th Workshop On User MObility and VEhicular Networks 2010 (IEEE LCN On-MOVE)*, USA, 2010.*

[1] The contributions that are included in the Ph.D. thesis are marked with '*.

9. S. Ickin, K. Wac, and M. Fiedler, "Demonstrating the stalling events with Instantaneous Total Power Consumption in smartphone-based live video streaming", *The Second IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT)*, Italy, 2012.*

10. K. Wac, S. Ickin, J. H. Hong, L. Janowski, M. Fiedler, and Anind K. Dey, "Studying the experience of mobile applications used in different contexts of daily life", *ACM SIGCOMM Workshop on Measurements Up the STack (W-MUST)*, Canada, 2011.

Licentiate Thesis and Book Chapter

11. Selim Ickin. Identification of Influential Factors on Android Smartphone-Based Video Quality of Experience. *Licentiate Dissertation*, Blekinge Institute of Technology. Sweden, 2013.

12. J. M. Pierson, Book Chapter in Large-scale Distributed Systems and Energy Efficiency: A Holistic View. "Green Wireless-energy efficiency in Wireless Networks", *Wiley Series on Parallel and Distributed Computing*, John Wiley & Sons, 2015.*

Other Published Material

13. S. Ickin, K. D. Vogeleer, D. Erman, and M. Fiedler, "Standardisation of user perception on mobile streaming applications in seamless network handover", *Joint FP7 PERIMETER & ETSI-HF Workshop*, France, 2011.

14. S. Ickin, K. D. Vogeleer, M. Fiedler, and D. Erman, "On the choice of performance metrics for user-centric seamless communication", *Third Euro-NF IA.7.5 Workshop on Socio-Economic Issues of Networks of the Future*, Belgium, 2010.

15. S. Ickin, "Energy metrics unique enough for smartphone-based video QoE evaluation?", *Quality of Experience: From User Perception to Instrumental Metrics. Dagstuhl Seminar 12181, Dagstuhl Reports*, Volume 2, Issue 5, pg 12, Germany, 2012.*

16. S. Ickin, K. D. Vogeleer, D. Erman, and M. Fiedler, Report on the conduct and conclusion of the standardization workshop, *Deliverable 8.7-PERIMETER, User-Centric paradigm for Seamless Mobility in Future Internet*, 2011.

17. S. Ickin, Implementation of Measurement Module for Seamless Vertical Handover, *Master's thesis*, 2010.

18. K. D. Vogeleer, S. Ickin, M. Fiedler, D. Erman, and Adrian Popescu, "Estimation of Quality of Experience in 3G-networks with the Mahalanobis Distance", In *The Fourth International Conference on Communication Theory, Reliability, and Quality of Service* (CTRQ), Hungary, 2011. [peer-reviewed].

19. K. D. Vogeleer, S. Ickin, D. Erman, and M. Fiedler, "Perimeter: A user-centric mobility framework", *IEEE 35th Conference on Local Computer Networks* (LCN'10), pages 625 − 626, USA, 2010. [peer-reviewed]

20. K. D. Vogeleer and S. Ickin, "A decentralized information service for Media Independent Handover (MIH - IEEE 802.21)", *Graduate or Senior Design Project Student Application Paper*, 2010. [peer-reviewed]

21. K. D. Vogeleer, S. Ickin, D. Erman, and M. Fiedler. Detailed design of user centric mobility in heterogeneous environments, *Deliverable 3.2- PERIMETER, User-Centric Paradigm for Seamless Mobility in Future Internet*, 2010.

22. K. D. Vogeleer, S. Ickin, D. Erman, and M. Fiedler. Technical report on UMTS limitations in Android, *PERIMETER, User-Centric Paradigm for Seamless Mobility in Future Internet*, 2011.

Manuscripts

23. S. Ickin, K. Wac, M. Fiedler, "Non-Intrusive Sledgehammer Implementation Towards Energy Saving on Smartphone". *

24. S. Ickin, L. G. M. Ballesteros, M. Fiedler, J. Markendahl, K. Wac, "MOS per Joule: QoE-based Energy Efficiency Metric for Smartphone-based Video Streaming". *

Publications Included Within the Thesis

This sub-section describes the thesis chapters with the corresponding involved publications. Chapters 3-8 might include parts from the content of our previously published materials that are itemized as 1, 2, 3, 4, 5, 8, 9, 12, 15, 23, and 24. The matching between the chapters and the corresponding publications are given in Table 1.

Publication	#1	#2	#3	#4	#5	#8	#9	#12	#15	#23	#24
Chapter 3	✓										
Chapter 4						✓					
Chapter 5		✓									
Chapter 6							✓	✓			
Chapter 7				✓	✓		✓		✓		✓
Chapter 8			✓							✓	

Table 1: The matching between the thesis chapters and the included publications

Acronyms

2G	2nd Generation
3G	3rd Generation
3GPP	3rd Generation Partnership Project
4G	4th Generation
ABC	Always Best Connected
ACR-HR	Absolute Category Rating with Hidden Reference
ACR	Absolute Category Rating
API	Application Programming Interface
BDP	Bandwidth Delay Product
BER	Bit Error Rate
BMU	Battery Monitoring Unit
BTH	Blekinge Institute of Technology
BW	BandWidth
CapEx	Capital Expenditures
CBR	Constant Bit Rate
CCDF	Complementary Cumulative Distribution Function
CDMA	Code Division Multiple Access
CDN	Content Delivery Network
CI	Confidence Interval
CIST	Consumption Improvement by Stuffing Time
CL	CLumped packet size
CLBB	Cross-Layer Burst Buffering
CPU	Central Processing Unit
CSS	Context Aware Software
CZD	CZekanowski Distance
DB	Database
DCH	Dedicated Channel
DCR	Degradation Category Rating
DJ	Delay Jitter
DNS	Domain Name Server
DRM	Day Reconstruction Method
DRX	Discontinuous Reception
DSCQS	Double Stimulus Continuous Quality Scale
DSIS	Double Stimulus Impairment Scale
DSS	Darwin Streaming Server
e2e	end-to-end
EARTH	Energy Aware Radio and neTworking tecHnologies
ECDF	Empirical Cumulative Distribution Function
EDGE	Enhanced Data GSM Environment
EMR	Electromagnetic Resonance
ESM	Experience Sampling Method
ETSI	European Telecommunication Standards Institute
EU	European Union
Euro-NF	European project on the Network of the Future
EVDO	Evolution-Data Only
EWMA	Exponentially Weighted Moving Average
FACH	Forward Access Channel
FG-IPTV	Focus Group on IPTV
FP7	Seventh Framework Programme
FPR	False Positive Rate
FR	Full-Reference
GOP	Group of Pictures
GPL	GNU General Public License
GPRS	General Packet Radio Service
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRE	Generic Routing Encapsulation
GSM	Global System for Mobile
GUI	Graphical User Interface
HCI	Human Computer Interaction
HLS	HTTP Live Streaming
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed Uplink Packet Access
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HVS	Human Visual System
ICMP	Internet Control Message Protocol
ICT	Information and Communications

	Technology
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IF	Influential Factor
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISP	Internet Service Provider
ITU-R	International Telecommunication Union - Radio communication
ITU	International Telecommunication Union
JRQ-MMQA	Joint Rapporteur's Group on Multimedia Quality Assessment
JSON	JavaScript Object Notation
KDE	Kernel Density Estimation
KLS	Kernel Level Shaper
KPI	Key Performance Indicators
L2TP	Layer-2 Tunneling Protocol
LCD	Liquid-Crystal Display
LGPL	GNU Library General Public License
LTE	Long Term Evolution
M2M	Machine-to-Machine
MAS	Measurement Assessment Tool
MBS	Maximal Burst Size
MH	Mobile Hotspot
MLDS	Maximum Likelihood Difference Scaling
MLE	Maximum-Likelihood Estimation
MNO	Mobile Network Operators
MOS	Mean Opinion Score
MOVIE	Motion Based Video Integrity
MP3	MPEG-2 Audio Layer III
MPEG-4	Moving Picture Experts Group 4
MPMT	Mobile Power Monitoring Tool
MPQM	Moving Picture Quality Metric
MSS	Maximum Segment Size
NCL	Number of Clumped Packets
NGN	Next Generation Network
NR	No-Reference
OpEx	Operating Expenditures
OR	Outlier Ratio
OS	Opinion Score
P2P	Peer-to-Peer
PC	Personal Computer
PCC	Pearsson Correlation Coefficient
PCH	Paging Channel
PDF	Probability Density Function
PDR	Packet Drop Rate
PDU	Protocol Data Unit
PDV	Packet Delay Variation
PESQ	Perceptual Evaluation of Speech Quality
PEVQ	Perceived Evaluation of Video Quality
PL	Packet Loss
PLT	Page Loading Time
PRR	Packet Reordering Rate
PSNR	Peak Signal-to-Noise Ratio
PSQA	Pseudo-Subjective Quality Assessment
QoE	Quality of Experience
QoL	Quality of Life
QoS	Quality of Service
R	Transmission Rating Factor
RAN	Radio Access Network
RB	Ring Buffer
RF	Radio Frequency
RMSE	Root Mean Square Error
RNC	Radio Network Controller
RR	Reduced-Reference
RRC	Radio Resource Control
RRC	Radio Resource Controller
RSSI	Received Signal Strength Indication
RTCP	RTP Control Protocol
RTP	Real-time Transport Protocol
RTSP	Real Time Streaming Protocol
RTT	Round Trip Time
$\mathbf{S}_P$	Standard Deviation of Instantaneous Power Consumption

SCC	Spearman Correlation Coefficient
SCRI	Signaling Connection Release Indication
SDO	Standard Developing Organizations
SI	Spatial Information
SIM	Subscriber Identity Module
SLA	Service Level Agreement
SMA	Simple Moving Average
SMS	Short Message Service
SNMD	Sensor Node Management Device
SNR	Signal to Noise Ratio
SOS	Standard deviation of Opinion Scores
SPDY	SPeeDY
SRT	Server Response Time
SSCQE	Single Stimulus Continuous Quality Evaluation
SSIM	Structural Similarity Index Model
STPM	Self-tuning Power Manager
STREP	Specific Targeted Research Projects
SVT	Software Visualisation Tool
TCP	Transport Control Protocol
TI	Temporal Information
TOP	Tail Optimization Protocol
TPR	True Positive Rate
TR	User Response Time
TTL	Time-to-Live
UDP	User Datagram Protocol
UDPIP	UDP over IP
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
UR	User Rating
URL	Uniform Resource Locator
UTRAN	UMTS Terrestrial Radio Access Network
VBR	Variable Bit Rate
VDP	Visible Difference Predictor
VIF	Visual Information Fidelity
VPN	Virtual Private Network
VQA	Video Quality Assessment
VQEG	Video Quality Expert Group
VQM	Video Quality Metric
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Networks
Z2T	Zero Throughput Time

Table of Contents

Chapter 1

Introduction

"*The greatest energy of movement will be obtained when synchronism is maintained between the pump impulses and the natural oscillations of the system.*" –Nikola Tesla

---

1.1 Preamble

The computing power grows exponentially as the number of transistors in a dense integrated circuit doubles every two years [1]. The vast majority of devices and things that benefit from being connected will be always-connected to Internet. Internet connectivity and access to Internet services have already become a fundamental need to cope with real life challenges and to contribute to society. Both academia and industry invest to expand the Internet coverage and range of services. For example, "internet.org" project [2] aims to provide Internet access to critical and ideally to all regions in the world. Worldwide, in 2020, 50 billions of things are expected to be connected to the Internet, there will be five billions of mobile broadband subscriptions worldwide [3]. Improvement of processing power on devices and the growing availability of Internet connectivity lets high number of small size mobile ubiquitous devices available for users to launch diverse apps including social networking, instant messaging, and video conferencing. Given that majority of the applications are cloud-based, this gives rise to the amount of mobile network-based services and applications, and eventually causes an exponential growth in Internet traffic. Thus, opportunities (*e.g.*, increased services and applications) yield more experience, demands, and eventually increase user expectations. In close future, personalized big data, driverless cars, virtual reality, and Inter-

net of Things (IoT) will generate an augmented data. In parallel, enormous amounts of Internet data will create more and various challenges, which have to be managed to prevent degradation of the quality perceived by the end-user. Thus, connectivity alone is not sufficient; the delivery of network data needs to be managed as well to assure the application performance. By 2017, it is forecasted that 85 % of the world's population will be covered with 3G-based (*i.e.*, WCDMA/HSDPA) network [3].

Smartphones are the most popular mobile devices enabling mobile Internet. According to Cisco Visual Networking Index (VNI) [6], globally in 2013, the number of mobile devices have reached seven billions (with half a billion increase since 2012), 77 % of which are smartphones. Amongst the smartphones, the Android OS has the market share of 80 % in 2014 [4]. Average smartphone data traffic in 2013 was 529 MB per month [5], and it is expected to grow exponentially (reaching 2 GB per month by 2018) particularly due to video streaming traffic. In addition to the connectivity, the perceived quality of diverse interactive smartphone applications and services have become critical to the user acceptance. The subscribers expect from the applications or services to provide them the best quality at any context and time. Otherwise, the users whose quality expectations are not fully satisfied, may give up using the applications or switch to another network or service provider. In order to prevent the risk of customer churn, the service/network providers need to consider the user-centric approach, and assure the user's expected and perceived quality.

1.2 Introduction to Quality of Experience (QoE)

We define QoS and QoE for mobile computing systems as follows. QoS is defined to understand the underlying reasons for poor application performance at the technical level, *i.e.*, mainly focusing on the metrics related to the layers below the transport layer in the Internet protocol suite. QoE has a larger scope than QoS, and it is an emerging term defined to understand and maximize (or at least maintain) the user-perceived quality for a particular application or a service [16]. It comprises the complete end-to-end effects, *e.g.*, from the generation of data till the consumption of it by the user, of the system. The state-of-the-art underlying technical network-based factors such as delay often determines the performance of mobile applications. QoS, influencing QoE, can be critical to the user's QoE especially for highly interactive multimedia applications that necessitate very high data rates. QoS highly focuses on provider-centric approach, *i.e.*, based on data as measured at the core network or at the access base stations to predict the quality of the network level. QoS contributes to QoE, but is not sufficient to define it. Studying the end-user perceived quality from various aspects (*e.g.*, network, application, and energy) and on the end terminals (*e.g.*, smartphones of

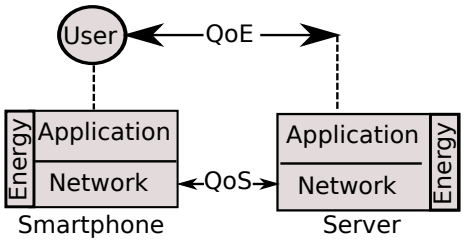


Figure 1.1: QoE from network, application, and energy perspectives are depicted.

users) in a multi-disciplinary manner is necessary. The evaluation of QoE so far has mainly focused on an application's usability, which is evaluated in studies conducted for a limited time in controlled laboratory environments, under conditions that do not resemble users' natural daily environments. The results of such evaluations help to discover serious and immediate usability issues, but they are unlikely to help in recovering from issues that are relevant to real-life situations outside the laboratory (*e.g.*, variable network connectivity).

In this thesis, we study the QoE from network, application, and energy perspectives and also present the inter-relation of the measured metrics as depicted in Figure 1.1.
Network perspective: In this thesis, we study various definitions of delay jitter including the one-way Packet Delay Variation (PDV) and the Maximal Burst Size (MBS) size. One-way packet delay variation is the variation of end-to-end (*i.e.*, between the streaming server and the terminal) one-way packet delay, which utilizes the timestamps of packets upon arrival/departure to/from the terminal/streaming server. Maximal Burst Size (MBS) is a metric that enables to understand the degree of clumping exhibited by packets arriving to the terminal. In addition, we study metrics such as Server Response Time (SRT) (as measured in the application-layer) and Round Trip Time (RTT), which are highly influenced by the network conditions. SRT is measured in this thesis as the duration between when the HTTP request is sent from a smartphone to a weather forecast web server (located in the US), and the HTTP response (with a simple text data with weather information) is received back to the smartphone. RTT is the duration between the ECHO request and ECHO reply messages measured at the smartphone.
Application perspective: In the application-layer, there are many metrics that can be studied including application's state like data traffic and the events in the user interface. In this thesis, we consider a set of application-layer metrics such as the duration in-between the display of the pictures in a video stream, user's interaction with the user interface (*e.g.*, touch events, user's QoE indication), and the running applications.

Energy perspective: We measure the power consumption of a smartphone and its applications, *e.g.*, the duration that an application is in a particular state and its energy consumption for this state. Instantaneous power consumption is defined as the product of voltage (U) and the current (I) drawn from the smartphone's battery. In this thesis, we consider the average power during one second.
Inter-relation between the three perspectives: The power state of a network-based smartphone application depends on the application state which might be directly influenced by the network conditions. For example, if there is not enough bandwidth, the delivery of packets to the application gets delayed, which eventually causes interruptions in displaying the content of the packets to the user due to buffer underflow. When there are no packets to process in the buffer, the power consumption of smartphone might drop to a lower power level, thus the energy might be wasted during that time interval. Thus, all the three perspectives are inter-related and in this thesis we research these inter-relations. Next we summarize the research questions in Section 1.3.

1.3 Research Questions

The research questions (RQ) that are researched within this thesis are itemized as follows.

- R.Q.1: What is the methodology to assess the user's QoE on a smartphone?
- R.Q.2: What are the most influential factors for the user QoE on smartphones?
- R.Q.3: How can the video QoE on smartphones be assessed?
  - R.Q.3.1: What is the method to assess video QoE in the network-level on smartphones?
  - R.Q.3.2: What is the most indicative packet-level QoS metric for video QoE?
  - R.Q.3.3: What is the method to assess video QoE at the application level?
  - R.Q.3.4: What is the relationship between the QoE and the delay in-between the displayed pictures, *i.e.*, inter-picture time, on the smartphone screen during network-based video streaming?
  - R.Q.3.5: What is the power consumption pattern during video streaming application, *i.e.*, how can QoE be revealed in the power level?
  - R.Q.3.6: How can the power consumption metric help in developing energy-efficient QoE assessment tools?

- R.Q.4: What are the advantages and disadvantages of the external hardware and internal software energy measurement tools for smartphones?
- R.Q.5: How can the energy consumption be reduced while user is video streaming on the smartphone?
  - R.Q.5.1: Is there always a trade-off between the video QoE and the energy consumption on the smartphone? What are the scenarios where the improvement in QoE and the decrease in energy consumption are correlated?
  - R.Q.5.2: What are the tradeoffs (with respect to energy consumption and QoE) for real-time network-based video streaming vs. local video streaming, *i.e.*, "download-first-and-watch-later" approach?
  - R.Q.5.3: How can multiple file downloads be scheduled to reduce the energy consumption on the smartphone?
- R.Q.6: How can the smartphone energy consumption be reduced in a simple-to-deploy user-centric manner, when the user is not interacting with the smartphone?
  - R.Q.6.1: What is the mathematical model for a screen-based smartphone interaction?
  - R.Q.6.2: What is the impact of bursty traffic during screen OFF?
  - R.Q.6.3: How much energy can be saved if smartphone connectivity is controlled based on user behaviour when the screen is OFF?
  - R.Q.6.4: How can we extend the OFF duration of the network data module without degrading the user QoE?

1.4 Overview of the Thesis

In this section, the important contents and the flow of the thesis are given. We summarize the content of each chapter in separate sub-sections, and in parallel we present the inter-relations of chapters. Each subsection includes a figure that summarizes the corresponding chapter's motivation, methods, addressed research questions, and the most important outcomes. Chapter 2 introduces the concepts related to QoE, and also presents the technical background. We start to present our core research work in Chapter 3 of this thesis.

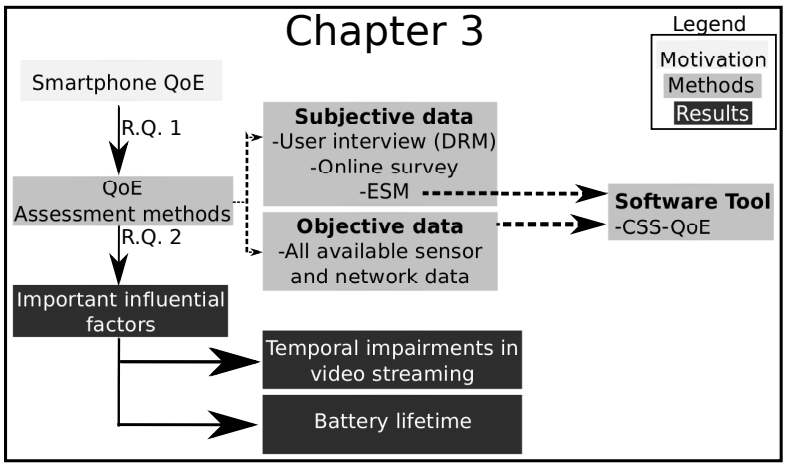


Figure 1.2: Chapter 3 overview: identifying the influential factors for smartphone QoE.

1.4.1 Identification of Influential Factors on Smartphone QoE

The thesis starts with the identification of the most important factors influencing the smartphone QoE (Figure 1.2). Research questions R.Q.1 and R.Q.2 are addressed. A software measurement tool, QoE-CSS, is developed and used for collecting both objective and subjective QoE metrics. The objective data comprises autonomously collected samples from the smartphone sensors and network conditions. Various research methods such as Experience Sampling Method (ESM), online survey, and Day Reconstruction Method (DRM) are employed to collect subjective QoE rating data. ESM consists of user's Mean Opinion Score (MOS), and data regarding user's momentary context. We use two different notions, MOS and User Rating (UR) throughout the thesis, where the MOS is used to address the overall averaged user rating, while the UR is sampled more frequently and indicates the instantaneous opinion score of the user (*e.g.*, frequent user rating during video streaming). The concepts and the definition of the methods are given in Chapter 2.

Amongst many, the most important influential factors for the end-user perceived smartphone QoE are identified. The first one is the *battery life* that might even prevent the user from the usage of the smartphone, as the operation time is limited to the amount of energy left on the battery. Battery lifetime is defined as the amount of duration a smartphone can run on a single charge for a given usage pattern [8]. Aside from the economical (*e.g.*, charging mobile devices) or environmental (*e.g.*, CO_2 foot-

prints) aspects, the most important expectation for a smartphone relates to its capability and energy to complete any task requested by its user [67]. The advancements in battery technology, which enables the applications or services on the mobile devices, stay rather slow as compared to the recent developments in ubiquitous computing. Consequently, energy consumption of applications on smartphones is a key factor determining the smartphone QoE.

The second factor influencing user's QoE is related to specifics of video streaming, *temporal impairments*, *e.g.*, video freezes (or stalling events). Their excessive occurrence might even prevent users to watch real-time network-based video on the smartphone. Some examples of temporal impairments are spottiness or occasional discontinuities in the video playout, which are often caused by unstable end-to-end network throughput. In parallel, video streaming is also known to be one of the most energy-consuming applications on smartphones. This is due to its high resource demands such as an active wireless technology interface for downloading data with a high bitrate, high Central Processing Unit (CPU) utilization to process/decode/render/adapt high amounts of video content, and an adequately bright Liquid-Crystal Display (LCD) screen for displaying the content to the user [99]. In parallel, the energy consumption of video streaming application is highly influenced by the temporal impairments. Thus, the influence of video freezes on QoE are twofold, (i) video freezes caused by mobile context and fluctuations of end-to-end network link quality, might degrade QoE immediately and (ii) video freezes may increase the energy consumption (due to extended stream duration), thus indirectly influence the long-term QoE on the smartphone. In this thesis, the inter-relation between the two factors affecting the QoE is also researched in details.

1.4.2 Temporal Impairments in Video: Packets and Pictures

As explained in the previous section, the QoE of video streaming applications may be low due to the occasional disturbances at network level that manifest themselves at the application level as stalling events. The study of video streaming starts with the identification of the network-based influential metrics. First, as a part of the PERIMETER measurement subsystem implementation [11], a user study is conducted to find out how the delay variation and the bursty video data traffic influences the end user perceived quality during the video streaming on the smartphone. Network measurements, via objective metrics such as PDV and MBS, are obtained from within the smartphone kernel-level and then matched with the subjective user ratings (*e.g.*, UR) collected via ESM. We represent the relationship between the UR and the PDV and MBS via power-law models. The summary of this study is given in Figure 1.3, and the details of the study is given in Chapter 4. Research questions R.Q.3.1 and R.Q.3.2 are addressed.

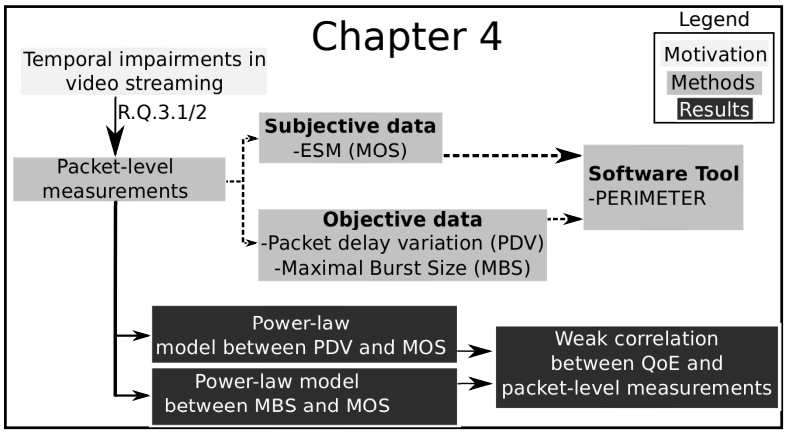


Figure 1.3: Chapter 4 overview: identifying video QoE via network-level measurements.

As merely weak correlations were detected in-between the packet-level metrics and the user perceived quality, it is followed by another user study with the focus on the application and user interface metrics. To this end, the tool VLQoE is developed to record and measure the delay in-between the displayed pictures (inter-picture time) on the smartphone screen, *i.e.*, the user interface. This made possible to obtain direct comparison between the user perceived quality (measured via MOS and freezes indicated by user) and the inter-picture time. We complement the study with additional face-to-face user interviews and online surveys, and obtained qualitative feedback. There is a set of outcomes of this study, and the most important one is that the inter-picture time on the video display can be represented as a two-state ON/OFF exponential model. The study is summarized in Figure 1.4, and the details are given in Chapter 5. The research questions R.Q.3.3 and R.Q.3.4 are addressed.

1.4.3 Measuring Energy Consumption on Smartphones

Our research on the smartphone battery life comprises energy measurements on smartphones. Existing energy measurement methods are studied, and amongst many techniques, the Monsoon power monitoring tool [170] is chosen and used in the further energy studies in this thesis. The main reason for choosing the Monsoon power monitoring tool is shortly its capability to obtain rigorous power measurements on the smartphone, *e.g.*, with 5000 power measurement samples per second. This also helped us to

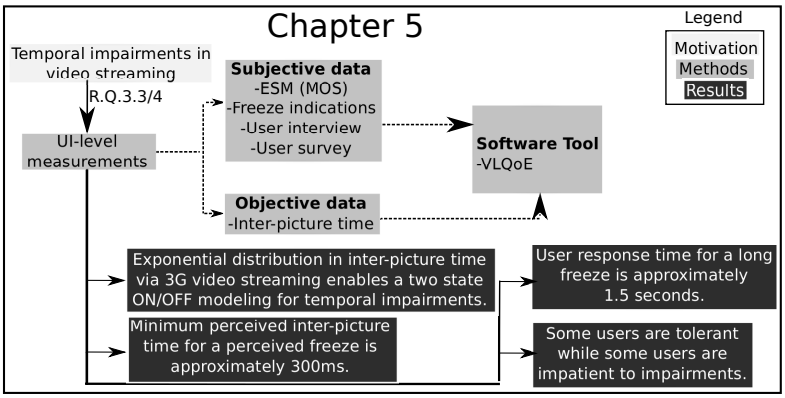


Figure 1.4: Chapter 5 overview: identifying video QoE via the application-layer measurements.

study in details (anomalies) of the power consumption for the power-hungry network-based real-time applications such as video streaming. We have compared measurements obtained via both software- and hardware-based methods. One outcome is that software-based power measurement tools (*e.g.*, PowerTutor) may not measure accurately the power consumption of devices, they induce a significantly high power overhead on the running mobile device and low sampling rate. The study is summarized in Figure 1.5, and further details about the study are given in Chapter 6. The research question R.Q.4 is addressed in Chapter 6.

1.4.4 Energy Saving Approaches in Video Streaming

We conducted QoE and power measurements during video streaming on the smartphone, and eventually propose a set of energy saving approaches. Developing non-intrusive video QoE assessment tools, particularly to acquire packet-level measurements, are often complex. In addition, continuous probing of the received and transmitted packets induces extra processing overhead on the device, which highly increases the energy consumption. Towards this aim, the software-based VLQoE tool is utilized during the energy measurements. This helped to visualize and match the anomalies in the power consumption (measured via hardware-based Monsoon power monitoring tool) and the anomalies in the user interface. Anomalies in the user interface are measured both via subjective freeze indications by the users as well as the objective inter-picture time metric. This provides smartphone QoE researchers to consider the power

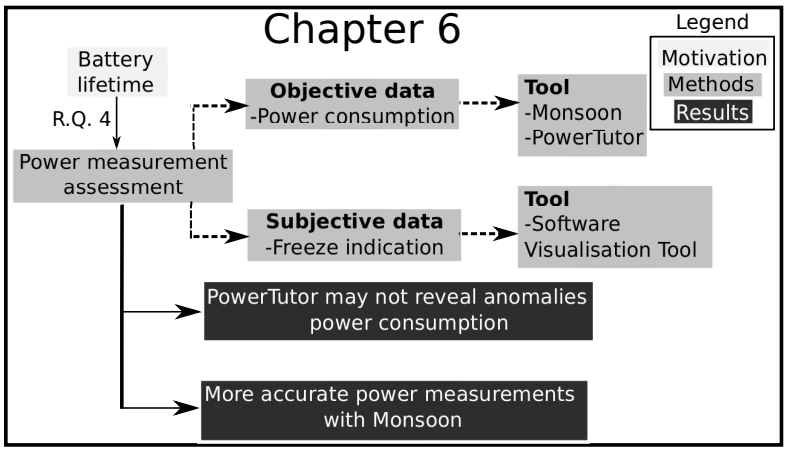


Figure 1.5: Chapter 6 overview: choosing the suitable tool to conduct energy measurements on the smartphone.

fluctuations as an indicative metric, in the design and implementation of QoE assessment tools. If a built-in light-weight battery sensor is available on smartphones, then the energy measurements can be integrated into the QoE assessment framework. We present that, specific measurement settings can reveal the anomalies at the video user interface, *e.g.*, freezes. Therefore, utilizing the power sensors on the smartphones and utilizing them into energy-efficient and rather simple QoE assessment tools is practical. For example, instead of continuous probing on all QoE metrics, power data collected by the sensors can pinpoint the anomalies in the power consumption, and only if potential QoE drop is detected, it can then start the detailed investigation process on the network layer.

In addition, the temporal anomalies in the user interface typically manifest themselves as reduced power consumption due to the adaptive behavior of the system, *i.e.*, releasing the radio data resources, *e.g.*, fast dormancy, when there is no data activity, of the real-time video streaming mechanisms such as in HTTP Live Streaming (HLS). However, during the video streaming, the total power consumption is not reduced completely down to zero, as the LCD display is still ON and the CPU remains active for a particular duration. During this time interval the energy is "wasted", given that the user might wait for the video to resume playing again. This behavior causes the energy consumption to increase with the increased video streaming session duration. Therefore, better management of specific QoS metrics can lead to less interrupted video sessions,

e.g., less stalling events. Therefore, accurate QoS metrics eventually yield less energy consumption due to the reduced time to complete the requested task, *e.g.*, shorter video sessions. Furthermore, the amount of energy saving depends on the user behaviour, *e.g.*, whether the user gives up watching a video after a long freeze, or s/he keeps calm expecting that the video starts streaming back again.

We also study the influence of the freezes on the power consumption with different scenarios. For instance in case of a Transport Control Protocol (TCP)-based video stream, if the network performance is poor, there are often freezes due to re-transmission delays. In this case, the video content is not skipped, thus there is no picture jump. In parallel, if it is a User Datagram Protocol (UDP)-based video, then occasional picture jumps might be experienced as there is no re-transmission mechanism. Thus, in case of a temporal impairment on the video stream and assuming that the user watches the video until the end, the video streaming duration increases if it is TCP-based, but is unchanged if it is a UDP-based video stream. This study enables to gain insight on the energy consumption for different scenarios as well as is helping to develop approaches on leveraging power consumption patterns for QoE management.

We compare the real-time mobile cellular-based video stream and the local-based video stream from the energy consumption perspective. In the case when there are multiple files to be pre-downloaded via the 3G wireless access network, we propose approaches to save energy on the smartphone. We research a use case in which the user downloads multiple large size files, *e.g.*, multimedia podcasts to a smartphone, when there is reliable Internet connectivity, and consumes them later, *e.g.* while traveling on the train. We conducted a study to understand how the downloads should be performed to save the battery. A set of controlled experiments are conducted which comprise simultaneous (*i.e.*, in parallel manner) and sequential (*i.e.*, in serial manner) file downloading consisting of various sizes (*e.g.*, 10 KB to 20 MB). We present that scheduling the downloads save energy on the smartphone, as compared to downloading them individually (one by one) over a period of time. Eventually, a set of suggestions is provided based on different volumes of data activities, and regarding how the data activities should be scheduled. The study on power consumption measurements, and approaches for energy saving during video streaming are summarized in Figure 1.6. The details of the study is given in Chapter 7. Research questions R.Q.3.3, R.Q.3.4, R.Q.3.5, R.Q.3.6, and R.Q.5 are addressed in Chapter 7.

1.4.5 Energy Saving When User is Not Interactive

Video streaming is used actively by smartphone users in real time, and contributes to high portion of energy consumption on a device. However, the energy consumption of a smartphone is also significant when it is not being used actively, *e.g.* when the smart-

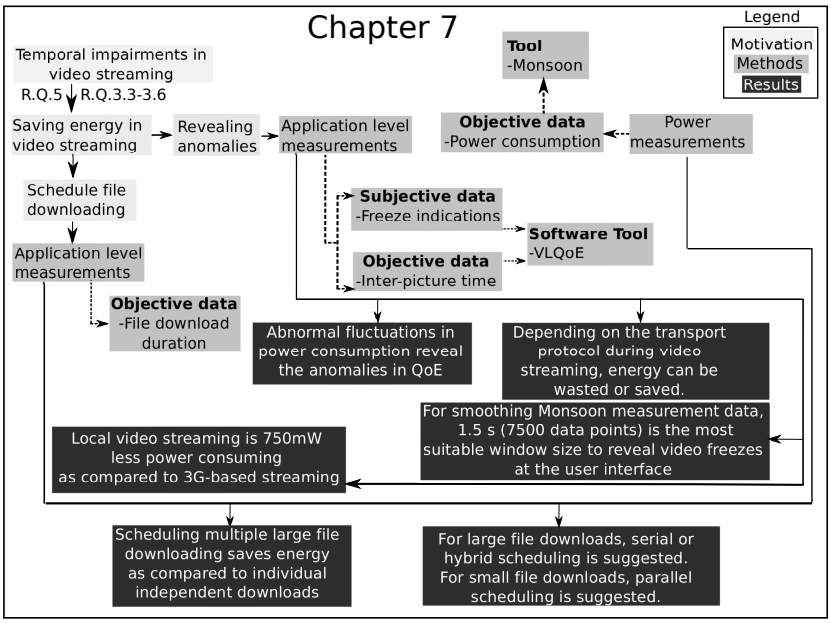


Figure 1.6: Chapter 7 overview: revealing the video QoE anomalies via the power consumption metric and reducing energy consumption for video streaming.

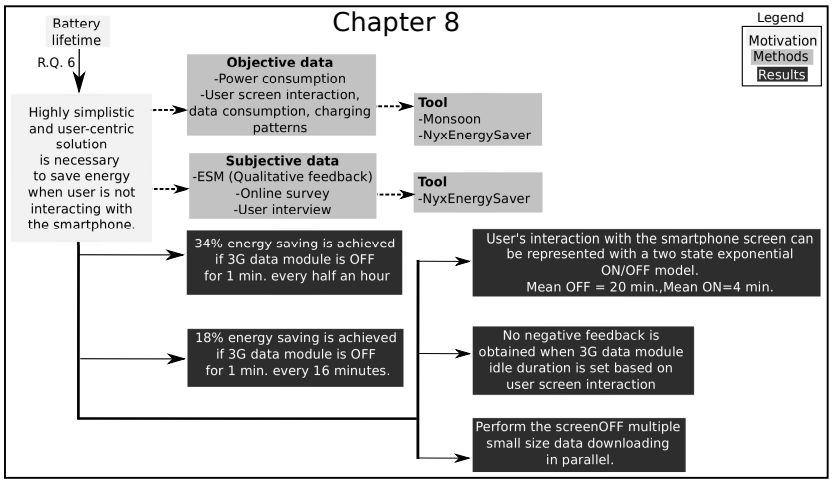


Figure 1.7: Chapter 8 overview: Increasing the smartphone energy saving when there is no user interaction.

phone screen is OFF. Thus, we further synthesize a set of solutions to increase energy savings and indirectly the long-term QoE, while maintaining the short-term QoE. Differently from the previous studies, it has been studied whether the user-smartphone interaction models obtained from the user studies can help to suggest directions for 'intervention', *i.e.*, QoE-based energy-efficiency on the smartphone. The first solution comprises a software solution powered by a smartphone tool that provides context-aware energy saving on the smartphone. One of the most power consuming components on the smartphone is the cellular data module. The smartphones are usually running various cloud-based applications and services that demand Internet access for keeping synchronized with the most up-to-date information. Therefore, the data activities of smartphone applications cause the wireless access network interface, *e.g.* 3G, to oscillate in-between ACTIVE and IDLE states. The cloud-based applications on smartphones execute the data activities independently from each other, thus the number of asynchronous data activities increase with the number of installed/running cloud-based applications. Applications might run advertisements (*e.g.*, banners) or background services to send user-specific data to other apps or websites, which in parallel highly increases the application's network data traffic. Involuntary data activities of a smartphone are also not desired by the users. One reason is that it might lead to undesired high data rates being charged by the operator (assuming that the data plan is different

from 'flat rate', *i.e.*, billing is dependent on the data usage). In addition, there are also some users who do not desire to be notified and interrupted by the application updates while they do not interact with the phone [196]. Therefore, controlling the network data activities on a smartphone when the user is not interacting with them is necessary. However, this needs to be done without degrading the user QoE perceived for the applications. We research a non-intrusive and rather simple energy-saving approach. Our solution consists of a light-weight algorithm (*i.e.*, a user-centric sledgehammer algorithm), which controls the cellular data module (switches it ON/OFF) when the user is not actively interacting with the smartphone. This is done by first studying the statistical user model, *i.e.*, smartphone-user screen interaction patterns (screen state changes on the smartphone), while the smartphone is being used in daily life in the user's natural environments. Having the user's individual models, energy saving can be achieved without degrading the user perceived QoE.

Preventing the data activities by switching OFF the cellular data module and switching it ON at particular intervals might have some drawbacks. Enabling bursty data traffic from all applications and service data traffic is beneficial from the energy saving aspect, but it might add extra latency due to TCP interference and congestion. We discuss this use case by considering the download time and energy measurements obtained in Chapter 7. The measurements obtained while downloading files in parallel or in sequential manner with small file sizes help to understand the consequences (w.r.t. energy and download time perspectives) of bursty traffic caused by multiple but small-sized (*e.g.*, 10 KB to 500 KB) data activities. We concluded that the bursty (*i.e.*, simultaneous) traffic is in fact yielding less download time and energy consumption as compared to the scenario with asynchronous or sequential downloads. The summary of the study is given in Figure 1.7, and the details regarding the study are given in Chapter 8. Research question R.Q.6 is addressed in Chapter 8.

1.4.6 Summary

The complete flow of the thesis is explained, and the superposition of each chapter and study are given in Figure 1.8. There are in total nine chapters in this thesis.

1.5 Research Methods

Various research methods are used in this thesis. We first describe each of the methods in details, and categorize them based on the corresponding sampling rate, the type of collected data (*e.g.*, objective or subjective) and the data collection context, *i.e.*, in a controlled lab, or in a user's natural daily life setting (*i.e.*, "in-the-wild).

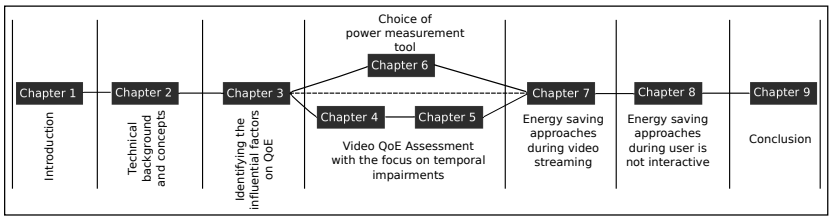


Figure 1.8: Thesis overview: flow of chapters.

1.5.1 In-the-wild Smartphone User Studies

The preferred experimental setting of Human Computer Interaction (HCI) studies is in user's daily life environments, *i.e.*, "in-the-wild", because a controlled lab setting can not fully represent the context of the user (*e.g.*, natural user behavior might change in a lab [22]), and the natural environment (*e.g.*, mobility, location). To acquire the real life data, a software with particular technical configurations (depending on the goal of the study), is employed on users own smartphones for some period of time (*e.g.*, a week, a month, or more). The challenge is the lack of design guidelines for in-the-wild smartphone-based user studies, as we research and present in [251].

1.5.2 Collection of Subjective Data

Qualitative data helps much to understand the context of the collected data [178]. In the sequel, we explain how the subjective data is collected in this thesis.

Online Survey

An online survey enables collecting both subjective quantitative and qualitative data, and identifying the user background and demographics. They are collected only once prior to the start of user studies while the users are in their natural daily life settings (*e.g.*, at home, in the office). Various online surveys are prepared and provided to the participants with different goals in different studies in this thesis.

User Interviews and Day Reconstruction Method (DRM)

User interview is a research technique enabling to collect qualitative data from the users that are involved in the user study. The collected data is often collected via face-to-face interviews in the lab. It typically helps to obtain information regarding the subject

background, occupation, and the use of technology before the start of a new user study. In our studies, in addition to those items, the user opinions on the smartphones and details regarding the previous experiences with the smartphone and the QoE in general are collected. In long and large-scale user studies, the user interviews can also be conducted in a periodic manner, *e.g.*, weekly, during the running study. This helps to validate the collected objective and subjective data, and to confirm it with the users.

During the user interviews, we employed DRM [111] that is a method to help users to provide an opinion on the last 24 hours, while pinpointing the emotions (*e.g.*, delightful or annoying moments) and experiences with respect to the phenomena at each recalled activity, *e.g.*, video streaming on YouTube, video conferencing on Skype, walking in the street with Google Maps [75]. The reason for limiting it to 24 hours is to minimize the memory effect [191] (*e.g.*, serial position effect, primacy/recency effect), or the fading effect [192] (*e.g.*, user might over- or under-estimate the effects or emotions in a long-term).

Experience Sampling Method (ESM)

Traditional QoE assessments are often performed in the form of post-view subjective scores. In other words, for a video QoE assessment for instance, the users are asked to watch a particular video and then rate the overall perceived quality after the video ends. However, for long duration videos, this kind of post-view based assessment may not capture the instantaneous (momentary) quality changes (but rather averaged results), which eventually does not capture critical impairments or fluctuations. In order to capture the in-situ experiences in a more frequent manner, another method called ESM is used. ESM is a research method that is used to assess phenomena at the time they occur, from the human perspective, in order to maximize the validity of the data [65] [66]. Lutille *et al.* in [65] claimed that the best time to ask the user about the preferences and/or the user needs is in the midst of the actual activity being closely inspected. This method can be used to identify the most critical moments for the QoE user experiences. This helps to identify the end-user perceived quality on spot in ad-hoc manner, while delivering the service to the user, and to relate it to factors influencing it. The collection of data via software-based questionnaire is performed in user's natural daily environment (in-the-wild) approximately no more than ten times a day, each lasting no more than two minutes [20]. The collected data is of highly subjective nature.

Mean Opinion Score (MOS)

In evaluation of subjective QoE factors, the user perceptions are represented by numerical values. Table 1.1 depicts a single stimulus Absolute Category Rating (ACR) method with five-grade Mean Opinion Score (MOS) scale [30]. The user-perceived quality is translated into numbers ranging from 1 to 5. Although some drawbacks exist for MOS (*e.g.*, smoothing/averaging out the ratings given by the individual user), it is commonly used in subjective assessment of the user-perceived quality.

Table 1.1: ITU-T scale of media quality impairment, also referred to as User Rating (UR).

Scale	5	4	3	2	1
Quality	Excellent	Good	Fair	Poor	Bad

1.5.3 Collection of Objective Data

Autonomous Logging

Various autonomous software logger tools have been developed and used to address the research questions of this thesis. The common characteristics of the tools are that they collect objective data from the network, application, and available sensors on smartphones. The sampling rate is varying in-between the tools, typically in the order of minutes. The developed tools are listed as QoE-CSS (Chapter 3), PERIMETER measurement subsystem (Chapter 4), VLQoE (Chapter 5), DownloadTrainCatcher and IOVidEoQ (Chapter 7), and NyxEnergySaver (Chapter 8). The data is collected both in-the-wild and in the lab. The tools contributed to different studies, and they will be described in more details in corresponding chapters. The user studies with PERIMETER, VLQoE, DownloadTrainCatcher, and IOVidEoQ tools are conducted on the same unique smartphone (as some functionalities necessitate root access); and the studies conducted with QoE-CSS and NyxEnergySaver are employed on users' own smartphones.

Power Measurements

We have used Monsoon power tool [170] for all energy measurements in various studies in this thesis. The Monsoon power monitor device contains the power monitor

hardware and the power tool software, running on Windows XP and Seven, which can provide robust measurements on any device that uses a single Lithium (Li) battery. The measurements are obtained and can be saved with a sampling rate of 5 kHz. The tool supplies the power to the device, thus the device battery is bypassed. The Monsoon external power-monitoring device is typically used for ground-truth measurements [170]. Further comparison between an internal software tool PowerTutor and the external Monsoon power monitoring tool are presented in Chapter 6, in order to motivate the choice of Monsoon power monitoring tool.

In Fig. 1.9, the research methods used in this thesis are categorized with respect to their usage for objective or subjective data collection as well as the periodicity of data collection.

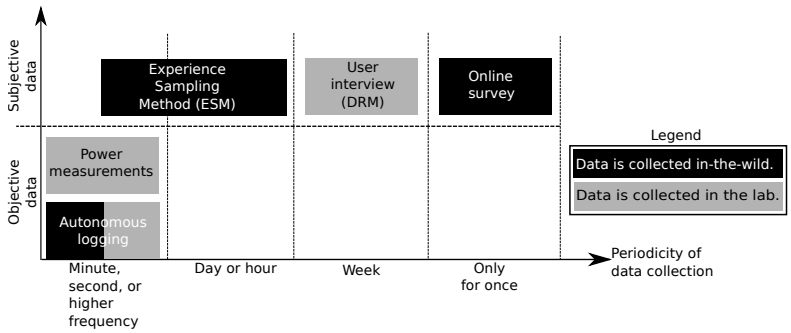


Figure 1.9: Research methods used in this thesis

1.6 Main Contributions

There are in total eight main contributions in this thesis, and they are itemized below. The research questions that each contribution addresses are given.

- **C.1:** For the first time, we thoroughly explore the area of smartphone QoE in relation to the energy consumption. This contribution addresses R.Q.1 and R.Q.2,
- **C.2:** We present a novel methodology to assess user-perceived QoE in-the-wild from the applications on the smartphone. This contribution addresses R.Q.1,
- **C.3:** For the first time, we show through qualitative research that the energy consumption on smartphones is one of the most influential factors on QoE. This contribution addresses R.Q.2,

- **C.4:** We developed new tools, focusing on different layers of the network stack, to assess video QoE on the smartphone in-the-wild. This contribution addresses R.Q.3,
- **C.5:** We perform extensive QoE and energy measurements both on users smartphones in-the-wild as well as controlled measurements in the lab, and provide empirical data. This contribution addresses all of the research questions,
- **C.6:** We study models of user QoE based on user interaction, and performance of application and network. This contribution addresses R.Q.2, R.Q.3, R.Q.5, and R.Q.6,
- **C.7:** In contrast to the complex technical solutions in the literature, we provide rather simple user-centric algorithms for saving energy on smartphones. This contribution addresses R.Q.6,
- **C.8:** We show that the energy saving on the smartphone is possible with rather simple technical solutions while maintaining the user QoE via leveraging our models. This contribution addresses R.Q.5 and R.Q.6.

1.7 Thesis Outline

This Ph.D. thesis is organized in the form of a monograph based on a set of peer-reviewed and/or published papers since 2010. This thesis is structured as follows. In Chapter 2, the concepts such as QoE assessment methods, video streaming, and existing energy consumption measurement methods are introduced. Comprehensive technical background information is given concerning the above items. In Chapter 3, a set of most important influential factors on QoE on Android OS smartphone are given in details. In Chapter 4, the study to identify the influence of network-level metrics such as packet delay variation on the smartphone video QoE, is presented. Another user study on smartphone-based video QoE with the focus on metrics captured in the user interface is given in Chapter 5. In Chapter 6, the advantages and shortcomings of a set of software-based and hardware-based power measurement tools for smartphones are presented. The accurate choice of a power tool later helps to reveal the anomalies in QoE (*e.g.*, video freezes and corresponding low user ratings) with the anomalies in the power measurements. In Chapter 7, the results from the power consumption for various video streaming scenarios are presented. Extensive experiments are conducted both for real-time video streaming and local-based video offline streaming, *i.e.*, "download-and-watch-later". The relation in-between the QoE and the energy consumption on the

smartphone are discussed both from the traditional tradeoff and also from the win-win aspects. Some approaches to save energy during video streaming, when the smartphone is actively used by user, is proposed. In Chapter 8, we present approaches to save energy when the smartphone is not actively used, *e.g.*, when smartphone screen is OFF. An easy-to-deploy and rather simple algorithm is presented enabling to save energy on smartphone without degrading user perceived quality while the smartphone screen is OFF. We also confirmed with the results from Chapter 7, that the scheduling small-size application network data activities in parallel, and forcing them to be performed within bursty traffic at a later time does not negatively impact the download duration during the bursty intervals. In Chapter 9, the thesis is concluded by summarizing the main contributions, and presenting the future directions.

Chapter 2

Concepts and Technical Background

"*In theory, theory and practice are the same. In practice, they are not.*"–Albert Einstein

---

2.1 Introduction

In this chapter, the concepts and the technical background are given related to two core items: *Quality of Experience (QoE)* and *energy consumption* on smartphones. We briefly discuss mobile devices and applications running on them in Section 2.2. We present the QoS and QoE definitions in Section 2.3. Section 2.4 focuses on video streaming, and the application-layer protocols, together with the influential metrics on the video QoE. The related work on the temporal impairments in video streaming are given in Section 2.5. The state-of-the-art on the standardization in the area is given in Section 2.6. The existing QoE measurement methodologies are presented in Section 2.7. Some related work on the existing QoE video management mechanisms are given in Section 2.8. In Section 2.9, we explain energy consumption on smartphone, review a set of state-of-the-art energy/performance tradeoffs, and then elaborate on the energy consumption's crucial role on the QoE. In Section 2.10, we review the existing smartphone energy measurement tools, and also present related work on energy-saving approaches on the smartphone. In Section 2.11, we review the related work on the user

interaction with the smartphone screen, as well as the influence of the energy consumption on the user behaviour. Section 2.12 summarizes the chapter.

2.2 Mobile Devices and Applications

2.2.1 Network, Application, and Energy Perspectives

Mobile devices are different from the fixed PCs and laptops in many aspects. They have limited screen size and the right content (minimum and most important information rather than the full content, *e.g.*, as a thumbnail) has to be presented to the user at the right time without high delay [13]. Context is defined as *"any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves"* [14]. The mobile devices are used in diverse contexts and locations, thus the user may be highly interrupted by external events, and sometimes it is necessary that the content and the task at hand have to be completed as quickly as possible before the user switches her attention or before her context switches.

Mobile device characteristics are many, therefore including their location awareness, high dependency on network connectivity, its performance, limited device capabilities such as battery, and support for a wide variety of user interfaces and platforms [19]. There are many types of mobile devices in the market, thus the variation of hardware components and characteristics including screen size, sensor type and resolutions, user input method (*e.g.*, non-keyboard gestures, swype, touch, or pinch), might be a challenge for testing [13], *i.e.*, adapting the application to any mobile device. Thus, the diverse characteristics of mobile devices make the mobile application testing complex and difficult to perform. Additionally, the costs of the dataplan (associated with a subscription) of smartphones can influence the user QoE, thus the data consumption of devices need to be evaluated in detail. There is variable network throughput which might highly influence the network-based application performance, *e.g.*, real-time video streaming apps, browsers, thus, testing efforts need to consider corresponding solutions to maintain user QoE in such undesired situations. Availability of network is another common problem as there might be locations with limited mobile connectivity, or for some reason a person may not have a cellular data plan and find her/himself in a place without a WiFi access. Furthermore, the apps should also be tested along with the functionality of other existing apps in a mobile device. For example, what happens when a user is using an application, and she receives a call, or a text message? How is the application being handled?

Another challenge could be an operating system being open-source, thus being less costly, or enabling access to low-level objects in the operating system. Identifying the root causes of malfunctions and bugs in mobile apps can sometimes be very difficult, thus considerable efforts must be put to replicate the exact scenarios that cause these bugs. Such examples include *e.g.*, pinpointing the precise root cause for privacy or security violations. For example, recent studies of the Android and iPhone platforms discovered that some apps transmit user data undesirably to third-party advertisement servers. This behaviour has also drawbacks in terms of usage of smartphone resources such as bandwidth and energy [17]. In [62], it is stated that it is indeed challenging to test the correctness of a software on mobile devices. Testing all combinations of software and hardware, which is referred to as combinatorial explosion in [63], is described as impractical.

Additionally, one of the most common challenges is to represent the user's experience in real-world as the applications are often developed in labs or offices. There is often a lack of context-aware testing tools that handle context-aware features. There exists also a tradeoff between the accuracy and the scalability of mobile applications. On one hand, it is required for apps to be scalable to any incremental features that are able to adapt to any environment, and functional premises. On the other hand, it is hard to test the accuracy of all enhanced components of the application after each adaptation.

2.2.2 Infrastructure Perspective

Mobile applications are being developed at various points in the whole system including front-end devices and also on the backend servers. Thus, the infrastructure of the system, *i.e.*, the communication protocol between the server and the application or the application server capacity, might also play a crucial role on the overall application performance. A holistic approach on the development and comprehensive tests that involve both front-end, back-end and intermediate interfaces might be necessary to be performed before releasing network-based applications. The lack of complete mobile application quality testing on different aspects (due to the aforementioned challenges) results in poorly perceived application performance.

One of the acceptable and accurate ways of testing application performance is by assessing the user experience via collecting direct feedback from the users. However, the challenge is that the user experience is highly subjective, thus to obtain a general QoE model that fits every user is difficult.

As can be seen, the mobile application quality spans over many domains. We discuss only a set of smartphone-based issues related to the network, application, and energy in this thesis, which might be beneficial both for operators and application de-

velopers to consider to improve perceived quality of applications and services provided to users.

2.3 QoS and QoE Definitions

QoS is the ability of a network to provide an assured service level with the focus on parameters that exist in network and application level [68], [239]. QoS's main objective is to prioritize particular data packets on the network, so that each service quality meets the expectations depending on the needs and Service Level Agreement (SLA). The diagnosis is often performed based on network-level measurements with technical metrics such as bandwidth, packet loss, packet delay, or packet delay variation. The ITU-T recommendation E.800 [24] defines QoS as "the collective effect of service performance, which determines the degree of satisfaction of a user of the service". This definition assumes that the user-perceived quality is increasing with the better underlying technical performance. It is also limited with "acceptance of a service", which does not go beyond such as "delightfulness" or "annoyance". For example, probing and relying on individual technical measures, *e.g.*, the Key Performance Indicators (KPI), can quantify performance degradation on the network level, however, they do not fully describe the user-perceived quality. European Telecommunication Standards Institute (ETSI) defines the QoS challenge as "meeting customer's quality expectations, and providing availability and reliability, while maintaining the network operator's flexibility to new technology" [118].

As seen from the network perspective, the implication and the importance of QoE have evolved over the years from end-to-end QoS, since QoS alone was not powerful enough to express the perceived quality in a communication service [15]. Therefore, the user-centric QoE has overtaken the role of network-centric QoS by increasing the importance of user-perceived quality instead of technical performance of the overall service. Mobile Network Operators (MNO)'s have already begun to conduct QoE studies to diagnose the potential problems in advance and to minimize the rejected communication service products by customers in the market [68], [241].

In Figure 2.1, the difference between the QoS and QoE is sketched. QoE depends on many factors related to user, technical, context, as well as business and economical perspectives, while the QoS only focuses on the network perspective. In the evaluation of QoE, technical groups focus on network and service performance based on QoS models; business groups focus on the revenue, cost, and customer churn rate based on economical models; and the social scientists emphasize QoE as happiness, and relate it to experiences [21]. These studies in different domains are highly inter-related with each other.

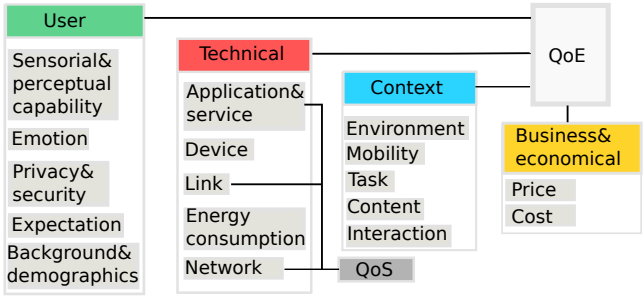


Figure 2.1: QoE is a function of many potential factors.

There are a set of definitions of QoE. In the DSL forum in 2006, QoE is defined as "a measure of end-to-end performance at the services level from the user perspective and an indication of how well the system meets the users needs." [250]. The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) defines QoE with the focus on user as "The overall acceptability of an application or service, as perceived subjectively by the end-user" [16]. In [23], QoE is described as "a measure of user performance based on objective and subjective psychological measures of using a service or product". Most recently in the Qualinet (COST Action IC 1003) white paper [15], a successor QoE definition has been stated, " The degree of delight or annoyance of the user of an application or a service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user's personality and current state". Despite all the aforementioned definitions of QoE, it is claimed that there is still a lack of a solid theoretical and practical QoE framework [15].

According to Kilkki [21], the quality ecosystem model for QoE consists of end users with diverse roles. The ecosystem is identified where the person is categorized as customer, user, or a group member. A holistic framework in [28] proposes that the network efficiency, user acceptance, and the technical perspectives are corners of a QoE triangle. Khalil *et al.* [25] describes the QoE model in a communication ecosystem that can adapt to many specific contexts and integrated cross-domains such as technical aspects, business models, and human behaviour into one framework. The ecosystem is defined as the systematic interaction of human, technology, and business in particular context. The QoE definition includes all human subjective and objective quality expectations and experiences during interaction of a person with technology and with business entities.

The main characteristics of QoE is that it is subjective, user-centric, holistic, and multi-dimensional. Therefore, it is highly susceptible to the influential factors such as user context, location, content, together with the technical QoS metrics. The definition of *influence factor on QoE* is stated in [15] as "any characteristic of a user, system, service, application, or context whose actual state or setting that may have influence on the QoE of the user". It can be grouped into Human Influential Factor (IF), System IF, and Context IF, and they are studied in details in this thesis.

In modeling the usage of pervasive technology and the user's QoE, Jaroucheh *et al.* [103] have emphasized the importance of considering the historical and the current user context and the flexibility of user behaviour. Similarly, according to Hassenzahl *et al.* [105], some influential factors on the user experience are stated such as the user's internal state, the characteristics of the designed system, the context within which the interaction occurs, and the meaningfulness of the activity. Based on the survey results, Korhonen *et al.* [104] concluded that the mobile device, task at hand, and social context are the most influential factors of a user's QoE. Park *et al.* [106] indicated usability and usefulness; Shin *et al.* [107] added enjoyment and network access quality as the factors influencing QoE.

2.4 Video Streaming

In this section, we emphasize on the network-based video streaming application. First, the IP packets with the media content are downloaded via the network link from the destination. Then the downloaded packets are processed, rendered, and presented as the video content to the user on the device display. The processes can be performed either sequentially or in parallel based on the type of video streaming, *e.g.*, progressive downloading, download-and-watch-later, real-time streaming. During this process, there exists signaling traffic between the source and the consumer video player in various levels at the network stack. The video download patterns depend on, amongst many other factors, the link type, *e.g.*, 3rd Generation (3G) or Wireless Fidelity (WiFi), the smartphone type, the video player, and the application-layer protocol.

2.4.1 Application-layer Protocols

Video clips on mobile terminals are often streamed via two main application layer protocols: Real Time Streaming Protocol (RTSP) [122] or Hypertext Transfer Protocol (HTTP) [123]. The RTSP video streaming is mainly based on the unreliable UDP transport protocol. The data packets are sent from the server to the smartphone as they are generated by the video/audio codec [124]. The flow control is done by the RTSP

in the application layer with periodic control messages being sent from the client to the streaming server. RTSP is implemented for live multimedia streaming. On the other hand, the HTTP video stream is based on the reliable TCP transport protocol and the packets arrive at the client (*e.g.*, the sink) in segments. The size of the segments depends on the available bandwidth and vary with the window size of the TCP packets, which in turn influenced by RTT and Bandwidth Delay Product (BDP) [125]. HTTP streaming is used for multimedia streaming and it commonly works based on progressive download, *i.e.*, the video is downloaded from a webserver and temporarily stored at the cache of end-user device and then displayed on the screen. The characteristics of each protocol with respect to the network layer, *e.g.*, burst size and burst duration, are hypothesized as different, and this difference is expected to influence the application performance, and eventually the end-user perceived QoE.

2.4.2 Video Streaming Application Mechanism

A typical sketch that summarizes the network-based video streaming mechanism is given in Fig. 2.2. In the figure, a constant video playout buffer is assumed. There is a

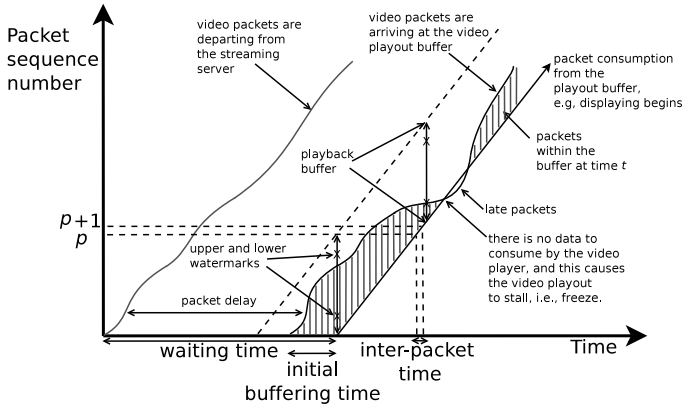


Figure 2.2: Illustration of the video playout and variables.

delay in-between when the video packets are departed from the source (video server), and when they are received over the network on the other end (player). There is also an extra delay between the times when the packet is arrived to the receiving end (*i.e.*, video player or consumer), and the time when the packet is consumed by the video player (*i.e.*, rendered and displayed on the video screen). In the figure, a high jitter

is depicted, (*i.e.*, deviation of the delay between the time when the packets are sent from the server and received at the video player). This causes the content of the video playback buffer to starve, and buffer to be emptied (in the case of very high delay). The latter case causes picture freezes. There are low and high water marks that indicate the status of the video playout buffer, which is used in adaptation of the link throughput based on the video playout in order to prevent buffer over-/under flow in advance. For example, when the buffer gets highly overloaded in RTSP-based video streaming, the video player might send back to the server a signal indicating to reduce the sending rate of packets. This also holds for the opposite case; when the buffer content becomes too low, it sends signal to the server to speed up sending video packets.

2.5 Temporal Impairments in Video Streaming

The video quality perceived by the end user is highly susceptible to the discontinuity such as stalling events (or a freeze) in the video. A freeze is highly depending on the human perception [114], and is recognized when the next picture in the video is not displayed within the duration expected by the user. This highly depends on metrics in the network layer including the throughput (*i.e.*, received bits per second), as well as on crucial factors in the application such as the video playout buffer. If the size of the video playout buffer is too small, then it is highly probable that the buffer will be emptied (*i.e.*, the video packets will be consumed by the video player), which eventually causes a freeze in the playout. Thus, large video playout buffers help to reduce the frequency of discontinuities (provide smooth video playout), on the other hand they are costly (in terms of space and computation) and also not applicable to real-time streaming. This is simply because there will be extra buffer delay while presenting the video content to the user. According to Bang *et al.* [129], TCP generally provides good streaming performance, when the achievable TCP throughput is roughly twice the media bitrate. However, it is also important to quantify a "good performance" from the user perspective based on measurement layers of the network stack.

There are multiple performance indicators to consider while assessing the QoE of a particular service. The influential factors for the QoE of the video streaming application in Internet are categorized in four categories: context (environment, social/cultural background, purpose of using the service), user (expectations of user, memory and recency effects, usage history), system (transmission network, devices, screens, video buffering strategies), and content level (video codec, format, resolution, duration, content, type, motion pattern) [126] [128].

The frame rate, *i.e.*, the periodicity of the displayed video pictures on the screen, and the network bandwidth are important influential factors on the video QoE [130]

[131]. Yang *et al.* [132] and Quan *et al.* [133] studied the frame rate metric, *i.e.*, the number of displayed pictures per second, to quantify jerkiness during video streaming. The results are obtained via fixed computer-based testbed within standardized lab environments. It is clear from the results that the MOS increases with the frame rate. An exponential model is suggested in-between the jerkiness of a video and the frame rate per second. Hossfeld *et al.* [115] studied the influence of a stalling event, namely *picture freeze*, on the end-user perceived QoE by user opinion score ratings on the five-level ACR scale, yielding MOS, while considering the freeze duration and the number of their occurrences. The authors concluded that the stalling of video stream impacts the video experience of the user independent of the actual video characteristics [115]. Another outcome of the same study was that based on the exponential QoE mapping functions to the stalling frequency and the durations, one freeze with two seconds duration within a 30 s-long video clip maps to a MOS value 3 in desktop settings. Despite, further studies on smartphone settings and with longer video clips without any restrictions of video length need to be studied to obtain more reliable models. In addition, the distribution of the freezes in a video is often not uniform, thus a varying number of freezes with varying lengths would yield a much more realistic scenario. Van Kester *et al.* [135] have investigated the average end-user perceived QoE (via five-level MOS) on 20 seconds long video sources based on the number of freezes and concluded that the acceptable freeze duration (MOS > 3.5) is 360 milliseconds, and they noted that the frequency of the freezes must also be taken into consideration in QoE assessment. In addition, no significant difference in the video QoE was observed between when the video is stopped with or without skipped frames. Zinner *et al.* [139] stated that the Structural Similarity Index Model (SSIM) can be used to quantify the influence of packet loss on the QoE; while the Peak Signal-to-Noise Ratio (PSNR) can not. Furthermore, by using SSIM, the authors also showed that a low-resolution video with smooth playout obtain high QoE as compared to a high resolution video with disturbed playout showing the very high influence of stalling events on the QoE, related to the "Provisioning-Delivery Hysteresis" [140].

One other important influential factor on the end-user perceived video quality related to temporal impairment is the initial waiting time [201], which is defined as the time gap between when the request is issued and when the request is sufficiently visible to the user. In [49], a logarithmic relationship between the QoS and QoE has been presented. The characteristics of the logarithmic function is that if the stimuli is increased by a factor of k, then its influence on the function is additive. Thus, the influence in the change of impairment on the MOS is highly depending on the current state of the impairment. The number and the duration of delays, for web browsing and HTTP video streaming have been studied in [136], and the authors have presented a logarithmic relationship between the waiting time and the QoE. The authors stated that

the longer the waiting time, the less satisfied the user becomes. The location of waiting time, *e.g.*, before/after the video has started displaying, is important as such. Hossfeld *et al.* [126] showed that initial delays before the video has started has no severe impact on QoE.

In this thesis, the user experiments are conducted via video streaming sessions on the user devices. As there is no particular and ground-truth methodology to assess real-time video QoE on a smartphone, we develop our own tools and methodology and conduct comprehensive user studies.

2.6 Standardisation Groups in Video QoE

There are also organizations such as the Video Quality Expert Group (VQEG) [43] [44] that bring together the experts from industry, academia, government organizations, *e.g.*, the International Telecommunication Union (ITU), and Standard Developing Organizations (SDO), with the goal to advance the field of video quality of television and multimedia applications. VQEG not only performs subjective video quality experiments, but also validates the objective video quality models and develops new techniques. Eventually, detailed test plans for evaluating QoE in a subjective way are provided by this organization. ITU involves in itself study groups such as SG9, SG12, and the Joint Rapporteur's Group on Multimedia Quality Assessment (JRQ-MMQA) that studied the QoE requirements and QoE assessment methods for multimedia services [283]. Again within the ITU, the Focus Group on IPTV (FG-IPTV) studies the relationship between QoS and QoE with the focus on QoE requirements, traffic management mechanisms, application layer reliability solutions, and performance monitoring for IPTV services.

2.7 Video QoE Assessment

This thesis focuses mainly on the video streaming application for the smartphone; thus, we present a set of state-of-the-art objective and subject assessment methods in Sections 2.7.1 and 2.7.2, respectively. The summary of all methods and standards used in QoE assessments are summarized in Table 2.1. In the table, the QoE assessment methods are classified based on their characteristics such as being "objective" or "subjective" (as presented in the first column) as well as their stack level (as presented in the second column). The parameters that are being studied in each class are presented in the third column of the table. The corresponding methods and the standards are given together with the references in the fourth column of the table. Additionally, we discuss

the challenges in subjective QoE scales in 2.7.3.

Table 2.1: VIDEO QOE ASSESSMENT METHODS

Type	Stack Level	Parameters	Methods & Standards
Objective	User	brainwaves, heart rate, nerve act., eye tracking, skin conductance, camera, engagement	[32–35, 38] [27, 109, 280]
	Media	image quality level	(MOVIE, PEVQ, MPQM, PSNR,VIF, SSIM, VSNR, VQM) [29, 48, 51, 147] [52, 53], ITU-T (P.862,J144,J.MM,J.VQHDTV), ITU-R (BS1387, J.148)
	Packet	packet-loss, bitrate	(A-PSQA, YoMo, Pytomo) [137, 142, 143]
		buffering events, init. delay	ITU-T P.564, ITU-T P.NAMS
	Bitstream	network (*e.g.*, packet header and encoded payload), application (*e.g.*, codec type, video format), terminal (*e.g.*, monitor size)	E-Model [31] ITU-T (P. NBAMS, G.1070)
	Hybrid	all or combination of above	
Subjective	User	MOS, SOS	(DSISM, DSCQS, SSCQE, MLDS, crowdsourcing) ITU-R BT.500, P.910, ITU P.800 [30], [156] [57] [199], [64] [138]

2.7.1 Objective QoE Assessment in Video Streaming

The objective assessment of QoE in video streaming comprises specific tools for monitoring physiological signs like via brain waves [32, 33], heart rate [34], nerve activity, eye tracking [35], and skin conductivity [38]. The tools are classified based on their intrusiveness, response time, and sensitivity level. According to [27], EMG (Electromyogram), which measures nerve activity, is relatively more sensitive to stimuli as compared to skin temperature; on the other hand it is known to be more intrusive to the user. Reichl *et al.* evaluated QoE in real user environments by capturing user interaction with the smartphone, and the user context via the help of two cameras mounted on the user's hat [109]. Although this method enables capturing realistic data, it might in parallel become highly intrusive to the user carrying the hat. Although the objective measurements might be costly from the economic perspective, due to the use of expensive devices, they are considered to be less time-consuming as compared to the subjective assessment methods.

There are other objective quality assessment metrics such as user engagement, which is based on for example the viewing time or the number of views of a video [284]. These metrics alone can not indicate accurately QoE as those are influenced by many factors including the user's mood, or the popularity of video content [279] [280].

According to [165], the objective methods are categorized with the criteria such as the targeted service, *e.g.*, IPTV, video conferencing, web browsing; model type *e.g.*, whether there is a reference signal or not; application, *e.g.*, network planning, monitoring; model input *e.g.*, reference signal; model output, *e.g.*, MOS; and modeling approach, *i.e.*, modeling human perception system or obtaining the system characteristics via experiments. The categories are as follows [283]:

1. **Media-layer models:** Rely on the video signal to predict QoE. They are fur-

ther categorized into Full-Reference (FR), Reduced-Reference (RR), and No-Reference (NR) based on the availability of the original source to make further comparisons between quality levels. FR metrics are suitable for measuring offline video quality, *e.g.*, Perceptual Evaluation of Speech Quality (PESQ) [29]), and NR and RR metrics are used for in-service video prediction [41, 42], *e.g.*, online monitoring. Rigorous user methodology is necessary to assess the quality in the case of NR metrics, *e.g.*, training sessions, randomizing quality levels.

Motion Based Video Integrity (MOVIE) [45] evaluates dynamic video fidelity and integrates combination of spatial and temporal aspects of distortion assessment. Moving Picture Quality Metric (MPQM) [47] calculates the subjective quality metric based on the inputs, namely the original video sequence and its distorted version. PSNR [48] is a similar measure to probe the noise level of a compressed image based on the original image, and it is a fidelity metric to evaluate the spatial quality of a video. It also disregards the viewing conditions and the characteristics of human visual perception [40, 42]. Migliorini *et al.* [147] presented QoE results based on the experiments done with a packet-based simulator with the focus on PSNR and channel conditions. Visual Information Fidelity (VIF) [51] is a measure for assessing video quality, however it requires stochastic models for the source, distortion and the Human Visual System (HVS) [246]. SSIM index [52] is a method for measuring the similarity of the image quality while taking the initial distortion-free image as reference. However, these methods are machine-measurable metrics and it takes few seconds for SSIM to compute the quality of a video. V-factor is another video quality rating score based on the content format and processing types such as compression type, Group of Pictures (GOP), and quantize levels.

There are more objective quality assessment methods such as VSNR and VQM [55], however all the above-mentioned assessment techniques are not suitable for real-time video quality assessment in mobile platforms, as they necessitate the original version of the videos for comparison, *i.e.*, the original media signal. In addition to the storage constraint, it will also cause extra computation overhead on the mobile devices, which eventually might affect the energy consumption. Thus, such methods are of limited practicability in a mobile environment. Perceived Evaluation of Video Quality (PEVQ) [53], a full reference algorithm, computes MOS scores of the video quality for fixed and mobile video and considers the degraded video signal output from the network. It estimates end-user perceived quality of pre-recorded videos, *i.e.*, MOS, with respect to the spatial impairments. However that tool fails to deliver good estimations in case of temporal impairments such as picture freezes [148]. Some existing standards

within this model category are ITU-T P.862 (for speech), ITU-R BS1387 (for audio), ITU-T J144 (for video), ITU-T J.VQHDTV (for video), ITU-T J.MM (for video), and ITU-R J.148 (for multimedia).

2. **Parametric packet-layer models:** Assessment of QoE based on the packet-header information, *i.e.*, when no content or payload is taken into account, has less computational load. It is used for assessing the impact of packet-loss, which is a network-level metric. In [137], a no-reference QoE assessment measure, A-PSQA, is proposed as a performance evaluation method, based on the analysis with respect to the packet-loss percentage. Bitrate, buffering events, and initial delay metrics can be categorized within this group. There are a set of related works on the packet-layer models. PC-based based measurements on the VLC player were conducted in [141], where the video/audio buffer utilization based on the Real-time Transport Protocol (RTP) and UDP streaming is studied. The drawback of that study is that it does not consider the picture freezes and the corresponding end-user perceived quality. YoMo [142] is a Java tool with a Firefox plugin and it estimates the video player buffer status on application layer. Pytomo [143] is a Python-based client-level tool that analyzes the playback performance of particular video sources on YouTube. It obtains the cache Uniform Resource Locator (URL) and Internet Protocol (IP) addresses of the hosting video server, ping/download/playback statistics, initial buffer, stalling event, and buffering duration of the selected videos. Despite its great contribution on the objectively measured video quality, it lacks detailed definition on how the end-user perceived quality is assessed. The existing standards are ITU-T P.564 (for speech) and ITU-T P.NAMS (for video) that use only packet header information to estimate audio, video and audio visual quality.

3. **Parametric planning models:** Imply use of the quality planning parameters for networks, terminals, and environmental factors. Some existing standards include ITU-T G.107 (for speech), where the E-Model is used for VoIP based services [31], and it is defined as a "computational model for assessing the combined effects of variations in several transmission parameters". However, from our experience, we know that the poor performance of a few important parameters is enough to obtain poor QoE, thus not necessarily the combined effect of all parameters is necessary to predict QoE. Other standards in this group focused on the video such as ITU-T G.1070, and in this recommendation multimedia quality is predicted via network, *e.g.*, packet loss pattern, application (*e.g.*, codec type, video format) and terminal equipment parameters, *e.g.*, monitor size.

4. **Bit stream layer models:** Use the encoded payload bit stream data and packet

header information. It estimates the characteristics of the missing data via the bit-stream information in the preceding and the succeeding packets, thus the assessment depends on the data content. ITU-T P.NBAMS is a related standard that aims to predict audio, video, and multimedia quality from the IP stream. Bit stream layer models are more accurate, as compared to parametric layer models, to estimate quality at the cost of computation power as they extract partially the payload information.

5. **Hybrid models:** Are made of combinations of the above models.

Objective QoE assessment can assess the quality without the existence of actual users, thus it can be less time consuming as compared to the subjective assessment. On the other hand, objective assessment methods are known to be complex, and if the conditions are not representative for the user's real context then the validity of the results are highly impacted [155], [156]. As the objective measurements alone might not cover the whole dimensionality of the phenomena, subjective assessment methods (in which the users are actually involved in the assessment) are necessary to complement and to validate the objective measurements.

2.7.2 Subjective QoE Assessment in Video Streaming

The subjective assessment evaluation is performed by presenting particular stimuli to human subjects and then collecting user feedback based on the corresponding user-perceived quality. In the video streaming, stimuli is typically either a spatial or a temporal impairment in the video playout. This assessment needs careful planning, and it comprises a multidisciplinary approach, including human cognitive aspects. ITU-R BT.500 [55] and ITU-T P.910 [156] are recommendations that detail the test settings, and rating scales to be used in the controlled-environment tests. The most common way of representing the user perceptions are by numerical values. The five-grade scale is typically used for Opinion Score (OS) ratings, *i.e.*, for Mean Opinion Score (MOS). Other methods defined by ITU P.800 are double stimulus impairment scale method (DSISM), threshold definition (TD), and pair comparison (PC). ITU-R and ITU-T [156] are various standards with regards to viewing conditions, criteria for selection of observers, test equipment, procedures to follow during and after the data collection for subjective quality assessment. Next, further commonly used subjective methods are discussed.

Double Stimulus Continuous Quality Scale (DSCQS) is a test that involves screening of degraded and original versions of short video clips that are shown to the user in random order. The users are asked to rate the quality of the video based on a five-grade quality scale which is later normalized into a 1(Bad)−100(Excellent) scale. The

difference in the assessment values are calculated, and then averaged on all subjects to obtain the DSCQS value. This method is used when not all samples for each quality are available to show to the user. The participants are notified about the order of the reference and degraded versions of video sequences. The video sequences are shown only once to the participants.

Single Stimulus Continuous Quality Evaluation (SSCQE) is another test recommended for long duration, *i.e.*, $20-30$ minutes long, video sequences with processed quality. The video clips used in this method do not need to be standard. The participants rate the user-perceived video quality continuously on an ACR scale. It is a single stimulus method, where the user is asked to rate the video quality based on the processed video sequence on a score recording device, for predefined conditions such as the record-sampling period and the position of the slider.

More methods exist for video QoE assessment in addition to the above suggested ones for in-lab measurements with different trade-offs. Crowdsourcing is one approach [56] that moves the testing effort from the controlled lab environment into Internet [57]. The user feedback and the ratings are collected while the users experience a given task by using their own computers or mobile handheld devices that are connected to the Internet anytime and anywhere. In return, the users earn money or some kind of tokens by completing small tasks and providing corresponding feedback based on their perceived QoE. Thus, it is not necessary to allocate extra time and lab room for participants. Within this approach, reaching large number of users with various demographics in small time is possible. Therefore, the crowd-based video quality assessment approach is promising. However, the challenges still remain due to non-standard test equipment, lack of controlled experiments, and reliability issues. For example, the users might want to finish a given task as fast as possible without concentrating on it much in order to earn more money, which results in 'cheating'. Although there are methods to minimize the influence of cheating (such as additional questionnaire methods), the evaluation of crowdsourcing-based data remains as a challenge. The diversity in crowdsourcing platforms and users are discussed in [199], *e.g.*, the offered tasks, completion time of the tasks, working times of the users, and the user profiles.

There are advantages and disadvantages for objective and subjective methods. On one hand, more representative data (with lower cost within shorter time) can be collected via the objective methods, on the other hand, it might be necessary to validate the results with users via subjective tests as the objective methods (*e.g.*, a very popular video due to its content might indicate high user engagement) do not necessarily indicate an acceptable video quality. Therefore, QoE methods that involve user interviews and tools that can record momentarily QoE in daily life user environments (*e.g.*, without being obtrusive to the user) are more reliable in assessing QoE, and it is recommended

to validate objective metrics with the subjective ones.

2.7.3 Challenges with the Subjective QoE Scales, *e.g.*, MOS

Divergence in manifestation of reactions with respect to identical stimuli exists. Disparity in user ratings, due to varying retrospective data, is a key challenge to consider in QoE studies. Thus, in a user study, to know how to assess the users' reaction is hard, and can depend on each individual due to the dissimilarities of reactions with respect to identical stimuli. Some people can react violently, *i.e.*, punishing the quality with grade 1 on the MOS, while others might be more patient and tolerant.

The attributes and the type of QoE assessment data to be collected should be well defined prior to the experiments. The target variable, *e.g.*, MOS, should hypothetically relate to the collected influential attributes. There are challenges, while assessing the user-perceived quality based on MOS, and the challenges with respect to the end-user perception of the quality levels such as "OK", "Poor", "Fair", and "Good" are pointed out in [58–60]. In [61], the authors argue that the ITU recommended methods for subjective quality assessment of speech and video are not appropriate for new services and applications. MOS is valid only for particular experiment setting and one needs to be very careful while applying MOS to a particular targeted service. In addition, it might be influenced by the "forgiveness effect" [200].

As averaging user ratings might smooth out the low user ratings, the authors of [64], have introduced a new term, Standard deviation of Opinion Scores (SOS) to have a better understanding on the fluctuations of users' perceived quality. Furthermore, due to the high variation in opinion scores, the subjective ratings are valuable if they pinpoint the low qualities in particular. Therefore, it is important to investigate those worst-case scenarios, *e.g.*, scenarios with long freezes that can be statistically described as outliers and extreme values of the opinion score dataset. Nevertheless, R. Schatz *et al.* [165] recommended Absolute Category Rating with Hidden Reference (ACR-HR) test method with five-point quality scale as the mean value and the confidence intervals of the results from different methods do not deviate much. There are other methods such as the Maximum Likelihood Difference Scaling (MLDS) approach used by Menkovski *et al.* [138], which relies on relative quality assessment that utilizes the direct comparison mechanisms rather than rating. Relative quality assessment is indeed a good method to understand the perceived quality differences.

Despite some drawbacks of MOS scale, it has been applied to some user studies on this thesis, as it is commonly used in the area. However, the weaknesses of MOS is covered by other methods such as face-to-face or online user interviews and user surveys in this thesis. In this thesis, we focus on metrics objectively measured from the network- and application-levels, with the energy perspective. Then, we relate them to

MOS and qualitative user feedback obtained from users.

The Role of Human Perception in QoE

The aforementioned reviewed models are related to the biological and cognitive factors behind the Human Visual System (HVS), and predicting the visible factors that degrade the user perception has studied via Visible Difference Predictor (VDP) [74]. The outcomes of VDP studies help to understand the HVS and help to conduct insightful QoE assessment. In [157], a set of threshold durations have been proposed in web browsing. It is stated that 100 ms is defined as the maximum tolerance threshold for the user to feel that the system is reacting instantaneously. It has also been stated that disruption less than one second is recognizable by human but the user might still feel uninterrupted.

The human's perception of a difference depends on the current magnitude of perception, thus some changes in quality may not be perceived by human, unless they are above some threshold. Weber-Fechner [46] introduced the formula based on "just-noticeable difference" in Eq. 2.1 that links the human perception to the relative change in stimulus. dP is the differential perception and is proportional to the relative change $\frac{dS}{S}$ of a physical stimulus, S.

$$dP = k\,\frac{dS}{S} \tag{2.1}$$

Logarithmic relationship between the QoS and the QoE are revealed in the context of web browsing [54] and file downloading [72].

The IQX hypothesis model [69, 71], is used to describe QoE as an exponential function of single QoS impairment factor. IQX relates the impact of absolute stimulus change to the current perception level. In other words, the change of QoE caused by a change in QoS, *e.g.*, packet loss, depends on the actual current level of QoE, *i.e.*, reflecting the actual level of expectation. The resulting equation becomes as depicted in Eq. 2.2. In [69], it is shown that QoS metrics such as the reordering rate (caused by delay jitter) and packet loss are easily matched to the QoE metrics MOS.

$$\frac{\partial QoE}{\partial QoS} = -\beta(QoE - \gamma) \tag{2.2}$$

which has the solution,

$$QoE = \alpha \cdot \mathrm{e}^{-\beta \cdot QoS} + \gamma \tag{2.3}$$

Similarly in [138], a non-linear relationship between the video bitrate and the perceived quality by the user is stated. In this thesis, we present our QoE models via similar non-linear models.

2.8 Video QoE Management Mechanisms

Together with QoE monitoring, it is also necessary to manage and intervene the underlying technical setup or mechanism in order to improve the perceived quality during video streaming. Mekovski *et al.* [149] proposed a QoE control framework for adaptive video streaming that takes into account the temporal and spatial characteristics of the video. Similarly, Latre&Turck's work on QoE management in [150] used the already existing QoE metrics and techniques such as traffic flow adaptation (by changing the transport layer configuration), admission control (by admitting/blocking new connections to avoid congestion), and video rate adaptation (by changing the actual video quality level). The authors' work focuses on the low layer design for autonomic management architecture, and emphasizes the need to design control loops at higher levels. The presented outcomes of the studies are a result from simulators with the focus on full-reference video quality metrics such as SSIM. Also the end-to-end evaluation of performance metrics, *i.e.*, measurements on the end user devices is missing. The proposed methods and algorithms are great contributions to obtain QoE insights, however are not much applicable to real life scenarios, where end-to-end QoE is vital.

2.9 Energy and Its Relation to QoE

Energy is, by Oxford definition, "the strength and vitality required for sustained physical or mental activity" [76]. The energy must be sustained in a wise manner in any activity, therefore the products and the services must ideally be energy-efficient, *i.e.*, they must gain the ability to do the same task by using less energy. Amongst the numerous benefits of energy-efficiency, some important ones that are related to this thesis work are: (1) reducing air pollution and CO2 emissions while generating energy (*i.e.*, environmental); (2) reducing customers' energy bill (*i.e.*, cost); and (3) increasing the service time of the electronic ad hoc or other mobile devices such as smartphones that are running with limited battery life time (*i.e.*, availability). The Information and Communications Technology (ICT) comprises up to 10 % of the global annual energy consumption and about 2 % of the global CO2 emissions [10]. High energy is being consumed, while a large amount of data is being generated, processed, and transmitted over the network [7]. $60-80$ % of the consumed energy in mobile networks is

consumed by access technology.

The energy studies in this thesis are on some portion of operational activities in the wireless technology. The infrastructure nodes of wireless technology such as the base stations or the routers are connected to the power grid, where the operational costs are the major problem. This motivates Telco operators to put extra efforts to save energy on the access networks, given the rising volumes of data traffic expected for the upcoming decade.

From the end-user point of view, the mobile ubiquitous device users benefit from the applications and the services as long as those devices have the required energy to fulfill the users' requested tasks. In this context, *i.e.*, for the devices that run on limited battery and disconnected from the power grid, the availability of energy might be more important then its cost. This implies that the energy consumption might be potentially an influential factor on the perceived QoE on the mobile devices such as the smartphones. Application developers often neglect the power consumption of the their applications/services as their priority is on usability. They often want to see the applications in the market immediately [100] to turn the investment into profit as quick as possible. It is estimated that more than half of 200 new Apple's iPhone applications available daily, do not achieve a critical mass of user acceptance and are withdrawn from the store's offer within months from the launch. One reason for users uninstalling a particular application from their smartphones might be their energy consumption. Thus, energy management can play a crucial role in improving application acceptance by users.

Energy measurements have to performed in parallel with the QoE measurements. Ultimate goal is to minimize energy consumption without diminishing much the QoE.

2.9.1 Existing Energy/Performance Trade-offs

Energy savings are often achieved with the cost of degradation in the performance. Some trade-offs can be listed as follows [36, 37, 205].

- **Deployment/Energy Efficiency Trade-off:** in-between the deployment costs such as Operating Expenditures (OpEx), Capital Expenditures (CapEx), and the overall energy consumption of the network,
- **Spectrum/Energy Efficiency Trade-off:** in-between the throughput per bandwidth unit, *i.e.*, bandwidth utilization, and the overall energy consumption of the network,
- **Bandwidth/Energy Efficiency Trade-off:** in-between the available bandwith and the consumed energy to transmit data,

- **QoS/Energy Efficiency Trade-off:** in-between QoS parameters such as packet delay, or packet loss, and the consumed energy to transmit data,
- **QoE/Energy Efficiency Trade-off:** in-between the user-perceived QoE, and the consumed energy

Studying the trade-offs needs to be done with great care, as there might be some scenarios that the items at both sides of the the trade-off becomes correlated. One example is the extra power being consumed to provide high bandwidth. On the other hand, this might cause extra interference in the wireless transmission medium, and eventually causes reduction in the QoE. The relation between the QoE and energy-consumption can be a trade-off or a win-win. In this thesis work, a set of scenarios on the smartphone for the both cases are studied in Chapter 7.

Amongst the five listed approaches, this thesis focuses on the last one, *i.e.*, QoE/Energy efficiency trade-off. An approach to reduce the energy consumption while not influencing the user-perceived QoE is presented in Chapters 6, 7, and 8.

2.9.2 Impact of Network Applications on the Energy Consumption

The cloud-based applications are expected to provide service anywhere and whenever requested by the end-user, but constrained by the remaining energy. The factors influencing the energy consumption of the smartphones are cellular module [77] [78], the display with LCD panel and touchscreen, the graphics accelerator/driver, and the backlight [99]. It is stated in [260] that not only the cell environment, radio channel conditions, carrier frequencies, but also application data traffic patterns have significant impact on the battery lifetime. The energy consumption of smartphones is proportional to the usage of online services. Most network-based applications are interactive, and it is stated in [176] that the instantaneous total power consumption is doubled upon touching the phone screen. The limitations in battery life of smartphones reduce the service time, which degrades the QoE [67].

Downloading data is the most energy-consuming process in smartphones [229]. Not only during interaction with the smartphone, but even when smartphone is in passive mode, *e.g.*, when screen is OFF, *i.e.*, the data activities still exist. This fact is often neglected by most of the developers. ScreenOFF smartphone data traffic contributes up to 58.5% of the total radio energy consumption [79]. There are multiple reasons, such as the inefficient and asynchronous information polling intervals of popular cloud based applications, or sneaky network behavior of the available free apps with advertisements.

The number of 4G subscriptions is expected to reach one billion in 2017, and accordingly the importance of battery lifetime on smartphones will grow. The smart-

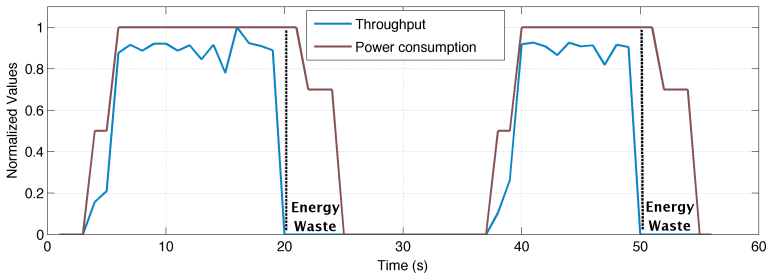


Figure 2.3: High-level illustration of wasted energy for a typical network activity on 3G cellular data module.

phone applications necessitate a robust management of data transmissions to reduce the energy consumption caused by cellular data traffic. Additionally, research shows that the user-satisfaction based from the battery performance for 3G-enabled smartphones is higher (6.7/10) than that in 4G-enabled smartphones (6.1/10) [97]. Based on the previous work [67], it has also been concluded that some users even avoid using 4G to save energy.

The frequent and asynchronous network activities and data flows from different apps cause the cellular data module of a smartphone to oscillate in-between these power consumption states. Fig. 2.3 illustrates the power consumption and the throughput during a typical network activity via a 3G cellular. The power consumption shifts to the active, *i.e.*, most power-consuming, state as the throughput increases. However, when there are no packets being transferred (*e.g.*, after the black vertical dotted line in Fig. 2.3), the power consumption does not immediately drop back. Managing the timeout periods for the transitions amongst the power states, *i.e.*, predicting the timeout periods based on the data patterns, is an ongoing work in the mobile energy optimization field denoted as fast-dormancy [96]. In addition to the timeout periods, there exists also some transition delay while switching the power states. The transition delays depend on many factors, *e.g.*, the congestion in the cell, although they are reduced in Long Term Evolution (LTE)-Advanced, as compared to LTE and 3G [95]. Overall, it is important to minimize the number of transitions in-between the power states, and execute the data traffic in bursts. While the smartphone is connected via the 3G interface, the power consumption varies based on the Radio Resource Controller (RRC) power states such as IDLE, Paging Channel (PCH), Forward Access Channel (FACH), or Dedicated Channel (DCH) (in the order from least to highest power consumption). As a representative example from one test run (see Fig. 2.4), even during one RTT, *e.g.*, 1429 ms (due

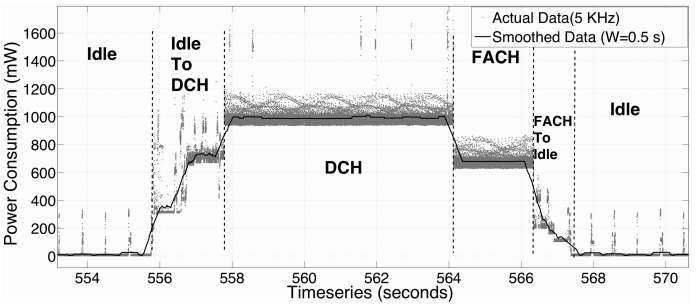


Figure 2.4: Example of the power consumption of different states of the cellular radio module in Samsung Galaxy S for a single Ping execution with RTT = 1429 ms [196].

to the delay when the RRC power state promotes from IDLE state) of a 64 Bytes ICMP packet, the duration when the radio module state is not IDLE, is roughly more than 10 seconds. A few seconds of delay is observed when the RRC power state shifts from IDLE to DCH state as expected. Observe that the solid black line is the Simple Moving Average (SMA) [197] with window size of 0.5 s, *i.e.*, smoothing over 2500 power data samples. In SMA, all the values within a timewindow have the same weight, in contrast with the non-linear moving average methods such as Exponentially Weighted Moving Average (EWMA). Thereby, the energy is wasted during the remaining duration, *e.g.*, IDLE-To-DCH and FACH-To-IDLE periods. In addition, not all the time spent at the *DCH* state is due to the data activity, rather due to the inactivity timeout periods.

The power consumption of the cellular data technologies is different amongst themselves. According to Huang *et al.* [185], 10 MB data downloading via 3G and LTE require roughly 30 and 1.6 times the energy of WiFi, respectively. According to [229], being IDLE while connected to a GSM network costs 41% less energy than being connected to the 3G Universal Mobile Telecommunication System (UMTS) network. Although UMTS networks are efficient in terms of large data transfers, this is not true when there are multiple small ones scattered over a long duration. Overall, 4G LTE has much higher bandwidth (downlink median: 13 Mbps, uplink median: 6 Mbps) as compared to the 3G access technologies at the cost of being 23 times less power-efficient [185].

2.10 Energy Models and Measurements for Smartphone

The energy consumption measurements are investigated from the application and the network perspectives.

2.10.1 Energy Consumption Model for Cellular Radio

The cellular radio is designed to minimize the energy consumption such that it shifts between the most power-consuming state to a less power consuming one depending on the network traffic. Radio Resource Controller (RRC) protocol in UMTS [202] has a state machine with three states (or four based on the manufacturer and network provider) that have different power consumptions, and these states depend on the data rate and the uplink/downlink queue sizes [203]. The IDLE state is assumed when no radio resource is allocated, no connection between User Equipment (UE), *e.g.*, smartphone, and Radio Network Controller (RNC) is established, and the UE cannot transfer any data. The DCH state is assumed when UE-to-RNC connection is established, and the UE is allocated with the dedicated DCH transport channels for both downlink and uplink with the highest bandwidth and power. The FACH is the state for very low data throughput rate application requirements, and in FACH, the RRC connection is established but no channel is associated. The power consumption decreases in the order of DCH, FACH, and IDLE [204].

Cellular State Demotion:

When there is data activity, if the cellular module is at DCH state, it preserves the state for a particular timeout period. If there is no packet received within the inactivity duration (depending on the device and network settings), User Equipment (UE) sends a Signaling Connection Release Indication (SCRI) signal to the Radio Network Controller (RNC), indicating that the currently assigned radio resources are not needed. Then, the RRC state demotes to IDLE or indirectly through FACH and PCH states [282].The cellular radio module shifts from DCH to IDLE state first by shifting into an intermediate state (FACH) depending on the tail time (can last up to 15 seconds [204]), which depends on the network provider, the traffic volume, or the smartphone manufacturer. The amount of the web content to be polled and the connection speed are important factors for the energy consumption during the polling periods. In addition, there exists a delay between the state transitions, which are due to the high number of control messages being transmitted and received between UE and the RRC.

Cellular State Promotion:

The cellular module might promote a state change from IDLE to DCH or FACH, *e.g.*, due to an increased data rate. According to the state of the art 3G power consumption models, it takes a few seconds for RRC state to switch from IDLE to a DCH state with the exchange of 30 signaling messages [87].

Similar RRC power state models exist for 4G LTE networks [95]. For instance, the LTE RRC state machine consists of ACTIVE, Short Discontinuous Reception (DRX), Long DRX, and IDLE states. The active state implies continuous reception, if there is no data transfer, the Active state demotes to Short DRX state. The power state promotes back into Active if there is a data transfer before the Short DRX timeout expires, otherwise the state demotes to Long DRX. Similarly in Long DRX, if there is data activity before the Long DRX timeout expiry, the state promotes to Active, else it demotes to the IDLE state.

2.10.2 Smartphone Power Model for Video Streaming

The total power consumption during video streaming is a sum of involved processes such as downloading the packets via the wireless module, decoding the received video and audio packets, and rendering/displaying the decoded frames on the screen. The power consumption during the processes depend on multiple factors, *e.g.*, the power consumption during downloading depends on the available throughput and the used hardware of the wireless interface, the decoding power depends on the user codec type, and the display power depends on the display size, the display type, and the hardware's chipset.

Downloading process

Downloading is the process when the data is being received to the smartphone. Indeed, this might also highly susceptible to many factors including the type of the communication protocol. When the downloading is performed via a stateful transport protocol such as TCP, the process involves both downloading (the video content payload) and uplink (the control messages back to the remote video server such as ACK and NACK). However, the impact of small sized ACK messages on the power consumption is small, thus this effect can be ignored. The power consumption during downloading and uploading a file via the smartphone's wireless interface can be represented by a linear equation as in Eq. 2.4-2.5, respectively. a_{d} and a_{u} are the downlink and uplink power consumption (per Mbps) parameters, while a file is being actively downloaded or uploaded (*i.e.*, when throughput is not zero). b is the bias power, *i.e.*, the power consumption when the

corresponding network module is switched on but not active (*i.e.*, when the throughput is zero). In [185], the parameters to the models are provided as in Table 2.2 for the three wireless technologies and for downlink and uplink.

$$P_{\text{downlink}} = a_{\text{d}} \cdot t + b \tag{2.4}$$

$$P_{\text{downlink}} = a_{\text{u}} \cdot t + b \tag{2.5}$$

According to the table, the following conclusions can be made with respect to the

	a_{u} [mW/Mbps]	a_{d} [mW/Mbps]	b [mW]
LTE	438.39	51.97	1288.04
3G	868.98	122.12	817.88
WiFi	283.17	137.01	132.86

Table 2.2: Power model for data transmission [185].

power models during the data transmission process. First of all, the uplink power consumption per Mbps is much higher as compared to the downlink. Second, the LTE power consumption is much higher than in 3G and WiFi when there is no throughput. Third, when the downlink throughput is nonzero, the power consumption of LTE per Mbps is less than half of the power consumption of 3G and WiFi.

Decoding process

The decoding process's power consumption is considered to be negligibly small as compared to the display and data transmission process [186]. The authors in the previous research stated that with the same display content, bitrate, and frame rate, the video decoding power consumption stays the same when the video is stretched to 1280×720 from 640×360. However, the power consumption during decoding process is highly susceptible to the resolution of the video, *i.e.*, high decoding power is needed to decode video with higher resolution. To summarize, the decoding power consumption increases significantly with the frame rate and the resolution. No significant effect of bitrate is observed on the decoding power.

Display process

The display power consumption contributes much to the overall smartphone power consumption. The display module is also another factor influencing the power con-

sumption as for example Super AMOLED Plus is more energy efficient than a Super AMOLED [186]. Due to the characteristic of the display modules, the power consumption for particular colors might be different as well. Lighter colors (*e.g.*, white) consume much higher power as compared to the darker ones (*e.g.*, black).

2.10.3 Energy Consumption Metric

Energy consumption is defined by power consumption measured over the time. Energy is calculated as the sum of the (per second) average power, $\bar{P}_t$ (Watt= Joule/second) between time t (in seconds) and $t-1$ (in seconds), multiplied by the unit second, as presented in Eq. 2.6. T depicts the duration of a measurement.

$$E(T)\,[\mathrm{J}] = \sum_{t=1}^{T} \bar{P}_t\,[\mathrm{W}] \cdot 1\,[\mathrm{s}] \tag{2.6}$$

Energy-per-bit (E_b) is another metric to express the energy-efficiency of a particular work, *e.g.*, transmitting data over the wireless access network. It is calculated in Eq. 2.7 as the amount of energy consumed during transmitting one bit of the transferred data. I is the number of bits transfered for which the energy E is used. Simplest example is the comparison of the energy-efficiency in-between the General Packet Radio Service (GPRS) and the UMTS. The power consumption of the UMTS is much more than GPRS, however the transmission time of certain bytes of data is much less via UMTS than transmitting the same amount of data via GPRS. Therefore, the reduced transmission time in UMTS eventually provides less energy consumption, in overall, than the GPRS. This makes UMTS much more energy-efficient in large data transmissions as compared to 2G [229]. Therefore, energy-per-bit is a highly important metric to consider to compare the energy-efficiency of different systems.

$$E_b\,[\mathrm{J/bit}] = \frac{E\,[\mathrm{J}]}{I\,[\mathrm{bit}]} \tag{2.7}$$

2.10.4 Energy Management Efforts in Information and Communications Technology (ICT)

The efforts in academia and industry to reduce energy consumption on the overall ICT are many. Energy Aware Radio and neTworking tecHnologies (EARTH) is a project that focused on developing energy-efficient products, components, deployment strategies, network management strategies particularly in future LTE systems. GreenTouch [187] is another example with a similar goal; it aims to reduce the carbon foot-

print and energy consumption of ICT devices by transforming today's communication and data networks. The focus of these large-scale long term projects some approaches are mostly on the base station side, *e.g.*, detaching the base stations (*i.e.*, demoting to IDLE state) while not significantly decreasing the coverage when there is low traffic. This way, according to [10] [12], up to 53 % energy savings are achieved. In this thesis, the focus is narrowed down to energy consumption only on smartphones.

2.10.5 Energy Measurement Techniques, Methods and Tools on Smartphones

There are various techniques to measure power consumption on smartphones [205]. Power consumption is acquired either via physical measurements or via measurement based estimations. The first measurement technique is by probing directly the USB wireless interface, *e.g.*, WiFi, LTE, via an external digital meter; the second technique is by using development boards and kits to measure only the broadband module's power consumption; the third technique is by intercepting the battery terminals via an external high precision power measurement devices that measures the overall power consumption of the devices, Monsoon with 5kHz sampling rate or Fluke 8845A with 50kHz sampling rate [190]; and the fourth technique is via built-in sensors, *e.g.*, fuel-gauge IC, within the devices that enable power profiling. Fuel-gauge IC indeed seems promising as it provides measurements such as voltage, temperature, and current every second, and stores it into a specific memory. It can also control the charger for a more efficient charging depending on the battery capacity.

There are software-based energy measurement tools available for smartphones with the common aim of making the power measurements transparent to the app developers as well as to the users, so that they can take appropriate action to minimize their smartphones power consumption. PowerTutor [172] is used in research, and it presents the power consumption of all available components (*e.g.*, CPU, network interface, display, GPS) and the running applications to the users and the application developers. It works accurately only on particular phones such as HTC G1/G2. PowerTutor receives the current values from the driver and then multiplies the value by the voltage that is basically the phone battery (typically 3.7 V or 4.5 V depending on the phone type). PowerTutor estimates the energy consumption of applications and services based on the processing times.

2.10.6 Smartphone Resource Profilers

Profiler is a software or a complete system that enables to get energy usage of applications, services, or hardware components on a device. EnergyBox [253] uses the

pre-obtained models and estimates power consumption values based on the different states of the device. However, these models are often limited to the conditions that the physical measurements were conducted. ProfileDroid [207], characterizes behaviors of a set of Android apps at multiple layers such as app specification, user, OS, and network layer; however the work does not involve the behavior investigation from the energy perspective. Eprof [254] is another Android-based profiler that is able to analyze the power states of applications. PowerScope [255] is an energy profiler that is able to identify energy leaks, and estimates the application energy consumption by measuring the duration and frequency of states. JouleWatcher [256] is another energy profiler that estimates the energy consumption of the whole systems as well as the individual threads. Self-tuning Power Manager (STPM) [257] is a resource management framework that collaborates apps and systems in order to use the I/O devices in an energy-efficient manner. This is done by monitoring the intentions of apps to use the I/O devices. Koala [258] predicts the energy consumption of each running software, and allocates resources to provide the optimum energy-efficient operating condition. Llama [259] adjusts the application QoS to achive the energy requirements by also considering usage behavior. PowerProf estimates the power consumption of individual hardware components, and it aims to obtain a power model. CoPoMo [260] is a context-aware power consumption model, simulated with realistic scenarios and metrics. Carat [261] is a research-based free app, empowered by UC Berkeley, that aims to provide helpful feedback to the user regarding how to save more battery based on the smartphone usage. In order to do that, Carat collects a set of metrics from the user phones such as the name of the running apps, the percentage of remaining battery, memory and CPU utilization, the unique device ID, the battery state, the OS version, and the phone model. Sesame [262] is a power modeling scheme for the overall system power. It utilizes a built-in current sensor to estimate the overall energy consumption. AppScope [263] is an Android-based energy estimator system, designed as a kernel module, which monitors the kernel level hardware usage of applications. Devscope [264] is an Android-based online tool that controls components according to the built-in Battery Monitoring Unit (BMU) (*i.e.*. fuel-gauge IC), and derives the component power model automatically by analyzing the changes of power state. Nokia Energy Profiler [266] is a Symbian based tool that is able to monitor the energy consumption of 2G and 3G on Nokia N95, with applications such as text messaging, data traffic, and voice.

Profilers enable to understand energy consumption, but it is still up to the user to act upon to obtain a good management of energy resources in her/his device. Even if certain tools seem to do power modeling (*e.g.* Sesame [262]), it seems to be left to the user in most cases.

2.10.7 Energy Management with Energy-aware Tools in Operating Systems

ECOSystem [252] is a Linux-based energy-centric operating system that fairly allocates energy consumption to hardware components and applications, by tracking the applications' resource needs. Odyssey [275] adapts the QoS and the energy consumption based on the available energy and resources (and also by utilizing historical data), by monitoring the availability of resources and the demands of running applications. Cinder [274] is another OS for mobile devices that aims to allocate energy by scheduling threads based on the available resources, *i.e.*, by controlling taps and reserves. Applications often use the available resources (*e.g.*, sensors) independently from each other. Context-dependent energy saving approaches such as ErdOS [218] (an extension to Android OS) manages the computing resources for applications based on the user's demands, *e.g.*, usage patterns and context. CondOS [217] is responsible for managing the available resources (*e.g.*, sensors) for all running applications. Openmoko [265] is a free mobile phone project with an open source software stack, Neo FreeRunner, and its circuit schematics are publicly available. This enables a researcher to analyze the energy consumption of each hardware component of the phone with the help of an external power meter, and to identify the most power-consuming ones.

2.10.8 Commercial Battery Saving Apps

There are various energy saving apps based on manually configured user preferences, including JuiceDefender [82], Auto3G BatterySaver [83], EnergySaver [84], BatterySaver [85]. These apps do not collect data related to the user behaviour, thus are not based on any user behavior models. BlueFi app [86] allows a communication either via voice or text over WiFi or Bluetooth, with the limitation that the two peers will be within each others or within the same WiFi router's range. This can be both an energy-efficient and economical solution for communication between two smartphones as compared to communication over the cellular networks.

2.10.9 Network-based Energy Saving on Smartphones

Various studies were conducted to reduce energy consumption with the focus on the network. Some important previous work so far are related to leveraging the 3rd Generation Partnership Project (3GPP) standards by proposing fast-dormancy. This is basically forcing the transitions such as DCH to IDLE or FACH to IDLE without waiting for the inactivity timeout periods. Tail Optimization Protocol (TOP) is proposed by Qian et al [213] to minimize the timeout periods, *i.e.*, inactivity times, between

the states by invoking the fast dormancy support. On the other hand, fast dormancy might have drawbacks. According to research performed by AT&T [204], for particular apps, reducing the tail timer by three seconds reduce the resource usage by 40 %, but increase the number of state promotions by 31 %. TailEnder [81] schedules the network data transmissions (either via prefetching or delaying based on the application type), in order to reduce the duration of the device in its high-power consuming state. TailTheft [268] is another scheme that utilizes the tail time (unutilized and wasted time) of applications by scheduling a set of extra transmissions. The RadioJockey system in [87] predicts the end of communication messages of applications by analyzing program execution traces and saves energy by invoking fast dormancy.

Other approaches are related to scheduling the network traffic. SALSA [281] is an algorithm, using the Lyapunov optimization framework, that decides on the time of the data transmission. AT&T used app profiler to diagnose the small data transmissions that are scattered over a long duration, bundled the scatter-burst transmission of small packets into a single transmission, and achieved 40 % energy saving. The missing part in the research is how the suggested solution influences the end-user perceived QoE. Vergara *et al.* [230] studied a Cross-Layer Burst Buffering (CLBB) algorithm that schedules the background 3G traffic of a smartphone in order to fully utilize the network link when active and in parallel increase the IDLE duration. Junxian *et al.* [79] studied the screenOFF traffic of 20 users for a period of five months, and compared the data traffic patterns both when the screen is ON and OFF. The screenOFF data traffic comprises of a large number of small packets, which in turn comprises 35.84% of all captured packets from all users during the five months of study. The average burst length during screenOFF was 1.37 seconds. The authors propose a solution based on "fast dormancy" and "batching" of traffic of particular applications like streaming music or Facebook. The authors show via simulations that it is possible to save up to 60.92% of the network energy for applying these algorithms, however they do not discuss the consumed energy during the execution of those algorithms running in the background of a smartphone. Also the assumption is that the operator network could be configured by a device for its fast dormancy interval; appears to be / is impractical. The particular parameterization of the algorithm proposed by the authors would be much more complex than presented in the simulations.

The advantage of our own research is that we do not try to build a model per application, but per user. We leverage human factors of interaction with technology towards building an accurate model for energy saving, for a given individual, additionally to the fact that our modeling focuses uniquely on the user. We use the data collected exclusively on a users smartphone and we do not attempt to change any network parameters, as authors of [79] suggest. The authors do not address the issue of how the algorithm would perform in real life user settings, *e.g.*, without studying the resulting user experi-

ence. This is what distinguishes our work in Chapter 8 of this thesis from the previous work. Other related works to optimize delay-tolerant data transfers in the operating system include research of Balasubramanian *et al.* [81] on a distribution of energy consumption of smartphones for email, news feeds, and web search and based on their efforts to model the "long tail for this consumption, they have developed an algorithm that either schedules applications updates on regular times (*e.g.*, email) or pre-fetches a multimedia content needed by a user later on. The experiments are mainly simulation-based (with simulations parameterized with results of real lab-based measurements on smartphones). The authors as do not address the issue of how the algorithm would perform in real life user's settings, as we do in this thesis.

Moreover, the RadioJockey solution by Athivarapu *et al.* [87] focuses mainly on leveraging the fast dormancy state of a smartphone and invokes it based on the prediction of the end of data exchange burst on a smartphone. This way, the phone saves energy while sleeping in between the data bursts. However, the authors do not consider the user model in their optimisation, and particularly not the screenON and screenOFF models, as we do. Their solution and results are strictly focused on predicting the communication bursts for a given application. Similarly, an optimisation for only email application has been researched by Xu *et al.* [80], where the authors analyse current email clients for Android and Windows platforms (for a smartphone) and propose its protocol optimisation, proving via simulations that their solution may enable to cut the energy consumption for email by half. They evaluation includes in-lab measurements and they do not involve real users for evaluating their perception and experience, as we do. Finally, we acknowledge that the design decisions for own work is similar to long-standing research on batching work and slack times in resource-constrained systems [89], however these are purely based on the scheduling of resources and have little to do with human factors and individual user models, as we propose.

On the other hand, for real-time applications that are not delay tolerant, approaches such as PSM-Throttling are proposed [269]. The PSM-Throttling adapts the streaming rate and utilizes the available link (*i.e.*, via bandwidth throttling) better to reduce the transmission energy consumption. YouTube video works based on a progressive download manner, and the video content is downloaded in two download phases [145]. The first one consists of an initial burst typically lasting for 40 s, and during this phase the application uses the maximum available bandwidth. In the second phase, *i.e.*, after 40 seconds, the application uses a throttling algorithm to reduce the download rate equal to 1.25 times the video clip encoding rate. During the second phase the video data is received as 64 KB data chunks. Kernel Level Shaper (KLS) is another proposed algorithm to save energy by scheduling independent applications' network traffic within the kernel. In order to reduce the initial latencies, Gerber *et al.* [204] pipelined HTTP requests into a single TCP connection rather than sequential HTTP requests. There

are example applications such as Google Music that also transfer data as large bursts. SPeeDY (SPDY) [231] is an application-layer protocol developed by Google in order to transport web content with low latency. It is multiplexing HTTP requests in a single TCP connection in order to increase the link utilization. On the other hand, bundling the network traffic might have drawbacks. For example, Blenk *et al.* studied the tradeoff; while single TCP flows typically do not utilize the available bandwidth on a cellular link due to variable capacities, parallel transmissions yield an increased average transmission duration [233].

Choosing the low power consuming wireless interface is another alternative to reduce energy consumption. Sharma *et al.* presents an energy-efficient architecture, Cool-Tether [232] in order to utilize the available smartphones with Mobile Hotspot (MH)'s existing Internet access. Cell2Notify [270] is a system proposed by Microsoft Research, particularly for VoIP calls, that turns on the WiFi interface of smartphone only when a VoIP call is received. This is done as it is claimed that the WiFi power consumption is less and gives a better performance as compared to the cellular networks. Blue-Fi [271] and ZiFi [272] are systems that predict the availability of WiFi access points based on the Bluetooth contact-patterns, ZigBee interference, or cell-tower information. These methods reduce the energy consumption by preventing the long IDLE periods and number of scans of WiFi interface when WiFi is not accessible. There are other studies as such done by Rahmati *et al.* [273], to reduce the energy consumption of smartphone by estimating the right network interface (without switching on the interfaces), only based on the context information and historical data.

All of the aforementioned existing tools and operating systems that aim to save energy are summarized in Table 2.3

2.11 Smartphone User Behavior

There are diverse smartphone user interactions. In this section, we review the related work on the user's charging behaviour, user's smartphone screen interaction, and the corresponding state of the art models. Next, we review the related work on the sociological aspects such as the sociological and psychological influence of the application's data traffic on the users, and how the user behaviour might be influenced by the application's network activities. We also present a set of existing apps to control application notifications, which are intended to reduce the influence of application's state on user's state.

Table 2.3: Existing tools and operating systems for energy management [18].

RESOURCE PROFILERS	PowerTutor, PowerScope, Joule Watcher, STPM, Koala, Llama, ProfileDroid, DevScope, BMU Neo FreeRunner, Carrat, Sesame, Nokia Energy Profiler, KLS, CoPoMo
ENERGY-AWARE OPERATING SYSTEMS	EcoSystem, Odysseey, Cinder, ErdOS, CandOS
WIRELESS INTERFACE AND PROTOCOL OPTIMISATIONS	TailEnder, TailTheft, TOP, KLS, SPeeDy, PSM-Throttling, Cell2Notify, Blue-Fi, RadioJockey, ZiFi, Context-for-wireless, SALSA, CLBB
EXTERNAL HARDWARE POWER MONITORING TOOLS AND SENSORS	Monsoon, Fluke 8845A, other sensor board-based prototypes 16F684 microprocessor, DS2782 Stand-Alone Fuel Gauge IC
BATTERY SAVING COMMERCIAL APPS	Auto3G BatterySaver, JuiceDefender, BlueFi EnergySaver, Comodo

2.11.1 Smartphone Charging Patterns

Charging patterns are influenced by the availability of power to complete the user's tasks. The charging patterns may differ from one user to another. The authors of [208], based on a large-scale Blackberry study, clustered users as: opportunistic chargers, light consumers, or night-time chargers. In [209], the authors show that the charging patterns do not depend on the amount of the energy left on the smartphone, but rather on the opportunity, device, context, time of the day, and location. In other words, the users tend to charge their phones even when the battery level is not very low, which may indicate that the energy is an issue, and that the users tend to avoid low battery level on their smartphones. The battery level of a smartphone varies almost by a factor of 2.5 between the business user interacting a lot with it, and the smartphone permanently at IDLE state [99].

2.11.2 Smartphone Screen Interaction Patterns

According to Google Statistics, smartphones are the backbone of our daily media interactions with the highest percentage, 38 %, as compared to other devices such as tablets, and PC's. The statistics also show that 54 % and 33 % of the total operation time of smartphones are spent in communication and entertainment, respectively. The average interaction time on the smartphone screen is 17 min (minutes) per day [210]. Another research states that the most frequent users interact with the smartphone every 7 minutes during day time [88]. Inevitably, the statistics are highly varying depending on the country and the users. This makes "one-model-fits-all" almost impossible. Falaki *et al.* in [276] claim that the mechanisms to improve user experience or energy consumption should be done in an idiosyncratic way, *i.e.*, per a user, which means adapting to the individual user behaviour. They showed that number of daily user smartphone interactions varies from 10 to 200, with daily interaction durations varying between 30 min to 500 min. Another finding of the study was that many short interactions likely drain more energy than a few long interactions, as those interactions likely change the state of the relevant components such as network data modules, CPU, or memory. The variation in the interaction with the smartphone causes variation in the energy consumption as well as in the data usage. In parallel, the cellular data plan with a network operator influences the network traffic and smartphone usage patterns according to the Ericsson Mobility Report [50].

2.11.3 Mathematical Models for User Smartphone Interaction

Falaki *et al.* presents in [276] that the current and the past interaction duration (*i.e.*, screen state duration) with the smartphone do not exhibit any strong correlation. Despite, the screenOFF times are matched to a Weibull distribution. The Complementary Cumulative Distribution Function (CCDF) plot helps to visualize the accumulative frequency of the events. The CCDF of a Weibull distribution is given in Eq. 2.8 [211], and if $a < 1$, then it is assumed that the chance of a particular event to occur reduces over time. If $a > 1$, then it is assumed that the chance of a particular event to occur increases over time. This means for screenOFF duration that, if $a < 1$, the longer the screen is OFF, the higher the probability that it will stay OFF. An exponential distribution is used to model the time until an event occurs in continuous time stochastic processes. An exponential (memoryless) distribution is the special case of Weibull distribution when the a is 1, and in this case the chance remains the same independent of time. Pareto distribution, on the other hand, has rather long tail, *i.e.*, decays slow, as compared to Exponential distribution. This means that there is probability that very long screenOFF durations may occur. It asymptotically approaches to the linear axes.

$$\mathrm{CCDF}_{\mathrm{Weibull}}(t) = \mathrm{e}^{-(bt)^a} \tag{2.8}$$

$$\mathrm{CCDF}_{\mathrm{Exponential}}(t) = \mathrm{e}^{-bt} \tag{2.9}$$

$$\mathrm{CCDF}_{\mathrm{Pareto}}(t) = (a/t)^b : t > a \tag{2.10}$$

2.11.4 Sociological and Psychological Aspects

Social media facilitates people to communicate, fetch information, learn, entertain, and accomplish tasks remotely. However, technology is of advantage only when it is used adequately. An excessive usage of smartphones might have disadvantages, *e.g.*, preventing face-to-face interactions and weaking the bonds between peers [90], perceiving others as having better lives and thus causing unhappiness [278], reducing user's emphatic skills [277], or misinterpreting others' behaviour on social media [92]. Social networking apps have changed user behaviour, and may cause users to become addicted to it. Davidow in [221] states that users should control the tools to accomplish tasks, rather than letting the tools control users. The Unicef Tap Project aims to provide clean water to kids while motivating users not to touch their smartphones. It is claimed that 10 min without interaction with the phone provides water to a child for a day [91]. Overall belief is that being connected to Internet should be done while not being

disconnected from the real world, and technology should facilitate benefits to the users.

There exist approaches to control the notifications and "disturbance" by smartphones. Notification Center is an app [214] that can control the application notifications based on user preferences without disturbing the user much. Android Jelly Bean provides users to toggle the "show notifications" ON and OFF. There exist apps [215] such as Addons Detected that identifies the apps that work with push notification services. AwayFind [216] is another app that notifies the user when email has been received from preferred senders. Although these solutions can prevent the users to receive undesired notifications and minimize user disturbances, they do not save radio power as the occasional network activities from the apps are still granted.

2.12 Summary

In this chapter, the core concepts in the thesis such as QoS, QoE, and energy consumption are presented. The technical details and resource requirements of the most energy consuming application on smartphone, *i.e.*, video streaming, are explained. Network-based video streaming quality highly depends on the status of the available throughput of the active wireless access technology. Thus, technical background on the power consumption of various states of wireless access technology is given. We also discuss the state of the art methodologies and tools in assessing the end-user perceived quality, *i.e.*, QoE and power consumption on smartphones. The existing internal software-based and a hardware-based external power monitoring tools and techniques are summarized. As this thesis bridges the gap between the QoE and the energy consumption on smartphones, and the user behaviour highly influences the energy consumption on smartphones, state of the art on the user behavior on the smartphones are also given. The existing studies that focus on the influence of application behaviour on smartphones, *e.g.*, undisciplined and inefficient resource management of applications, on the energy and the psychological aspects are also presented.

Chapter 3

Factors Influencing QoE on Smartphones

"Give me a place to stand, a lever long enough, and I will move the world." – Archimedes of Syracuse

---

3.1 Introduction

The usage of mobile applications and services in the daily life activities has been growing as they support the needs for information, communication, or leisure. More than fifty thousand apps are released in the Apple's App Store per month in 2014 [94]. Since the beginning of App Store, more than 20 % of new apps do not achieve a critical mass of user acceptance and are withdrawn from the stores' offer. In order for the applications and services to be embraced by the users, they need to be evaluated, with respect to their perceived experience, possible privacy violations, and the appropriateness of the application to the user's context and needs. Factors influencing this experience are known to date. Hence, this chapter presents a four-week-long 29-Android-phone-user study, which comprises a collection of both Quality of Experience (QoE) and the underlying network's Quality of Service (QoS) measurements through a combination of user, application, and network data on the users' phones. The main goal of this chapter is to derive and improve the understanding of users' QoE (and factors influencing it) for a set of widely used mobile applications in users' natural environments and differ-

ent daily contexts. The data acquired in this study is presented together with the follow up discussion on the implications for mobile applications design.

There are no robust scientific methods for evaluating applications' perceived smartphone QoE in the user's natural environment. Instead, there are qualitative methods for usability evaluation in the HCI community [100]; and there are quantitative methods for the evaluation of the QoS and performance of the underlying network infrastructures in the networking community [102]. Due to the dichotomy between these approaches, there are no robust methodologies that combine both types of methods. The approach is to measure QoE and QoS through a combination of methods with a goal of improving the understanding of factors influencing QoE, and enabling us to derive implications for mobile application design and QoE management.

The most important difference between the study presented in this chapter and the existing studies is the following. We measure users' perceived QoE in a minimally obtrusive manner on users' personal smartphones, for a set of mobile applications used in their natural daily environments. The aim is to increase the understanding of factors influencing smartphone QoE. We first present the methodology, and then the results that covers the identified influential factors on the smartphone-based QoE, and the impact of QoS on QoE.

3.2 Methodology

The focus during this study is on already implemented and operational interactive mobile applications, which are available and commonly used on a typical smartphone. The design of the methodology is given in this section. To recall the definitions of the methods, please see Section 1.5.2.

3.2.1 Overview

The following qualitative and quantitative methods are used in a four-week-long user study:

- Obtaining preliminary knowledge via an online survey on the user's background, demographics, and cumulative experience with smartphone,
- Employing continuous, automatic, and unobtrusive context data collection on the user's personal smartphone with the help of logger (CSS-QoE) software,
- Gathering user feedback on the perceived QoE via an Experience Sampling Method (ESM) executed multiple times per a day,

- A weekly interview with the user along the DRM method.

3.2.2 Participants And Data Collection

Recruitment: Online Survey

First, an online survey is conducted and the information gathered helped to recruit appropriate users for the study and get baseline background information. The information obtained answered to the questions such as:

- How long have they been using a smartphone?
- What phone type do they have?
- Through which network provider are they connected?
- What are their usual usage patterns for voice and data communication?
- Which applications do they use?
- What are the users' experiences in general with their own smartphones?

Moreover, the participants' socio-economic statuses are collected for each user. The major criteria for the selection of 30 participants were to own an Android OS smartphone and to use it frequently in various conditions in daily life. Therefore, 30 users were selected randomly from 430 potential candidates.

Automatic Continuous Measurements via CSS-QoE

The CSS-QoE application unobtrusively collects the information from users' Android smartphones regarding cellular network, Bluetooth connectivity, WiFi connectivity, WiFi Received Signal Strength Indication (RSSI), sent/received data throughput (KB/s), number of calls and Short Message Service (SMS)'s, accelerometer, screen orientation and brightness, running applications, and user location. Most of the data is collected only when the sensor value changes, *i.e.*, the Android Opinion Score (OS) updates the CSS-QoE with data. As the QoS indicator, the median Round Trip Time (RTT) for an application-level control message (64 bytes) is measured. The control message is sent every 3 minutes from the mobile device through the available wireless access network technology to a dedicated server that is deployed at the university campus. In addition, the Server Response Time (SRT), which is calculated as the time it takes for an HTTP request to get a response (with updated weather information from a dedicated weather application server to the Android smartphone), is monitored. As any new update is

detected by the CSS-QoE, the updated data is immediately recorded to the smartphone storage. This way, we minimize the memory allocation throughout the data collection process and minimize the risk of data loss.

QoE Ratings and Context Logs: Experience Sampling Method (ESM)

The ESM [66] is employed to gather users' QoE ratings. This is implemented in the form of a short survey running on a mobile device, which is presented to the subject after using an application. It is designed in a way that it does not influence the user experience and behaviour. Therefore, the survey does not appear, *i.e.*, does not pop up, after each application usage, but rather randomly (with a maximum of 8 - 12 surveys per day). The survey poses questions about:

- User's rating for the application QoE based on the 1-to-5 MOSscale [108],
- User's location (home, office/school, street, other indoor, other outdoor),
- User's social context (alone, with a person, with a group),
- User's mobility level (sitting, standing, walking, driving, other).

While rating the same application throughout the study or even for a given day, the users are requested to do their best to provide ratings, while keeping in mind that a rating is a purely subjective QoE, episodic assessment provided on the basis of the given perception of the specific episode of application use. It is aimed to capture QoE for a set of widely available mobile applications for entertainment, communication, or information purposes such as Internet-based radio, web browsing, online games, video streaming, email, and news. In total, it takes approximately five seconds for the user to complete each mobile survey. According to the tests confirmed by the users, the software in general (*i.e.*, automatic logging and ESM) did not negatively influence the performance of their smartphone.

Weekly Interview: Day Reconstruction Method (DRM)

The DRM method [111] is employed in order to study the possible relations and causality between QoE ratings, QoS, and the user context. The users are interviewed on a weekly basis regarding their usage patterns and experience on the mobile applications along with their previous 24 hour period. During the interview, users elaborated on their responses from the ESM. In parallel, the collected data via automatic logging and ESM from the smartphone are visualized and compared with the data obtained via DRM. This method has been used for fast identification of any inconsistencies in

between the methods, *e.g.*, ESM and DRM. This way, we identified the factors influencing smartphone QoE for any particular user.

3.2.3 Study Participants And Collected Data

The study was conducted for four consecutive weeks in between February to March in 2011. 31 Android users were recruited, and in total there were three types of Android smartphones (Motorola, HTC and Samsung). The users were subscribed to four providers: 23 participants with Verizon, four participants with Sprint, three participants with T-Mobile, and one participant with AT&T. Two participants ($S1$ and $S27$) dropped out in the first three days of the study due to battery issues on their old smartphones. $S11$ collected only one week of data and then dropped out due to an inconvenience. Three users ($S2$, $S8$, and $S9$) experienced data logging outages due to malfunctioning software on their smartphone or an explicit altering of the logging. Table 3.1 presents the study participants (from left-to-right) the participant ID; gender; profession; smartphone type; age range; overall MOS (as reported in the online survey); the MOS with the highest occurrence (perception as derived from the study) together with the percentage of its occurrence amongst all levels; number of occurrences of low (MOS$=2$ or 1) separated by comma; and the total number of MOS ratings collected by the user. None of the participants had accessibility problems related to their smartphone use, and none of them admitted that they were adversely affected by the Electromagnetic Resonance (EMR) health issues in mobile phone usage. In total, $17,699$ hours of data are logged, which represent $87.8\,\%$ of the hours for the overall 28 days of study duration. The missing data was ranging between 3 hours to 378 hours for some users due to occasional malfunctioning of the software.

3.3 Results For QoE And Context (ESM)

In this section, the results related to QoE ratings and the user context are given.

3.3.1 QoE Ratings

In total, 7804 QoE ratings are collected from all the participants with an average of 9.29 ratings per day and per participant. The high ratings (MOS$=4$ or 5) are much more frequent than the low ones (MOS$=1$ or 2) as illustrated in Fig. 3.2. In general, the participants find their QoE acceptable; they have explained that they learned how to maximize their mobile application usage along their routine activities. The participants

S	Gender	Profession	Phone Type	Age	QoE - Survey	QoE - Study	MOS= 2,1	Total No of Rat.
2	M	Customer service	Samsung Captivate	18-24	5	4(47%)	4, 4	218
3	M	Owner, moving company	Motorola Droid	25-35	2	4(61%)	3, 0	181
4	M	Driver	MyTouch 4G	25-35	4	5(77%)	4, 0	233
5	F	Research assistant	HTC Incredible	18-24	5	4(79%)	5, 2	227
6	F	Admin. higher educ.	G2	25-35	4	5(52%)	1, 0	323
7	F	ICT Consultant	Motorola Droid X	25-35	5	5(89%)	5, 1	390
8	M	Web developer	Motorola Droid	25-35	5	4(54%)	4, 0	143
9	F	Medical adm. assis	Motorola Droid	25-35	5	5(66%)	4, 0	197
10	F	Nanny	HTC Incredible	25-35	5	4(60%)	4, 0	543
11	F	Unemployed	Sam. Vib. Galaxy-S	25-35	5	5(68%)	3, 1	62
12	M	Unemployed	HTV Evo (WiMAX)	36-45	4	5(78%)	3, 6	620
13	M	Uni. program mngt	Motorola Droid	25-35	5	4(35%)	25, 3	254
14	M	Contractor	Motorola Droid X	25-35	4	5(63%)	8, 9	369
15	M	Accounts coord.	Motorola Droid	25-35	4	4(84%)	4, 1	196
16	F	Operations analyst	Motorola Droid X	25-35	5	5(57%)	7, 4	327
17	M	System analyst	Motorola Droid	36-45	5	5(48%)	4, 5	240
18	M	ICT consultant	HTC Evo (WiMAX)	25-35	4	5(62%)	5, 0	209
19	M	Teacher	Motorola Droid	25-35	4	5(68%)	4, 18	317
20	F	Admin. assistant	HTC Evo (WiMAX)	25-35	4	4(97%)	1, 1	296
21	M	Univ. student	Motorola Droid	25-35	4	3(57%)	7, 1	195
22	M	Grant admin	HTC Incredible	25-35	5	5(43%)	10, 1	276
23	M	Graduate student	Motorola Droid 2	25-35	2	4(83%)	1, 2	137
24	M	Systems analyst	HTC Evo (WiMAX)	25-35	4	5(94%)	2, 0	198
25	F	Univ. student	Motorola Droid 2	18-24	5	4(63%)	16, 10	386
26	M	Senior adm assis	Motorola Droid	25-35	5	4(55%)	11, 20	253
28	M	Graduate student	Motorola Droid X	25-35	5	5(30%)	34, 15	251
29	M	Paramedic	Motorola Droid	36-45	4	4(48%)	33, 12	213
30	F	Housewife	Motorola Droid X	36-45	5	5(83%)	3, 1	341
31	M	Registered Nurse	Samsung Captivate	25-35	5	4(52%)	1,1	209

Figure 3.1: Participants: demographics and QoE ratings. Participants *S*1 and *S*27 dropped out early in the study [67]

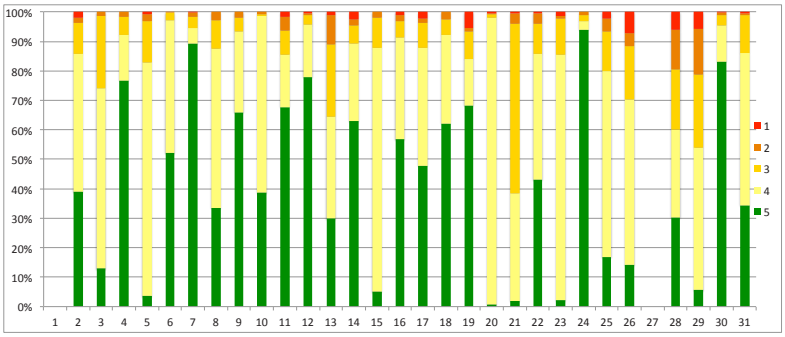


Figure 3.2: QoE ratings distribution for mobile users [67].

exhibited knowledge on circumstances that they can expect particular QoE depending on the network coverage and performance.

3.3.2 Applications

Based on the collected QoE logs, amongst the applications, there were standard Android built-in applications, *e.g.*, web or email applications, as well as a variety of specialized ones. After gathering all data from the study, the smartphone applications are identified (via Android Market) within the following 13 categories. They are presented in order of descending frequency of usage:

- **Communication:** Email, GMail, GTalk, Skype;
- **Web:** Default browser, Dolphin;
- **Social network applications:** Cooliris, Facebook, Foursquare, Okcupid, Tumblr, Touiteur, Twitter;
- **Productivity tools:** Astrid, Calendar, Callmeter, Outofmilk, Sandbox, Shuffle;
- **Weather apps:** Weather, Weather caching provider, Weatherservice;
- **News:** Espn, Foxnews, News, Newsfox, PenguinsMobile, PittFight, Reddit, Sports;
- **Multimedia streaming:** Listen, Lastfm, Pandora, YouTube;
- **Games:** Games, Touchdown, Words, Worldwar, WoW, Poker, Zyngawords;

- **Lifestyle apps:** Diet, Horoscope, Spark people;
- **Finance:** Stock;
- **Shopping:** Coupons, Craigslist, eBay, Starbucks, Starbuckscard;
- **Travel:** Locationlistprovider, Maps, Navigator;
- **Other:** Other applications.

3.3.3 Context

The applications were used mostly at “home”, “office/school”, and “indoor/other”, as depicted in Fig. 3.3. With respect to the social context, the applications were used on average 80 percent of the time when the users were “alone”. The dominant mobility levels for an application usage were “sitting”, “standing”, and “other” as specified by the user precisely as “in bed”. Low QoE ratings occur mostly when a person is at home or at school while being alone and sitting.

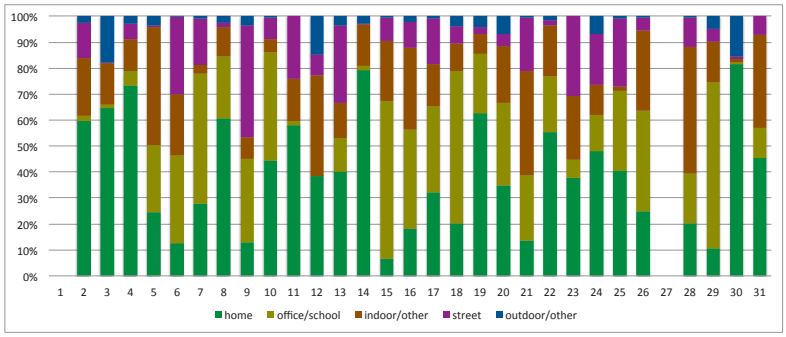


Figure 3.3: Mobile users’ locations distribution [67].

3.4 Factors Influencing QoE

In order to derive the factors influencing a user’s QoE, we have analyzed all DRM answers semantically. In order to give a richness of expressions, a word cloud is created that visualizes the word frequency in a user’s expressions as a weighted list. Font sizes are set in relation to the frequency of the corresponding word as illustrated in Fig. 3.4.

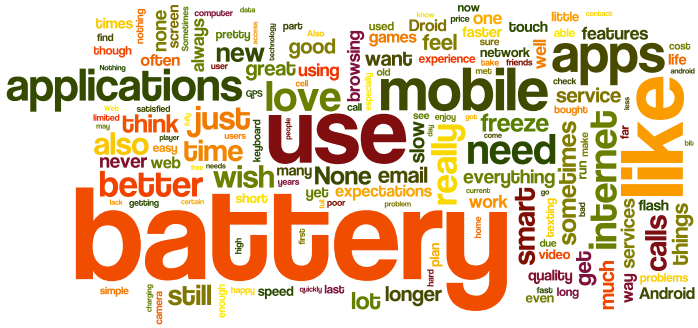


Figure 3.4: Expressions in user's interviews and surveys [67].

376 expressions from the 29 user's weekly DRM interviews, as well as the data from an online survey with 430 entries are used. The most frequently mentioned key words are applications, mobility, Internet, battery, performance (*e.g.*, "slow", "freeze"), as well as features such as Global Positioning System (GPS), camera, and Flash Player. Many of the user phrases are difficult to evaluate automatically. These words have been grouped into clusters by using the affinity clustering method, which then can be labeled along the identified factor. The coding and grouping of words into clusters were done by two independent researchers. We calculated the amount of phrases that are clustered to the same keyword by two reearchers, and their measure of agreement was 90 percent of the total clustered phrases. Most disagreements were related to the interface and application performance clusters, especially for cases where the participants were not clear on pinpointing the main issue. Finally, the following factors influencing the user's QoE have been distinguished.

3.4.1 Battery

Battery efficiency consistently influenced the experience of the smartphone users, as it limits their smartphone usage, especially at the end of the day, when the phone was completely discharged. *S*12 resolved that issue by carrying an additional battery. It can be concluded via Fig. 3.4 that the battery is one of, if not the most important influential factors on QoE based on the user interviews and the surveys.

3.4.2 Application Interface's Design

Application interface and the interaction were pinpointed very often; users did not like the position and the location of the buttons on the smartphone screen. They had difficulty with resizing or webpage scrolling, they did not agree with built-in dictionary items, and they complained about an inefficient manual input method, *e.g.*, the "fat finger" problem, thus a more adaptable user interface is suggested. Some users preferred interacting with a web-based interface of particular applications *e.g.*, Facebook, than with via the application widgets.

3.4.3 Application Performance

There were expressions used by the users while referring to low application performance such as "freeze", "sloppy", "sluggish", "speed", "performance", "usage of memory", and "sdcard". For some mobile applications, which users previously experienced on a fixed PC (*e.g.*, email), the expectations for performance were high. This resulted in low QoE. For those users who had a Personal Computer (PC) as an alternative device (*e.g.*, to receive/send emails), their rated QoE's were limited to reception of emails on the smartphone as they used the PC for sending emails. The reason for this is that typing on a real keyboard of a PC provides a better experience, especially for long text. On the other hand, some users preferred a smartphone to run most of the applications. For those users, mobile applications achieved enough usability to enable them not to use a larger and potentially more comfortable PC.

Two particular participants were adaptive, at the same time complaining about the applications such as instant messaging and security apps. *S*4 complained about the "stupid" autocorrect function of a messaging app, and *S*16 complained about a specific Virtual Private Network (VPN) application with poor usability to access corporate email.

Some mobile users expressed their tolerance for low application performance when they use it in mobile context. In parallel, users regularly mixed network performance with application performance metrics; for example, while saying "Skyping service is incredibly spotty", the concern is actually the underlying network connectivity of Skype service (*i.e.*, being spotty), not the application itself.

3.4.4 Phone Features

Smartphone users noted missing features of the phone, which then hindered their experience; for example, lack of a Flash Player, personalized alarm clock, special settings for vibrate-only mode, features for privacy, or a faulty GPS.

3.4.5 Apps And Data Connectivity Cost

In the online survey, many of the smartphone users indicated that the cost of applications and data usage prohibits them from experiencing these applications.

3.4.6 User Routine

The routines of the users implied that different sets of applications were used in the morning, in the evening before going to sleep, in the car, and outside the office. The user rating is influenced by the user's environment as well as the importance of the mobile application to the task at hand.

3.4.7 User Lifestyle

There were highly ranked applications that support a user's lifestyle choices, *e.g.*, sports, fashion, nutrition, and leisure. They are used on a smartphone due to their convenience of usage, *e.g.*, in the gym for logging the burnt calories, in the cafeteria for logging the caloric intake, or on the street while searching for a fancy restaurant.

3.5 The Role of QoS

Along the data analysis, not much evidence of the influence of QoS on the user's QoE were captured. There are low QoE ratings in our data, but there is no strong evidence for a low QoS. One of the reasons for that can be, as indicated previously, the fact that factors influencing a low user QoE are different than the QoS metrics that we have observed (*i.e.*, classicial metrics such as RTT and SRT). Another reason for that is the insufficient granularity of the QoS measurements with limited permissions, *e.g.*, non-rooted phones, to access network-level QoS metrics on user smartphones.

QoS is influenced by the choice of wireless access technology, that is, Wireless Local Area Network (WLAN), 2nd Generation (2G), 3G, 4th Generation (4G), and this eventually influences QoE. However, in this study it has been observed that the performance of the access network was not an issue, as users were often well-connected and had a choice of access networks (as ordered by an increasing nominal capacity): GPRS, Code Division Multiple Access (CDMA), Enhanced Data GSM Environment (EDGE), UMTS, Evolution-Data Only (EVDO), and High Speed Packet Access (HSPA). In addition, Worldwide Interoperability for Microwave Access (WiMAX) was available for selected users in selected urban locations: $S12$, $S18$, $S29$, and $S24$. Some users were connected over WiFi at home and at office, in order to assure better QoS. A diversity in

the connectivity through WiFi interfaces amongst the users is observed, ranging from 0 to 398.5 hours. In total, nine participants never used WiFi during the study, while six participants never turned the interface OFF. The latter allows the smartphone to connect to any available WLAN access point when detected. The influence of the WiFi signal strength on MOS was also studied, however no clear trends were observed.

The overall mean of RTT values that are collected in the study is 231 ms with a standard deviation of 73 ms. However, the values differ per MOS level; Fig. 3.5 presents the mean and the 95 % confidence intervals of SRT and RTT. It is observed that higher values of SRT and RTT correspond to lower MOS. For both measures, the confidence intervals get narrower as the MOS increases. The fluctuation in these measures observed for low ratings, especially for MOS values 1 and 2, is related to diverse influence of these measures on the application performance. The overlapping of the confidence intervals make the comparison difficult. It is observed that the recommended mean SRT for a smartphone application assuring the MOS level of 3 is 950 ms, while the MOS level of drops to 2 when the mean SRT increases to 1050 ms. Thus, roughly one second average SRT seems to be a critical point that the limits of acceptability. Similarly, the mean RTT of 220 ms corresponds to MOS level of 3, and the MOS value drops to 1 (a bad user experience), when the mean RTT increases to 263 ms. The throughput, *i.e.*, bytes received per second by the smartphone, on the user smartphones is also studied. Although we have observed that the mean throughput increases with the increase in MOS values, very wide confidence intervals exist. This indicates that there are many different throughput ranges resulting in the same MOS level.

The applications with many low QoE ratings, were multimedia streaming applications like Listen (audio feeds application), YouTube (video streaming), and Pandora (real-time radio streaming). Any participant in our study was using one of these applications in average 1.67 hours per day, which involved in average 50 MB of downlink and 1.7 MB of uplink traffic per day. A total of 1.38 GB of downlink and 0.5 GB of uplink traffic is observed in 28 days. A participant was running the "Listen" application in average for 0.8 hour, YouTube for 0.34 hour, Pandora for 0.5 hour per day, although most of the usage was observed within a fixed group of 10 participants with a varying distribution of population. The applications Listen, YouTube, and Pandora involved in average 32.7 MB, 8.15 MB, and 8.36 MB of downlink traffic per day, and 1.03 MB, 0.3 MB, and 0.36 MB of uplink traffic, respectively. Some of the mobile users using these applications rated 1 for MOS, being critical of its performance, but some other users were tolerant, knowing that they gain possibility of accessing these applications while being mobile, and paying a "performance price" for that. Low ratings can be related to the fact that these applications required high network capacity or, as learned from the participants, some application widgets were buggy and influenced the application performance. Additionally, some participants were streaming pre-downloaded

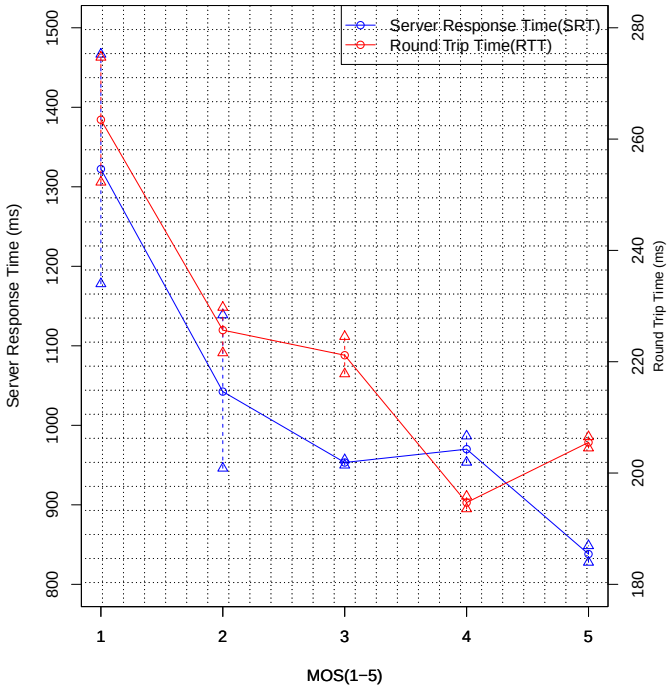


Figure 3.5: Mean RTT and SRT (95% CI) vs. MOS [67].

MPEG-2 Audio Layer III (MP3) files (*i.e.*, from local storage), while driving, riding the bus, or walking on the street in order not to get influenced by the unstable wireless network conditions.

The 4G (WiMAX) service was rarely used because of its unavailability, as *S*18 claimed: *"Unfortunately, I don't get 4G in (A). And when I'm in (B), the 4G connection keeps switching ON and OFF, and the notifications are just annoying. So I keep 4G switched off"*. *Another user (S*20*) said: "My phone can operate on a 4G network, but I usually keep it set to 3G because in my experience, the 4G is not considerably faster and just eats up my battery... Generally I keep 4G turned OFF unless I am doing something network intensive and I know it is available"*. It was surprising to hear that, because according to the results of performance measurements that are conducted for the 3G and 4G networks, use of the 4G network results in better QoS parameters than 3G. Thus, it is presumed that the applications used by this particular user worked sufficiently well on 3G. This contradicts with our initial hypothesis that a mobile user always wishes to have the best possible and fastest service.

Additionally, using the DRM results, it is discovered that the users choose network interface to connect to the Internet in order to save battery and to avoid occasional disconnectivities. Users who were able and willing to charge their phones often preferred the access technologies in the order of WiFi-4G-3G (*i.e.*, first choose WiFi, if that does not exist choose 4G, if the latter two do not exist choose 3G), while this changes to 3G-WiFi-4G for the ones who charged their phones less often. A common feeling amongst our users was that 4G was as good as WLAN but drains too much battery. In addition, 4G coverage is a problem, so users who are subscribed to the network providers with 4G support are not necessarily always within the 4G coverage. This causes connectivity oscillations between 3G and 4G, putting users at risk of instant disconnections, and also resulting in draining extra battery.

3.6 Study Limitations

The limitations of this studied are listed as follows. Firstly, the study has been conducted on self-selective group of smartphone users with self-selective set of applications. One reason is that we developed the automatic logging application only for the Android OS. The QoE ratings of participants in this study are studied and some participants who use many applications within a short time are identified. The maximum number of applications that were used within the last 3 minutes of the given QoE rating was 19. It is observed that the applications are rated with high MOS values, 4 and 5, if there are more applications used within the 3 minutes time period. It is hypothesized that either such a user is more advanced and uses many mobile applications,

which he/she is satisfied with; or by using so many applications in a short time, this user has no time to pay attention to details of his/her low experience, just grasping the essence of the information provided by the application for the given situation at hand. It would also be more contributive if it were possible to clearly pinpoint the type of access network technologies, *e.g.*, 3G, 4G, in relation to the MOS values. In addition, the extreme conditions, *i.e.*, very high RTT or SRT values, were not be able to be caught due to lack of granularity of the selected samples for the study. Capturing worst conditions in real-life scenarios is challenging and non-trivial, so a longer time of study could be suggested in order to capture all scenarios a user might experience in daily natural life. Moreover, there would have been a wider variety of measurable QoS network metrics, *e.g.*, delay jitter and packet loss if the smartphones were rooted and privacy concerns were not important.

3.7 Summary

In this chapter, the research towards understanding a mobile users' Quality of Experience (QoE) in their natural daily environments is presented. Our approach employs both quantitative and qualitative procedures, where the user becomes an active participant in the research. First, it requires gathering in-situ spontaneous information about the user's smartphone experience for a set of widely used smartphone applications; employing the Experience Sampling Method (ESM) for interaction with the user directly after each application usage. Second, it requires a retrospective analysis of the user's experience and the state of factors influencing it; employing the Day Reconstruction Method (DRM) to assist with the recollection of the past 24 hours. DRM was supportive in validating the collected data through the logger application (automatic logging and ESM). In this chapter, the analysis of the collected data is given by highlighting some factors that impact the user's QoE.

The novelty relates to the factors that influence QoE, including application interface design, application performance, battery efficiency, phone features, application and connectivity cost, user routines, and user lifestyle. These factors go beyond the usual usability, usefulness, and user value factors, as already indicated in the literature. The role of QoS was also studied, and it was indicated that increased Server Response Time (SRT) and Round Trip Time (RTT) values reduce the Mean Opinion Score (MOS) values. In general, it is observed that the users were well connected, and used their applications mainly while in fixed indoor position and while being alone.

One of the most important findings of this study was that there were many low user ratings that are associated with the multimedia streaming applications on the smartphones. One reason for this might be due to the occasional disturbances in the network

level that manifest themselves at the user interface as stalling events (*i.e.* freezes). As the QoS metrics studied in this chapter (*e.g.*, RTT and SRT) can not fully match with the end-user QoE, it is necessary to study the metrics with a more accurate measurement point. In addition, video streaming on smartphone applications are highly power consuming, in the long term influences negatively the battery lifetime. In the next chapter, we study the real time video streaming on the smartphone at the packet-level, *i.e.*, we study QoS metrics such as packet delay variation and maximal burst size, towards the goal of obtaining a better match between QoS and QoE in video streaming.

Chapter 4

Network-based Instrumentation of Smartphone Video QoE

"*The two most powerful warriors are patience and time.*" –Leo Tolstoy

---

4.1 Introduction

Maximizing the delight of customers, *i.e.*, their QoE, is the goal of all service and product providers, in order to obtain more customers and to reduce the customer churn rate. The customer is not likely to be interested in knowing QoS metrics such as PDV (*i.e.*, the difference between the inter-arrival time of packets), wireless signal strength, or packet loss, but rather in the perceived quality of the service or application. Customer feedback plays an important role in measuring the satisfaction level, and network/service providers research a set of measures that can predict the satisfaction of their customers without the need of direct feedback. QoE models can satisfy this need, however, it is often difficult to perform accurate measurements in user's natural environments.

In this chapter, we present and analyze the QoE model for streaming video over the Internet through a cellular radio network. We derive a QoE model that maps measurable QoS metrics for this service onto the satisfaction of an individual. The models are meant to be used by decision making engines for various use cases including network selection [243] and application monitoring. In particular, these models are used for the decision making engine in PERIMETER, a Specific Targeted Research

Projects (STREP) project granted by EU FP7.

PERIMETER's main objective was to establish a new paradigm for user centricity in advanced networking architectures [11]. PERIMETER aims to provide mobility that is transparent to the user (seamless mobility) in heterogeneous networks and tackles the problem from a user-centric perspective. Therefore seamless mobility can be controlled by actual user needs in addition to QoS requirements and business considerations. We present a measurement module that supplies the PERIMETER framework with QoS statistics about ongoing connections of a smartphone. With the information obtained from all available network links, PERIMETER framework takes decisions whether to handover to another network to maximize QoE and to maintain Always Best Connected (ABC) principle [243].

The QoS statistics are ideally measured by low-level packet handling mechanisms in the Opinion Score (OS). Providing accurate statistics to decision engines, *e.g.*, for handover, is vital, otherwise unnecessary actions may be triggered, which eventually increase the cost of backbone processes in the Internet infrastructure and delays towards the end-user [235]. Moreover, with the aim of achieving ABC transparently to the end-user, the processing time of the measurements plays a large role in real-time applications. While taking QoS measurements, the end-user preferably should not notice any degradation in performance. Also, privacy and security during the measurements must be maintained at all times. Thus, the preferred place to conduct network measurements is inside the OS, *i.e.*, in-kernel [248].

This chapter is organized as follows. We present the theoretical aspects in Section 4.2, implementation in Section 4.3, and brief explanation of the experimental testbed in Section 4.4. Then, we present real measurements via the conducted experiments to model PDV and User Rating (UR) in Section 4.5.1, the benefit of using the EWMA techniques on human perception statistics in Section 4.5.2, the definition of Maximal Burst Size (MBS) and observations regarding the effect of MBS on the user perception in Section 4.5.4. The chapter is summarized and concluded in Section 4.6.

4.2 Acquisition of QoS and QoE Metrics

In this chapter, the QoS metric of interest for our QoE model is the PDV. We present a QoS analysis of the PDV of a streaming video connection conducted in real-time. PDV is defined in [238] as the difference in one-way delay between packets, while ignoring any lost packets. Calculating the PDV can be done in various ways, *e.g.*, as described in [245] and [244]. RFC3393 [238], however, proposes the procedure outlined in Algorithm 1.

Before settling for a PDV algorithm, we experimented and compared the accuracy

Algorithm 1: Computing Packet Delay Variation (PDV)

Calculates the Packet Delay Variation (PDV) of a packet stream
For each received packet
while *Current time interval* **do**
  $T_{S,n} \leftarrow$ departure time of packet n at sender's side
  $T_{R,n} \leftarrow$ arrival time of packet n at receiver's side
  Calculate D_n, update D and N (see Eq. 4.1)
$PDV \leftarrow$ standard deviation of D_n values collected per time interval (see Eq. 4.2)

(*i.e.*, goodness of fit) of different PDV algorithms. The Ring Buffer (RB) algorithm yields the most accurate results with the fastest execution time. The RB is considered to be efficient in terms of buffer allocation during run-time. Other algorithms are storing packet parameters in a database, calculating the PDV with respect to each of the stored parameters at a later point in time. As a drawback the RB algorithm does not store packet parameters longer than one time interval.

PDV calculations require end-to-end measurements between the two peers (the video streaming server and the end user device running a video stream player application in Fig. 4.1). The unsynchronized one-way-delay of one packet D_n is calculated by the subtraction of the departure timestamp $T_{S,n}$ from the arrival timestamp $T_{R,n}$ obtained at both ends of the communication channel. The values of these metrics are stored at the reception of the first packets in each interval. The next packets' timestamps are compared with the initial stored values of the metric. Having the first packets as basis, we overcome the time synchronization problem.

$$D_n = T_{R,n} - T_{S,n} \qquad \text{[ms]} \tag{4.1}$$

We define the PDV as the standard deviation of the end-to-end delays between packets that are measured within a given interval. The packet delays are stored for each time interval, where each time interval is one second. The PDV is then given by

$$PDV = \sqrt{\frac{1}{N-1}\left[\sum_{n=1}^{N}({D_n}^2) - N\overline{D}^2\right]} \qquad \text{[ms]} \tag{4.2}$$

where D_n is the one-way-delay (in milliseconds), $\overline{D}$ is the average delay (in milliseconds), and N is the number of packets per second. PDV is updated each time a packet arrives. We note that the delay variation is being calculated by ignoring lost packets as described in the definition of one-way-delay variation in [238].

As a result of the real-life experiments applied to individuals, the PDV values are directly matched when the User Rating (UR) values are received. The user perception of the previous network condition directly influences user's later decisions. Instantaneous matching did not satisfy enough to prove that there is high correlation between the PDV and the UR values. The forget factor, the *recency effect* [247], and sudden change of PDV during voting are the possible reasons. Inclusion of the remaining effects of the previously obtained outputs into the calculation of the current output is made possible by the EWMA approach (as computed in Eq. 4.3), *i.e.*, via time-wise memory-based where the recent values are more memorable [240], [236]. When obtaining the UR, the EWMA is used for computing the correlation of instantaneous user perception against the QoS metrics. Thus, by using EWMA on PDV values, we tried to imitate the human perception process to a certain extent [240]. Similar approaches such as the M-Model [236], have been used previously. The positive impact of the EWMA on the quality of the matching between PDV and the user rating will become apparent in Section 4.5.2. The EWMA is computed as follows:

$$PDV_{\mathrm{EWMA}}(i) = (1-\alpha)\cdot PDV_{\mathrm{EWMA}}(i-1) + \alpha \cdot PDV(i) \qquad \text{[ms]} \tag{4.3}$$

where $PDV_{\mathrm{EWMA}}(i)$ is the current (at i^{th} interval) exponential weighted moving average PDV, $PDV_{\mathrm{EWMA}}(i-1)$ is the previous (at $i-1^{\mathrm{th}}$ interval) exponential weighted moving average PDV, and $PDV(i)$ is the current PDV. α is typically set to 0.25 [240], with the motivation that users assess the quality based more on their memory as compared to what they instantaneously perceive [236].

4.3 Implementation

A generic measurement module is used to assess the QoS metric (PDV and its statistics). The measurement data are computed from information located in protocol headers. Thus, the module can piggyback on any suitable protocol that allows for extensible headers. For example, IPv6 is a candidate for our measurement module as it allows for extensible headers by properly setting up the next header field. TCP has similar abilities to extend the header. Tunneling protocols are also targeted for our measurement module as they easily allow methods for customizing headers, *e.g.*, Generic Routing Encapsulation (GRE), UDP over IP (UDPIP), or Layer-2 Tunneling Protocol (L2TP). In our case, we implemented the measurement assessing algorithm on top of a User Datagram Protocol (UDP) tunnel. The measurement module needs the following two fields to work properly: the *sequence number* and the *time stamp*. These fields are preferably 32 bit long.

Figure 4.1: Schematic overview of the experimental set-up [171].

It is important to send the QoS metrics from the sender to the receiver's side since the actual PDV computation is done at the receiver's side. Storing a history of QoS metrics, and post computation of the data is not feasible. Upon the receipt of each packet, we rather update the average inter-packet arrival and departure times together with the deviation of the instantaneous values from the instantaneous means (see 4.2 and Algorithm 1). PDV is calculated by deploying these updated values at the end of each time interval [238].

Any interested entities, usually residing in user-space, can obtain the PDV metric by accessing the measurement module in the kernel. Contact with the measurement module can be established through a local UDP socket or another OS specific system, *e.g.*, the `/proc` file system in LINUX.

4.4 Experimental Testbed

We have set up a testbed to obtain live QoS and QoE measurements. These values allow us to define a QoE model. In our experimental setup, as depicted in Fig. 4.1, we streamed a video over the Internet from a server to an Android smartphone (HTC G1). The Android phone was connected to the Internet via 3G to the streaming server site. The 3G connection was a regular data subscription from a popular Swedish network provider. The video data is streamed from the server over the Internet to the smartphone, which is connected to the Internet via 3G.

We used the Darwin Streaming Server (DSS) framework [234] on a LINUX (2.6.27) Ubuntu 9.04 (installed on a fixed PC) for streaming media to the Internet. The streamed video is MPEG-4 compressed with dimensions 240×180 pixels, has 24 kHz AAC stereo sound, 23.97 frames per second (fps), and is streamed at a rate of 325 kbps. The RTSP protocol [249] is used for streaming the video from the server to the Android phone. After the RTSP session is initiated, the RTSP client periodically sends RTSP requests to the server as a feedback control mechanism. A session identifier is used to keep track of the sessions as needed, which eliminates the need for permanent TCP connections.

We implemented the QoE assessment tool for Android that consists of measurement module synchronized to our video streaming application. The measurements are recorded by our kernel module, extracted and written to a file by the streaming video application. The files were then retrieved and analyzed with statistics software.

Using this tool, we conducted a set of experiments, where a number of individuals rated a streamed video. While watching the video, in parallel they rated the video by pressing one of the five buttons on the touch screen corresponding to the UR. In total, one hour of data was recorded from 15 adult test subjects. We conducted the experiments in diverse environments. The test subjects were selected from various backgrounds.

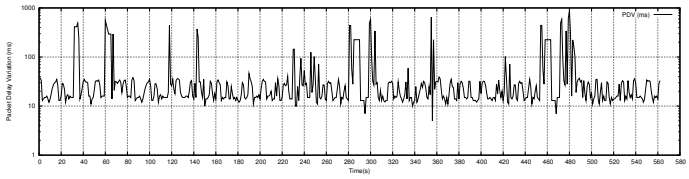
4.5 Results and Observations

4.5.1 Packet Delay Variation

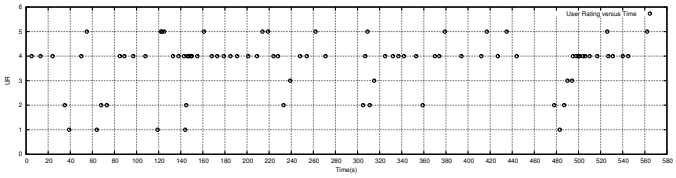
In Fig. 4.2, the relation between the measured PDV (QoS) and UR QoE values is depicted. The graphs show an excerpt of the time series obtained during our experiments. It is a typical example that shows that a user is reluctant to give high user ratings when the video quality improves, but immediately reacts with a poor user rating when the quality degrades. The model that best correlates the PDV and the UR is given by Eq. 4.4

$$UR = -0.88\ PDV/\mathrm{ms}^{0.27} + 6.38 \tag{4.4}$$

where PDV is measured in milliseconds (ms). Observe that the models in this chapter are valid for the user rating values within the (typical 1-to-5) MOS range. When high PDV values appear, the user would see a freezing video, *i.e.*, the most recently received video frame, until a new video frame is displayed. The latest frame would be played for a shorter time than expected in order to compensate the timing for the expected frame [237]. In case of packed loss, the lost frames will not be shown and the image will be skipped. Various queues on the way from the streaming server to the video player cause the PDV. Moreover, the 3G operator contributes to increases in PDV by attempting to compensate packet loss due to erroneous transmission by retransmissions. We also note that TCP is adequate in recovering from lost packets, but it amplifies the PDV while doing so. This is one of the reasons why TCP-based streaming is not preferred for real time video applications.



(a) Measured PDV values.



(b) Measured UR values.

Figure 4.2: Experiment excerpt of the time series of measured Packet Delay Variation (PDV) and User Rating (UR) values. There is a clear negative correlation between peaks in the PDV and negative peaks in the UR [171].

4.5.2 Exponential Weighted Moving Average

The UR obtained at a given time is strongly impacted by the previously measured PDV values. The goodness of the fit, the R^2 value for the power model, is improved from 0.25 to 0.51 when the EWMA technique is applied. This is an increase of goodness-of-fit of over 100 % as stated in Table 4.1. After EWMA was applied, the model evolved to Eq. 4.5, where PDV is in milliseconds (ms). From Table 4.1, we can conclude that PDV with EWMA yields, amongst the others, the most accurate QoE model.

$$UR = -9.10\, PDV/\mathrm{ms}^{0.08} + 16.18 \tag{4.5}$$

4.5.3 On-Off Flushing Behaviour

An oddity that we have observed during the experiments is the *on-off flushing* behaviour. Namely, the data transfer was suddenly stopped without any noticeable visual warning signs. After a seemingly random amount of time a burst of data, that was supposed to be sent during the outage, was delivered. One of the reasons for *on-off flushing* behaviour could be that the 3G network tries to recover from packet loss and only releases all packets when the lost packets are recovered. After the burst of data,

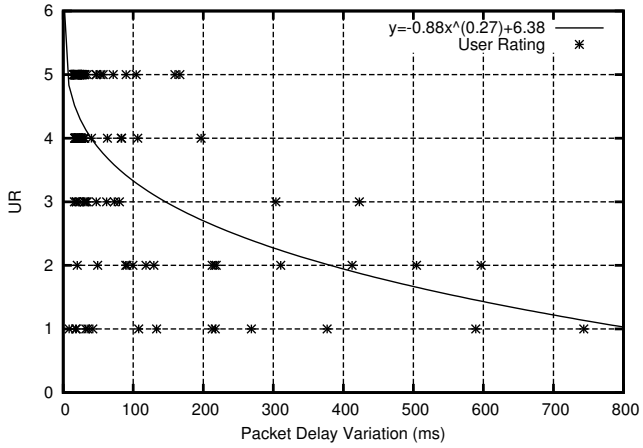
Table 4.1: Correlation between the User Rating (UR) and the Packet Delay Variation (PDV) with and without Exponentially Weighted Moving Average (EWMA) is illustrated amongst different models.

QoE Model	Linear	Log	Exp	Power
Coefficient of determination	R^2	R^2	R^2	R^2
PDV w/ EWMA	0.31	0.42	0.40	0.51
PDV w/o EWMA	0.17	0.19	0.22	0.25

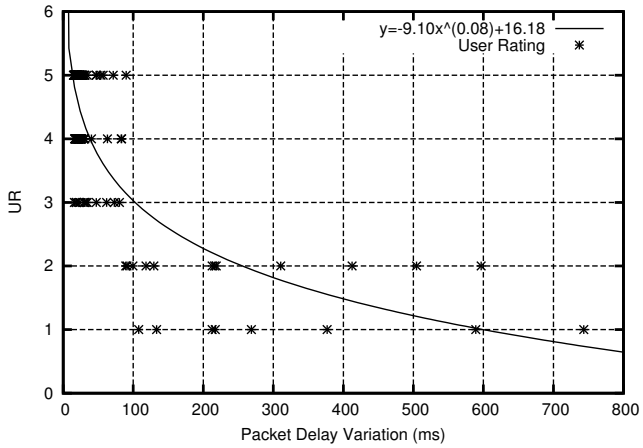
the data transfer continued as before. During the *off* timespan, the video on the screen stalled. As a consequence, very low UR values were registered.

In extreme cases no packets arrived for about up to 30 seconds. This *on-off flushing* behaviour degraded the correlation between the UR and the PDV results, drastically. Fig. 4.3 depicts two datasets, where we fit the PDV values against the UR. In Fig. 4.3(a), an excerpt of the unaltered dataset is plotted and fitted against a power function. The power model is chosen since it gives the maximum least square value, thus the best fit amongst other models as shown in Table 4.1 and Table 4.2. The goodness-of-fit, R^2, for the whole dataset is 0.51. Collection of consecutive identical values indicates that the PDV remained unchanged and no packets are received during those intervals. This is the case when data loss occurs, and we eliminate these values as follows. If we receive the identical PDV value from the measurement module for four consecutive times, we remove that PDV value from the dataset. Thus, with the new data set, we obtain Fig. 4.3(b), and the new R^2 value is computed as 0.68. With this operation, in 4.3(b) a better correlation of determination (R^2) value is obtained; improved from 0.51 to 0.68.

Referring to the large difference in R^2 values, instances of *on-off flushing* behaviour can be seen as unwanted anomalies in the dataset. Another factor that could affect the correlation in a similar way is known as the recency effect [247]; user's perception changes faster from *good* to *bad* than from *bad* to *good*. This is reflected in the UR dataset as a skewed and in particular quicker reaction to QoS degradation, thus decreasing correlation between the UR and QoS. We haven't yet investigated in detail the recency effect in our dataset.



(a) Including *on-off flushing* behaviour.



(b) Without *on-off flushing* behaviour.

Figure 4.3: Fitting of the UR against the Packet Delay Variation (PDV) with the raw data, *i.e.*, with the *on-off flushing* effect, shown in 4.3(a), is improved to 4.3(b) when the replicated PDV values (affected by *on-off flushing* behaviour) are filtered out. Corresponding model equations are denoted on the top right corner on both graphs [171].

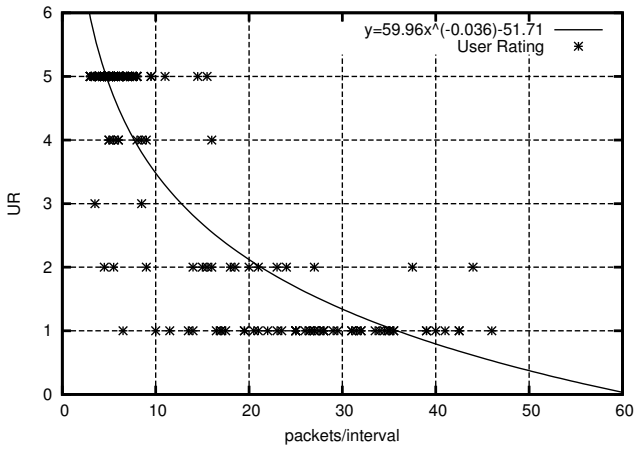


Figure 4.4: Fitting of the UR against the Maximal Burst Size (MBS) [171].

4.5.4 Maximal Burst Size (MBS)

In previous research, measuring the delay variation is based on the assumption that the inter-departure time of the packets is uniform. This assumption does not tell, however, whether QoS degradation originates in the network or if the application layer is responsible. Inter-departure times of packets were not observed to be uniform in our experiments. For this reason and out of curiosity, we observed the MBS. The number of departed packets per millisecond is traced. We then referred to MBS as the maximum amount of packets received within this time interval, which can also be considered as the *clumping size*. The MBS was plotted and fitted against the UR in Fig. 4.4. The model that we found is given by Eq. 4.6.

$$UR = 59.96MBS^{-0.036} - 51.71 \tag{4.6}$$

where MBS is measured in packets per interval. The UR is decreasing as the MBS is growing. The peaks in the number of transmitted packets per given time interval have a negative impact on the UR. This is a somehow surprising effect and can be explained by the fact that large MBS values indicate a *flush* following an *off* timespan. When we look at *pcap* traces, we can identify MBS peaks. We distinguish two cases; short trails of identically-sized and very large sized packets with different content, and long trails of differently sized large packets. Only in the second case we see a severe video

interruption as packet transmission is stopped when the streaming application sends feedback to the streaming server.
Our first guess is that the streaming video application, at least on Android, is constrained to process a certain amount of data during a given time. Larger amounts of information, reflected in the MBS, will result in deterioration of the video quality. Thus the application might be the bottleneck.

Table 4.2: The correlation for the UR in relation to the PDV and MBS amongst different models.

QoE Model	Linear	Log	Exp	Power
Coefficient of determination	R^2	R^2	R^2	R^2
PDV w/ on-off flushing	0.31	0.42	0.40	0.51
PDV w/o on-off flushing	0.47	0.63	0.54	0.68
MBS w/o on-off flushing	0.73	0.77	0.75	0.78

4.6 Summary

In this chaper, we presented the results and observations from a QoE assessment tool that consists of an in-kernel measurement module and an interactive video streaming application. We conducted a set of experiments, focusing on PDV and showed that the system is adequate to predict UR values, related to the QoE of streaming video users. Fitting the QoS on QoE metrics in a mathematical model enables a decision maker of QoS level for which application would need to make a decision, *e.g.*, for vertical handover, to perform seamless communication with the goal of maximizing QoE.

The importance of utilizing EWMA during the analysis of data related to the human perception is presented, *i.e.*, it emulates human memory. From the results of our experiments, we concluded that applying EWMA techniques to our collected data increases the goodness-of-fit of the power model by 100 %. In addition, *on-off-flushing* behaviour and its effects on the correlation between the PDV and the UR values are presented.

An additional parameter, the MBS, is measured and correlated with UR values. MBS, which is measured on the receiving side shows the density of the packets departed from the streaming server within very short time interval. The analysis regarding MBS is under investigation. Appropriate QoE models for these QoS metrics are being evaluated by statistical inference.

Even though the experiments and observations in this chapter merely focus on 3G network so far, the study can be complemented with QoS and QoE models from all available wireless networks of a device. As a support for vertical handover mechanisms, the complete information, obtained from within different UDP tunnels which are bound to WLAN, 3G or 4G interfaces, can be simultaneously provided to decision maker mechanism to choose the best available network.

In this chapter, we presented that the QoS metrics obtained from the kernel-level such as PDV and MBS indicate the perceived quality. However, still these models need to be improved, as packet level measurements do not fully tell about the user perceived quality. Kernel level measurements are not also practical, as deployment of kernel modules on any smartphone is a difficult task, due to the needs of root access and specific configurations. Thus, in the next chapter, we study the video streaming this time on an open source video player application, the VLQoE video player. We measure and relate the user's perceived quality of experience with respect to the measurements obtained rather at a point in the stack closer to the user, the user interface. This way, we are able to match how the video is actually presented to the user, and how the user perceives the quality and rates it. We focus on the application and user interface metrics, and study smartphone video QoE in the next chapter.

Chapter 5

Application-based Instrumentation of Smartphone Video QoE

"*The energy of the mind is the essence of life.*" –Aristotle

---

5.1 Introduction

A traditional network-centered QoS approach can capture the key influential factors, such as packet loss or delay, on the end-user perceived quality. However, the recent research suggests that QoS has to be complemented with a user-centric approach [112], namely QoE, to satisfy end-user requirements and expectations. QoE assessment is preferably conducted close to the end-user, *e.g.*, at the user interface. This way, as compared to the network-centered QoE approach, it is less complicated to interpret the relationship between the impairments at the user interface and the subjectively perceived QoE.

Amongst the available services and/or applications, the most bandwidth demanding ones are the video streaming applications [113] as they have large size end-to-end data delivery requirements. The end-user perceived QoE of the video applications is highly depending on, amongst others, the quantity and the duration of the freezes during the video streaming sessions [115] [117]. There are existing approaches to

improve perceived quality such as "download-and-watch-later", *e.g.*, YouTube, at the cost of initial download/waiting time. However, those approaches do not suit for real-time video streaming such as live-broadcast of a soccer match, as the user expects to receive the video content at the same time as the content is broadcast. In order to study QoE on the smartphone-based real-time video applications, it is important to consider smartphone user interface, as it is the location of the concrete evidences, *e.g.*, temporal impairments, which can directly be perceived by the users. The reason for occasional short-term or long-term temporal freezes [119] in the video playout might be due to the impairments at low layers in the network stack. In previous studies [120], the network traffic is studied within the network-layer in two different states: ON during a packet flow, *i.e.*, burst; and OFF when there is no packet flow, *i.e.*, zero throughput. It is necessary to complement the network-based two-state ON/OFF model with the user interface measurements, and then find out the distribution of the inter-picture times D_{p}, *i.e.*, the time gaps between two consecutive pictures displayed on the smartphone screen, in the user interface, which later might help to understand the relationship between different layers in the network stack. Yet, and to the best of our knowledge, there is no smartphone-based QoE tool that can simultaneously record potential QoE metrics at different layers of the network stack together with the sensor data.

In this chapter, QoE instrumentation of the smartphone-based VLC player [121], VLQoE, is introduced. Then, the studies conducted with VLQoE, which are based on the inter-picture time, are presented in two parts. The first part of the study involves an in-depth analysis of the inter-picture time metric, where the inter-picture times of the video is studied with a two-state (ON/OFF) model. ON state is referred to the state during a smooth video playout; and the OFF state is referred to the state during a video picture freeze. The second part of the study involves the subjective study with the focus on the inter-picture time metric quantifying a picture freeze. The influence of the inter-picture time on the end-user perceived quality of the video, measured with five-level ACR scale (1-bad, 2-poor, 3-fair, 4-good, 5-excellent), is presented. The user response time, *i.e.*, the time it takes for user to react to a long picture freeze, is investigated in order to understand the user tolerance levels to the inter-picture times. The user interface also enables a user to indicate a freeze event by interacting with the freeze button, which helps us to quantify a freeze from the picture delay perspective.

The remainder of this chapter is structured as follows. Section 5.2 presents the definition of proposed metrics as well as the detailed description of the VLQoE tool. Section 5.3 presents the details of the VLQoE tool together with its validation. Section 5.4 describes the experiment testbed and the experiment methods of our study. Section 5.5 presents the results of the two-state ON/OFF modeling based on the inter-picture time, and also the results of the subjective study that identifies the relationship between the

inter-picture time and the end-user perceived quality. The study limitations and the summary of the chapter are given in Section 5.6 and 5.7, respectively.

5.2 Inferring the Video QoE on the Smartphone

This section presents the studied metrics at the user interface in our work, and explains in details how they are related to QoE. The primary metric in this study is the inter-picture time, D_{p}. Afterwards, on the basis of D_{p}, two-state modeling is employed in the first part of our study. This is followed by a second part including a user study, where the minimum perceived D_{p} for each user is investigated. This section is concluded with a study on the user response time that reflects the time it takes for the user to react to a perceived video *freeze*.

5.2.1 Inter-picture Time (D_{p})

Video is delivered to the smartphone as a series of pictures and then displayed on the smartphone screen. The inter-picture time D_{p}, the time gap between two consecutive pictures displayed on the smartphone screen, are calculated as in Eq. 5.1. $T_{\mathrm{p}}(k)$ is the timestamp when the k^{th} picture is displayed on the smartphone screen.

$$D_{\mathrm{p}}(k) = T_{\mathrm{p}}(k) - T_{\mathrm{p}}(k-1) \tag{5.1}$$

5.2.2 Two-state (ON/OFF) modeling

In the first part of this study, the video streaming is investigated with the assumption that the video stream follows a two-state model, *i.e.*, ON/OFF, throughout a video streaming session, as shown in Fig. 5.1. The *ON* (time interval of a smooth playout) and *OFF* (time interval of a freeze) states are defined based on the D_{p} metric. An exponential two-state model is considered for the ON and OFF durations, as the modulating ON/OFF process is assumed to be memoryless. In [157], it is stated that 100 ms is defined as the maximum tolerance threshold for the user to feel that the system is reacting instantaneously. Thereby, 100 ms has been chosen as the state boundary between ON and OFF.

The Maximum-Likelihood Estimation (MLE), developed by Fisher [151], is a standard approach for parameter estimation, *i.e.*, to find the probability distribution that makes the observed data most probable. It has optimal estimation properties such as sufficiency, consistency, and efficiency. Thus, the MLE of the durations spent in each state (ON/OFF) are calculated as in Eq. 5.2−5.3. $L(\lambda)$ is the likelihood function for

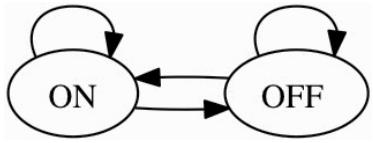


Figure 5.1: Illustration of transitions between the two states [114].

exponential distributed samples; n is the number of samples; $\hat{\lambda}$ is the MLE for the rate parameter, *i.e.*, it is used to calculate the value that is most likely to occur. $\bar{x}$ is the mean of the samples in exponential distribution.

$$L(\lambda) = \lambda^n \exp(-\lambda\, n\, \bar{x}) \tag{5.2}$$

$$\hat{\lambda} = \frac{1}{\bar{x}} \tag{5.3}$$

In our study, each sample x represents a duration instance of a state. Thereby, two different sets of instances are obtained, *i.e.*, ON and OFF.

5.2.3 Inferring QoE based on Inter-picture Time

In a realistic setting, users often watch videos that are at least longer than a few minutes. Thus, it is challenging to assess the instantaneous user-perceived quality at a particular point in time.

In this study, an extended ACR scale is used, complemented with extra user indications against the perception on the visual temporal impairments such as "freezes". Firstly, all the user indications with respect to the stalling events are studied together with the user ratings on the five-level ACR scale. The maximum inter-picture time values , $D_{\mathrm{p_{max}}}$, amongst all D_{p} values that lie in-between the two consecutive user indications, T_i and T_{i-1}, are studied. Thus, for each user indication there exists a unique $D_{\mathrm{p_{max}}}$ value. The display timestamp of the picture with the maximum display duration, $D_{\mathrm{p_{max}}}$, is denoted as $T_{D_{\mathrm{p_{max}}}}$ (see Fig. 5.3). In addition, the time gap in-between $T_{D_{\mathrm{p_{max}}}}$ and the corresponding user indication, T_i, is calculated as in Eq.5.4.

$$\Delta(T_{i-D_{p\mathrm{max}}}) = T_i - T_{D_{\mathrm{p_{max}}}}; T_{i-1} < T_{D_{\mathrm{p_{max}}}} < T_i \tag{5.4}$$

First Low User Rating Time ($T_{\mathrm{LowRating}}$) and a Number of Alarms: In the literature [117] [126], it is stated that the influence of stalling event on QoE shows

significant differences with the video length. Therefore, a sufficiently long video is used in this study, and the influence of stalling events is investigated, while considering the streaming duration before the first low user rating. In the impairment definition of five-level ACR scale, the ratings below 4 reflects an annoyance of the user [55]; and MOS = 4 is the lower boundary for an acceptable voice quality [158]. In this study, the *Low Rating* is considered as the rating less than 4 (*i.e.*, UR $<$ 4), together with the underlying temporal reasons. $T_{\text{LowRating}}$ is the time it takes from the start of the video until the time of the first low user rating (UR $<$ 4) per video session, and we study this metric in details. The low user ratings (UR $<$ 4) are matched to the underlying temporal reasons measured objectively at the user interface. The frequency of the freezes and the corresponding subjectively perceived quality are explored. The video pictures with the display duration higher than 100 ms were considered as *freezes*. The number of freeze instances (*alarms*), each with inter-picture time, D_{p}, higher than 100 ms, are recorded. In parallel, the streaming duration until the first low user rating, $T_{\text{LowRating}}$, is recorded. The analysis is performed for each user and for both streaming protocols, RTSP and HTTP for the purpose of generalization of the results.

5.2.4 Minimum Perceived Inter-picture Time

Pastrana *et al.* [159] stated that a single 200 ms long frame freeze was detected by all users in a 10 seconds-long sequence with CIF resolution. The perceived video freeze depends on the duration of the total duration of the video sequence. Staelens *et al.* [117] used short and long video sequences that comprise video freezes, which last up to 400 ms, and the authors claim that they have obtained different results with respect to the percentage of the detected video freezes when the subjective tests were done with videos with different durations. Therefore, the topic is still challenging, and it is important to understand the minimum perceived inter-picture time of the end-users during a video stream on the smartphone, which directly relates to the freezes and consequently to the end-user perceived QoE. Although the human visual perception depends on many factors including the context, time perception, memory effect [161], illumination, content [138], we assumed that the subjects indicated the freezes as they perceived them. The users were asked to indicate a perceived freeze by using the "Freeze" button located on the user interface during the video playout. The detailed procedure of the study is given in Section 5.4.2.

Finding out the exact temporal impairment, *i.e.*, pinpointing the exact D_{p} value for an indicated freeze by a user, is challenging. In other words, there might be many high (higher than the nominal 40 ms for a 25 fps video) values, which the users might react upon. However, the user might not have reacted upon all of them for some reason. An example scenario is illustrated in Fig. 5.2. In the sketched scenario, suppose $T_{\text{p}}(k)$

is the timestamp of the last picture that is displayed; and $T_{i_{\text{freeze}}}$ is the timestamp of the subject's i'th indication. The subscript "freeze" stands for the type of user indication. It is hard to interpret the reason for the freeze indication, *i.e.*, upon which D_{p} values out of $\{D_{\text{p}}(k-n-1)$, ..., $D_{\text{p}}(k-1)$, $D_{\text{p}}(k)\}$ the freeze was indicated for within some interval C'_i. Thereby, in order to guarantee the minimum perceived inter-picture time of a user, the perceived user inter-picture time with the *MinOfMax* approach is analyzed and explained as follows.

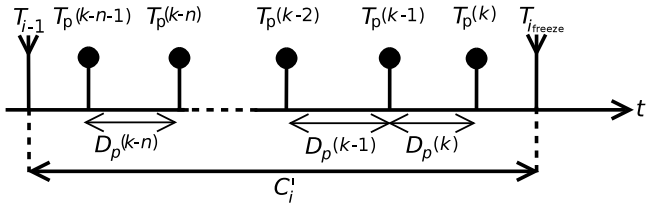


Figure 5.2: Illustration of a scenario where it is hard to interpret what the user rates [114].

MinOfMax: Firstly, for all freeze indications at $T_{i_{\text{freeze}}}$ by a user, the maximum D_{p} value in-between $T_{i_{\text{freeze}}}$ and T_{i-1} (*i.e.*, within interval C'_i) is calculated as shown in Eq. 5.5.

$$D_{\text{p}_{\text{max}_i}} = \max\{D_{\text{p}}(k)\},\ \forall k \text{ within the time interval } C'_i \tag{5.5}$$

Observe that T_{i-1} is not necessarily a *freeze indication* timestamp; it can be the timestamp of *any indication* amongst the six user feedback choices (one, two, three, four, five, and freeze). It is assumed that the user perceives and indicates a new freeze after the previous indication. This procedure is repeated for each freeze indication.

Secondly, the minimum of the maximum values received from user, S, are calculated in Eq. 5.6. We call this metric as the *minimum perceived inter-picture time*, $D_{\text{p}_S\text{min}}$, of user S. This way, the risk of wrong interpretation of a user's freeze indication is reduced.

$$D_{\text{p}_S\text{min}} = \min\{D_{\text{p}_{\text{max}_i}}\},\ \forall i \text{ of subject } S \tag{5.6}$$

Yet, the calculation of the *minimum perceived inter-picture time* for each user is defined. To complement the user perception investigation, it is also important to study the time it takes for a user to react to a stalling event; we name this metric the *user response time*.

5.2.5 User Response Time

The influence of the waiting times to the end-user perceived quality is studied previously [136], however, the duration until the user perceives a freeze and reacts to it has not been studied within the scope of QoE. This is important to know as QoE assessment depends on multidisciplinary parameters such as time perception and the "inner-clock" [163] of the user. Musser [162] states that the human consciousness lags 80 ms behind the actual events. Studying the time it takes for a user to react upon a freeze event, *i.e.*, to press the "Freeze" button, is complex and it depends on many factors. According to Grondin's work on time perception in [163], the actual time is different than the subjectively perceived time. In this study, it is assumed that the users indicate a freeze as they perceive it on spot.

The user response time is further investigated in two scenarios. The first scenario, *short*, is when a user indicates a past freeze with a rather short duration during a smooth playout. The second scenario, *long*, is when the user indicates a freeze while the stall is going on.

First Scenario – Short: Let's assume that $T_{D_{\mathrm{pmax}}}$ is the timestamp of the picture with the longest display time amongst all D_{p} values within the interval C_i' as shown in Fig. 5.3. Assuming that the user reacted and pressed the freeze button with respect to $T_{D_{\mathrm{pmax}}}$, the user response time is calculated, as shown in Eq. 5.7.

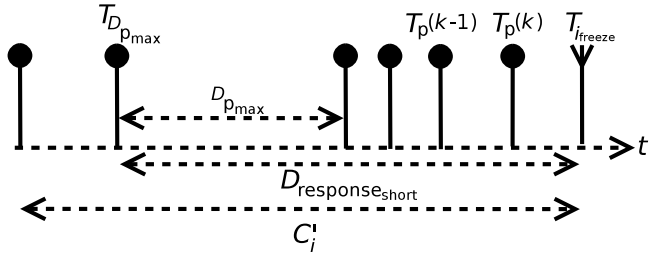


Figure 5.3: Illustration for the calculation of $D_{\mathrm{response_{short}}}$ [114].

$$D_{\mathrm{response_{short}}} = T_{i_{\mathrm{freeze}}} - T_{D_{\mathrm{pmax}}} : T_{D_{\mathrm{pmax}}} \in \{T_{\mathrm{p}}(k)\} \forall\, k \,\mathrm{within}\, C_i' \tag{5.7}$$

Second Scenario – Long: The freeze indications by the user for the *long* freezes are studied, *e.g.*, pauses [119], as this metric might signify the *end-user's tolerance level* to the long-term freezes. This scenario is when the user intervenes the video stream

by pressing the freeze indication button at $T_{i_{\text{freeze}}}$, before the next picture k is displayed at $T_{\text{p}}(k)$, as shown in Fig. 5.4. $D_{\text{response}_{\text{long}}}$ is calculated as the time gap between the display timestamp of the frozen picture and the user's freeze indication timestamp as shown in Eq. 5.8.

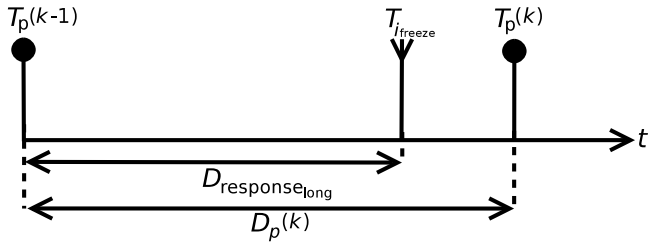


Figure 5.4: Illustration for the calculation of $D_{\text{response}_{\text{long}}}$ [114].

$$D_{\text{response}_{\text{long}}} = T_{i_{\text{freeze}}} - T_{\text{p}}(k-1) \tag{5.8}$$

5.3 VLQoE Tool Description

The VLQoE tool is implemented, as a mobile version of open-source VLC Media Player, while adding extra functionalities for video QoE assessment.

5.3.1 VLC Media Player

VLC is a packet-based media player [164] developed by Video LAN, first released in year 2000, available as open source, with GNU General Public License (GPL) and GNU Library General Public License (LGPL) software licenses. Amongst other available players, VLC is available for the most variety of operating systems, and also for Android-based terminals. According to the "sourceforge.net", VLC has been downloaded more than 17 billion [134]. VLC supports many video formats including mp4, and streaming protocols including HTTP and RTSP.

New functionalities are added to the VLC player source code in order to record picture display timestamps, the user context data, and the perceived quality ratings (*i.e.*, UR) while a video is being streamed. More details on the VLQoE tool is given in the next section.

Table 5.1: List of collected parameters by VLQoE.

COMPONENTS	PARAMETERS
USER INTERFACE	displayed video picture(frame), user controls, freeze indication, user rating (UR), display orientation and brightness, screen touch events
APPLICATION	re-buffering events
NETWORK AND PHYSICAL	interface type, service provider, signal strength(RSSI), packets/sec
OTHER	GPS coordinates, battery level, unique device ID

5.3.2 VLQoE Tool

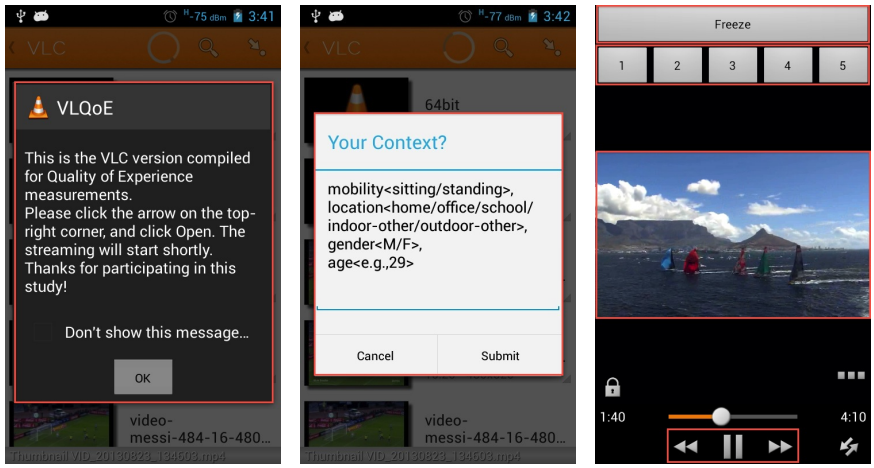
The original beta version of the smartphone-based VLC player consists of the video pane that displays the video; a set of video control buttons such as play, pause, rewind, and forward. We developed additional functionalities on top of VLC player and applied to different player components as detailed in Table 5.1. Additional functionalities are grouped into User Interface, Application, Network and Physical, and Other. The detailed description of the collected metrics is listed in the following subsections.

User Interface

We added additional buttons in the user interface in order to enable user to register in-situ continous feedback such as opinion score or freeze indication. This is especially important from the ESM perspective, and minimizes the memory effect. The snapshots from the user interface of the player are presented in Fig. 5.5. When the user launches the player, s/he sees a welcome message (see Fig. 5.5(a)), and s/he is required to click the "OK" button. Afterwards, the user is asked to type in and submit information regarding her/his mobility, location, gender, and age (see Fig. 5.5(b)).

Video Pane: During a video stream, the timestamps of the displayed pictures on the device display (after decoding and rendering) are recorded. Then, the D_{p} values are calculated. The video pane is shown as a rectangular box in Fig. 5.5(c), and it is 196×117 pixels with a fixed vertical display view.

User Rating Buttons $(1,2,3,4,5)$: For QoE measurements, the user is instructed to press one of the horizontally aligned five user rating (UR) buttons (based on ACR scale), as shown on top of Fig. 5.5(c), whenever the user feels like rating the quality.



(a) Welcome message (b) Context and Demographics (c) Freeze, UR, Display, and Control

Figure 5.5: Snapshots from the VLQoE player [114].

This helps to match the underlying application metrics to the application QoE, however it does not give information whether a possible low QoE is caused by a temporal impairment. Therefore, an additional button is needed that helps the user to indicate a freeze when required while the video is being streamed. When any of the rating buttons is pressed by the user, the event is recorded together with the corresponding timestamp.

Freeze Button: The freeze button is located on top of the user interface (see Fig. 5.5(c)) to enable users to press whenever s/he perceives a picture freeze on the video display. When the freeze button is pressed, the event is recorded together with the corresponding timestamp, *i.e.*, the data is labeled.

User Control Buttons: There are three buttons (play/pause, rewind, forward) at the bottom of the video pane, as shown in Fig. 5.5(c). We added the functionality to record the interactions with these buttons together with the corresponding timestamp.

Screen Display: The orientation (vertical/horizontal) and the brightness level (%) of the smartphone screen are recorded with the corresponding timestamp.

Screen Touch Events: The users' touch events on the display are recorded with the corresponding timestamps, while the video is being streamed. This is especially important to understand how the user reacts and/or interacts with smartphone upon a temporal impairment.

Application Layer

Rebuffering Events: During the video streaming, the rebuffering of the video content might cause undesired picture freezes [136]. Thus, the rebuffering events upon the playback buffer starvation are recorded with the corresponding timestamp.

Network and Physical Layer

Packets: The received and transmitted number of packets and bytes to/from the smartphone are recorded every second with the corresponding timestamp.

Interface Type: The active wireless interface of the device (3G or WiFi) is recorded once when the application is first launched.

Network Provider: The Subscriber Identity Module (SIM) card operator name is recorded once when the application is first launched. The reason is that there is no handover involved in our experiments and all of the users performed the tests on the same smartphone with the same SIM card. In the next VLQoE version, the logging of network provider can be further extended for future tests involving mobility (particularly handover amongst operators).

Wireless Signal Strength: The signal strength that is measured at the active wireless interface (3G or WiFi) is recorded for once, when the application is first launched. Next, the signal strength is recorded sporadically when the circumstances change, *e.g.*, only when a different WiFi RSSI is triggered by the Android OS on the smartphone.

Other

Other recorded metrics belong to the following components:

GPS: Users' GPS coordinates are recorded when the application is launched for the first time. The minimum time gap between the GPS polling is set to one minute. In addition, the minimum distance change with respect to the previous location is set to

arbitrary 10 meters. Based on the conditions, the software records the new coordinates with the new timestamp. The user should manually enable the GPS and Network (via cell tower locations) to calculate the location on the "Settings" menu of the smartphone. If none of these features are enabled, VLQoE pops up a notification and encourages the user to enable it. The type of the active GPS component (Network or GPS) is also recorded. This way, it is possible to get the mobility information of the user while streaming video. Within the scope of this study, we aimed to conduct the experiments while the users are at fixed position, thus we did not think of any particular reason for choosing this specific setting.

Battery Level: The battery change is triggered by the Android OS of the smartphone and is recorded on the device. The events are recorded with the corresponding timestamp.

Device ID: The unique device ID of the smartphone is recorded for once when the application is launched for the first time. The names of all data files are labeled with the device ID. This is helpful to distinguish in-between the users.

FTP Upload: In addition, a *FTP Upload* option is provided on the "Settings" menu of VLQoE, so that the data files that comprise the aforementioned metrics can be uploaded manually by the users to the server located at Blekinge Institute of Technology (BTH), Sweden or at some other server for research purposes. This is implemented to support future studies to enable remotely capturing a wide variety of data from various users at various locations on Earth.

Although, we implemented the above functionalities within the VLQoE, we do not investigate all of the collected metrics. In this study, we studied the timestamp of the displayed video pictures, the timestamp of the user ratings, and the timestamp of the freeze indications by users.

5.3.3 Validation of VLQoE

The inter-picture time measured via the VLQoE tool is validated in this section. For that, a video with 25 frames per second, with 500kbit/s bitrate is used. Thus, the nominal inter-picture time is $\frac{1}{25\text{fps}} = 40\,\text{ms}$. The video is streamed directly from the local storage of the smartphone in order to discard the network influence on the streaming. The validation is conducted on three different smartphones with different branches, *i.e.*, Samsung Galaxy S, Samsung Galaxy S4, and ZTE T40.

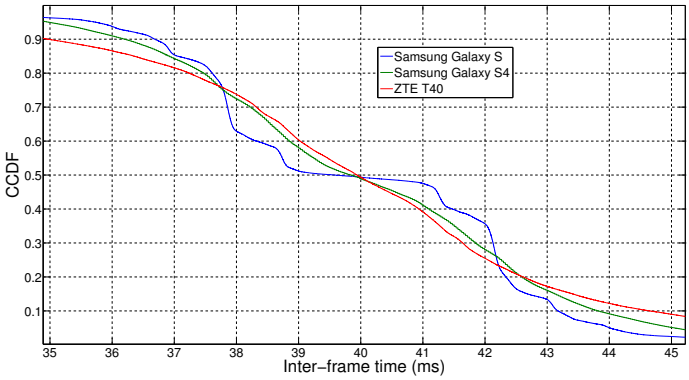


Figure 5.6: CCDF of inter-picture delay measured on three different smartphones for a 25 fps video clip.

The CCDF for the inter-picture time measured on three different smartphones are given in Fig. 5.6. Three different datasets collected from three different smartphones are presented in one plot. The calculated mean, median, confidence interval (97.5%), and the corresponding number of samples per each dataset is depicted in Table 5.2. The median inter-picture time are 39.93 ms, 39.58 ms, 39.85 ms for ZTE, Samsung Galaxy S, and Samsung Galaxy S4, respectively. The measured values are very close to the nominal 40 ms, and we conclude that the VLQoE tool can be used to accurately and timely measure the inter-picture time.

Table 5.2: VLQoE's inter-picture time statistics measured at three smartphones.

Phone type	Mean [ms]	Median [ms]	97.5% CI [ms]	Number of samples
ZTE	48.77	**39.93**	± 3.28	362396
Samsung S	47.84	**39.58**	±2.98	383841
Samsung S4	47.53	**39.85**	±3.03	387412

5.4 Experiment Settings and Methods

In this study, the VLQoE tool has been used primarily for performance analysis with respect to the D_{p} metric, while considering the user feedback. The experiments are

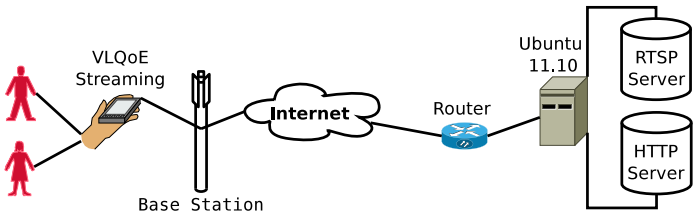


Figure 5.7: Experiment testbed [114].

conducted in two parts. The first part consists of a study without user involvement to establish a ground truth and to understand the video streaming characteristics with respect to the ON/OFF modeling. The second part is the user study designed to capture the circumstances when the video QoE has occurred, *i.e.*, UR and freeze indications with respect to the inter-picture time (D_{p}) values.

The first part of the experiments is the "ON/OFF Study" and it is introduced in Section 5.4.1; and the second part of the experiments is the "User Study" and it is introduced in Section 5.4.2. We used a common experiment setting in both parts, which is detailed as follows.

Common Experiment Settings: The VLQoE software is installed on Samsung Galaxy S (Android 4.2.2). The video was located at a dedicated Ubuntu 11.10 streaming server within the BTH university campus (City of Karlskrona, Sweden), and it was streamed to the smartphone via RTSP and HTTP protocols. The streamed videos were displayed on the smartphone screen with a resolution of 196×117 pixels. The same video has been used in both parts; the streamed video content was a 250.44 seconds-long water-sports theme video clip with a sequence of 6251 pictures, nominal frame rate of 25 fps, and was encoded with a bitrate of 1000 kbit/s. The video consists of various scenes comprising cheering fans, interviews with the sportsmen, and racing scenes of the sailing boats.

The Scene Complexity (SC) is a metric as the average of the spatial and the temporal details of all frames of a video clip, as shown in Eq. 5.9; where $F(n)$ is the luminance channel of the n^{th} video frame; the temporal complexity (TI) is calculated as the root mean square (RMS) of frame-to-frame image changes as in Eq. 5.10; and the spatial complexity (SI) is calculated as the RMS of the Sobel filtered luminance channel of the frames, as shown in Eq. 5.11 [166] [165], ITU-T P.910 [156] .

$$SC = \log_{10}(\mathrm{mean}_n[SI(n) \cdot TI(n)]) \tag{5.9}$$

$$TI(n) = \mathrm{RMS}_{\mathrm{space}}[F(n) - F(n-1)] \tag{5.10}$$

$$SI(n) = \mathrm{RMS}_{\mathrm{space}}[\mathrm{Sobel}(F(n))] \tag{5.11}$$

The *SC* of the video clip used in this study is calculated as 7.04 with a high *TI* value of 67, and *SI* value of 16. Based on the rough estimates of Ramos *et al.* [167], a video with *TI* greater than 38 could be considered as a video with high motion, but with some exceptions. In addition, it has observed that the calculated *SC* value is comparably higher than all of 25 ANSI standard test scenes [166].

5.4.1 ON/OFF Study (Part 1, no user involvement)

In the ON/OFF study, our main goal is to investigate and obtain a user model with respect to the D_{p} metric. During the first part of the experiments, video streaming is tested at different locations, including places where the video streaming quality was poor, *e.g.*, comprising university library, apartment basements in the Karlskrona city. The purpose was to replicate the worst-case scenarios with distortions during a real-time video stream. This part was conducted without the user involvement (no user-interaction), and exactly the same video content was streamed consecutively using two streaming protocols, *e.g.*, RTSP and HTTP via 3G. A test supervisor started the video stream and did not press any other control buttons, as this part of the experiments aims to model D_{p} at the user interface during network-based video playout. Two experiments were performed; one for each protocol (HTTP or RTSP); and each experiment comprises 30 repetitions.

Based on the D_{p} values, the durations spent at the ON and OFF states are investigated. The state boundary between ON and OFF states is defined as 100 ms: the video is considered being in the ON state when the inter-picture time is less than 100 ms; and it is assumed in the OFF state when the inter-picture time is greater than or equal to 100 ms. The ON and OFF durations are collected separately in two different data sets called ON and OFF, and investigated the distribution as well as the Maximum-Likelihood Estimation (MLE) of the durations within the states by using Matlab. No extra artificial disturbances have been applied on the link between the streaming server and the smartphone, but instead the data has been collected in a similar context that the users might experience in real life.

5.4.2 User Study (Part 2, with user involvement)

In the second part of the study, 30 subjects were met at various locations in Karlskrona, Sweden, and the users were asked to watch the streamed video on the smartphone. A more detailed background description on the selected subjects is presented in Table 5.3 in Section 5.5.2. Each user first watched the RTSP-based video and then the HTTP-based video. In total, 60 (30 users $\times$ 2 protocols) user experiments are conducted.

During the experiments, the ITU-T P.910 [156] recommendations for quality in video conferencing and video-on-demand were applied as much as possible. However, due to the inconsistency in the definition of experiment settings in different standards documents (ITU-R BT.500, ITU-T P.910/1/2, ITU-R BT.1438), this is indeed still a challenge [165]. One of the strengths of this study is that the subjective tests are conducted in peoples' natural daily life settings. For example, the test supervisor did not set any restrictions such as the distance between the phone and the user. Instead, the users were asked to hold the smartphone at a comfortable distance and take a comfortable position in a silence room with convenient illumination level, *i.e.*, at a familiar physical space in daily life. The users were asked to press the "Freeze" button anytime to indicate an evidence of a visual freeze on the display, whenever they recognize a freeze on the video. In addition each user was encouraged to rate the temporal quality based on the five-level MOS scale, while pressing one of the five user rating buttons at her/his own will during the playout.

As the focus is on the influence of temporal impairments on the users, not all QoE aspects are considered in this study. For example, an additional "Freeze" button on top of the ACR scale was used to collect the user reactions upon stalling events. ITU-P.910 [156] recommends that the source signal, recording environment/system, scene characteristics, and the spatial/temporal information of the video needs to be taken into consideration during the configuration of the user experiments. The standard is of high importance in analysis of content level influential factors on QoE. However, in this study, the focus is more on the context, system, and the user level aspects and the temporal impairments are considered on a single video source with a length of more than four minutes in realistic smartphone settings. The traditional methods ask for user feedback on the quality such as ACR, after the video is completed, however our aim in this work is to study and sample the experience on-the-fly as the video continues to stream in real-time. The users streamed exactly the same video content and the video is displayed on the video pane with 196×117 pixels resolution. The videos were muted to let the users focus only on the visual impairments. This way, the influence of other impairments such as the discontinuity of sound is minimized. For simplicity, the orientation (vertical) and the brightness of the display were kept constant for all users.

Furthermore, a short baseline interview with each subject is conducted just after

the experiments. Demographics such as age, gender, and occupation are collected. In addition, information that addresses the following questions: (1) How often are the users streaming video over the network on their own smartphones per day (less than two minutes, less than five minutes, less than ten minutes, less than thirty minutes, less than one hour, more than one hour); (2) Which network interface are they using while streaming video (3G or WiFi); (3) What is the brand name and the operating system of the users' smartphone; (4) Via which wireless network interface are they experiencing freezes the most, namely Often Experienced Freezes (OEF) on (3G or WiFi). After each user watched the same video with two different streaming protocols, at the end of the experiments, the test supervisor asked to rate the overall quality based on both videos in the five-level MOS scale.

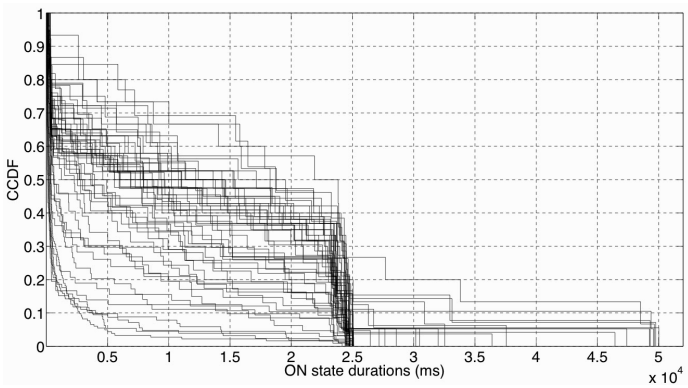
5.5 Results

The presentation of the results are structured in two parts. The results for the ON/OFF study, *i.e.*, two-state modeling of the inter-picture times (D_p), are given in the Section 5.5.1. The results of the second part of the experiments, *i.e.*, the user study, are presented in Section 5.5.2.

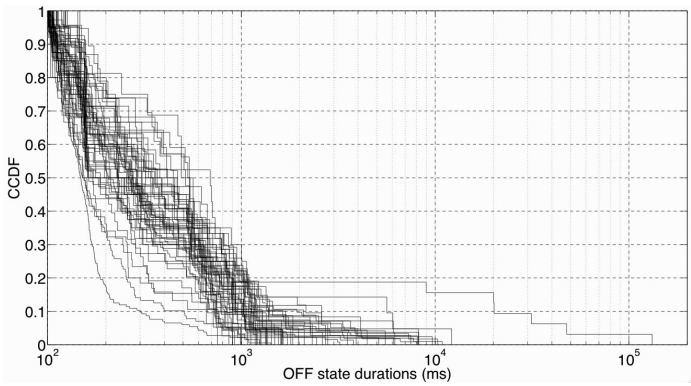
5.5.1 ON/OFF Study (Part 1)

The duration of the ON and OFF states is calculated throughout 58 iterations amongst the 60 iterations. During two iterations (RTSP run #29 and HTTP run #27), the video stopped in the very beginning due to very bad network coverage, which made the video impossible to stream. Thus, two iterations from our dataset are excluded. The duration of ON and OFF states is studied. First, it has been assured that the distribution of the ON and OFF state durations in all 58 runs could fit into an exponential curve by obtaining the Coefficient of Determination (R^2) values, as it can show how well our hypotheses is correct, *i.e.*, how well the collected data points fit an exponential curve. Exponential modeling is chosen as it is assumed that the ON and OFF durations are memoryless, *i.e.*, the current state of the video streaming does not depend on its previous state. The mean R^2 values were calculated as 0.93 and 0.81 for OFF and ON states, respectively. Next, the MLEs for the durations of ON and OFF states for all 58 iterations are calculated, then the mean of all MLE values are obtained. The mean MLE of OFF and ON durations is calculated as 642 ms and 9.7 s, respectively. CCDF plots visualize random, rare, and extremely high values. The aim here is to emphasize on the high OFF durations that might cause low end-user perceived QoE. Thus, the ON and OFF durations are visualized for all runs separately via CCDF plots as given

in Fig. 5.8(a) and Fig. 5.8(b), respectively. Slight deviations are observed in-between the CCDF plots, as the network conditions were not identical for all runs. However, the R^2 on exponential fits of all runs for ON and OFF scenarios were above 0.8. Long



(a) ON state durations (ms). Linear x-axis.



(b) OFF state durations (ms). Logarithmic x-axis.

Figure 5.8: CCDF plots for the duration in ON and OFF states for 58 runs [114].

OFF durations followed by very short inter-picture durations are observed. This shows that the pictures were clumped and then displayed as many back-to-back pictures to the smartphone screen at once in very short duration. This behavior eventually manifests itself on the user interface as fast-forward for a short period of time.

5.5.2 User Study (Part 2)

In this subsection, the results with respect to the user study including the user demographics and background obtained via online survey, user feedback via interview, automatically captured objective data via VLQoE, the user ratings (QoE) and the freeze indications collected via the user interface of VLQoE, the minimum perceived inter picture time, and the user response time. Before presenting the results, elaboration on the corrupted (due to corrupted timestamps) or missing data is given.

Corrupted or Missing Data: The freeze indications by the user are necessary in order to identify the minimum perceived inter-picture time. In this study, some user data was corrupted, or did not help in identification of freezes due to the missing freeze indications. Amongst the 14 out of 60 user experiments, the users did not press the freeze button. We were not able to capture the corrupted/missing data during the experiments, thus we could not diagnose. From the lessons learned, we suggest to check the collected data starting from the early phases of the study, as also suggested by Oliver [168], in future studies. This can increase both the quantity and the quality of the collected data.

Short Interview with the Participants

In total, there were 30 participants in the user study. 28 participants were within 21-30 years age range; $S13$ and $S30$ were within 31-40 years age range. Five subjects amongst all participants ($S2$, $S5$, $S16$, $S17$, $S22$) were female. Five subjects watched the videos at home, and 25 subjects at the university campus.

The background information regarding their experiences on their own smartphones, particularly on video streaming, is collected via online survey. This is helpful to select subjects from different nationalities and gender. In addition, it is aimed to select subjects that stream videos on their own smartphones in daily life in different context. The background information of the subjects are presented in columns $2-5$ in Table 5.3. The first column is the subject ID; the second column is the nationality of the user. The subjects were from six different nationalities. The third column shows the users' total daily video streaming durations via wireless network interface on their own smartphones in daily life. 23 out of all subjects watch a video on the smartphone with a duration of at least five minutes per day. The fourth column shows how the users watch video on their own smartphones, *e.g.*, streaming via only 3G; via only WiFi; via only *local* storage of the smartphone; or via *Both* WiFi and 3G. 13 subjects stream video via only WiFi interface; five subjects stream via only 3G; six subjects stated that they use both interfaces occasionally to stream video; four subjects use only local streaming;

and two subjects ($S13, S29$) never watch a video on the smartphone. Six subjects ($S5$, $S13$, $S21$, $S23$, $S27$, $S29$) claimed that they never watch a network-based video on the smartphone in daily life. Often Experienced Freezes (OEF) during video streaming is given in column 5 of Table 5.3 and it is categorized as either 3G or WiFi. In total, 17 subjects claimed that they often experienced freezes while streaming via 3G; and only four subjects stated that they experienced freezes while streaming via WiFi. $S13$ and $S23$ state that the reason for not watching network-streamed video is due to their previously perceived low quality of experience of the freezes via 3G streaming.

ESM Results

On the sixth column of Table 5.3, the average evaluation (in five-level ACR scale, *i.e.*, MOS) of subject on the temporal quality of the videos is presented. Five subjects rated 3-Fair, 23 subjects rated 4-Good, and one subject rated 5-Excellent for the average temporal quality of the streams. The average ACR received from one subject, *i.e.*, $S5$, was lost. Although the subjects have experienced occasional freezes during the video sessions and registed in-situ 1's or 2's, which is detailed in the next section, the overall MOS values were higher than 2-Poor. This might be due to the cognitive factors such as the human memory effect [161] [149].

After studying the user-perceived quality based on the short interviews, the user ratings (via ESM) collected during the video stream are examined in more details. In the last column of Table 5.3, the most frequent user rating received during each streaming session is presented together with its percentage of its occurrence amongst all other ratings. For example, 4 (50%) means that the number of ratings with UR$=4$ comprises 50 % of all ratings, which is also the highest percentage amongst other ratings, given by a particular subject. The values are stated for the two streaming protocols (separated by a comma).

The maximum inter-picture time value during the period between the current and the previous user indication (user rating or a freeze) is studied. The CCDF plots of the maximum inter-picture time values for the user ratings $1-5$, and freeze indications collected from all users are given in Fig. 5.9. The plot clearly shows that 60 % of the D_{p} values (where the y-axis is at 0.4) are less than or equal to approximately 65 ms, 80 ms, 150 ms, 500 ms, 500 ms, 700 ms for "UR 5", "UR 4", "UR 3", "UR 2", "UR 1", and "Freeze", respectively. The corresponding exponential models for all user ratings are given in the Appendix A. Van Kester *et al.* [135] states the acceptable (UR > 3.5) freezing time as 360 ms. According to the results of this study presented in Table 5.4, if the acceptable boundary is considered to be UR> 2, then the mean $D_{\mathrm{p_{max}}}$ should not exceed 321 ms for an acceptable quality. The CCDF plots of "UR 1" and "UR 2" are almost the same.

Table 5.3: Participants demographics and statistics summary. Unavailable: "-".

S	Nationality	Net. Stream per day	Stream via	OEF via	Interv. MOS	VLQoE UR RTSP, HTTP
1	Pakistan	30 min	Both	3G	4	4(50%), 4(46%)
2	Pakistan	$>$ 1 hour	3G	3G	4	3(32%), 4(34%)
3	Iran	5 min	WiFi	3G	4	-, 4(63%)
4	Turkey	$<$ 2 min	WiFi	3G	4	4(50%), 4(28%)
5	China	Never	Local	-	-	4(33%), 4(25%)
6	China	$>$ 1 hour	WiFi	WiFi	5	5(59%), -
7	Turkey	30 min	Both	3G	4	-, -
8	Sweden	$>$ 1 hour	3G	-	4	5(26%), 3(34%)
9	China	5 min	WiFi	3G	4	4(28%), 2(25%)
10	India	30 min	WiFi	-	4	4(36%), 4(33%)
11	Sweden	30 min	Both	3G	4	5(62%), 4(42%)
12	Sweden	5 min	WiFi	3G	3	4(33%), 5(38%)
13	Sweden	Never	-	3G	3	5(27%), -
14	India	$>$ 1 hour	Both	WiFi	4	4(46%), -
15	Pakistan	30 min	3G	3G	4	-, 5(51%)
16	Sweden	5 min	3G	-	4	5(80%), 5(66%)
17	Bangladesh	5 min	WiFi	-	4	4(33%), 3(63%)
18	Bangladesh	$>$ 1 hour	3G	3G	3	4(27%), 3(49%)
19	Pakistan	30 min	WiFi	3G	4	5(45%), 4(58%)
20	Sweden	1 hour	Both	3G	4	5(36%), 5(76%)
21	Pakistan	Never	Local	-	4	2(39%), 4(35%)
22	Sweden	5 min	WiFi	3G	3	-, 4(46%)
23	Pakistan	Never	Local	3G	4	5(53%), 5(65%)
24	Pakistan	10 min	WiFi	WiFi	4	2(41%), 4(47%)
25	India	10 min	WiFi	-	3	4(58%), 4(57%)
26	Pakistan	5 min	WiFi	WiFi	4	4(39%), -
27	Pakistan	Never	Local	-	4	5(54%), 3(33%)
28	Sweden	30 min	Both	3G	4	5(31%), 4(37%)
29	Sweden	Never	-	-	4	-, 3(25%)
30	Sweden	30 min	WiFi	3G	4	5(62%), 4(33%)

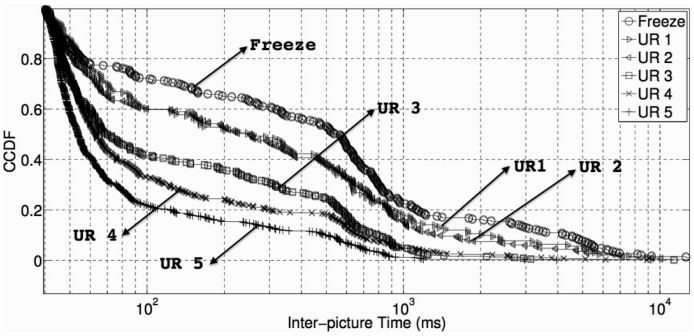


Figure 5.9: Inter-picture time distribution of the corresponding user indications [114].

The more detailed statistics of the $D_{\mathrm{p_{max}}}$ values before a user indication are given in Table 5.4. The inter-picture time values for $D_{\mathrm{p_{max}}}$ values were exponential distributed with R^2 values given in Table 5.4. The dataset for the $D_{p_{\mathrm{max}}}$ values for all user indications were fit to an exponential distribution with R^2 value of higher than or equal to 0.84. The mean of the maximum inter-picture time values were calculated as 152 ms, 282 ms, 321 ms, 768 ms, 831 ms, 1289 ms for "UR 5", "UR 4", "UR 3", "UR 2", "UR 1", "Freeze", respectively. There is a clear trend that the UR increases, *i.e.*, user perceived quality improves, as the $D_{p_{\mathrm{max}}}$ value decreases.

The time gap between the user indication, T_i, and the displayed picture with the maximum display time, $T_{D_{\mathrm{p_{max}}}}$ is calculated. The statistics on the latter are given in the last three rows of Table 5.4. The mean $\Delta(T_{i-D_{p\mathrm{max}}})$ values were calculated as 2109 ms, 2650 ms, 3267 ms, 3808 ms, 3809 ms, and 2603 ms for "Freeze", "UR 1", "UR 2", "UR 3", "UR 4", and "UR 5", respectively. As the mean time gap between the user's indication and the user rating (in the order of freeze, 1, 2, 3, 4) increase, the corresponding user rating gets higher. One reason could be the fact that users are willing to rate the bad quality immediately, and are forgetting to rate when the quality improves. Similar behavior has also been observed during a separate user study in Chapter 4. For some reason, the $\Delta(T_{i-D_{p\mathrm{max}}})$ value drops to 2603 ms for UR $= 5$, which is under investigation.

First Low User Rating Time $(T_{\mathrm{LowRating}})$ **and Number of Alarms**: In this part, the pictures that are displayed for more than 100 ms duration ($D_{\mathrm{p}} > 100$ ms) are categorized as *alarms*. The number of alarms is visualised along with the corresponding low user ratings (UR < 4) [158], and this is done for all user experiments in Fig. 5.10. Each

Table 5.4: Number of data points, goodness-of-fit, mean, median, standard deviation values are presented in rows 2-6 for $D_{\mathrm{p_{max}}}$ together with the corresponding $\Delta(T_{i-D_{p\mathrm{max}}})$ values in rows 7-9.

User Indication	Freeze	UR 1	UR 2	UR 3	UR 4	UR 5
# of Data Points	266	142	229	305	471	515
R^2**(Exponential)**	0.96	0.94	0.94	0.84	0.90	0.94
Unit	ms					
Mean$\{D_{\mathrm{p_{max}}}\}$	1289	831	768	321	282	152
Median$\{D_{\mathrm{p_{max}}}\}$	550	253	253	70	63	54
Std$\{D_{\mathrm{p_{max}}}\}$	2495	1611	1666	758	706	295
Mean$\{\Delta(T_{i-D_{p\mathrm{max}}})\}$	2109	2650	3267	3808	3809	2603
Median$\{\Delta(T_{i-D_{p\mathrm{max}}})\}$	1332	1457	1686	2387	2527	1335
Std$\{\Delta(T_{i-D_{p\mathrm{max}}})\}$	3147	3358	4220	4133	4016	3412

point on the plot represents the result of one video experiment session. $T_{\mathrm{LowRating}}$ for each user are shown on the y-axis; and total number of alarms that are detected until $T_{\mathrm{LowRating}}$, are shown on the x-axis. Some subjects were reluctant to rate, *i.e.*, a low user rating at the 25$^{\mathrm{th}}$ second of a video session is received with respect to the 37 alarms recorded by the application within this interval. In contrast, some people were rather eager to rate, *i.e.*, seven users rated the quality with UR < 4 within the initial 30 seconds of the video, although there were no alarms with $D_{\mathrm{p}} > 100$ ms.

Minimum Perceived Inter-picture Time

The minimum perceived and reported inter-picture time values for each subject (subject ID at the first column) are given in Table 5.5 at the columns 2 and 5 for HTTP and RTSP streaming, respectively. The total number of freeze indications (#i) by each user are given in columns 4 and 7, respectively. The minimum perceived inter-picture time events that were reacted by the users at least 80 ms after $T_{D_{\mathrm{Pmax}}}$ were considered as stated by [162]. There were two user experiments ($S6$ and $S13$ during RTSP streaming) out of 60, in which $D_{\mathrm{response_{short}}}$ were less than 80 ms. Therefore from those two datasets, the first $D_{\mathrm{response_{short}}}$ that is higher than 80 ms were selected. The calculated perceived inter-picture times for each user are presented in Table 5.5 ranging between 40 ms and 4542 ms. The distribution of the measured inter-picture time values can be represented with an exponential function with $R^2 = 0.82$. The mean of the per-

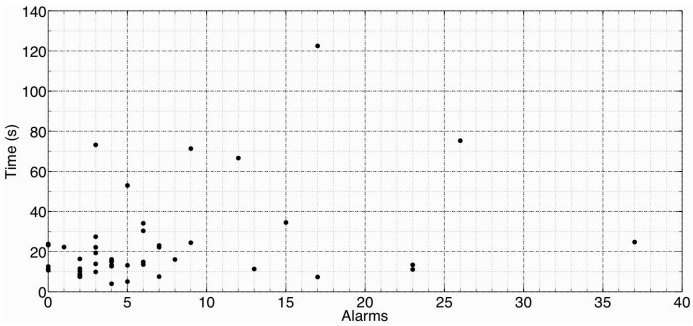


Figure 5.10: The user ratings (UR < 4) are visualized with respect to the corresponding relative time from the start of the video, and the number of alarms raised by the application [114].

ceived inter-picture time based on all subjects is 328 ms. Some subjects indicated many freezes such as *S*6 (with 39 freeze indications) during RTSP streaming, while some did not contribute in any freeze indication such as *S*3. The lack or low amount of freeze indications can be caused by many reasons, some factors are: (1) the subject forgets to indicate a freeze, (2) the subject does not perceive the freezes, or (3) the video playout is nominal (*i.e.*, not much higher than the nominal 40 ms inter-picture delay).

User Response Time

The user response time is calculated for each user while considering the first and the second scenario, separately.

First Scenario − Short: In the *short* scenario, the user indicates a perceived freeze by pressing the freeze button as the video plays out. The results with respect to the user response times for the short freezes as defined in Section 5.2.5 are given in Table 5.5. The third and the sixth column depicts the $D_{\text{response}_{\text{short}}}$ values for RTSP and HTTP streaming experiments, respectively. The values are in-between 218 ms and 10596 ms for HTTP streaming with 2985 ms average; and the values are in-between 102 ms and 6496 ms for RTSP streaming with 1515 ms average.

Second Scenario − Long: In the *long* scenario, the user indicates a freeze while the video is being stalled. The $D_{\text{response}_{\text{long}}}$ values that are obtained from each subject are different. As the CCDF plot can conveniently reveal the very high user response time

Table 5.5: Minimum perceived inter-picture times for each user. Unavailable: "-".

	HTTP			RTSP		
S	$D_{\mathrm{p}s\mathrm{min}}$	$D_{\mathrm{response}_{\mathrm{short}}}$	$\#i$	$D_{\mathrm{p}s\mathrm{min}}$	$D_{\mathrm{response}_{\mathrm{short}}}$	$\#i$
1	49 ms	4451 ms	2	55 ms	4995 ms	4
2	53 ms	7075 ms	10	67 ms	318 ms	11
4	57 ms	10596 ms	8	43 ms	313 ms	7
5	41 ms	454 ms	5	43 ms	539 ms	5
6	-	-	-	41 ms	705 ms	39
7	51 ms	3538 ms	12	-	-	-
8	41 ms	218 ms	14	86 ms	111 ms	7
9	-	-	-	45 ms	1862 ms	2
12	103 ms	4401 ms	6	318 ms	2548 ms	7
13	-	-	-	515 ms	1399 ms	7
14	-	-	-	40 ms	385 ms	1
16	41 ms	275 ms	10	83 ms	534 ms	41
17	-	-	-	389 ms	6496 ms	8
18	50 ms	431 ms	4	50 ms	1124 ms	2
19	4542 ms	4969 ms	1	-	-	-
20	117 ms	4394 ms	9	45 ms	117 ms	9
21	43 ms	3367 ms	20	59 ms	666 ms	3
23	681 ms	1300 ms	3	2880 ms	2704 ms	2
24	53 ms	3959 ms	13	249 ms	1352 ms	1
27	474 ms	1493 ms	3	-	-	-
28	45 ms	2127 ms	4	49 ms	102 ms	15
29	2924 ms	319 ms	1	-	-	-
30	46 ms	369 ms	4	-	-	-

in the data, the CCDF plot of $D_{\mathrm{response}_{\mathrm{long}}}$ is given in Fig. 5.11. $D_{\mathrm{response}_{\mathrm{long}}}$ values can be represented by an exponential distribution with R^2 of 0.98 ($CCDF = \exp(-0.0007 \cdot D_{\mathrm{response}_{\mathrm{long}}}/\mathrm{ms})$). The mean of the $D_{\mathrm{response}_{\mathrm{long}}}$ values is calculated as 1533 ms. Again, based on the results plotted in Fig. 5.11, the user response times vary from user to user.

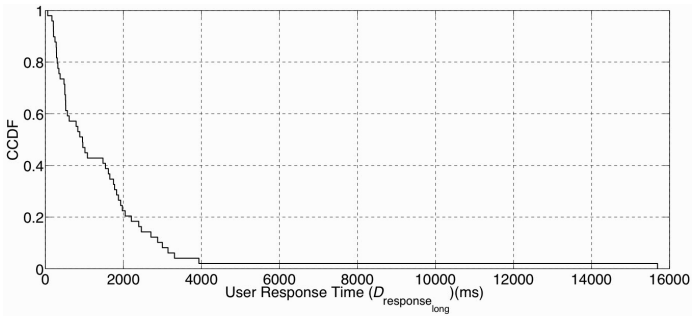


Figure 5.11: CCDF plot of User Response Times, $D_{\text{response}_{\text{long}}}$, for the long freezes [114].

5.5.3 Summary of the Results

A smartphone-based video assessment tool (VLQoE) is developed and deployed in the user studies. VLQoE can can record the metrics that are detailed in Section 5.3.2. The tool enables to analyze in detail all the collected parameters related to QoE either during or after the experiments. The user studies with VLQoE are conducted in two parts: the first part aims to model the inter-picture time in realistic settings during 3G streaming; while the second part aims to find out the video QoE via subjective study. The most important findings of this work are stated as follows:

- ON/OFF modeling based on the inter-picture time during video streaming is studied, and exponential models (with mean R^2 values 0.81 and 0.93 for ON and OFF states, respectively) are obtained. The mean ON duration is calculated as 9.7 s; and the mean OFF duration is calculated as 642 ms during the streaming experiments. (Part 1, Section 5.5.1)

- The minimum perceived inter-picture time is studied. The mean of the minimum perceived inter-picture times is calculated as 328 ms. (Part 2, Section 5.5.2)

- The mean of the highest acceptable (UR $>$ 3) inter-picture time values is calculated as 282 ms. (Part 2, Section 5.5.2)

- The mean of $D_{\text{response}_{\text{short}}}$ values for the HTTP- and the RSTP-based streaming are 2985 ms and 1515 ms, respectively. (Part 2, Section 5.5.2)

- The mean of User Response Time ($D_{\text{response}_{\text{long}}}$) during a long freeze scenario is calculated as 1533 ms. (Part 2, Section 5.5.2)

- The time it takes for the subjects to give a low user rating, *i.e.*, UR < 4, from the start of the video is studied. It has been found that, this duration is varying for each user. Some users are reluctant, and do not respond immediately despite the high number of alarms (freezes) raised by the video player; while some users are rather willing to rate low even for low number of alarms. (Part 2, Section 5.5.2)

5.6 Limitations

The obtained results of this chapter are valid for the particular video, for the particular users involved in the study in particular conditions as stated. Further tests are under investigation for a wide variety of videos and with the participation of more users. The accuracy of the timestamping of the displayed pictures might be a few milliseconds varying from the actual values. This depends on the clock accuracy, and drifting as well as on the timestamping of the programming environment and operating system, and also to the implementation related factors, *e.g.*, particular location of the timestamping function in the code. Moreover, the accuracy of the minimum perceived inter picture time values were based on the number of freeze indications per user as well as other factors related to human factors such as cognitive bias, user concentration level, etc. The improvement of the tool with respect to this aspect is ongoing.

During the experiments, corrupted or missing data for some subjects are detected, *e.g.*, some users did not press the freeze button. Based on this experience, a periodic check of the data during the study is recommended to detect and prevent similar anomalies in future studies.

In the current version of VLQoE, only a subjective method with the five-level ACR scale is employed as it is a commonly used one. On the other hand there are other recommended assessment methods as such some focuses on the relative quality [138]. The task of studying other QoE methods on the user interface is scheduled for future work, and QoE researchers are encouraged to modify the VLQoE for their target studies.

5.7 Summary

In this chapter, the extended version of the popular media player, VLC-media player called VLQoE is presented, and it is aimed to help researchers to conduct smartphone-based video QoE experiments. This is achieved by adding an extra functionality to an open source VLC media player. VLQoE records a set of metrics from the user interface, network- and physical-layer, and the available sensor metrics from the Android

OS. The network- and physical-layer measurements enable the collection of lower layer QoS related metrics such as signal strength and the transmitted/received packets/bytes. The measurements at the user interface enable the collection of direct user feedback with respect to the temporal impairments perceived by the users; and the sensor/GPS components enable the collection of user context during video streaming. The VLQoE tool can be used for user-centric QoE modeling as a function of wide variety of collected metrics including the smartphone battery level, player's rebuffering events, numbers of received and transmitted video packets, smartphone screen orientation/brightness, smartphone network interface signal strength, and the location/context of the end-user.

In this study, VLQoE tool is used to further study the temporal impairments of a network-based video stream. The approach is presented in two parts. In the first part, the video streaming session is modeled in a two state ON/OFF model with the assumption that 100 ms is the boundary inter-picture time for a video playout to be considered either in ON (smooth playout), or in OFF (video picture freeze) state. The duration of ON and OFF states with exponential models is presented together with the corresponding inferential statistics. In part 2, the focus was on the QoE centric modeling and a user study is conducted in order to find out the boundary inter-picture times between the ON and OFF states. To do that, the minimum perceived inter picture times is studied, *i.e.*, the minimum display duration of a video picture that is perceived by a user. Next, it is concluded that this metric varies from user to user (max. 2880 ms, min. 40 ms).

Next, the influence of the inter-picture times on the end-user perceived QoE is presented as measured at the user interface. Based on the results obtained from the user study, the mean of the maximum inter picture time values were 152 ms, 282 ms, 321 ms, 768 ms, 831 ms, and 1289 ms for "UR 5", "UR 4", "UR 3", "UR 2", "UR 1", "Freeze", respectively. There is a clear trend that the maximum inter-picture time increases as the user rating decreases; and the highest mean of maximum inter-picture times have matched with the "Freeze" indications, as expected. Almost similar maximum inter-picture time distributions are observed for "UR 1" and "UR 2". Moreover, the first low ("UR $<$ 4") ratings that were received by user during the video streaming sessions are studied. The occurrences of the high (greater than 100 ms [157]) inter-picture times, *i.e.*, alarms, are studied together with the streaming time until the reception of the first low user rating. In parallel, the overall user response time of the subjects in two scenarios (short/long freezes) are analyzed. The mean user response time with respect to the short freezes are calculated as roughly 3 s and 1.5 s in HTTP- and RTSP-based streaming, respectively. The user response time with respect to the long freezes can be represented via the exponential distribution, and the mean user response time for a *long* freeze is calculated approximately 1.5 s.

The ultimate goal of video QoE experiments is to detect degradation in the quality of video stream, and to react in order to minimize the influence of it on the end-user perceived quality. VLQoE tool has potential to provide metrics with high information gain to the machine learning mechanisms that will benefit in terms of improvement of the future streaming quality, *e.g.*, adapting the player buffer size in real time. The content presented in this chapter can be extended with more number of test subjects (*e.g.*, via Google Play), with long-duration studies, and could shed light on construction of new methodologies for smartphone-based subjective studies and eventually QoE modeling on the smartphone.

We show that the VLQoE tool has great potential for further improvements for mobile-based QoE assessments. It can further be enhanced with more variety of QoE assessment methods, video content, and eventually to be put in the Android application store in order to reach a larger number of subjects. The tool can be improved by adding the options that helps the researchers to choose the desired UR collection methodology, *e.g.*, % slider instead of five buttons, choose/upload new test videos (*e.g.*, VQEG video clips), to be used in the video experiments. The outcomes of testing different existing assessment methodologies and videos will help in standardizing smartphone-based QoE assessment.

Assessment of QoE in real life user studies while being minimally obtrusive to the user is challenging. Obtaining the user data non-obtrusively, *i.e.*, without explicitly asking the user to press the button, can be done by the accelerometer data of the device during the video streaming. This might help to identify the user behavior (*e.g.*, jiggling the mobile device) with respect to the video freeze. Similarly, the obtained user data on the screen touch events may potentially reflect the user perception in a less obtrusive and more objective way. These items can be studied in future work. On the other hand, continuous recording of accelerometer data might be resource consuming. Therefore, one recommendation is an automatic recording of the accelerometer data once a high inter-picture time at the user interface has been detected.

Some further future works could be applied; in fact, the study could be re-designed in order to increase the engagement of the subjects to the study, *e.g.*, tournament to detect freezes. The VLQoE tool can be turned into a game platform where the user feedback with respect to the quality of the video would be collected in minimally intrusive away. This way, the number of participants, the variety of context, and consequently the quality of the data might be improved. The score of each user based on the game can be presented as public and the subjects will be able to compete with each other to detect and indicate the highest number of freezes with the cost of uploading their data to a public QoE database. The collected data can then be fed into machine learning mechanisms so that the application itself will be trained and then suggest the "good-enough" levels of the parameters for the video source, which could then be fed into the

control loop.

We have so far studied inter-picture time with the focus on network-based video streaming, and the collected data depends on the real-time condition (available bandwidth) of the network link during the experiments. Therefore, we plan to complement our work with controlled experiments, *e.g.*, predefined disturbances, to have ground truth, and stream videos that are pre-recorded from the local storage of the smartphone.

In Chapter 4 and 5, we have studied the video streaming at the network and the application level in order to model QoE. We have addressed the temporal impairments such as video freezes (one of the most important influential factors on smartphone-based video QoE as identified in Chapter 3), throughout extensive user studies. Next, we study another very important influential on smartphone QoE that is the energy consumption. We study the power consumption of network-based video streaming. However, before conducting actual measurements, the choice of the power measurement tool is crucial especially because we want to study the power anomalies to extract potential energy saving approaches. In the next chapter, we study two different energy measurement tools: a software tool that internally runs on a smartphone, and an external hardware tool.

Chapter 6

Choosing the Right Power Measurement Tool

"If I had asked people what they wanted, they would have said faster horses". –Henry Ford.

6.1 Comparing Power Monitoring Tools

Power consumption measurements on mobile handheld devices have *cons* and *pros* particularly with respect to intrusiveness and accuracy. For instance, a software tool can obtain measurements without influencing the actual usage behavior of a mobile device, however it might not obtain accurate measurements. Hardware measurement tools are considered as ground-truth due to their high precision capabilities in measurements, however they are poor in portability that makes them not possible to be used in user studies, *e.g.*, while users are using their own smartphones in daily life environments. The software-based energy tools are necessary to reveal the anomalies in energy while the applications are running on the smartphone to match energy consumption with QoS and QoE. Therefore, it is necessary to compare these two different approaches before choosing the right tool for measurements. This section provides the comparison between a *hardware-based* and a *software-based* measurement tool with respect to precision of measurements as an illustrative example. The pros and cons of each tool, Monsoon [170] as a hardware tool, and PowerTutor [172] as an internal software tool), are compared.

Measuring the power consumption directly on the terminal is difficult due to measurements' influence on the device power (*e.g.*, as performing accurate power measurements on a battery-constrained mobile device might yield immense usage of resources), and non-trivial as some phones restrict the access to the power sensor within the operating system kernel. Thus, developers often prefer estimation-based software measurements (by probing the application's process time and allocated resources on the device) to predict the individual application's approximate energy usage. The software tools provide the overall picture of the power and energy consumption of the applications being running on the smartphone including the network interfaces, CPU, and display. They do not provide ground truth measurements, but only can provide estimations. The usage statistics of hardware components are gathered via *procfs* [152] for detailed process information, *BatteryManager* [153] API's or *upower* [154] to obtain battery statistics, which might be inaccurate due to the dependence on the hardware (*e.g.*, processor) architecture. Some drawbacks can be overcome by modifying the kernel, however then it might impact the portability of the tool. In addition, it might be highly resource-consuming as it keeps the CPU busy in order to sample current with high precision. The estimation of energy consumption is often done by preliminarily obtaining physical energy measurements on the device with respect to different power states, *e.g.*, while transmitting data via 3G interface, and then obtain energy models basically via fitting the physically measured data.

Monsoon Power Monitoring Tool: The Monsoon power monitor device contains the power monitor hardware and the power tool software, running on Windows XP and Seven, which can provide robust measurements on any device that uses a single Lithium (Li) battery. The measurements are obtained and can be saved with a sampling rate of 5 kHz. The tool supplies the power to the device, thus the device battery is bypassed. The Monsoon external power-monitoring device is typically used for ground-truth measurements [170].

PowerTutor: PowerTutor is a smartphone application; developed by a collaboration of academic and industrial entities, which displays the power consumed by a set of system components such as CPU, network interface, display, GPS, and other applications. The aim of its development was to make the power measurements transparent to the app developers as well as to the users, so that they can take appropriate action to minimize their smartphones' power consumption. PowerTutor receives the current values in mA from the driver and then multiplies the value by the voltage that is basically the smartphone battery (typically 3.7 V or 4.5 V depending on the phone type). PowerTutor estimates the energy consumption of applications and services based on

the processing times, and is only available for specific phone types. Although these software tools provide the overall picture of the power and energy consumption of the applications being running on the smartphone, the interfaces, CPU, display, etc., they do not provide ground truth measurements on all type of devices, but only can provide estimations.

For power measurements, choosing the "right" sampling rate is necessary in a way that the tool collects enough data for the purpose, without influencing the behavior of the system [173]. Therefore, the power measurement process needs to minimize the impact on the battery life during the measurement process as the energy consumption. In order to perform further statistical tests, we modified the PowerTutor in such a way that it writes the obtained measurements directly to the smartphone's internal storage with a sample rate of 1 Hz. The choice of the power measurement tool depends on the application to be measured. The sampling rate of software measurement tools that run on smartphones need to be kept limited in order to minimize resource usage, if they are running on the battery-powered devices.

We conducted a set of tests to identify the differences between PowerTutor and Monsoon. PowerTutor is installed on the HTC G1 as it is recommended particularly for the Google phones, and in parallel, the Monsoon power-monitoring tool intercepted the battery of the smartphone. This enables simultaneous measurements, and enables to observe the differences between the two tools. We streamed a video (with 500 Kbit/s bitrate) to the smartphone via 3G interface, and monitored the measurements from two tools, simultaneously. Slight inconsistencies between the obtained measurements through Monsoon and PowerTutor are found; PowerTutor measurement values can drop down to zero occasionally as depicted in Fig. 6.1. The instantaneous power

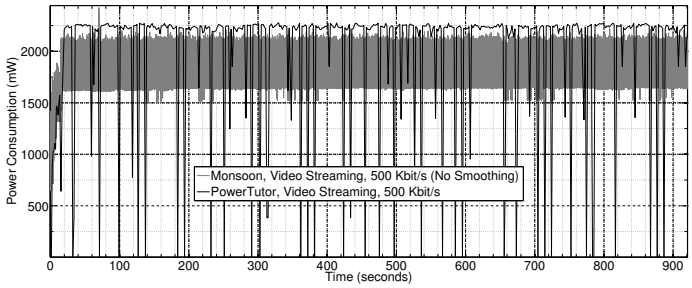


Figure 6.1: Measurements obtained via Monsoon and PowerTutor during video streaming [205].

consumption values (with 5 kHz sampling rate) obtained via Monsoon are within the

Bitrate	Tool	Max.	Min.	Std.	Mean	Med.	N
Kbit/s	With PowerTutor						
150	Monsoon	2449.3	1351.2	94.0	1786.2	1772.3	4425001
150	PowerTutor	2278.0	0	517.5	2078.8	2227.0	958
300	Monsoon	2404.0	1489.9	94.2	1762.5	1745.0	4375001
300	PowerTutor	2287.0	0	562.3	2047.6	2231.0	453
500	Monsoon	2423.4	1499.1	90.8	1793.6	1776.4	4425001
500	PowerTutor	2278.0	0	649.8	1993.8	2238 .0	442
Kbit/s	Without PowerTutor						
150	Monsoon	2198.8	1486.2	110.4	1727.4	1719.6	4425001
300	Monsoon	2170.1	1490.1	107.6	1739.0	1730.0	4425001
500	Monsoon	2201.9	1482.5	99.6	1753.1	1739.7	4425001

Table 6.1: Power measurements obtained through Monsoon and PowerTutor (mW).

robust 1600 mW−2200 mW range as shown in Fig. 6.1. Next, a video is streamed (with three different bitrates: 150 kbit/s, 300 kbit/s, and 500 kbit/s) to the device *with* and *without* the PowerTutor. In both scenarios, the power consumption measurements are recorded via Monsoon. It is concluded that PowerTutor consumed extra power within the range 23 mW and 59 mW. The descriptive statistics are presented in Table 6.1 for both scenarios, *i.e.*, *with* and *without* PowerTutor. Monsoon can provide highly accurate power measurements; yet, these battery-interception based measurements are highly obtrusive and can not be used in user studies. On the other hand, PowerTutor is minimally obtrusive to the user and can provide power models based on the device usage, but it relies on power measurements that are not as accurate as Monsoon due to factors such as unavailability of reliable sensors, or rather low sampling rate. Hence, the measurement tool should be carefully chosen depending on the purpose of the study and the limits should be reported in any discussion of the results.

In Section 6.2, we demonstrate simultaneous power consumption measurements and user indications with respect to the video quality variations during a network-based video streaming use case on a smartphone. For measuring the power consumption, we use PowerTutor and Monsoon power monitoring tools. We obtain the user's freeze indication directly from the user interface.

6.2 Demonstrating the Stalling Events with Power Measurements

Recent studies mostly focus on the averaged values of overall power consumption of applications in order to diagnose and increase the battery performance of handheld devices. Yet, there is still lack of focus on the variations in the power measurements. In network-based applications, the communication stack consists of standardized functions distributed into different protocol layers that consume energy on the communication systems. Thus, during the play-out of a video streaming application, any abnormal interrupt on one of those layers influences the instantaneous power consumption values.

Most popular video applications work based on transmission-controlled streaming, and a stalling event, so-called *freeze*, is a common impairment and it is considered as a key influence factor in user's perceived video quality [169]. Probing the underlying network-layer metrics during user studies, in order to identify the influential factors for poor user experience, needs high-energy demanding and hard-to-deploy monitoring tools. In addition, an instantaneous increase in the delay and packet loss rate does not always cause a video streaming being interrupted, *e.g.*, by a stalling event, due to its dependency on the size of the jitter buffer. The freezes and the corresponding fluctuations in the measured power values might be caused for different reasons. For example, during video streaming on a mobile terminal, when there is a disturbance (*e.g.*, long duration with no throughput) in the network traffic, some HLS clients with long play-out buffer are known to deactivate their network module. Derivation of robust power models that can represent the worst-case network scenarios can empower implementation of energy efficient QoE measurement tools.

In this section, firstly, it is demonstrated that the power consumption metric has potential to identify the misbehaviours in the communication stack during video streaming that have consequences such as stalling events. The live simultaneous measurements that match the energy consumption and the QoE is visualised. This is done while a live video is streamed and displayed on the Android smartphone. During the video streaming, the instantaneous power consumption values are collected and visualized through two different tools in parallel. The first one is a Monsoon's ground-truth hardware-based power-monitoring tool that can sample at a 5 kHz sampling rate called Mobile Power Monitoring Tool (MPMT) [170]. The second one is a modified version of a software-based power-monitoring tool called PowerTutor with a rather low sampling rate of 1 Hz. The MPMT measurements are directly visualized by the Monsoon Software that is running on a PC with WindowsXP and connected to the MPMT through a USB cable. The PowerTutor measurements are first transmitted over the 3G

interface to a dedicated web server, and then the collected data is visualized via a visualization tool called Software Visualisation Tool (SVT) in another platform. This way, the power measurements conducted at the same device with two different measurement tools are visualised, simultaneously.

6.2.1 Methodology

The experimental setup is summarized in Fig. 6.2. There are three main components: (i) HTC Dream G1 smartphone with Android 1.6, *i.e.*, the device that the measurements are conducted on, (ii) SVT for the software-based power measurements that gather the data from a software-based measurement PowerTutor tool, (iii) Monsoon power-monitoring tool for power measurements via external hardware. The smartphone streams a live video on a video player that is implemented for video QoE evaluation. The RTSP protocol is used for streaming via 3G interface via the Internet through the dedicated Darwin Streaming Server (DSS) running on a MacOSX (v.10.6.8). The video is MPEG-4 compressed, and it is displayed on a 176×144 pixels video pane with a frame rate of 25 fps. Since it is more probable to experience and demonstrate the stalling events while streaming at higher bitrates as compared to the lower ones, a video is streamed with a bitrate rate of 481 kbit/s during the demonstration. The applications and services on the smartphone that might influence the power consumption measurements are switched off. Together with the video player application, the modified version of the open source project PowerTutor is installed on the device. P_n is measured as the total instantaneous power consumption of the smartphone. The reason for describing the measurement as "instantaneous" is due to the very high sampling rate of MPMT. The following testbed is designed to demonstrate the stalling event with the live P_n values.

6.2.2 Hardware Measurements

Hardware measurements are conducted and visualized through the Monsoon's MPMT. It has its own software to visualize the measurements. The experimental setup is established while smartphone's battery is intercepted by MPMT. It is connected to PC for visualisation, and works with high precision [170], *i.e.*, it generates 5000 measurement samples (in milliwatts) per second. The consumed power at each sample is measured as the product of the instantaneous current and voltage.

The measurements through MPMT is considered as ground truth, however it has disadvantages in terms of portability. Especially in QoE studies, conducting non-obtrusive experiments is vital. In addition, with this setup alone, it is a challenge to

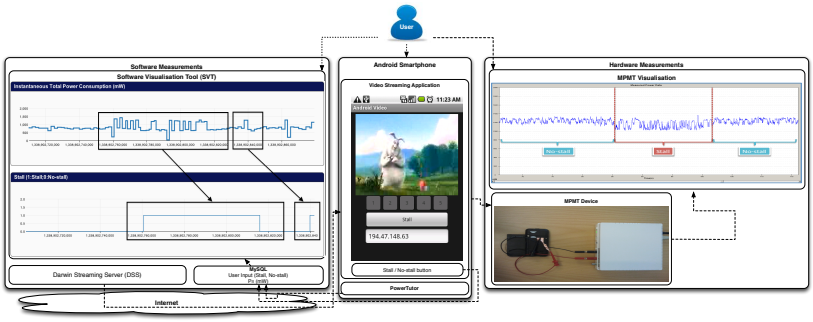


Figure 6.2: Experiment setting used for visualising the simultaneous power consumption values via SVT (left-hand-size) and MPMT (right-hand-size) during live video streaming [160].

synchronize the timestamps of the power consumption values, QoS, and the user behaviour metrics, simultaneously. Therefore, the experiments are complemented via software-based measurements.

6.2.3 Software Measurements

Software measurements consist of two components; the modified version of the open source project PowerTutor [172] for measuring the power consumption, and the SVT for visualisation. PowerTutor runs a background service that samples the power consumption values per second and sends to a dedicated MySQL server through the Internet in the form of JavaScript Object Notation (JSON) objects. The visualization tool SVT, is implemented via Javascript. It fetches the values stored in the database and visualises it on the HyperText Markup Language (HTML) web page.

6.2.4 Demonstration

The testbed in Fig. 6.2 is used during the demo. The audience is asked to watch a two minutes long video that is streamed via the 3G interface of the smartphone. During the experiments, the user is asked to press the "Stall" button when a stall is visible to the user in the video. Similarly the user is asked to press the "No-stall" button when the corresponding previously indicated stall ends. For the demonstration, we visualized the power measurements via MPMT and SVT, together with the user's indication of a stall region. On the right hand side of Fig. 6.2, P_n values obtained via Monsoon are presented

in a snapshot video streaming scenario that consists of a stalling event. *Stall* (a freeze) and *No-stall* (a smooth playout) regions are depicted with red and blue, respectively. Increased fluctuations of P_n values around a lower average is visible during a *Stall* region as compared to the *No-stall* one. On the left hand side of Fig. 6.2, the live measurements via PowerTutor are visualized via SVT with the corresponding region marked as *Stall* by a user. The SVT tool labeled the timestamps at different time scales, thus at a first glance, the user reaction looks delayed with respect to the fluctuation region in the power consumption.

The stalling events and their influence on the power measurements are demonstrated through MPMT and SVT. Energy savings are observed during the stalling events while streaming live video to the Android smartphone. On both measurements, increased fluctuations on the power consumption values during a non-user-interactive live video streaming session at its steady-state on a smartphone likely tells that either there exists a stalling event, which was caused by the communication channel, or due to another unexpected reason related to the application. Energy measurements are done via built-in applications in almost all mobile devices, therefore once robust power models for those misbehaviours are identified and deployed, the need of continuous collection of other network indicative QoS metrics with intrusive and hard-to-deploy tools may diminish in the future. Then, new QoE research methods involving power consumption can be suggested for QoE studies. By this way, those user studies on network depending applications can be done in a non-obtrusive and more energy efficient way.

6.3 Summary

Yet, the comparison between the two power measurement approaches is presented. We observed that detecting the QoE anomalies is possible via the power measurements, and the software based measurements (*e.g.*, PowerTutor) are not as good as detecting them as compared to the external hardware monitoring one, Monsoon. The reason is that the software-based PowerTutor has a low power-sampling rate. In order to capture the anomalies and then build models based on different network states, first, it is important to rely on ground truth measurements. Thus, we chose Monsoon power-monitoring tool and used it in further energy studies. In Chapter 7, we present extensive power measurements that are conducted during video streaming (via VLQoE) on a smartphone. We study the anomalies in the power consumption and the objectively measured metric (*e.g.*, inter-picture time) to identify the stalling events, and then suggest corresponding energy saving approaches for video streaming.

Chapter 7

Energy Saving During Video Streaming on the Smartphone

"If I have seen a little further, it is by standing on the shoulders of Giants." –Sir Isaac Newton.

7.1 Introduction

Desktop computers are replaced by smartphones as they enable those applications to be used in diverse context and environment. For example, applications such as instant messaging or video streaming/communication do not necessitate much extra accessories such as keyboard or mouse, thus they do not require the user to be on the desk while interacting with the application. In the case of video streaming, users can stream video on smartphones in diverse context, *i.e.*, anytime, anywhere (*e.g.*, while the devices are unplugged from the electrical power source), as long as the end-user's Quality of Experience (QoE) is satisfied. Video streaming applications running on the smartphones are a very popular form of entertainment, on the other hand they are highly energy-consuming, and their excessive usage decreases the operation time of the battery. Based on the findings presented in Chapter 2, the high energy consumption of applications eventually reduces the time it takes for the smartphone battery to be emptied, and this phenomenon is one of the most important influential factors influencing the user's QoE. The video streaming applications are also highly network bandwidth-demanding and are susceptible to degradations of the QoS metrics in the

end-to-end communication link. Eventually, these factors might reduce the level of the end-user perceived quality, and in fact might cause users being discouraged using the application. Thus, investigating the end-user perceived quality of those applications is important and while doing that we include the consideration of energy consumption in order to diagnose any possible energy waste in parallel.

Studying the high-level energy models of applications is a common approach to understand the overall energy consumption of an application that influences the battery life in the long-term. However, in case of streaming video, understanding the influence of occasional temporal impairments (*e.g.*, freezes) on the smartphone's instantaneous power consumption during a video stream, helps to relate the perceived QoE to the *anomalies* (*i.e.*, the abnormal significant drops) of the instantaneous power consumption values. Identifying this inter-relation between power consumption and the freezes can in turn help us to find out approaches to reduce the energy consumption during network-based video streaming. The freezes in a video stream might cause the CPU to transit into the "IDLE" state to reduce the energy consumption. When the video pauses or stops, the power consumption might drop as there is no data to be decoded and rendered to the screen display. When there is a poor network performance, one approach might be that the video player can change the total duration of the video stream while delaying or skipping particular parts of a video clip while maintaining the QoE.

In case of streaming video, the initial waiting time (timegap between when the user registers the command to start the video, and the video starts displaying on the screen) might reduce the QoE and increase the energy consumption. This initial waiting time highly depends on the end-to-end delay of the service, *i.e.*, the delay in the initial signaling of the IP packets. Thus, the increase in the initial signaling duration might also yield energy waste on the smartphone.

This chapter is structured as follows. We first present the energy consumption during smartphone-based video streaming and also investigate the freezes of a video streaming application in details in Section 7.2. In Section 7.3 and Section 7.4, we provide energy saving approaches for the case when single or multiple video streams are performed via the *first download and watch later* approach from local storage. In that case, we compare the download time and the energy consumption with respect to different file download scheduling approaches, and we recommend energy-efficient solutions for different file sizes. In Section 7.5, we study the relation in between the anomalies in the power consumption and the anomalies in a metric (*i.e.*, the inter-picture time) measured at the user interface of the smartphone during video streaming. In Section 7.6, we study the inter-relation between the energy consumption and the QoE during video streaming. In Section 7.7, we present our user study (with realistic video streaming settings on a smartphone in-the-wild) that aims to find out how much energy saving can be achieved while maintaining QoE. The limitations in studies conducted within this

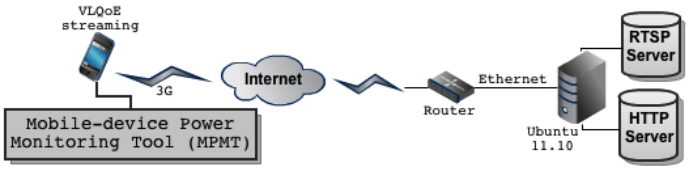


Figure 7.1: Testbed used during the study [206].

chapter are presented in Section 7.8, and the chapter summary is given in Section 7.9.

7.2 Energy Consumption During Video Streaming

We use VLQoE on the smartphone to stream video and collect application-layer measurements. In parallel, we measure instantaneous power consumption via Mobile Power Monitoring Tool (MPMT) (also called Monsoon device). The experiments are conducted in three different scenarios: local streaming (streaming from the local storage of the smartphone), RTSP streaming via 3G, and the HTTP streaming via 3G on the smartphone. We compare the energy consumption of all scenarios.

The display screen of smartphone is the closest point of the video streaming application to the user, thus the user interface is the preferred location to objectively quantify the freezes, and relate it to the user-perceived quality. The freezes are studied at the user interface with the objective metric of inter-frame (or also called inter-picture) time, D_{p}, as the time gap between the two consecutive displayed pictures on the smartphone screen. In addition, the metrics such as the initial signaling duration, D_{s}, on the network-level and the inter-packet time, D_{pkt}, are studied as they might influence the power consumption of the smartphone and the waiting time of the user.

7.2.1 Experiment Testbed and Method

The experiment testbed is designed as given in Fig. 7.1. The VLQoE tool is used in the experiments (see 5.3 for more information about the tool). A 250 s-long video clip (with a sailing race theme, 1000 kbit/s bitrate, 25 fps frame rate, 6251 pictures, 31.8 MB total file size), is streamed to a Samsung Galaxy S with Android v. 4.2.2. The videos are streamed via the 3G interface of the smartphone using a well known Swedish network provider, from a dedicated Ubuntu 11.10 machine located at BTH university. The Ubuntu machine was running two servers: (1) a DSS [234] for RTSP streaming, (2) a VLC media player server for HTTP streaming. The experiments are repeated with

exactly the same video via three different scenarios: via RTSP, via HTTP, and via the local storage of the smartphone. The local streaming is considered to be a baseline for the comparison with the other two network-based scenarios. The experiments are completed with more than 10 iterations for all scenarios. During each video session, the timestamp for each video picture is recorded (on the local storage of the smartphone) together with the corresponding inter-frame (or inter-picture) time. In addition, the *tcpdump* tool is executed on the smartphone to record all the packets that are transmitted/received from/to the device over the network. The packet-based statistics of the video streaming traffic is studied, as it might also impact the total power consumption. The time gap between the consecutively transmitted/received packets from/to the smartphone are investigated separately for both RTSP- and HTTP-based streaming. The incoming and the outgoing traffic of the smartphone during the video streaming is distinguished for the RTSP and the HTTP streaming. The incoming traffic is denoted by *ToPhone*; and the outgoing traffic is denoted by *FromPhone*.

The brightness level is fixed to a minimum constant level in all experiments. The timeout for the screen to shift to a sleep mode was set to a value longer than the video duration. Any interaction with the smartphone including touching the screen during the experiments is avoided. The VLQoE is restarted before every experiment run to flush the application cache. It is also assured that other services and applications that might influence the power consumption are stopped on the phone during the experiments.

7.2.2 Tools and Implementation for Metrics

There are four metrics being focused: the instantaneous power consumption, P_n; the inter-frame time, D_{p}; the inter-packet time, D_{pkt}; and the initial signaling duration, D_{s}.

Instantaneous Power Consumption (P_n)

The Samsung Galaxy S smartphone has a standard 3.7 V Li-ion battery [177]. The instantaneous power consumption (P_n) of the n^{th} sample is calculated in Eq. 7.1 as the product of the battery voltage, U, *i.e.*, 3.7 V (that is set through the MPMT software), and the instantaneous current, I_n, drawn by the smartphone and measured at the n^{th} sample via MPMT.

$$P_n = U \times I_n[\mathrm{mW}] \tag{7.1}$$

The high sampling rate of MPMT is required to distinguish the anomalies (unusual fluctuations of power consumption over time) from the normal behavior of the instantaneous power consumption pattern. However, MPMT has a sampling rate of

5 kHz (equally spaced samples), which might be higher than required. Thus, the *actual data* contained high frequency noise that made it hard to process the raw data.

'Moving average' techniques are used commonly by technical analysts to view the true underlying trend of erratic data [197]. We applied SMA to the power consumption dataset as shown in Eq. 7.2 in order to smooth out the high frequency noise from the actual data, and construct a separate dataset with the *smoothed data*. Amongst other moving average techniques, we chose SMA as it is producing less processing overhead while being as accurate as other more complicated smoothing algorithms [219]. The window size, *i.e.*, the number of samples under observation, should be set so that the data is smoothed, while preserving the characteristics of the power consumption pattern, *e.g.*, without losing the abnormal fluctuations. Thus, various window sizes (W) such as 2500 (0.5 s), 5000 (1 s), 7500 (1.5 s), 10000 (2 s), and 15000 (3 s) are studied while smoothing the actual data. The calculated SMA is given in Eq. 7.2, and the smoothed power consumption ($\bar{P}_n$) is considered for the later anomaly detection study.

$$\bar{P}_n = \text{SMA} = \frac{P_n + P_{n-1} + ... + P_{n-(W-1)}}{\text{mW}} \tag{7.2}$$

Based on the $\bar{P}_n$ values, two phases are defined during the video streaming session.

Phase 1 is defined as the first initialization region where the VLQoE initiates the streaming just after the "Play" button is pressed by the user. This region is when the $\bar{P}_n$ value starts increasing from a lower steady state region to a higher one as shown in Fig. 7.2. After *Phase 1*, the $\bar{P}_n$ follows a steady state region in *Phase 2*, and this continues until the end of the video session. However, in *Phase 2*, the steady state behavior of $\bar{P}_n$ might be impacted by occasional freezes during the video playout, and those regions are identified as "Freeze Regions". Fig. 7.2 illustrates the abnormal fluctuations that are detected as "Freeze Regions" in *Phase 2*.

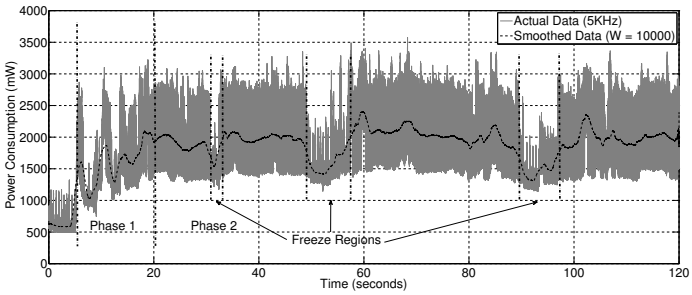


Figure 7.2: Instantaneous power consumption during 3G-based video streaming [206].

Inter-frame Time (D_{p})

To recall from Chapter 5, the inter-frame (or inter-picture) time, $D_{\mathrm{p}}(k)$, is studied as the time gap in-between the k^{th} and the $(k-1)^{\mathrm{th}}$ displayed picture, while ignoring the skipped pictures (Eq. 7.3). $T_{\mathrm{p}}(k)$ is the time when the k^{th} picture is displayed on the phone screen.

$$D_{\mathrm{p}}(k) = T_{\mathrm{p}}(k) - T_{\mathrm{p}}(k-1)[\mathrm{ms}] \tag{7.3}$$

High inter-frame time values manifest themselves as long-term freezes during the video playout. This way, it is possible to construct a direct relationship between what the user actually sees and actually perceives with respect to the QoE of the application. The abnormal peaks of the D_{p} and the $\bar{P}_n$ values are sketched in Fig. 7.3. A snapshot of

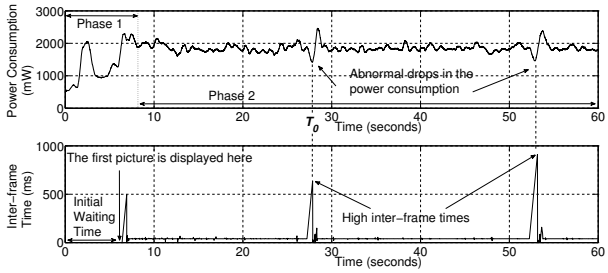


Figure 7.3: Abnormal behaviors are presented from the two datasets: *smoothed* instantaneous power consumption (top) and the inter-frame time (bottom) [206].

upwards D_{p} peaks and their relation to the downwards $\bar{P}_n$ peaks are sketched. The first video frame is displayed on the smartphone screen approximately six seconds after the user presses the "Play" button. Approximately at the 28^{th} s, *i.e.*, T_0, a high peak for D_{p} and a low peak for $\bar{P}_n$ are conferred.

The P_n and the D_{p} metrics are recorded on two separate platforms, thus the synchronization between the recorded timestamps is cumbersome. The timestamp for both datasets are reset with respect to the first abnormal peaks (the first local downwards peak in the $\bar{P}_n$ metric at Phase 1, and the first local upwards peak at D_{p}) to an initial time, T_0, as depicted in Fig. 7.3. Then, a set of local maximums for D_{p}, and a set of local minimums for $\bar{P}_n$, are identified and matched. Next, the inter-anomaly time for D_{p}, $\Delta T_{\mathrm{A(picture)}}$, and the inter-anomaly time for $\bar{P}_n$, $\Delta T_{\mathrm{A(power)}}$, are calculated. Inter-anomaly time for inter-picture time and power consumption are the time gap between the j^{th} and the $(j+1)^{\mathrm{th}}$ abnormal peaks for D_{p} and $\bar{P}_n$, respectively.

Inter-packet Time

The inter-packet time, $D_{\text{pkt}}(k)$, is calculated as the time difference between the k^{th} and the $(k-1)^{\text{th}}$ packets captured by the *tcpdump* tool during the network-based video streaming as shown in Eq. 7.4.

$$D_{\text{pkt}}(k) = T_{\text{pkt}}(k) - T_{\text{pkt}}(k-1)[\text{ms}] \tag{7.4}$$

D_{pkt} has been analyzed in four combinations, *i.e.*, two protocols and two directions: RTSP (*ToPhone*), RTSP (*FromPhone*), HTTP (*ToPhone*), and HTTP (*FromPhone*). The initial signaling packets that are received and transmitted at the beginning of the video stream are also investigated.

Initial Signaling Duration (D_{s})

The initial waiting time for a video streaming application is the time gap in-between when the user presses the "Play" button and when the first picture of the video is displayed on the smartphone screen. The initial waiting time might be influenced by the high D_{p} values at Phase 1, as we referred to as the total signaling duration, D_{s}. The D_{s} exists both for RTSP- and HTTP-based streaming. This duration is important, because if D_{s} gets too high, it might augment an extra energy consumption in the video stream as the total energy consumption within Phase 1 is proportional to the duration. In other words, a longer initial signaling period means higher energy consumption. The signaling network traffic for the RTSP streaming is different than the HTTP streaming traffic. In RTSP streaming, the signaling duration, D_{s}, is considered as the time difference between the first packet that is sent to the server and the first RTP packet that is received at the smartphone. The first RTP packet is received at the mobile terminal just after the *RTSP/1.0 OK* message being received as a reply to the *PLAY* message that was previously sent to the streaming server from the smartphone. In HTTP streaming, the signaling duration is considered as the time difference between when the first TCP packet sent to the server from the smartphone, and when the first Protocol Data Unit (PDU) is received at the smartphone as a reply to the *GET* message that was previously sent to the streaming server. 41 iterations were executed both for RTSP and HTTP streaming to get statistically significant data. Then, the packet-based initial signaling durations are analyzed for both protocols.

7.2.3 Results

Occasional freezes, *i.e.*, relatively high inter-frame times, are observed during the video playout when the video was streamed via the 3G interface of the smartphone. There are

differences in the frequency and the duration of those freezes, which results in variation in the total streaming time and displayed number of packets amongst the experiment iterations. The overall statistics with respect to the durations and the number of pictures that are displayed on the smartphone screen per video streaming session are given in Table 7.1. From left to right; the mean, standard deviation, minimum, and the maximum values regarding the durations and the number of displayed pictures are presented for the three scenarios: local, HTTP, and RTSP streaming. Local streaming is considered to be a baseline for the comparison with the other two network-based scenarios. The mean of the total video stream duration, and the mean number of the displayed pictures on the smartphone screen are the highest for HTTP. The standard deviation of the number of displayed pictures is the highest for the RTSP-based video stream, which can provide an indication about the high deviation in the number of skipped pictures due its underlying transport layer protocol, *i.e.*, UDP.

Table 7.1: Overview on the duration and number of pictures.

Scenario	Duration [s]				Displayed Pictures			
	Mean	Std.	Min.	Max.	Mean	Std.	Min.	Max.
Local	250	0.50	248	250	6239	17	6192	6250
HTTP	253	1.79	250	254	5978	35	5907	6047
RTSP	250	0.20	249	250	5937	82	5830	6075

Instantaneous Power Consumption (P_n)

Exactly the same number of data points (eight million) are collected for all scenarios during Phase 2 of the video stream. The fluctuations of P_n in the beginning, *i.e.*, Phase 1, and in the end of the video streaming are not included in the data set. This is done to mask the influence of the initial and the terminating signaling traffic. The P_n values are calculated for all three scenarios, and the CCDF plots for all the three scenarios are given in Fig. 7.4. The median is 1000 mW in local streaming, while it is 1750 mW in case of HTTP and RTSP streaming.

The highest observed P_n value was 2800 mW for local streaming, while the highest values for HTTP and RTSP streams were 3400 and 3550 mW, respectively. RTSP- and HTTP-based video streaming yield higher power consumption as compared to the local streaming scenario, which might be due to the factors such as the active state of the 3G cellular data module because of continuous data transmission.

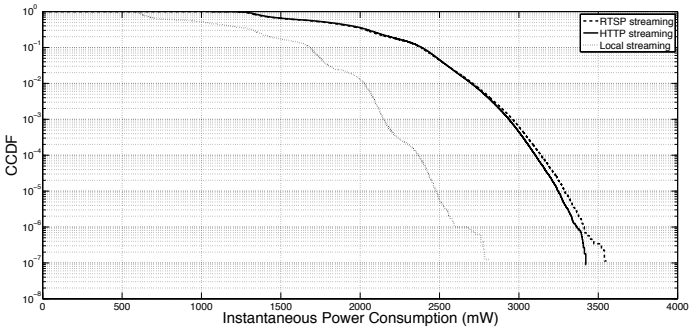


Figure 7.4: Power consumption during Phase 2 [206].

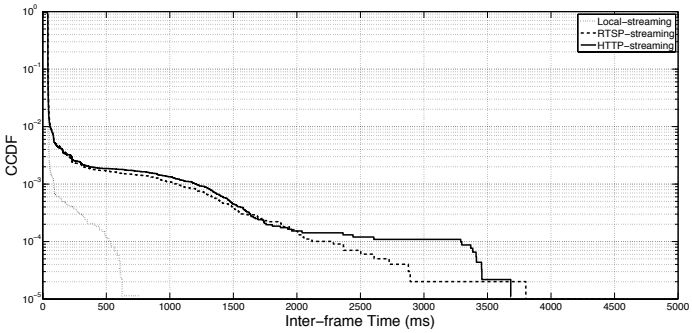


Figure 7.5: CCDF of inter-frame time in three scenarios [206].

Inter-frame Time (D_p)

CCDF plots for the inter-frame time (D_p) are presented in Fig. 7.5. CCDF plot enables us to visualize the very high and rarely occurring inter-picture time instances, such as OFF phases. Although, no freezes are expected for local streaming, still a few, short (duration up to 700 ms) freezes are observed. The reason for the freezes might be due to the sporadic latency in the application's decoding process. RTSP- and the HTTP-based video streaming show similar D_p distributions for the range up to 2 s. However more frequent high ($\geq$3 s) inter-frame time values (*i.e.*, longer freezes) are observed in the HTTP stream as compared to the RTSP stream.

Inter-packet Time (D_{pkt})

The mean, standard deviation, minimum, and the maximum number of packets per video streaming session are listed from left to right in Table 7.2. The number packets of the four streaming traffic scenarios are listed from top to bottom: HTTP from smartphone, RTSP streaming from smartphone, HTTP streaming to smartphone, and the RTSP streaming to smartphone. In average, there are much more packets transmitted from the smartphone to the streaming server during HTTP streaming as compared to the RTSP streaming as expected. The sum of the mean number of received and transmitted packets for HTTP and RTSP streaming are 26062 and 25236, respectively.

Table 7.2: Overview on the packets-based statistics.

Scenario	Packets			
	Mean	Std.	Min.	Max.
HTTP-FromPhone	3761	32	3713	3828
RTSP-FromPhone	126	3	122	132
HTTP-ToPhone	22301	15	22248	22309
RTSP-ToPhone	25110	152	24907	25267

The CCDF plots of the inter-packet time for the four scenarios are presented in Fig. 7.6. 90% of the D_{pkt} values of the transmitted packets, *i.e.*, (FromPhone), are less than or equal to 130 ms in HTTP streaming, while this value corresponds to 4 s in RTSP streaming. This is due to the periodic transmission of ACK messages to the streaming server, and this periodicity of the RTP Control Protocol (RTCP) messages in UDP-based RTSP streaming is much lower as compared to the TCP-based HTTP streaming. Therefore, less signaling packets are involved in the transmission during the RTSP-based video stream as compared to the HTTP-based one.

Initial Signaling Duration (D_{s})

The statistics for the initial signaling duration, D_{s}, obtained through 41 successful iterations, for both streaming protocols is given in Table 8.5. The standard deviation of D_{s} in HTTP streaming is much higher as compared to RTSP streaming. A maximum D_{s} of 19.06 s is observed during the HTTP streaming, while D_{s} did not exceed 4.73 s in RTSP streaming. The mean D_{s} values are 3.98 s and 4.55 s for RTSP- and HTTP-based streaming, respectively.

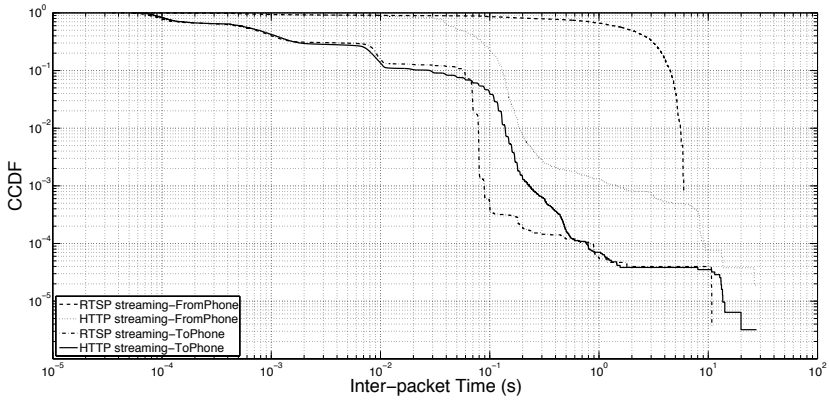


Figure 7.6: CCDF of inter-packet time [206].

Table 7.3: Initial signaling duration statistics.

Protocol	Signaling Duration [s]						
	Mean	Median	Std.	Min.	Q1	Q3	Max.
RTSP	3.98	4.31	0.75	0.73	3.54	4.41	4.73
HTTP	4.55	4.39	2.74	1.93	3.32	4.56	19.06

Yet, we have presented results from our measurements during real time video streaming on the smartphone. It is also necessary to take into consideration other video streaming scenarios such as "download first and watch later". In this scenario, the full video file is first downloaded from the source and stored in the device, then the user initiates the stream of the downloaded file via the local device storage. We measure the total energy consumption and the total duration for this scenario in the next subsection. This in turn helps to understand the cons and pros of the two scenarios (*i.e.*, real time video stream and local stream) with respect to energy consumption, the total time it takes to finish watching a video, and the experienced distortions in the video stream.

7.2.4 Video File Downloading: Energy and Download Duration

We measure the power consumption and the download duration while downloading file via the 3G network, *i.e.*, without a real time stream. We measured the power con-

sumption with Monsoon power monitoring tool. The timestamps when the downloads are initiated and when they are completed, are recorded by our download software tool, *i.e.*, DownloadTrainCatcher, on the local storage, *i.e.*, sdcard, of the smartphone. The experiments are conducted with Samsung Galaxy S smartphone via 3G SIM card (from a popular network provider in Sweden) with 6 Mbit/s downlink peak rate. The experiments are conducted randomly during different times of the day to improve the representativeness (*i.e.*, generalizability) of the download duration. In total, we have performed 36 download iterations, and we identified one outlier (174 s). After the outlier is filtered out, we obtained the statistics shown in Table 7.4. In the table, the mean, standard deviation, minimum, maximum, and the number of data points are given. Based on our measurements, the mean download duration for a 31.8 MB video file is 83.5 seconds.

Table 7.4: File-downloading Metrics.

Metric	Mean	Std.	Min.	Max.	Samples
Duration [s]	83.5	18.6	53.9	123.3	35

The power consumption during a file download consists of various Radio Resource Controller (RRC) states based on the throughout on the 3G interface. A snapshot from a power measurement is given in Fig. 7.7. The power measurements are conducted via

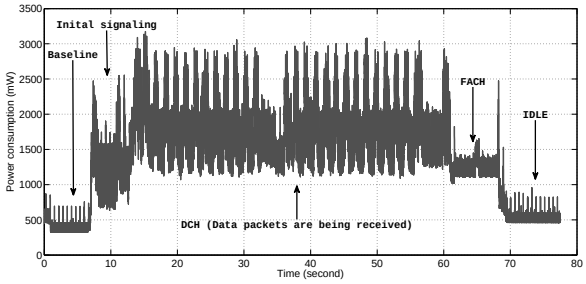


Figure 7.7: Power consumption while downloading the video file via smartphone's 3G.

10 iterations, and we extracted the DCH power consumption (Phase 2) values. The power consumption during downloading the video file are given in and Table 7.5. The median Phase 2 power consumption was measured as 1580 mW.

Table 7.5: Power consumption while downloading the video file.

Metric	Median	Std.	Min.	Max.	Samples
Power [mW]	1580	431	1033	3154	1670016

Now, we have the parameters from the measurements for the model, and we can compare two scenarios: (1) Real time video streaming via the 3G network, (2) downloading the same video file and streaming it from local storage.

7.2.5 Comparing the 3G streaming and the Local streaming

We first show the power consumption and the download duration in two different scenarios. The first one is when the video is being streamed directly via 3G. We use the measurements in the previous section (Table 7.1), *i.e.*, the average video streaming duration during RTSP and HTTP streaming, 252 seconds. Based on the measurements, we assume that the mean power consumption during streaming a video via the 3G is 1750 mW. For the local streaming scenario, we measure the mean power consumption as 1000 mW (while the cellular data module being disabled) with a mean streaming duration of 250 s. In the local streaming scenario, we also add the initial downloading phase. For simplicity, we assumed that the download duration is 84 seconds (is measured as 83.5 s in Table 7.4). We also neglect the initial signaling duration on all scenarios, and assumed the mean power consumption only during the actual file-downloading phase. We also assume that the local video stream starts immediately after the file is downloaded to the local storage. We calculate the total energy consumption in the two scenarios as given in Eq. 7.5. In local streaming we add the initial downloading phase to the local streaming phase; however in the 3G streaming, there is no initial downloading thus $T_{\text{downloading}}$ and $P_{\text{downloading}}$ are zero.

$$E_{\text{total}} = \bar{P}_{\text{streaming}} \cdot \bar{T}_{\text{streaming}} + \bar{P}_{\text{downloading}} \cdot \bar{T}_{\text{downloading}} \tag{7.5}$$

In Fig.7.8, the energy consumption is illustrated with respect to the time for the two scenarios. The total energy consumption increases linearly with the time. In 3G streaming scenario, the data transmission and display of the video are performed simultaneously, thus the slope (*i.e.*, power) of the line is steeper. In local streaming, there is the initial downloading phase with a rather less steep (*i.e.*, due to less power consumption) until the 84$^{\text{th}}$ second, and the energy consumption increases with a lower slope while streaming the downloaded file from the local storage. The total energy consumption

is calculated as 441 J and for 3G streaming scenario, and 383 J for the local streaming scenario. Although there is a 58 J energy saving if the stream is downloaded and then streamed through the local storage (which might be good for long term QoE). However, the total duration is increased, which might influence the short term, *i.e.*, recent, QoE. The average streaming duration for the 3G-based streaming is 252 s, while it takes 82 seconds longer (in total 334 s) to download and locally stream the same video on the smartphone. In addition, additional freezes occur during 3G streaming, while there are no freezes in the download and stream scenario. The additional freezes in 3G-based streaming is expected to increase the energy consumption and the streaming duration. 3G-based stream is recommended if the user wants to start watching a video immediately, *i.e.*, without waiting for it to be downloaded, and local-based streaming is recommended if the user prefers to save energy and also do not want to experience any freezes during the video stream. Table 7.6 summarizes the pros and cons of the

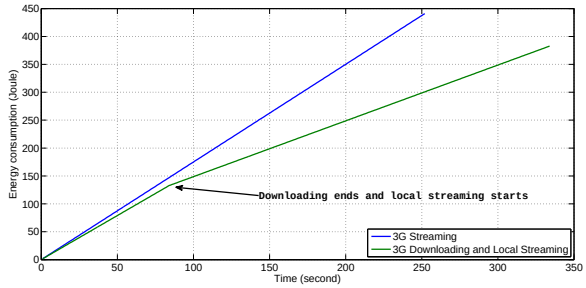


Figure 7.8: 3G Streaming vs 3G Downloading + Local Streaming (considering only Phase 2).

3G-based and the local streaming.

Table 7.6: Comparing the 3G streaming and the local streaming.

Scenario	Energy	Duration	Initial delay	Occasional freeze
3G-based	Higher	Lower	Lower	Yes
Local	Lower	Higher	Higher	No

We have discussed the energy consumption during video streaming, and eventually compared the energy consumption and the video streaming duration of 3G-based and

the local-based streaming use cases. During the 3G-based real-time video streaming, the files are downloaded and displayed simultaneously. In local-based streaming, the file is first downloaded and then streamed. In the next section, we study the energy consumption during file downloading, and recommend approaches to save energy especially when there are multiple files are desired to be downloaded.

7.3 Saving Energy During File Downloading

Most of the energy is consumed on a smartphone while sending and receiving data over the network, with the clear domination of 3G data module's power consumption [99]. The amount of data exchanged in-between smartphones and corresponding servers, the used access technology, as well as the timing of data transmission have a significant impact on the overall energy consumption of the smartphone. The oscillation of RRC states are basically caused by the asynchronous data activities of different applications over the cellular data interface. During the RRC state transitions, also due to the necessary signaling overhead, additional transition delays are induced. These additional delays increase the total energy consumption. In addition, during the active (DCH) state of RRC, the available network bandwidth is not fully utilized due to rather low volume data traffic per application. In order to avoid the high number of oscillations amongst the RRC states, it is important to schedule the network data activities of different applications. This is done by scheduling the network data activities such that the RRC state is kept IDLE for a longer period of time, but it is utilized as much as possible while it is fully active, *e.g.*, in DCH state. The scheduling can be done by delaying particular network activities, and performing/clumping them. This way, both the overall energy consumption and data transmission duration can be reduced. Hence, it is appealing to transmit data from multiple applications simultaneously and thus reducing the number of state transitions. Multiple TCP-based data flows can be transmitted in a serialized or in a parallel manner.

In this section, we focus on file downloading (*e.g.*, downloading videos/documents for later offline usage) on smartphones. We first measure the energy consumption and the download duration when multiple files are downloaded on a smartphone via 3G asynchronously, and then when the downloads are scheduled (*e.g.*, either in parallel or in serialized manner). Next, we repeat the measurements when the downloads are performed through WiFi tethering via another smartphone. The former measurements help to decide on the best way of download scheduling to reduce energy if the user wants to download multiple files immediately via the available 3G. The latter measurements help to identify the energy gain when the same files are downloaded via rather low energy consuming WiFi, but with the cost of extra waiting time assuming

that the WiFi is not available in-situ. We measure the total power consumption via Monsoon power monitoring tool [170] (see Chapter,6), and the download duration via our download tool in a realistic environment. In the measurements, we involve two different smartphones, and focus on large-sized file-downloading from a dedicated file server. The overall download duration and power consumption highly depend on the Domain Name Server (DNS) resolution time (if any), initial TCP three-way handshake time, packet latency, and the available end-to-end network bandwidth. Thus, we break down and study the energy consumption in three different phases.

This section is structured as follows. In Section, 7.3.1, further details regarding the download scheduling mechanism of files are given. In Section 7.3.2, the testbed and the data collection are discussed in details. The measurement results for different downloading scenarios are given in Section 7.3.3.

7.3.1 Scheduling Network Traffic on Smartphones

3G UMTS networks are efficient for large data transfers [229]. If there are multiple files to be downloaded with some time-gap in-between larger than the timeout values, oscillation might occur in-between the RRC states which causes high energy consumption.

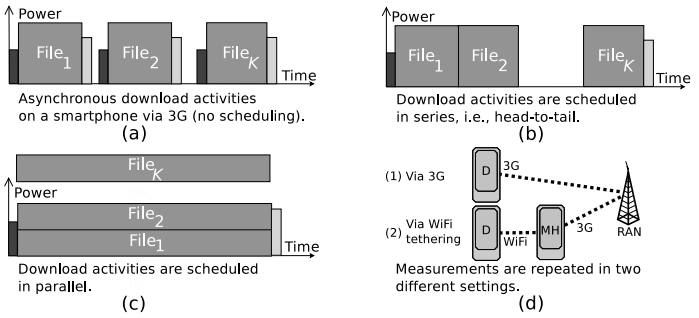


Figure 7.9: The energy consumption at Phase 1 (black), Phase 2 (gray), and tail phase (white) are depicted for asynchonous downloading (a), serialized downloading (b), and parallel downloading (c), respectively [181].

The sketches for different file-downloading scenarios via 3G are given in Fig. 7.9a, Fig. 7.9b, and Fig. 7.9c. Fig. 7.9a depicts the scenario when file downloads are initiated as scattered over a long duration with no scheduling (*e.g.*, asynchronously) on a single smartphone. We break down and study the energy consumption within three different

phases. The area of the black rectangles represent the total energy consumption on the smartphone during channel allocation via RRC, and establishing a connection with the server. This is referred as *Phase 1*. During Phase 1, the three-way handshake takes place after the DNS resolution (if any). If there is no traffic due to the lack of bandwidth, the RRC state keeps at Phase 1 for some duration, which is a waste of energy. Thus, it is preferred to keep the Phase 1 duration as short as possible. *Phase 2* starts with arrival of the first TCP segment that contains the file content. Phase 2 energy consumption of different files are represented with gray rectangles. The CPU utilization and the throughput increase as compared to Phase 1, which cause the RRC to stay at DCH state. After the last data packet is received at the smartphone, Phase 2 waits for an extra inactivity period, then the *Phase 3* (also called the tail phase) begins. There is an extra tail duration, comprising an inactivity timer to release the resources in-between the smartphone and the Radio Access Network (RAN). Phase 3, depicted with white rectangles appended to each download in Fig. 7.9a, causes inefficient usage of resources if there are multiple scattered requests [185]. For all individual downloads in a smartphone, Phase 1 (black) and the tail phase (white) exist, and thus induce high energy consumption, which is actually energy waste. In Fig. 7.9b, the energy consumption is depicted when multiple file downloads are scheduled in series. As the second file download starts immediately after the previous one, the number of Phase 1 and Phase 3 states reduces. Fig. 7.9c depicts the download scenario when multiple files are downloaded in parallel. We refer to the formed energy consumption area, as a result of multiplexing different file download activities, as the *energy train*. The length of the train is the sum of all phase durations; and the height depicts the corresponding mean power consumption of the phases.

7.3.2 Measurement Method

Energy consumption is a function of the power consumption and duration of a download session. The power consumption and the duration are measured on the smartphone during serialized and parallel downloading in two parts: (i) via 3G, (ii) via WiFi tethering over another smartphone, Mobile Hotspot (MH) (see Fig. 7.9d). In the first part, D (tethering device) is directly connected to the Internet via Telenor's 3G (HSDPA and HSPA+) SIM card with 6 Mbit/s downlink peak rate. In the second part, D is connected to the Internet via the MH that uses the same SIM card. MH is a non-rooted Samsung Galaxy S4 with 1.9 GHz CPU running Android v.4.3 (kernel v.3.4); and D is a rooted Samsung Galaxy S with 1 GHz CPU running Android v.4.2.2 (kernel v.3.0). In order to automatize measurements, we developed a download tool for the smartphone to perform the HTTP downloads. For each file download, separate TCP connections (*e.g.*, sockets and threads) are created. We record the ground truth power measurements with

the MPMT [170]. The *tcpdump* tool helps to record the data packets during file downloading, however running it on the smartphone induces additional overhead and increases the power consumption. Thus, we have used the *tcpdump* tool to measure only the average Phase 1 duration. Throughout all experiments, the download durations are recorded directly via our download tool. During the experiments, screen brightness is set to minimum, and all irrelevant applications and services are switched off. We created multiple files with the identical size (with *mkfile -v XM <filename>*), and used them during the experiments. The choice of the file size was arbitrary. A representative file size of a three minutes-long podcast decoded with a nominal 1000kbit/s rate is 23 MB. We chose 20 MB files, assuming a slightly lower file size (for simplicity).

There are various scenarios in download scheduling experiment, and the scenario descriptions and the corresponding abbreviations are summarized in Table 7.7.

Table 7.7: Download scenarios and definitions.

Abbreviation	Definition	Abbreviation	Definition
P1	Phase 1	P2	Phase 2
tl	Tail phase	T_{download}	Total download dur.
MH	Mobile hotspot	D	Tethering device
1S	1 single download	K	Number of queued files
4x1S	4 individual single downloads	2x1S	2 individual single downloads
2P	Two parallel downloads	2S	Two serial downloads
4P	Four parallel downloads	4S	Four serial downloads

Download Duration

The timestamps when the downloads are initiated and when they are completed are recorded by our download tool on the local storage, *i.e.*, sdcard, of the smartphone. Accordingly, we calculated the total download duration for the serial and parallel scenarios as the difference between the last and the first timestamps. We divided the total download duration, T_{download}, into two phases. Phase 1 is the initial connection duration, T_{P1}, *i.e.*, in-between when the first SYN is sent from the smartphone and the first TCP segment with the file content is being received. We recorded the relevant metrics via the *tcpdump*. The duration of Phase 2 is calculated as the difference between the total download duration measured by the download tool (T_{download}) and the Phase 1 duration (T_{P1}) measured by *tcpdump*. The DNS resolution durations are neglected. The

download durations depend on the available link capacity at the "time of the day" of the experiment. In order to minimize this effect, we iterated each experiment for more than 30 times in a random order. For parallel downloading, we have validated that all downloads are performed simultaneously 98 % of the time. In serialized downloading, we have also validated (via 31 iterations) that the difference between the serialized downloads is in the range 90 ms to 400 ms for 90 % of the time. However, when the file downloads are serialized, *i.e.*, queued, the download completion time of a file depends on the download time of the previous files, *i.e.*, the files in the front. Thus, we define the overall processing time as the mean of download completion time of each serialized file, k. The average processing time, for K files in series (*i.e.*, in queue), $\bar{T}_{\text{processing}}$ is computed as in Eq. 7.6, where T_{download} is the total download time it takes to download K files. We show that the average processing time is a necessary parameter to study as it also considers the user action during the waiting time. In other words, the user can do tasks with the downloaded files, while waiting for other file downloads to be completed. This could guide us to approach the problem from the QoE perspective.

$$\bar{T}_{\text{processing}} = \frac{1}{K} \cdot \sum_{k=1}^{K} \left(\frac{k}{K} \cdot T_{\text{download}} \right) \tag{7.6}$$

If the file download is not queued, such as for parallel downloading, requests do not have to wait, thus processing time is equal to the total download time since $K = 1$. We also calculated the average per-file downloading time as the mean of download duration of all individual files.

Power Measurements

Power measurements are conducted using the Monsoon power monitoring tool. Between each experiment, we wait until the RRC state switches to IDLE. This way, we reduce the possible side effects between the measurements. Firstly, the power consumption of D is measured while downloading the files via the 3G. The power consumption pattern is studied for the three different phases: Phase 1, Phase 2, and tail phase. The tail phase duration is not measured, as this is highly depending on the operating system and the network configuration [282]. Thus, we assume the state-of-the-art tail phase duration, 12 s [228].

The power measurements differ for D and MH, while downloading via WiFi tethering. For D, the power is consumed by transmitting data via the WiFi interface. Here, we only consider Phase 2, since the other phases are very short and thus their impact is negligible. If there is no data activity, the power consumption drops within significantly shorter time as compared to 3G. For the MH, the energy measurements depend on the data sent and received via the WiFi and the 3G interface. Accordingly, all three

phases are considered. In order to reduce the measurement error, sufficient number of samples are collected, for each download phase, during steady state.

Energy Consumption: Downloading via 3G

On a single smartphone, the file downloads are executed for different scenarios on D via the 3G interface: (1) one download, 1S; (2) two files being simultaneously downloaded, 2P; (3) two files being downloaded in a serialized manner, 2S; (4) four files being simultaneously downloaded, 4P; (5) four files being downloaded in a serialized manner, 4S. In serialized downloading, the download tool executes the downloading of different files in a sequential order, and the download of a file is immediately triggered when the download of the previous file is completed. We calculate the total energy consumption for a multiple file-downloading scheme via 3G on a single smartphone, D. E_{total} is the total energy consumption during file downloading. "P1", "P2", and "tl" are the subscripts for the power consumption of Phase 1, Phase 2, and the tail phase, respectively (see Table 7.7).

$$E_{\text{total}} = \bar{P}_{\text{P1}} \cdot \bar{T}_{\text{P1}} + \bar{P}_{\text{P2}} \cdot \bar{T}_{\text{P2}} + \bar{P}_{\text{tl}} \cdot \bar{T}_{\text{tl}} \tag{7.7}$$

$\bar{P}_{\text{tl}}$ and $\bar{T}_{\text{tl}}$ are the mean power consumption and the duration of the tail phase, respectively. $\bar{P}_{\text{P1}}$ is the mean power consumption of the smartphone during Phase 1, and $\bar{T}_{\text{P1}}$ is the mean duration of Phase 1. $\bar{T}_{\text{P1}}$ highly depends on the initial duration including the RRC state switching delay from IDLE to DCH. $\bar{P}_{\text{P2}}$ and $\bar{T}_{\text{P2}}$ are the mean power consumption and the mean duration at Phase 2 during all simultaneous downloads. When the downloads are scheduled, there is only one Phase 1, and one tail phase, as they are clumped forming an energy train. All the parameters are measured for serialized and parallel downloading, separately.

Energy Consumption: Downloading via WiFi-tethering

The measurements for WiFi tethering were conducted using one smartphone acting as MH, and one acting as D, which actually performs the downloads. MH uses the 3G for Internet access, and in parallel shares its Internet access with smartphone D via WiFi tethering. In this use case, D downloads the files from a dedicated server on the Internet via WiFi tethering over MH that has 3G. We measure the power consumption and the download duration at D and as well as the power consumption on MH on the same scenarios as in Section 7.3.2, *i.e.*, serialized or parallel downloading. The power consumption of MH is measured in different scenarios: when WiFi tethering is OFF (baseline measurements); when WiFi tethering is ON with no connections; when

WiFi tethering is ON and D is connected; when WiFi tethering is ON and D is downloading a single file via MH, 1S; when WiFi tethering is ON and D is downloading two files in parallel via MH, 2P; and when WiFi tether is ON and D is downloading four files in parallel via MH, 4P. The total energy consumption during WiFi tethering is the sum of energy consumed at MH and the energy consumed at the connected device, D. The MH power consumption is expected to increase with the number of parallel downloads. Although the total energy consumption is expected to increase, there might be energy saving at individual WiFi tethering smartphones, D.

7.3.3 Measurement Results

We measured the download duration and the power consumption while downloading with respect to a set of scenarios in realistic environment. The collected steady-state power consumption values are normally distributed ($R^2 > 0.9$). The power consumption is broken down into Phase 1, Phase 2, and the tail phase. Phase 2 and Phase 1, in 3G, have the highest and the least mean power consumption, respectively.

Power Consumption During File Download

First, the power measurements when D downloads files via its individual cellular data interface (3G) are given. Next, we discuss the power measurements during WiFi tethering. The measurements are presented through Table 7.8-7.9 as the 95 % confidence levels are negligibly low (mean: 0.47mW) due to high number of samples (see last columns of the tables).

Downloading via 3G: The measured power consumption values on D for different file-downloading scenarios via the 3G interface are given in Table 7.8. In the table, from left to right, the mean, standard deviation, minimum, maximum of the power consumption values, and the corresponding number of samples at each scenario are presented. The power consumption statistics at Phase 1 and at the tail phase are given at rows $1-2$, respectively, and the corresponding mean values are lower than the mean of Phase 2 power consumption values as presented at rows $3-7$. Row 3 shows the power consumption values when there is only one download (1S). In rows $4-5$, which belong to the serialized downloading, no significant difference in the power measurements are observed when different numbers of files are scheduled. This is expected as there is only one established TCP connection at a time during downloading. On the other hand, when the downloads are executed in parallel, the power consumption slightly increases with the number of simultaneous downloads as given in rows $6-7$. In these scenarios, the number of established connections at a time is equal to the number of simultaneous

downloads, *i.e.*, threads. One reason for the increase in power consumption could be due to the increase in the CPU utilization.

Downloading via WiFi Tethering: The measured power consumption values of D for the same scenarios, while downloading exactly the same file via WiFi tethering over MH's 3G interface, are presented in Table 7.8 in rows $8-10$. D uses the WiFi interface, and the power consumption values are reduced as compared to using the 3G interface. When the downloads are executed in parallel, slight increase in the power consumption values is observed. However, this gives an extra overhead to MH. Next, we present the power consumption values on MH.

Table 7.8: Power Consumption (mW) on Samsung S (D).

Row	Scenario	Mean	Std.	Min.	Max.	Samples
1	3G, P1	1153	167	703	2370	275014
2	3G, Tail	1276	86	1063	2217	312511
3	3G, 1S, P2	1496	283	1003	3597	1300008
4	3G, 2S, P2	1508	298	1006	3799	1550004
5	3G, 4S, P2	1514	313	955	3928	3500005
6	3G, 2P, P2	1662	354	931	3717	1950007
7	3G, 4P, P2	1956	409	1114	4077	3150007
8	WiFi tethering, 1S	957	240	386	2950	1575017
9	WiFi tethering, 2P	1266	333	547	3278	1975008
10	WiFi tethering, 4P	1350	356	514	3330	2175009

Impact of WiFi Tethering on MH: The power consumption values measured at MH during different WiFi tethering scenarios are given in Table 7.9. Phase 1 and the tail phase power consumption values are given in rows $1-2$, respectively. Rows $3-5$ present the power consumption values when the WiFi tethering on MH is switched OFF, switched ON, and D is connected (without download), respectively. Rows $6-8$ present the Phase 2 power consumption values when D is downloading a single file, two files in parallel, and four files in parallel, respectively. There is a significant difference in power consumption between the scenarios when the WiFi tethering is enabled or not. However, there is no significant difference if the smartphone D is connected or not given that D does not download data. The power consumption of MH slightly increases with the number of concurrent downloads. As the focus is more on the battery lifetime

Table 7.9: Power Consumption (mW) on Samsung S4 (MH).

Row	Scenario	Mean	Std.	Min.	Max.	Samples
1	3G, P1	997	130	661	2889	475005
2	3G, Tail	1354	120	1047	3050	350006
3	3G, TetherOFF	549	106	371	2021	925004
4	3G, TetherON	845	179	628	2715	1475003
5	3G, D Connected	844	182	632	3112	700001
6	3G, TetherON, 1S, P2	1952	310	1304	4137	615005
7	3G, TetherON, 2P, P2	2146	288	1411	4391	625003
8	3G, TetherON, 4P, P2	2179	336	1374	4474	1325004

of multiple smartphones, one out of many smartphones can sacrifice its battery lifetime (or plug into a power grid), while multiple surrounding smartphones benefit.

Download Durations

We first cleaned the measurement data from a few outliers that were confirmed as measurement errors. The cleaned Phase 1 duration dataset comprises 31 iterations. Phase 1 duration is calculated as $4.6s \pm 0.3$ s with 95 % confidence level. The duration of a

Table 7.10: Download duration (s) for multiple downloads.

Row	Scenario	Mean	Std.	Min.	Max.	Iterations
1	3G, 1S	76.6	20.1	48.0	122.7	31
2	3G, 2P	129.2	42.2	65.3	235.7	32
3	3G, 4P	201.7	43.4	138.6	320.7	34
4	3G, 2S	129.7	15.8	96.5	158.6	31
5	3G, 4S	256.5	42.9	188.5	329.4	31
6	WiFi tethering, 1S	69.2	13.9	45.1	103.8	29
7	WiFi tethering, 2P	81.2	12.0	63.6	111.9	28
8	WiFi tethering, 4P	167.2	16.6	136.6	216.4	30

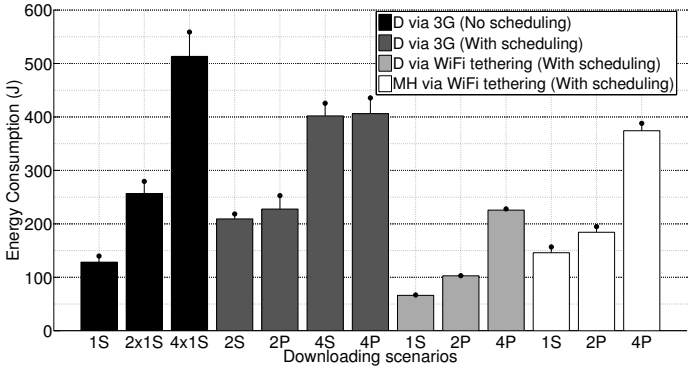
single file download via 3G is given in the first row of Table 7.10. The mean, standard deviation, minimum, maximum of the parallel download durations via 3G on D for two

parallel (2P), and four parallel (4P) are given in the second and third rows of Table 7.10, respectively. The download duration increases with the number of concurrent downloads on a single smartphone. The total download duration, during both the two serial and four serial download scenarios, are given in the fourth and fifth rows of Table 7.10. We applied one-way Anova test on the individual download durations during two and four serial downloads as the download durations are normally distributed ($R^2 = 0.98$). We conclude that there is no statistically significant difference in-between each file's download durations during serialized downloading. The total download duration for serialized file-downloading via the 3G interface is less than the total download duration for parallel downloading. For parallel downloading, this indicates that the available link is too much utilized, with high number of TCP connections at a time, causing packet losses, retransmissions, and higher *slow-start* phases. Eventually, the link becomes the bottleneck. TCP's slow-start has been considered as a robust mechanism, however it might cause unnecessary delays in transmission. Phase 2 durations are calculated as the difference between the total download duration, T_{download} and Phase 1 duration, T_{P1}. The mean $\bar{T}_{\text{P1}}$ is assumed to be 4.6 s for all smartphones that receive the data via the 3G interface. The last three rows, $6-8$, depict the durations of the same downloads via WiFi tethering over MH. In the tethering scenario, the download durations are less as compared to the scenario while downloading via the 3G. This might be related to the hardware enhancements in Samsung Galaxy S4, *i.e.*, due to the fact that Samsung Galaxy S4 allows a higher bitrate (*e.g.*, via HSPA+) as compared to Samsung S (via HSDPA).

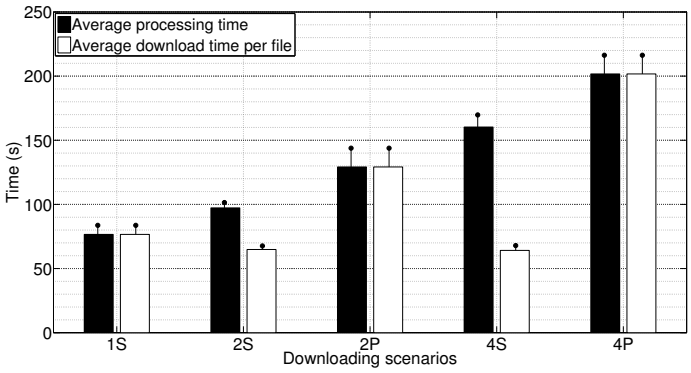
Energy and Processing Time

Based on the power and download duration measurements, the calculated energy consumption values are given in Fig. 7.10(a) together with the 95 % confidence levels. The energy consumption when two and four files are downloaded without scheduling are depicted with 2x1S and 4x1S, respectively. Energy consumption is reduced by 15 % and 22 % when two and four files are scheduled, respectively. The sum of the energy consumption of D and MH in the WiFi tethering is higher than in the 3G scenario, however the energy consumption of D decreases 48 % in the WiFi tethering scenario, as expected.

The average processing times and the download times per file, for serialized and parallel downloading on a single smartphone, are presented in Fig. 7.10(b) together with the upper 95 % confidence levels. There is no queuing in parallel downloading, but the average download time per file is higher as compared to the serial downloading. Thus, the average processing times are also 33 % and 26 % higher, when two and four files are scheduled, respectively.



(a) Energy consumption (different scenarios).



(b) Processing and download time (different scenarios).

Figure 7.10: Processing time and energy consumption [181].

7.3.4 Results Summary

This section presents the energy and download duration measurements for downloading multiple files (20 MB each) with different scheduling on a smartphone assuming realistic settings. Based on the measurements, when two and four files are scheduled (either in parallel or serial), in average 15 % and 22 % energy savings are achieved, respectively. Performing the downloads via WiFi tethering reduces the energy consumption, with the cost of waiting time for an available Mobile Hotspot (MH). The energy consumption of MH increases with the amount of download requests from the connected device, however MH can be connected to a power source. This way, it can provide Internet access to multiple smartphones, which are not connected to the power source. It has been known that WiFi tethering has been a great facility from the energy efficiency perspective, which needs to be managed and populated in a smart way. This also enables a reduction of the number of individual cellular connections to the RAN. This way, it is possible to decrease the per-device energy consumption and in parallel to increase the utilization of cellular data link per smartphone.

The average processing time is presented as it might influence the QoE. We presented that scheduling of two or four downloads (either via parallel or serial) consumes less processing time, less per-file downloading time, and less energy as compared to the scenario when the same files are downloaded individually. The average processing time is reduced by 33 % and 26 % in serial downloading as compared to the parallel downloading, when two and four files are scheduled, respectively. Per-file download time does not change in serial downloading, while it increases in the parallel downloading scenario.

There are a set of limitations in the study so far: the results obtained are specific to 20 MB file size, and we considered file scheduling based on only-parallel and only-serial downloading scenarios. Thus in the next section, we perform a similar study with various file sizes. We consider a scheduling consisting of four files, but this time also involving the use case when four files are downloaded in a hybrid approach (*i.e.*, combination of serial and parallel downloads).

7.4 Downloading Rather Small-size Files

In this section, we conduct file download experiments with smaller file sizes. We study a use case such that we have four identical files (assuming the users download same-size entertainment podcasts or educational material), *e.g.* video files, to be downloaded. In Section 7.3, it is observed that parallel downloading of large sized files (*e.g.*, 20 MB) yield higher average processing time as compared to serial downloading. One rea-

Table 7.11: Download scenarios and definitions.

Scenario	Definition	Scenario	Definition
4x1S	4 individual single downloads	2P2P	Two parallel downloads followed by two parallel downloads
4P	Four parallel downloads	4S	Four serial downloads

son might be due to the re-transmission, and reduced window size in TCP-based data transfer due to high congestion and limited bandwidth. However, for small size files, this might not be the case as the congestion window size in TCP data transfer may not saturate before the end of the file download. This might cause low utilization of the available network bandwidth, and in this case parallel downloading might yield a higher utilization with less average processing time. Thus, in this section, we repeat the measurements with different file sizes.

7.4.1 Measurement Method

The study consists of four downloading scenarios (individual, only-parallel, only-serial, hybrid) with nine different file sizes (10 KB, 50 KB, 250 KB, 500 KB, 1 MB, 2 MB, 5 MB and 10 MB). We enhanced the file download tool with new features including scalability for different combinations of scheduling. There are four scenarios: four files being downloaded: (i) independently, (ii) all in parallel; iii) all in serial; and (iv) two parallel download followed by a two parallel download. The scenarios and the corresponding abbreviations are summarized in Table 7.11. We used the "*mkfile -v XM <filename>*" command to create the files with the identical size, where *X* and filename are the desired file size in megabytes and filename. A 3G SIM card with a maximum downlink peak rate of 6 Mbit/s is used in all experiments. The download tool recorded the download duration of each downloaded file, basically as the difference between the time it is requested and the time it is finished as triggered by the Download Manager API. We used an external macro tool (that only runs on rooted devices) to run all of the scenarios in a sequential manner. This helps to perform all types of experiment scenarios at different times as scattered over a long duration, *e.g.*, in a day. All the experiments are conducted on a rooted Samsung Galaxy S. The automatized experiment procedure is given in Alg. 2. As previously, the timestamps are taken at the moment the file is requested (start timestamp) and the download is completely downloaded (end timestamp).

Algorithm 2: Experiment procedure for measuring the download durations.

```
For all file sizes {10 KB, 50 KB, 250 KB, 500 KB, 1 MB, 2 MB, 5 MB, 10 MB}
For all scenarios {4S, 4P, 2P2P}
while At least 100 iterations do
    Launch the download tool
    Start downloading [start timestamp]
    Wait until download finishes [end timestamp]
    Delete the downloaded files
    Close the application
    Wait for a cellular module demotion to IDLE
```

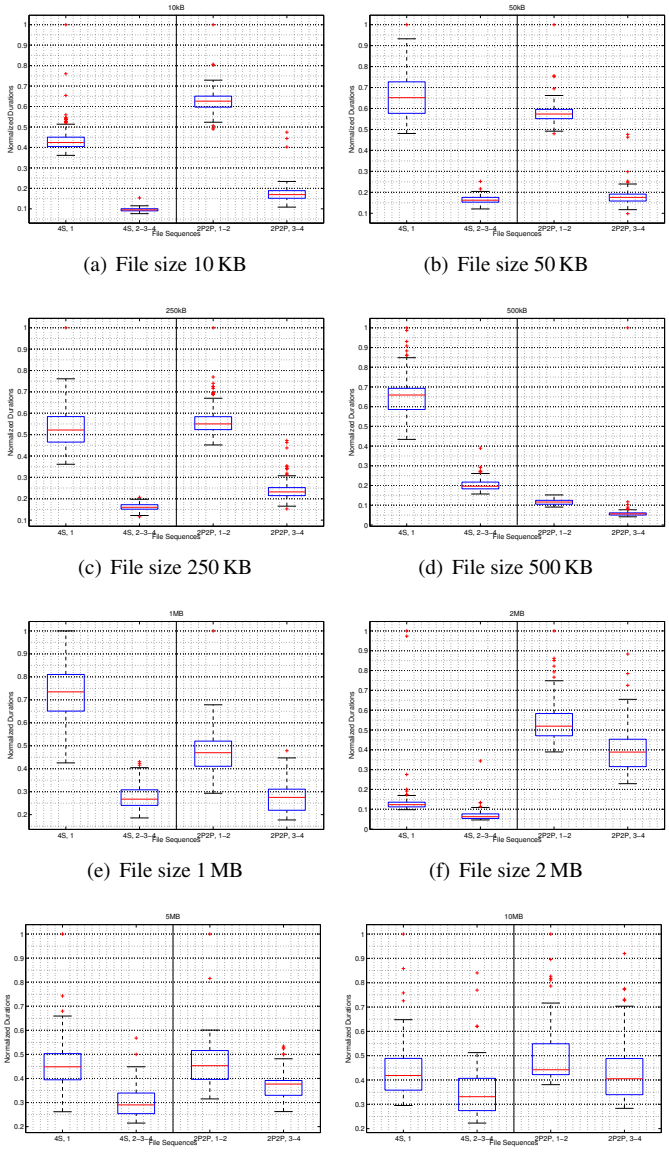
7.4.2 Observing the Per-file Download Time

We applied the one-way Anova test on the download duration of the files. We created two data sets. The first data set contains the download time of the files that are downloaded initially (*e.g.*, file 1 in the case of 4S scenario; file 1 and file 2 in the case of 2P2P scenario). The second data set contains the download duration for the files that are downloaded later (*e.g.*, files 2, 3 and 4 in the case of 4S scenario; files 3 and 4 in the case of 2P2P scenario). Our hypothesis is that the mean download durations of the two datasets are the same. The per-file download time for the two datasets for different file sizes are given in Fig. 7.11. We derived the statistical significance as given via the p-values in Fig. 7.12. The p-values grow with the file size; showing that the initial delay is becoming unimportant and being smoothed out as the actual file download increases. However in small size downloads, the initial duration is comparably significant to the actual file duration. Thus, to calculate the average processing time for smaller file sizes (*i.e.*, smaller than 20 MB), we have to consider a more general formula as the files first in sequence have statistically significantly different download durations than the remaining ones.

Acquisition of Average Processing Time: The average processing time is calculated as in Eq. 7.10. t_{begin} in Eq. 7.8 and t_{end} in Eq. 7.9 are the timestamp of the first and the last downloads, respectively. This considers the time when some files are downloaded, which the user can take action on them, while in parallel waiting for other downloads to be completed. Thus, this metric is important to consider within the QoE scope. Therefore, this is calculated as the minimum of all t_{begin}. t_{begin} and t_{end} are calculated for each file (assuming there are K files) for each run.

Acquisition of Total Download Time: The total download time is between when the user initiates the downloads and when the all file downloads are completed. This is calculated as the difference between the maximum t_{end} and minimum t_{begin}.

Acquisition of Energy Consumption: The energy consumption for 4P, 2P2P, and 4S scenarios are calculated as in Eq. 7.12, Eq. 7.13, and Eq. 7.14, respectively. The total energy consumption is the sum of energy consumption of Phase 1, Phase 2, and Phase tail. The energy consumption calculation for Phase 1 and Phase tail is straightforward, *i.e.*, by multiplying the mean phase duration with the corresponding mean power consumption. We assume the same Phase 1 and Phase tail power consumption and duration values as in Section 7.3. The Phase 2 energy consumption is calculated as the multiplication of Phase 2 duration and the Phase 2 power consumption. The calculations are slightly different for each scenario. For four parallel download scenario (4P), the Phase 2 duration is calculated as the subtraction of the earliest download begin time (minimum t_{begin}) and the Phase 1 duration ($\bar{T}_1$) from the latest download end timestamp (maximum t_{end}). The Phase 1 duration is subtracted as the download time measured by the tool includes the Phase 1 duration. The Phase 2 download duration for 2P2P scenario, *i.e.*, hybrid, is calculated as follows. Different from the previous scenario, the files are downloaded in two parts: the first 2P (*i.e.*, first two in parallel) and then the next 2P (*i.e.*, last two in parallel). As the files are downloaded in a sequence, the download duration of the files that are downloaded first (*e.g.*, first 2P), the Phase 1 download duration is subtracted from the total download duration which is the difference between the minimum begin time and the maximum end time of the files 1 and 2. In the second part (*i.e.*, the next 2P), there is no Phase 1 duration, thus we calculate only the difference between the minimum begin time and the maximum end time of the files 3 and 4. The Phase 2 download duration for 4S scenario is calculated as follows. As the Phase 1 duration influences the download duration of all four files, the Phase 1 duration, T_1 is subtracted from the download duration measured by the tool, *i.e.*, difference between the minimum start timestamp of all files and the maximum end timestamp of the all files.



(a) File size 10 KB (b) File size 50 KB

(c) File size 250 KB (d) File size 500 KB

(e) File size 1 MB (f) File size 2 MB

(g) File size 5 MB (h) File size 10 MB

Figure 7.11: Per-file download duration: the first in sequence (left) and the remaining files (right).

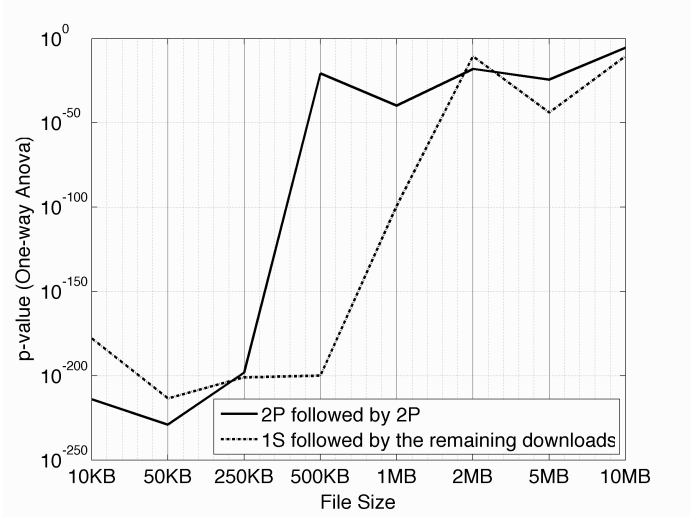


Figure 7.12: ANOVA *p*-values for different file sizes with two scenarios.

$$t_{\text{begin}} = \min\{t_{k,\text{begin}}\}\forall k. \tag{7.8}$$

$$t_{\text{end}} = \max\{t_{k,\text{end}}\}\forall k. \tag{7.9}$$

Average processing time:

$$\bar{T}_{\text{processing}} = \frac{1}{K} \cdot \sum_{k=1}^{K} \left(t_{k,\text{end}} - t_{\text{begin}} \right) \tag{7.10}$$

Total download time time:

$$\bar{T}_{\text{download}} = t_{\text{end}} - t_{\text{begin}} \tag{7.11}$$

Energy consumption:

$$E_{4\text{P}} = \underbrace{\bar{P}_1 \cdot \bar{T}_1}_{\text{Phase 1}} + \underbrace{\bar{P}_{4\text{P}} \cdot (t_{\text{end}} - t_{\text{begin}} - \bar{T}_1)}_{\text{Phase 2}} + \underbrace{\bar{P}_{\text{tl}} \cdot \bar{T}_{\text{tl}}}_{\text{Phase tail}} \tag{7.12}$$

$$E_{2\text{P2P}} = \underbrace{\bar{P}_1 \cdot \bar{T}_1}_{\text{Phase 1}} + \underbrace{\bar{P}_{2\text{P}} \cdot (\max\{t_{1,\text{end}}, t_{2,\text{end}}\} - \min\{t_{1,\text{begin}}, t_{2,\text{begin}}\} - \bar{T}_1) + \bar{P}_{2\text{P}} \cdot (\max\{t_{3,\text{end}}, t_{4,\text{end}}\} - \min\{t_{3,\text{begin}}, t_{4,\text{begin}}\})}_{\text{Phase 2}} + \underbrace{\bar{P}_{\text{tl}} \cdot \bar{T}_{\text{tl}}}_{\text{Phase tail}} \tag{7.13}$$

$$E_{4\text{S}} = \underbrace{\bar{P}_1 \cdot \bar{T}_1}_{\text{Phase 1}} + \underbrace{\bar{P}_{1\text{S}} \cdot (t_{\text{end}} - t_{\text{begin}} - \bar{T}_1)}_{\text{Phase2}} + \underbrace{\bar{P}_{\text{tl}} \cdot \bar{T}_{\text{tl}}}_{\text{Phase tail}} \tag{7.14}$$

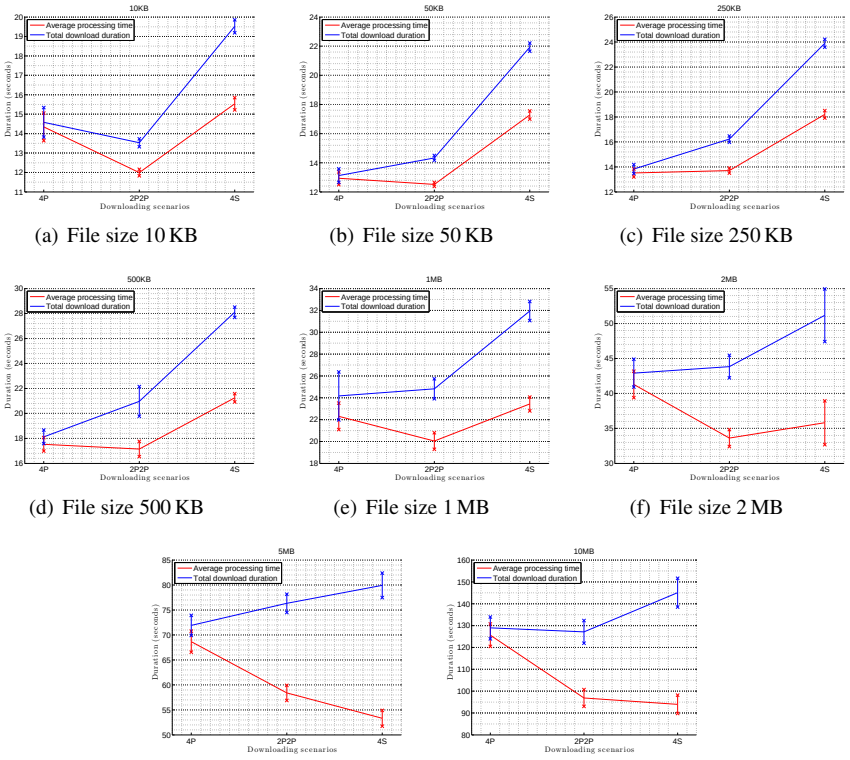
7.4.3 Results

Average Processing Time: The descriptive statistics (mean and 95 %confidence intervals, CI) regarding the average processing time for all three (4P, 2P2P, 4S) scenarios for nine file (downloded via 3G with 6 Mbit/s max. peak rate) sizes are given in Table 7.12. From left to right, file size, average processing time, total download time, and number of iterations are given. Fig. 7.13(a)- 7.13(h) visualize the average processing and the total download time for all file sizes. The average processing time increases drastically (statistically significantly larger than the other two scenarios) with the file size when the four files are downloaded in parallel. Thus, for large file sizes such as 10MB (or as also discussed for 20 MB in Section 7.3), downloading the files in parallel is not suggested, but it is rather recommended to download them either in series, 4S, or in 2P2P (two simultaneous downloads are followed by two simultaneous downloads) manner. For small file sizes, *e.g.*, 10 KB, the average processing time for serial downloading is statistically significantly larger than the other two scenarios, thus not recommended. In fact, when the files with small size are downloaded in a 2P2P manner, the average processing time outperforms the other two scenarios.

Total Download Time: When the files are downloaded in series, the total download duration is statistically significantly higher as compared to the other two scenarios. Thus, if a user prefers to download all files and complete the whole download process in the soonest manner, then we recommend user to initiate the downloading not in serial manner, but rather in parallel or in hybrid approach.

Table 7.12: Average processing and total downloading time for multiple file downloads.

Statistics	File Size	Average Processing Time [s]			Total Download Time [s]			Number of Iterations		
		4P	2P2P	4S	4P	2P2P	4S	4P	2P2P	4S
Mean	10 KB	14.35	12.00	15.54	14.58	13.53	19.52	160	160	161
CI (95 %)	10 KB	0.71	0.17	0.32	0.76	0.20	0.34			
Mean	50KB	12.94	12.51	17.28	13.12	14.33	21.93	175	175	176
CI (95 %)	50 KB	0.44	0.14	0.28	0.46	0.17	0.28			
Mean	250 KB	13.52	13.71	18.21	13.83	16.23	23.90	196	197	198
CI (95 %)	250 KB	0.32	0.21	0.30	0.36	0.25	0.32			
Mean	500 KB	17.52	17.15	21.25	18.13	20.97	28.10	184	184	186
CI (95 %)	500 KB	0.54	0.61	0.33	0.54	1.19	0.41			
Mean	1MB	22.30	20.05	23.44	24.18	24.83	31.94	111	107	111
CI (95 %)	1 MB	1.21	0.76	0.63	2.19	0.92	0.88			
Mean	2 MB	41.28	33.62	35.81	42.91	43.84	51.18	124	123	125
CI (95 %)	2 MB	1.89	1.21	3.13	1.98	1.61	3.76			
Mean	5 MB	68.65	58.40	53.32	71.93	76.33	79.94	145	148	146
CI (95 %)	5 MB	2.11	1.52	1.57	2.01	1.84	2.46			
Mean	10 MB	125.61	96.88	93.97	128.95	127.12	145.07	131	131	133
CI (95 %)	10 MB	5.13	3.85	4.18	5.01	5.17	6.60			



(a) File size 10 KB (b) File size 50 KB (c) File size 250 KB

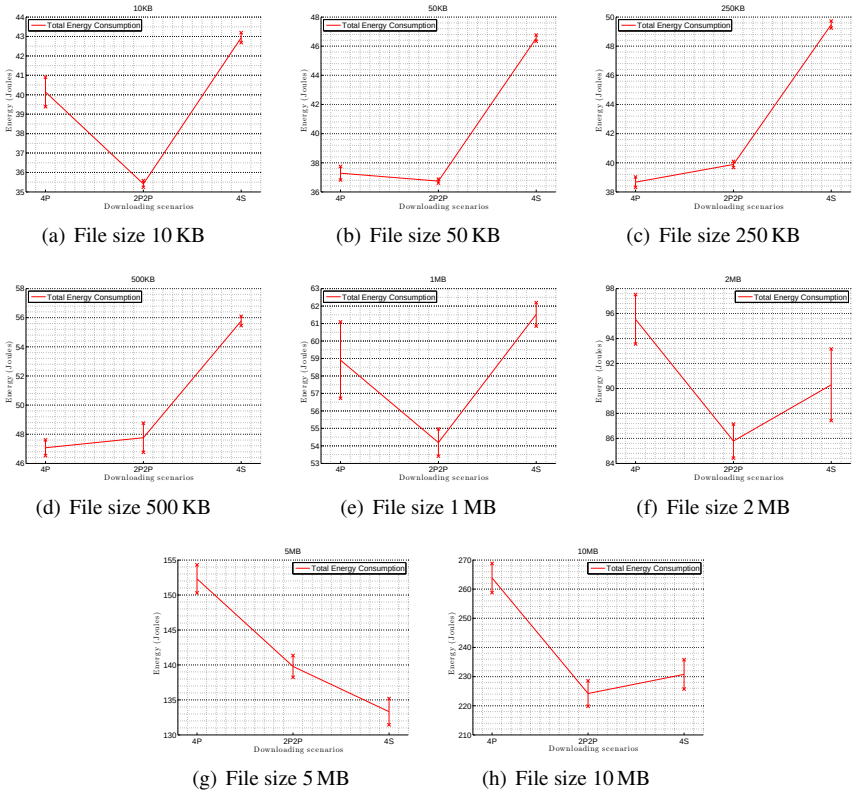
(d) File size 500 KB (e) File size 1 MB (f) File size 2 MB

(g) File size 5 MB (h) File size 10 MB

Figure 7.13: Average processing and total downloading time for three scenarios.

Energy Consumption: The total energy consumption for the three scenarios are given in Fig. 7.14. For small file sizes, the total energy consumption is statistically significantly higher in case of serial downloading as compared to the other two scenarios, *i.e.*, parallel or hybrid, thus serialized downloading of small-sized files is not recommended. As the file size increases, the total energy consumption difference between the parallel and the serial scenarios decreases. When the file size is increased to 2 MB, downloading in series becomes less energy consuming as compared to the parallel downloading. Thus, for file sizes larger than 2 MB, we recommend to conduct the download not in parallel manner in order to save energy.

Although the results obtained from these measurements are highly depending on the maximum available throughput (6 Mbit/s maximal peak rate), to date, the latter data rate is highly representative in the wild 3G connectivity and in realistic settings.



(a) File size 10 KB (b) File size 50 KB (c) File size 250 KB

(d) File size 500 KB (e) File size 1 MB (f) File size 2 MB

(g) File size 5 MB (h) File size 10 MB

Figure 7.14: Energy consumption for three sceanrios.

7.5 Energy-based Anomaly Detection

The abnormal behavior in the instantaneous power consumption values in case of the occasional freezes are investigated particularly during the network-based video streams. Although it is not very strong, there is a relationship in-between the duration of the freeze, *i.e.*, D_{p} and the power consumption, $\bar{P}_n$. The abnormal behavior of the video stream such as high inter-frame time manifests itself with long freezes, and is distinguishable with $\bar{P}_n$ for particular window size (W) values. The coefficient of determination (R^2) values of the exponential fits between the D_{p} and the $\bar{P}_n$ values are given for different W values in Table 7.13. Based on the R^2 values, the exponential relationship is the strongest both for RTSP and HTTP, when the actual data is smoothed with an SMA window size (W) of 7500 (1.5 s). When W is 15000 (3 s), the detection of

the abnormalities was not possible, as the smoothed data was departed from the original version too much, *i.e.*, abnormal fluctuations of P_n disappeared. As the D_p and the $\bar{P}_n$ datasets ($W =$

Table 7.13: R^2 values for different window size (W).

W	2500 (0.5 s)	5000 (1 s)	7500 (1.5 s)	10000 (2 s)
RTSP	0.0464	0.3062	0.5336	0.4709
HTTP	0.1174	0.2704	0.6016	0.3852

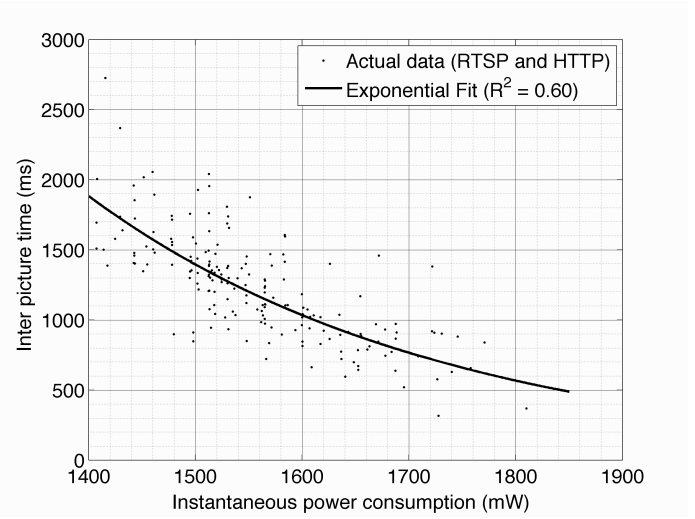


Figure 7.15: Abnormal D_p versus $\bar{P}_n$ values are illustrated ($W = 7500$, *i.e.*, 1.5 s) [206].

7500) from the RTSP and HTTP streaming are matched, an exponential relation in between the two metrics is suggested as presented in Fig. 7.15 according the Eq. 7.15. R^2 value was calculated as 0.60. Moreover, the statistics with respect to $\Delta T_{\mathrm{A(picture)}}$ and $\Delta T_{\mathrm{A(power)}}$ metrics are calculated as given in Table 7.14.

$$D_\mathrm{p}/\mathrm{ms} = 1.26 \cdot 10^5 \cdot \mathrm{e}^{-0.003 \cdot \bar{P}_n/\mathrm{mW}} \tag{7.15}$$

High delays in the network-level manifest themselves at the application layer as freezes in the video display screen. Eventually, the detected anomalies in the power consumption might be

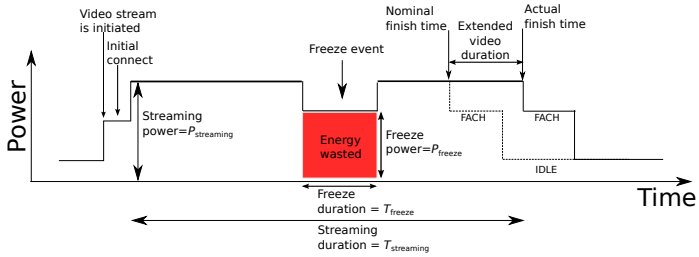
Table 7.14: Time gap between the anomalies with $W = 7500$.

	ΔT_A [s]				
	Min	Max	Mean	Median	Standard Deviation
Frame	23.47	26.10	25.02	25.04	0.52
Power	20.35	29.20	25.04	25.07	1.06

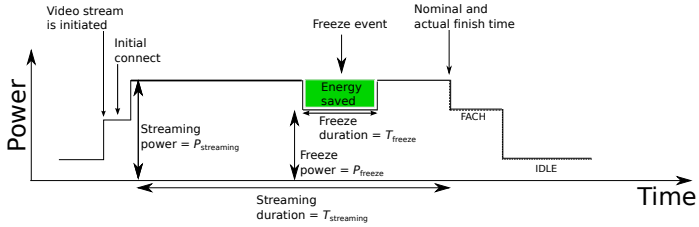
transformed into benefit and might be considered as an alternative approach for saving energy. In Section 7.6, we study the freeze regions in details.

7.6 Energy and QoE in Video Streaming: A Trade-off or a Win-Win?

The total power consumption of the smartphone comprises the power consumption of the individual components including CPU, LCD screen, and cellular data module. It is preferred that the energy consumption pattern of all components during the runtime of an application should adapt to save energy depending on the changing conditions of the system. During video streaming, there are mechanisms that adapt the power states of the cellular data modules based on the variations in the network throughput. Thus, the power consumption decreases when there is a stalling event, due to the state demotion in cellular data modules (*e.g.*, from active to idle). However, other components such as CPU and the screen keep their high power-consuming states unchanged. Therefore, during this time, *i.e.*, during user's waiting time for the video to continue playing, the consumed energy increases with time. In this scenario, the total energy consumed by the device is highly depending on the user's decision (*e.g.*, interrupting the play-out and giving up watching it, or keeping patience and wait) as well as the streaming protocol. The network protocol's role is also crucial when the video freezes. If the video is UDP-based, then there is a picture jump (*with-jump*), and the total video session duration does not depend on the freezes, thus the latter might be saving energy, at the cost of displaying less content. Accordingly in UDP-based video streaming, energy saving might be performed with picture jumps while also maintaining the QoE. If the video is TCP-based, then there are no picture jumps (*without-jump*), and the total video session duration increases given that the user watches the video till the end. The power consumption models for *without-jump* and *with-jump* scenarios are given in Fig. 7.16(a) and Fig.7.16(b), respectively. In a *without-jump* scenario in Fig. 7.16(a), the total video session duration increases with the total freeze duration. The actual finish time of a video session is a freeze duration more than the nominal finish time. The energy waste in a with-jump scenario is illustrated with the red region in Fig. 7.16(a). In a *with-jump* scenario in Fig. 7.16(b), the total video session is equal to the one when there is no freeze. In addition, during the freezes there is an energy saving proportional to the power saving and the total duration of a freeze. The energy



(a) Without-jump scenario



(b) With-jump scenario

Figure 7.16: Freezes are illustrated with the power consumption pattern during video streaming in two scenarios.

saving during the freezes is illustrated with a green rectangle.

We further discuss on the scenarios related to freezes as follows.

7.6.1 Problem Formulation

Without freeze: The total energy consumption, E_{total}, of a video stream without any freeze is calculated in Eq. 7.16 as the product of mean power consumption $P_{\text{streaming}}$, and the mean total duration $T_{\text{streaming}}$ of video streaming. In this scenario the streaming duration and the video session duration are equal.

$$E_{\text{total}} = \bar{P}_{\text{streaming}} \cdot \bar{T}_{\text{streaming}} \qquad \text{[W][s]} \tag{7.16}$$

With freeze, without-jump: The total energy consumption, E_{total}, for the scenario when there are freezes and the video pictures are not jumped (*i.e.*, not skipped), is calculated as in Eq. 7.17. E_{total} is the sum of energy during video streaming and also the additional energy consumption, E_{freeze}, (*i.e.*, wasted energy) during the freeze duration. Thus, the energy waste is

proportional to the freeze duration and the total power consumption of the freeze event.

$$E_{\text{total}} = \bar{P}_{\text{streaming}} \cdot \bar{T}_{\text{streaming}} + \underbrace{\bar{P}_{\text{freeze}} \cdot \bar{T}_{\text{freeze}}}_{\text{wasted energy}} \qquad [\text{W}][\text{s}] \tag{7.17}$$

When there is no picture jump, *e.g.*, in a TCP-based stream, the energy waste (with respect to a no freeze scenario) can be represented as follows. The total freeze duration, T_{freeze} during video session is formulated in Eq. 7.18, where N is the total number of freezes, and τ_n is the freeze duration of freeze n. $\bar{P}_{\text{freeze}}$ in Eq., 7.19 is the smartphone power consumption during a video freeze (*e.g.*, due to RRC FACH state or active CPU). The total energy waste is given in Eq. 7.20 as the multiplication of power and duration during freeze.

$$T_{\text{freeze}} = \bar{T}_{\text{waste}}\,[\text{s}] = \sum_{n=1}^{N} \tau_n \qquad [\text{s}] \tag{7.18}$$

$$\bar{P}_{\text{freeze}} = \bar{P}_{\text{waste}}\,[\text{W}] = \bar{P}_{\text{FACH}} + \bar{P}_{\text{CPU}} + \bar{P}_{\text{Display}} + \bar{P}_{\text{Other active components}} \qquad [\text{W}] \tag{7.19}$$

$$E_{\text{freeze}} = E_{\text{waste}}\,[\text{J}] = \bar{P}_{\text{freeze}} \cdot \bar{T}_{\text{freeze}} \qquad [\text{W}][\text{s}] \tag{7.20}$$

Assuming that the freezes are of the same length, the energy waste in a without-jump scenario can be further formulated as in Eq. 7.21.

$$E_{\text{wasted}} = \bar{P}_{\text{waste}} \cdot N \cdot \tau_n \qquad [\text{W}][\text{s}] \tag{7.21}$$

With freeze, with-jump: The total energy consumption for the scenario when there is a freeze and the pictures are jumped for a total freeze duration, is calculated as in Eq.7.22. In this scenario, the total video session duration is reduced by the total freeze duration. Thus, the total energy consumption during video streaming is reduced by $\bar{T}_{\text{freeze}}(\bar{P}_{\text{streaming}} - \bar{P}_{\text{freeze}})$.

$$E_{\text{total}} = \bar{P}_{\text{streaming}} \cdot (\bar{T}_{\text{streaming}} - \bar{T}_{\text{freeze}}) + \bar{P}_{\text{freeze}} \cdot \bar{T}_{\text{freeze}} \qquad [\text{W}][\text{s}] \tag{7.22}$$

$$E_{\text{total}} = \bar{P}_{\text{streaming}} \cdot \bar{T}_{\text{streaming}} - \underbrace{\bar{T}_{\text{freeze}}(\bar{P}_{\text{streaming}} - \bar{P}_{\text{freeze}})}_{\text{energy saving}} \qquad [\text{W}][\text{s}] \tag{7.23}$$

$$\text{where, } \bar{P}_{\text{streaming}} - \bar{P}_{\text{freeze}} > 0 \tag{7.24}$$

Thus, the energy saving is depending on the absolute difference between $\bar{P}_{\text{streaming}}$ and $\bar{P}_{\text{freeze}}$. By looking at Eq. 7.17 and Eq.7.23, it can be said that the less the total power consumption during a freeze, $\bar{P}_{\text{freeze}}$, the lower the energy waste in without-jump scenario, and the higher the energy saving in with-jump scenario. Assuming that the freezes are with the same length, the energy saving in a with-jump scenario can be further formulated as in Eq. 7.25.

$$E_{\text{saving}} = (\bar{P}_{\text{streaming}} - \bar{P}_{\text{freeze}}) \cdot N \cdot \tau_n \qquad [\text{W}][\text{s}] \tag{7.25}$$

7.6.2 Discussion on the Trade-Off and Win-Win

In UDP-based video stream, the QoE is influenced by the length and the number of freezes in a video stream. However, in contrast with the TCP-based stream, the video duration does not change as the stalled regions in the video content are skipped. The more pictures are skipped the less the energy consumption (*i.e.*, the higher the energy saving), but the less the QoE. Thus, the relationship between the QoE and the energy consumption becomes a trade-off.

The energy waste is proportional to the power waste during freezes, and 3G-based streaming is not energy efficient for a video stream with multiple freezes due to the RRC FACH state. In TCP-based video stream, considering an energy saving during a stalling event is far optimistic, and the video indeed consumes more energy given that the video session time is extended. Consequently, in TCP-based stream, the more freezes mean higher energy waste, and worse QoE due to the stalling events. The MOS decreases with the energy waste, E_{waste}, which is proportional to the length and the number of freezes. Therefore, the best scenario with respect to QoE and energy is that the video streaming is smooth, *i.e.*, without any freeze. This way, both the session time is kept minimum and indirectly the total energy consumption of the smartphone during video streaming, which can be considered as a win-win.

Relating the Scenarios to Previous Research on QoE Hossfeld in [115] studied the impact of the freezes (the stalling events) on video QoE. The streaming scenarios involve periodic freezes that are uniformly distributed and analyzed in every 30 seconds long intervals. It was concluded that one freeze with a maximum of 3 seconds in a video clip is the upper boundary that enables QoE at acceptable level, *i.e.*, MOS$\geq$ 3. Eq.7.26 is proposed to represent MOS with the length and frequency of the freezes, we obtained the MOS values from [116].

$$MOS(L,N) = 3.50 \cdot e^{-(0.15\cdot\tau+0.19)\cdot N} + 1.50 \tag{7.26}$$

We obtain the energy waste (via Eq. 7.21), the energy saving (via Eq. 7.25), and the MOS values

Table 7.15: Power consumption of smartphone during local streaming

Mean [mW]	Median [mW]	Std. [mW]
1558	1543	83.52

(via Eq. 7.26) for N and τ_n values each ranging from 1 to 6 (based on Hossfeld *et al.* [116]), and P_{freeze} values for approximately 1500 mW, and $P_{\text{streaming}}$ values for 1750 mW (based on our previous measurements in Section 7.2.3). Indeed, we had calculated median power consumption during a freeze as 1543 mW in Table 7.15, but we will use 1500 mW in our further calculations for the sake of simplicity. When the obtained MOS values and the energy values are matched, we obtain Fig. 7.17. The best fit was a power-law model with $R^2 = 0.92$. The energy waste (shown via blue colour) is much higher in without-jump scenario as compared to the energy saving in

the with-jump case (shown via red colour). This is expected as the power saving is assumed to be one-seventh $\left(\frac{1750\,\text{mW}-1500\,\text{mW}}{1750\,\text{mW}}\right)$ of the total streaming power consumption. In the without-jump scenario (shown via blue colour), the energy waste decreases when the MOS increases in a power-law model, which is a win-win situation. In the with-jump scenario (shown with red colour), the energy saving increases when the MOS decreases in a power-law model; which is a trade-off situation. The calculated numerical values for MOS vs. energy savings for with-jump

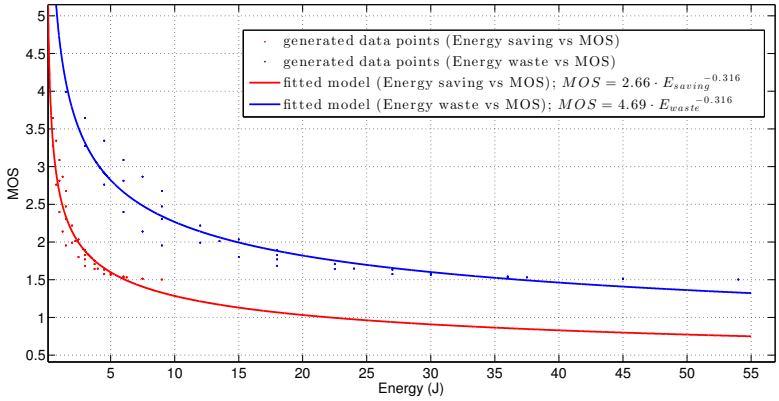


Figure 7.17: MOS vs. energy saving and MOS vs. energy waste for the with-jump and without-jump scenarios, respectively.

scenario and the MOS vs. energy waste for without-jump scenario are given in Table 7.16, where τ_n is the length of a freeze and N is the number of freezes (assuming that freezes have the same duration). When MOS of 3 is substituted into Eq.7.27, the maximum energy saving is achieved as 0.96 J (roughly 1 J) for a 30 s of video stream. In contrast, if there is no jump in the case of freezes, the energy waste is 1.15 J if *MOS* of 3 is substituted to Eq. 7.28, and the energy waste increases as the MOS further decreases. In this case, it is necessary to avoid freezes to save energy.

$$MOS = 2.66 \cdot E_{\text{saving}}^{-0.316} \tag{7.27}$$

$$MOS = 4.69 \cdot E_{\text{waste}}^{-0.316} \tag{7.28}$$

The model presented is based on the relation between the MOS and the freezes as presented in previous work [115]. As already mentioned, the model assumes that the freeze lengths are the same; indeed in real life this is not the case. We have observed in our previous work in Section 5 that the inter-picture time during a 3G-based real-time video stream holds a two state (ON/OFF)

Table 7.16: E_{saving} and MOS for various N, τ, and P_{saving} values.

					with-jump	without-jump
MOS	N	τ	P_{freeze}	$E_{\text{streaming}}$	E_{saving}	E_{waste}
3.99	1	1	1500	1750	0.25	1.5
3.64	1	2	1500	1750	0.5	3
3.34	1	3	1500	1750	0.75	4.5
3.08	1	4	1500	1750	1	6
2.86	1	5	1500	1750	1.25	7.5
2.67	1	6	1500	1750	1.5	9
3.27	2	1	1500	1750	0.5	3
2.81	2	2	1500	1750	1	6
2.47	2	3	1500	1750	1.5	9
2.22	2	4	1500	1750	2	12
2.03	2	5	1500	1750	2.5	15
1.89	2	6	1500	1750	3	18
2.76	3	1	1500	1750	0.75	4.5
2.30	3	2	1500	1750	1.5	9
2.01	3	3	1500	1750	2.25	13.5
1.82	3	4	1500	1750	3	18
1.70	3	5	1500	1750	3.75	22.5
1.63	3	6	1500	1750	4.5	27
2.39	4	1	1500	1750	1	6
1.99	4	2	1500	1750	2	12
1.77	4	3	1500	1750	3	18
1.64	4	4	1500	1750	4	24
1.58	4	5	1500	1750	5	30
1.54	4	6	1500	1750	6	36
2.13	5	1	1500	1750	1.25	7.5
1.80	5	2	1500	1750	2.5	15
1.64	5	3	1500	1750	3.75	22.5
1.56	5	4	1500	1750	5	30
1.53	5	5	1500	1750	6.25	37.5
1.51	5	6	1500	1750	7.5	45
1.95	6	1	1500	1750	1.5	9
1.68	6	2	1500	1750	3	18
1.57	6	3	1500	1750	4.5	27
1.53	6	4	1500	1750	6	36
1.51	6	5	1500	1750	7.5	45
1.50	6	6	1500	1750	9	54

exponential distribution. Thus, in the next session, we repeat our user study with similar settings in [115], and on the smartphone. Our aim is to improve the QoE model to represent a more realistic scenario, and to relate it to energy consumption.

7.7 QoE vs. Energy Case Study on Smartphone

The above scenario assumes that the temporal impairments are uniform, however this is not realistic. Thus, in the next study we present temporal impairments by considering realistic temporal impairment models of 3G (via *with* and *without* picture jump scenarios), which we obtained in our previous research as presented in Chapter 3. We consider an exponential ON/OFF modeling on the inter-picture times, with particular mean ON state and mean OFF state durations, and collect both user perceived QoE, *i.e.*, MOS throughout a user study, and the power measurements. In order to fully control the experiment testbed, we stream from the smartphone's local storage,

i.e., sdcard, and implement the video player to mimic the exponential ON/OFF behaviour of 3G. Eventually our goal is to obtain a relationship between the energy consumption and MOS based on the freezes in a video stream and compare the obtained model with the model in Fig.7.17.

7.7.1 User Study Method

There are three scenarios in this experiment. Scenario 1 is a smooth video playout without any temporal impairment, *i.e.*, freezes. Scenario 2 and scenario 3 include freezes; where in the second scenario the freezes occur without any picture jumps, and in the third scenario the freezes occur with picture jumps. In scenario 2, when a freeze event occurs, the video is paused and continued from the exact point where it has paused, thus all the video content is eventually presented to the user. In scenario 3, upon a freeze event, the video is paused for a particular duration, the video is skipped with exactly the OFF duration, and then is resumed from the skipped point of the video. This causes less content to be presented to the user. Observe that the IOVidEoQ tool in these experiments does not capture the inter-picture time but only the ON/OFF events. In other words, during an ON event, the video stream follows a smooth playout via the local storage.

Choice of ON/OFF state parameters

We applied various ON/OFF durations on the video streams during a preliminary pilot study. We would like the obtained MOS values to span over the 5-level MOS scale. The settings applied to the users during the pilot study are: Mean ON time = 7 seconds and Mean OFF time = 3 seconds for 3 random pilot users; Mean ON time = 8 seconds and Mean OFF time = 2 seconds for 4 random pilot users; and Mean ON time = 9 seconds and Mean OFF time = 1 second for 4 random pilot users. The reason for continuing the study with the "Mean ON = 8 seconds, Mean OFF = 2 seconds" pair is as follows. When 30% distortion with "Mean ON time = 7 seconds and Mean OFF time = 3 seconds" was set, the obtained MOS values did not contain any MOS = 1 and the average MOS was 3.56. When 10% distortion was applied (with "Mean ON time = 9 seconds, and Mean OFF time = 1 second", the obtained MOS values ranged between 2 and 5, and the qualitative feedback received from the users indicated annoyance related with too many interruptions. With the "Mean ON = 8 seconds and Mean OFF = 2 seconds", based on the four pilot users, the received MOS values spanned all of the MOS range, *i.e.*, 1 to 5, which helped to obtain a representative model. Thus, we further employed 19 more random participants, and applied the "Mean ON= 8 seconds and Mean OFF = 2 seconds" setting on them. Eventually, in total 23 participants watched the videos with the latter setting combination.

Choice of video content

We clipped a three minutes long action scene from a popular science fiction movie from year 2014 (from Blu-ray disc). We chose a scene that includes high movement level, and rich in temporal information in order to address a worst-case scenario for freezes. In other words, we assumed that the higher the temporal complexity of the video, the higher the probability that the

freezes will influence the QoE. The video is prepared with 6 Mbit/s with 25 frames per second, and converted into MP4 multimedia format.

QoE Experiments

We conducted a user study involving a user survey, face-to-face interview, and the user experiments with the video streaming tool, *i.e.*, IOVidEoQ. We recruited 38 users from four different countries (Sweden, Switzerland, and France), and obtained survey and interview data from all of them. We asked questions to the users regarding their daily life usage of smartphones and also collected user background information, *e.g.*, demographics. We collected information regarding the user details (age, gender, user's own phone type) and experience on video streaming on smartphones. We asked the user the following questions based on users' video streaming experience on the smartphone in daily life:

- How often do you watch a video on the smartphone per day? ($>$ 1 hour/day, $<$ 1 hour/day, $<$ 30 min/day, $<$ 10 min/day, $<$ 5 min/day, $<$ 2 min/day, $<$ 1 min/day, never)
- What percentage of your video sessions is network-based, *i.e.*, you stream the content over the Internet?
- While streaming over a network (if you do so), do you prefer WiFi or cellular data (*e.g.*, 3G/4G) while streaming over the network? Why?
- I normally watch a video on my smartphone... (at home, at office/school, at street, at other indoor, at other outdoor, while alone, while with a person, while with a group, while sitting, while standing, while walking).
- What is your overall perceived video quality based in MOS scale (5-excellent, 4-good, 3-fair, 2-poor, 1-bad) on network-based on video streaming on smartphones? Please answer for both WiFi and 3G.

After the user answered the questions, we explained the detailed instructions regarding the next steps in the study. Then, the user watched a video stream via our video streaming tool on two phone types: either Samsung Galaxy S4 or Nexus 5. Both phones have exactly the same display, *i.e.*, Super AMOLED, 16M colors, 1080 x 1920 pixels. The complete QoE experiment procedure is depicted in Fig. 7.18. During the video study with the IOVidEoQ tool, first, the test supervisor explained the user what the subject will do. The test supervisor enters the userID, age, gender of the user together with the mean ON and mean OFF durations for temporal impairments. Next, one of the three randomly selected video scenarios begins to display on the smartphone screen. After the end of the video session (*i.e.* three minutes), the MOS scale menu pops up, and the user is required to give a rating on the five-level ACR scale (where 5 is excellent and 1 is bad). Then, 15 seconds of gray screen is presented to the user. The user is asked to watch this gray screen, as it is recommended by ITU-T P.910 [156] standard and helps to prevent the memory effect, *i.e.*, to flush the memory from bias. Next, the second randomly chosen scenario starts playing. And this loop continues until the three scenarios with the same video content (no freeze, freeze without jump, freeze with jump) are watched and rated by the user.

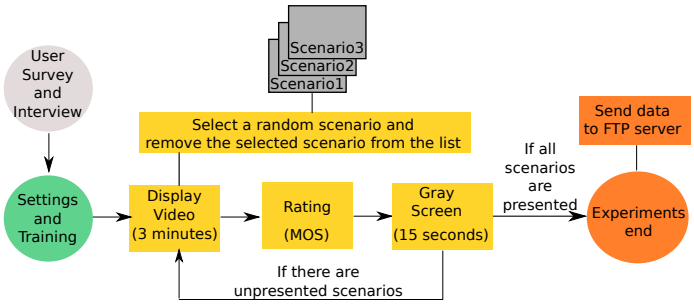


Figure 7.18: User Study Procedure

7.7.2 User Study Results

Before presenting the outcomes of the user study with respect to the data collected via our video streaming tool, we present a set of overall statistics (based on the interview and the short survey) on the participants' background and context while the video applications are being streamed in daily life. We also collect users's opinion on the overall quality perceived on different wireless access technologies in daily life with the corresponding reasoning.

Survey and Interview Results

The users' background is depicted in Table 7.18. The presented variables for each user are, from-left-to-right, the subjectID, age, gender, users' own phone type, occupation, country of residence, average duration of video stream in daily life, the MOS based on QoE on WiFi, and the MOS based on cellular data, *i.e.*, 3G or 4G. In total, there were 38 participants who attended to the study. The age range was between 22 and 51, with an average of 29. There were 24 males and 14 females in the study. The user profile comprises various occupations including researchers, students, administrative staff, and an antropologist. They were residents of three different countries: Sweden, Switzerland, and France. The participants use diverse phone types including Nexus 5 (11), Iphone 5 or 6 (10), Samsung (8), Moto-G (3), LG (2), Xperia (1), Xiomi (1), HTC (1), and Huawei (1).

3G vs WiFi: The mean MOS for WiFi and 3G are 4.3 and 3.5, respectively, and the figure that compares the MOS ratings with respect to two different wireless interfaces is given in Fig. 7.19. Fisher's Exact Test [70] can be used to understand whether the two datasets with nominal data are statistically significantly different from each other. We applied Fisher's Exact Test to compare the two data sets (*i.e.*, WiFi and cellular). We conclude that the two dataset means are statistically significantly different than each other with p value equals to 0.0034 (95 % confidence interval). It has concluded that WiFi is prefered more than the 3G for video streaming.

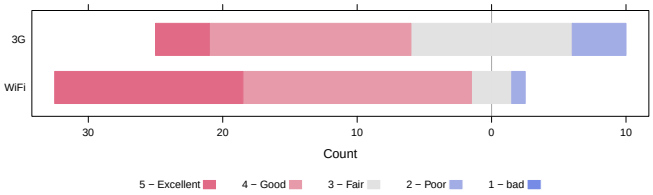


Figure 7.19: Overall comparison of MOS ratings between WiFi and 3G/4G cellular.

Qualitative Feedback on Wireless Interfaces: We classified the qualitative feedback received from the users regarding the reason behind perceiving a quality particular wireless network interface as better than the other. We classified the user feedback in six categories with corresponding keywords as in Table 7.17. The first column is the classified keyword, where the second column contains all the corresponding matched and classified keywords within user's qualitative data. The number of occurrences of each keyword is calculated and given in paran-

Table 7.17: Keywords from the qualitative feedback for the choice of network interface.

CLASSIFIED KEYWORD	MATCHING KEYWORDS
speed (17)	faster, speed, poor, delay, interruptions, quality, bad, load faster, consistent, capacity
cost (14)	costs, free, money, expensive, save, cheap
connectivity (3)	always connected, use in movement, disconnects
data plan (2)	no data plan, no cellular operator, wifi default
reliability (2)	more reliable, reliable indoor
usability (1)	difficult to change WiFi

thesis in the first row near the reasons in Table 7.17. In overall, the main motivation for choosing a wireless interface over another is related to the speed (for 17 users) and the cost of connectivity (for 14 users).

Context and Location: The users participated in the study watch videos in daily life in various contexts and location. We observed that 76 % of the participants watch video at 'home' when 'alone', and in 'sitting position'.

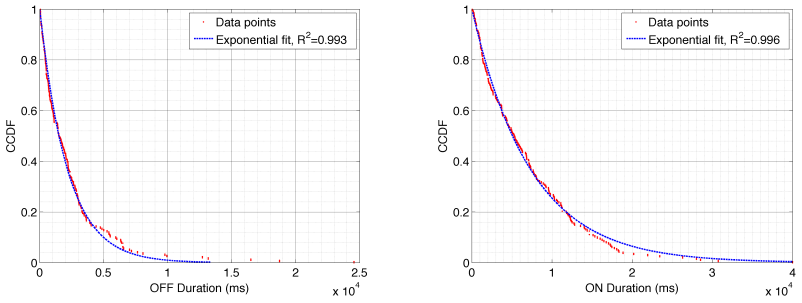
Smartphone-based Video Usage: The participants' usage of smartphone-based video per day is studied. Amongst the 38 users, 82 % of the users watch video on the smartphone for less than half-an-hour per day.

Table 7.18: User background and user's experience with smartphone-based video streaming.

SubjectID	Age	Gender	Phone Type	Occupation	Residence	Vid.Usage	MOS [WiFi]	MOS [3G]
S1	32	Male	Huawei	Student	Sweden	more than hour a day	5	3
S2	27	Male	Samsung S4	Student	Sweden	less than hour per day	5	4
S3	31	Female	iPhone5	Researcher	Sweden	more than hour a day	4	2
S4	36	Female	iPhone5	Researcher	Sweden	less than 10 mins per day	4	3
S5	28	Male	Nexus5	Researcher	Switzerland	less than 2 mins per day	4	3
S6	27	Male	Nexus5	Researcher	Switzerland	less than 5 mins per day	4	3
S7	26	Male	Nexus5	Researcher	Switzerland	less than 10 mins per day	4	3
S8	37	Male	Samsung S4 m.	Researcher	France	less than 5 mins per day	3	3
S9	23	Female	iPhone5	Researcher	France	less than 30 mins per day	4	2
S10	27	Male	Samsung S4 m.	Researcher	Sweden	less than 30 mins per day	4	4
S11	31	Female	Xperia Z2	Administrator	Sweden	lless than 1 hour per day	4	3
S12	28	Female	Nexus5	Student	Sweden	less than 5 mins per day	5	4
S13	27	Male	Nexus5	Student	Sweden	less than 1 mins per day	4	3
S14	51	Male	iPhone5	Student	Sweden	less than 5 mins per day	5	4
S15	33	Female	Samsung S4	Student	Sweden	less than 5 mins per day	5	5
S16	33	Male	Samsung S3	Student	Sweden	less than 1 min per day	4	3
S17	23	Male	iPhone5s	Student	Sweden	less than 10 mins per day	5	4
S18	22	Male	iPhone5	Student	Sweden	less than 5 mins per day	5	4
S19	28	Female	iPhone5	Student	Sweden	less than 10 mins per day	5	3
S20	26	Female	Samsung S3	Student	Sweden	more than an hour per day	4	3
S21	26	Male	iPhone5	Student	Sweden	less than 2 mins per day	5	4
S22	23	Male	iPhone6	Student	Sweden	less than 10 mins per day	4	5
S23	23	Male	Moto G	Student	Sweden	less than 5 mins per day	5	4
S24	25	Female	Nexus5	Student	Sweden	less than 10 mins per day	4	4
S25	33	Male	Nexus5	Researcher	Sweden	less than 5 mins per day	5	4
S26	34	Female	iPhone6	Antropologist	Sweden	-	-	-
S27	27	Male	Xiomi	Researcher	Sweden	less than 10 mins per day	3	2
S28	26	Male	MotoG	Researcher	Sweden	less than 10 mins per day	4	4
S29	27	Male	MotoG	Researcher	Sweden	less than 10 mins per day	4	3
S30	38	Male	LG G3	Researcher	Sweden	less than 30 mins per day	4	4
S31	28	Male	Nexus5	Researcher	Switzerland	less than 10 mins per day	4	3
S32	24	Male	Nexus5	Researcher	Switzerland	less than 5 mins per day	4	3
S33	26	Nevermind	Nexus5	Researcher	Switzerland	less than 2 mins per day	4	3
S34	34	Male	Samsung Note	Researcher	Sweden	less than 1 mins per day	4	4
S35	29	Female	Samsung S4 m.	Researcher	France	less than 30 mins per day	2	5
S35	29	Female	Samsung S4 m.	Researcher	France	less than 30 mins per day	2	5
S36	30	Male	GalaxyS3	Student	Sweden	less than 2 mins per day	4	3
S37	27	Female	LG G5	Researcher	Sweden	less than 5 mins per day	3	4
S38	38	Female	HTC	Researcher	Sweden	-	-	-

7.7.3 Validation of the Tool

We have selected 10 random datasets (each more than 200 data points) from user experiments in order to confirm that the applied ON/OFF distortions during the video playout are exponentially distributed. The ON and OFF durations in two datasets are fitted into exponential curves giving both $R^2 = 0.99$, as given in Fig. 7.20(a)-7.20(b). Thus, we confirm that the tool applies distributions within time durations that are exponentially distributed as expected.



(a) OFF distribution (b) ON distribution

Figure 7.20: CCDF's with respect to random test runs.

7.7.4 User's Video Experiment Results

MOS per each scenario

We compare the MOS values received with respect to the three scenarios: (Scenario 1) without any freeze; (Scenario 2) freeze without picture jump, and (Scenario 3) freeze with picture jump. The summary statistics of the ON and OFF durations that the 30 users experienced during Scenario 2 and Scenario 3 are given in Table 7.19. From columns left to right, the video state (*i.e.*, ON or OFF), mean duration, median duration, standard deviation of the duration, minimum duration, maximum duration, and the number of samples in each dataset is given. The differences

Table 7.19: Average ON and OFF durations that the users experienced during the experiments in Scenario 2 and Scenario 3.

State	Mean [s]	Median [s]	Std. [s]	Min. [s]	Max. [s]	N
ON	8.0	7.7	1.8	5.1	12.8	30
OFF	2.0	1.9	7.0	0.4	3.6	30

in between the MOS scores with respect to the three scenarios are given in Fig. 7.21. We applied pair-wise Fisher's Exact Test to the dataset with two columns (freeze with jump, freeze without jump) in order to understand whether they are statistically significantly different. We obtained a p value of 0.8784 (95 % confidence interval). We conclude that the two datasets are not statistically significantly different from each other. This outcome motivated us to conclude that there is no statistically significant difference in QoE if the video is interrupted with or without

skipping video content. The most important factor that influences the QoE is the occurrence of an interruption, *i.e.*, a freeze.

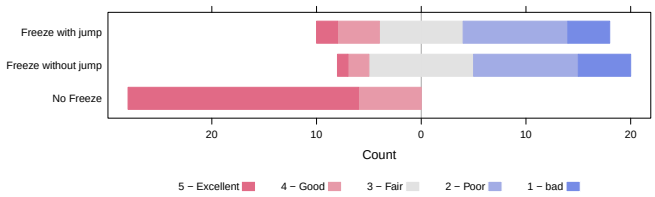
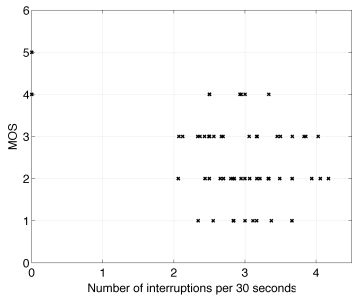


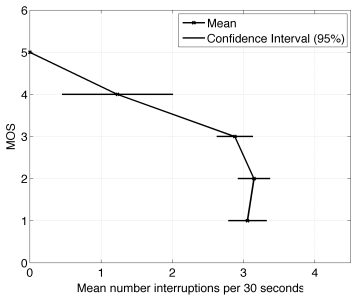
Figure 7.21: Comparing the Three Scenarios from the QoE perspective.

Number of Interruptions

We studied the average number of interruptions per 30 seconds during the video streaming session. The choice of 30 seconds interval is to match the previous work conducted in [116] as mentioned earlier in this Chapter. We matched the number of interruptions to the received MOS values per session, however we have not identified any strong trend as given in Fig. 7.22(a). The mean number of interruptions per MOS is calculated and depicted in Fig. 7.22(b) showing a more clear trend.



(a) Number of interrupions per 30 seconds



(b) Mean number of interrupions per 30 seconds per MOS

Figure 7.22: A set of metrics are given with the corresponding MOS scores.

Average ON Duration Per Session

We studied other metrics related to the ON and OFF states during a video session. In the two state ON/OFF exponential model, the average ON duration for each video session is matched to the corresponding MOS values. The relationship in between the MOS and the average ON duration can be represented via linear or exponential models as given in Fig. 7.23(a). The mean of the average ON duration is also calculated per MOS values together with the confidence intervals (95 %) as depicted in Fig. 7.23(c). The obtained models are presented in the second column of Table 7.20. The exponential and the linear models are given in rows $2-3$, respectively. Indeed, neither of the fits are strong, where the exponential and the linear fits yield $R^2 = 0.67$ and $R^2 = 0.68$, respectively.

ON Probability

Another metric that is studied on the obtained user data is the ON probability and its relation to the MOS scores. ON probability is calculated as the mean of ON duration divided by the sum of ON and OFF duration per cycle, *i.e.*, $\frac{\overline{ON}}{\overline{OFF}+\overline{ON}}$. An exponential law model is suggested with a strong correlation as given in Fig. 7.23(b). The mean of the ON probability for each MOS is calculated and depicted in Fig. 7.23(d). The first column of Table 7.20 presents the logarithmic, exponential and the linear fits for the ON probability. An exponential model can represent the relation between the ON probability and the MOS with a good fit ($R^2 = 0.74$), while the logarithmic ($R^2 = 0.66$) and the linear ($R^2 = 0.69$) models are slightly weaker.

Table 7.20: ON probability and average ON duration: Models and corresponding goodness of fit (R^2) values.

	ON probability		Average ON Duration [ms]	
Model	Equation	R^2	Equation	R^2
Log.	$MOS = 8.18 \cdot \log(P_{\text{ON}}) + 4.44$	0.66	-	-
Exp.	$MOS = 0.15\text{e}^{3.38 \cdot P_{\text{ON}}}$	**0.74**	$MOS = 2.32\text{e}^{0.00004 \cdot \overline{ON}}$	0.67
Lin.	$MOS = 9.84 \cdot P_{\text{ON}} - 5.33$	0.69	$MOS = 0.000013\overline{ON} + 2.28$	0.68

Average OFF Duration Per Session

In addition to the metrics related to the ON state of the video stream, we also study the similar metrics for the OFF states. The average OFF duration is the mean duration of the freeze events. The relation between the average OFF duration and the MOS is given in Fig. 7.24(a) with $R^2 =$

0.72 and $R^2 = 0.68$, for exponential and linear fits, respectively. The fitted equations are depicted in the second column of Table 7.21 for the both mathematical models.

OFF Probability

The OFF probability is calculated as the mean of OFF duration divided by the sum of ON and OFF duration per cycle, *i.e.*, $\frac{\overline{OFF}}{\overline{OFF}+\overline{ON}}$. The relation between the OFF probability and the MOS is given in Fig. 7.24(b) with $R^2 = 0.74$ and of $R^2 = 0.69$ value for exponential and linear fits, respectively. The corresponding fitted equations are given in the first column of Table 7.21.

Table 7.21: OFF probability and average OFF duration: Models and corresponding goodness of fit (R^2) values.

	OFF probability		Average OFF Duration [ms]	
Model	Equation	R^2	Equation	R^2
Exp.	$MOS = 4.67\mathrm{e}^{-3.38 \cdot P_{\mathbf{OFF}}}$	**0.74**	$MOS = 4.67\mathrm{e}^{-0.0003 \cdot \overline{OFF}}$	**0.72**
Lin.	$MOS = -9.84 \cdot P_{\mathrm{OFF}} + 4.51$	0.69	$MOS = -0.001 \cdot \overline{OFF} + 4.51$	0.68

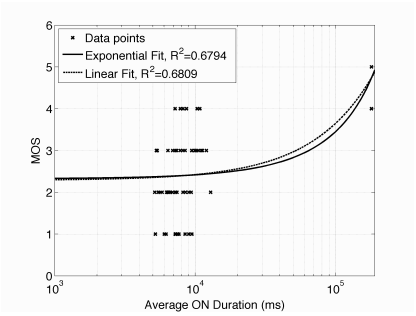
Impact of ON and OFF states on the MOS

We have also studied the impact of ON and OFF state metrics on the MOS. With the given cycle and the average ON and OFF states, the OFF duration influences MOS much more than the average ON state duration. We studied the relation between the three parameters, *i.e.*, average ON, average OFF, and the MOS value in Eq. 7.29. The fits with various parameters are given in Fig. 7.25(a) and 7.25(b). The table that depicts the R^2 values with different parameters are given in Table 7.22. When the weight of the average OFF state duration, *i.e.*, $\overline{OFF}$ gets much higher as compared to $\overline{ON}$, the goodness of the fit (*i.e.*, R^2 value) improves.

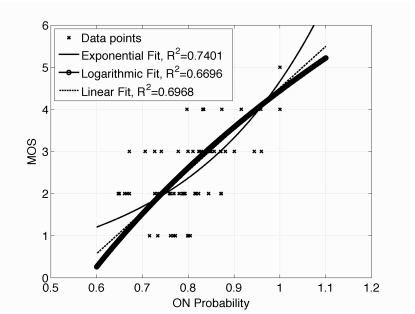
$$MOS = a \cdot \exp(b \cdot \overline{OFF}) + c \cdot \exp(d \cdot \overline{ON}) + e \tag{7.29}$$

Total OFF duration and Average OFF probability

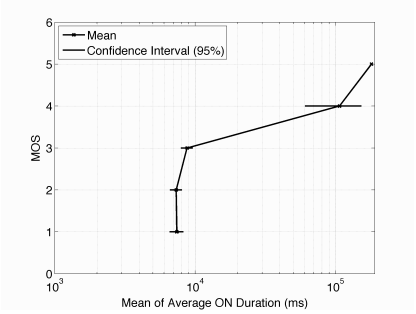
We have concluded in this section that as long as there is a video freeze during a video session (via streaming via the local storage based on the particular settings used in this study), the user perceived quality (quantified via MOS), is not different whether the content is skipped or not. This indicates that energy saving during the video freeze is possible. Based on Fig. 7.24(d), 0.2 is the maximum acceptable (MOS$\geq$ 3) OFF probability in our video streaming experiments. If



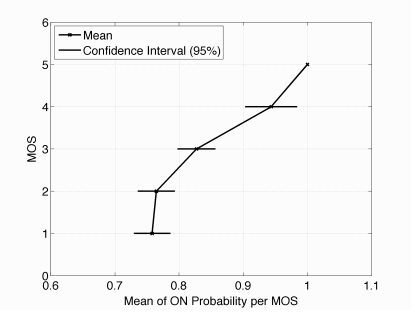
(a) Average ON duration (ms)



(b) ON probability

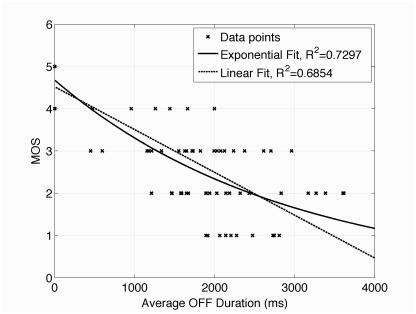


(c) Mean of Average ON duration per MOS (ms)

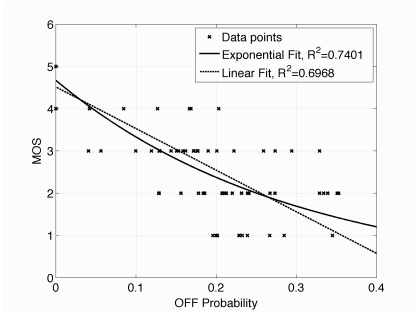


(d) Mean of ON probability per MOS

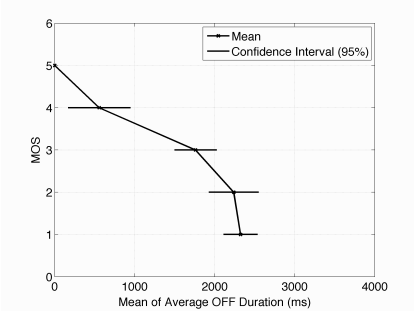
Figure 7.23: A set of metrics based on the ON state of the video streaming are given with the corresponding MOS scores.



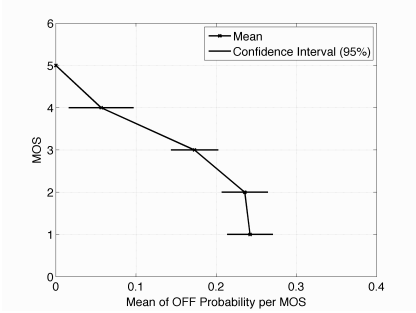
(a) Average OFF Duration (ms)



(b) OFF Probability



(c) Mean of Average OFF Duration per MOS (ms)



(d) Mean of OFF Probability per MOS

Figure 7.24: A set of metrics are given with the corresponding MOS scores.

Table 7.22: Impact of ON and OFF states On the MOS

a	b	c	d	e	R^2
0.25	-0.0006	0.75	7.62e-06	1.52	0.69
0.50	-0.0006	0.50	9.02e-06	1.70	0.70
0.75	-0.0006	0.25	1.18e-05	1.88	0.71
0.90	-0.0006	0.10	1.61e-05	1.99	0.71
2.99	-0.0005	1.33	1.57e-06	-0.04	0.74

0.2 is substituted to the linear equation Eq. 7.30 (with $R^2 = 0.95$), the acceptable total freeze

duration is calculated for a 3 minutes long video is roughly 40 seconds.

$$T_{\text{freeze}} = 208 \cdot P_{\text{OFF}} - 1 \qquad [\text{s}] \tag{7.30}$$

7.7.5 Energy Measurements

After the user QoE study, we conducted the energy measurements. The power consumption during video stream is obtained via Monsoon power monitoring tool. Next, we calculated the average energy consumption for each scenario by multiplying the mean power consumption with the mean duration. The video duration is the time from when the video playback is started until the time it has completed. Observe that there are no tail phases, thus there is no waste after the video playback as the video is streamed from the local sdcard of the device. The initial phase (time gap between when the play button is pressed and the first frame is displayed) can also be neglected as the video starts immediately after the button is pressed. Here the goal is not to provide generic models for energy consumption for local streaming but only to compare the energy consumption differences amongst the three scenarios.

In Fig. 7.26, the power consumption (with respect to the time) obtained via three different scenarios are given. The green line depicts the video session without any freeze, the red line depicts the video session with a freeze without a content jump, and the blue line depicts the video session with a freeze and when the content is skipped. Observe that the power consumption is extended when there is a freeze without a jump, thus more energy consuming. The power consumption values drop occasionally for the freeze scenarios as depicted with red and blue. We conducted 14 video streaming runs (each involving freezes in exponential distributed manner). We present the results from our energy measurements in three parts.

First, we visualize the smoothed power consumption values (via SMA with window size of 1500 ms) for all runs that involve occasional freezes. This helped to set the approximate thresholds for the power consumption and the freeze duration to be used in prediction mechanism. This way, we assure that the power consumption values can predict the duration of OFF events in the video stream. We synchronized the power measurements collected via the Monsoon power monitoring tool and the ON and OFF events captured by our video streaming tool. This is done by considering the first highest peak of power consumption value in the energy data, and the timestamp when the video is started as recorded by our video streaming tool. A snapshot of power consumption and the detected OFF events (together with the freeze durations) are given in Fig. 7.27. Based on the figure that visualizes the power consumption measurements and the duration of the OFF events, we set thresholds in order to find out the accuracy of prediction.

The threshold for the OFF duration is as 1500 ms, as 1500 ms the window size that is being used as an input to the window size of the SMA. In other words, smaller thresholds might not be visible at the smoothed data. A more detailed motivation on the choice of 1500ms is given in Section 7.2.2 Table 7.13. Based on the 1500 ms threshold between the ON and the OFF state, we intuitively selected 800 mW as the threshold as the correlation would look promising in Fig. 7.27. In other words, the red and the blue horizontal lines are the threshold for the OFF duration and the power consumption, respectively; and the red points above the red line and the blue power

measurement values below the blue horizontal line are likely correlated. Thus, based on the two threshold levels, true positive (*TP*), false positive (*FP*), true negative (*TN*) and the false negative (*FN*) values are identified, and the corresponding confusion matrix is given in Table 7.23. We calculated the precision, recall – also called as True Positive Rate (TPR) – and False Positive Rate (FPR) in Eq. 7.31-7.33, respectively.

$$Precision = \frac{TP}{TP+FP} \tag{7.31}$$

$$Recall = TPR = \frac{TP}{TP+FN} \tag{7.32}$$

$$FPR = \frac{FP}{FP+TN} \tag{7.33}$$

During a video stream, *TP* is detected when the power consumption is less than or equal to 800 mW and the OFF duration is higher than or equal to 1500 ms; *FP* is detected when the power consumption is less than or equal to 800 mW, and OFF duration is less than 1500 ms; *FN* is detected when the power consumption is above 800 mW and the OFF duration is higher than or equal to 1500 ms. *TN* is detected when the the power consumption is above 800 mW and the OFF duration is below 1500 ms. In the calculation of accuracy, we have only considered the instances where OFF values are recorded by our tool, *i.e.*, when the OFF events are triggered. The results (*i.e.*, the confusion matrix, the number of samples, *N*, and the precision, True Positive Rate (*TPR*) and False Positive Rate (*FPR*) values for each of the 14 runs (video sessions that involve occasional freezes) are presented in Table 7.24. The summary statistics from those runs, *i.e.*, the mean, standard deviation, minimum and maximum values of precision, recall (True Positive Rate (*TPR*), and False Positive Rate (*FPR*) are summarized in Table 7.25. As the calculated precision (average 0.89) and the recall (average 0.97) values are acceptable, we consider the 800 mW as the boundary between an ON and OFF state. In other words, we considered that the power consumption drops below 800 mW when there is a freeze event. If there is no freeze, *i.e.*, a smooth playout, the power consumption value is considered to be higher than 800 mW. We also did not consider the values above 1000 mW as these values (also seen via the CCDF plot in Fig. 7.28(a)) belong to the high peaks generated at the beginning and at the end of each video session. The separated datasets for the freeze (*i.e.*, OFF) region and the smooth playout (*i.e.*, ON) region are depicted Fig. 7.28(b) and Fig. 7.28(c), respectively.

	Power below 800 mW	OFF duration above 1500 ms
TP	✓	✓
FP	✓	×
TN	×	×
FN	×	✓

Table 7.23: Confusion matrix: description of TP, FP, TN, and FN.

Run 1, N =19	Power below 800mW	Power above 800mW	Precision	Recall = TPR	FPR
Freeze above 1500 ms	9	0	0.82	1	0
Freeze below 1500 ms	2	8			
Run 2, N =18					
Freeze above 1500 ms	9	2	1	0.82	0.22
Freeze below 1500 ms	0	7			
Run 3, N =26					
Freeze above 1500 ms	20	0	0.95	1	0
Freeze below 1500 ms	1	5			
Run 4, N =23					
Freeze above 1500 ms	15	0	1	1	0
Freeze below 1500 ms	0	8			
Run 5, N =25					
Freeze above 1500 ms	12	0	0.8	1	0
Freeze below 1500 ms	3	10			
Run 6, N =20					
Freeze above 1500 ms	11	0	0.85	1	0
Freeze below 1500 ms	2	7			
Run 7, N =23					
Freeze above 1500 ms	14	0	1	1	0
Freeze below 1500 ms	0	9			
Run 8, N =16					
Freeze above 1500 ms	7	1	0.78	0.88	0.091
Freeze below 1500 ms	2	10			
Run 9, N =22					
Freeze above 1500 ms	8	0	1	1	0
Freeze below 1500 ms	0	14			
Run 10, N =18					
Freeze above 1500 ms	1	0	0.85	1	0
Freeze below 1500 ms	2	5			
Run 11, N =23					
Freeze above 1500 ms	13	0	0.81	1	0
Freeze below 1500 ms	3	7			
Run 12, N =14					
Freeze above 1500 ms	6	0	0.86	1	0
Freeze below 1500 ms	1	7			
Run 13, N =23					
Freeze above 1500 ms	10	0	0.91	1	0
Freeze below 1500 ms	1	12			
Run 14, N =20					
Freeze above 1500 ms	11	0	0.92	1	0
Freeze below 1500 ms	1	8			

Table 7.24: Precision, Recall (TPR), and FPR values for prediction of OFF events by the power consumption are depicted for each video experiment run.

	Average	Std.	Min.	Max.
Precision	0.89	0.08	0.78	1.00
Recall (TPR)	0.97	0.06	0.82	1.00
FPR	0.02	0.06	0.00	0.22

Table 7.25: Summary of accuracy values of 14 runs.

7.7.6 Discussion on Energy Consumption and MOS

If Eq. 7.30, Eq.,7.34 and Eq.,7.35 are substituted into Eq.7.36 (from Table 7.21), then Eq.,7.37 and Eq. 7.38 are obtained for the *freeze with picture jump* and the *freeze without the picture jump* sceanarios, respectively. Accordingly, we present the exponential models between the energy saving vs. MOS (first row in Fig 7.29), and the energy waste vs. MOS (second row in Fig 7.29), for the freeze with picture skipping and the freeze without picture skipping scenarios, respectively.

$$T_{\text{freeze}} = \frac{E_{\text{saving}}}{P_{\text{saving}}} \tag{7.34}$$

$$T_{\text{freeze}} = \frac{E_{\text{waste}}}{P_{\text{waste}}} \tag{7.35}$$

$$MOS = 4.67\text{e}^{-3.38 \cdot P_{\text{OFF}}} \tag{7.36}$$

$$MOS = 4.67 \cdot \exp(-3.38 \cdot (((E_{\text{waste}}/\text{J}/0.185) + 1)/208)) \tag{7.37}$$

$$MOS = 4.67 \cdot \exp(-3.38 \cdot (((E_{\text{waste}}/\text{J}/0.728) + 1)/208)) \tag{7.38}$$

In the case of the video streaming scenario with picture freeze and with the picture jump, we assume an average power drop 185.1 mW (*i.e.*, the power consumption is dropping from 909.9 mW to 728.7 mW) during a picture freeze. The maximum achieved energy saving is 5 J which corresponds to a MOS value of 3. A further increase in the energy saving causes drop in the MOS value.

In the case of the video streaming scenario with the picture freeze and without the picture jump, we assume that the power drops to 728.7 mW. During the picture freeze, the power consumption value stays at this power level, thus there is corresponding waste. This shows that a freeze reduces the MOS value and in parallel increases the energy consumption. The total energy waste is linearly increasing with the duration of the freeze, and exponentially reducing the MOS value.

The models between energy and the MOS are also non-linear as the ones presented in Section 7.6.2, although the achieved energy values per MOS are slightly different. The reasons for this are as follows. The QoE models obtained from our recent study is based on the smartphone with particular experiment settings (*e.g.*, three minutes long video with exponentially distributed freeze durations). The models obtained in Section 7.6.2 are based on the QoE models obtained by the authors in [116]. Those models are obtained via video session lengths of 30 s long involving deterministic freeze durations and counts. The second reason is that in Section 7.6.2, we assumed that the power consumption drops by 250 mW during a picture freeze. In our recent measurements with local streaming, we measured the power drop as 185 mW.

The calculated energy saving is relatively very small as compared to other potential energy savings in the communication system, *e.g.*, at the base stations. The energy saving would scale up to much higher values given that more and more users watch videos on smartphones everyday

with longer durations, and the number of smartphones has already reached more than a few billion. Thus, the total energy saving during video streaming would scale up in total to gigajoules.

	Average [mW]	Std. [mW]	Median [mW]	N
OFF phase	728.7	40.4	724.8	3854892
ON phase	909.9	38.3	909.2	12053435

Table 7.26: Summary statistics for power consumption at ON and OFF phases.

7.8 Limitations

The experiments are conducted via particular streaming and downloading apps, which we developed, on particular smartphones that we have in our lab. It might be necessary to study the energy consumption on other similar network-based real-time multimedia applications, *e.g.*, implementations via Google's WebRTC, Skype, or audio streaming apps such as Spotify, Internet radio. In addition, more iterations of the experiments could yield more robust results. In calculating the energy consumption, we assumed a static power consumption for all file sizes. The results presented are specific to the experiment settings including, *e.g.*, particular SIM card from a popular telecom operator with 6 Mbps highest peak download rate, and particular video sequences.

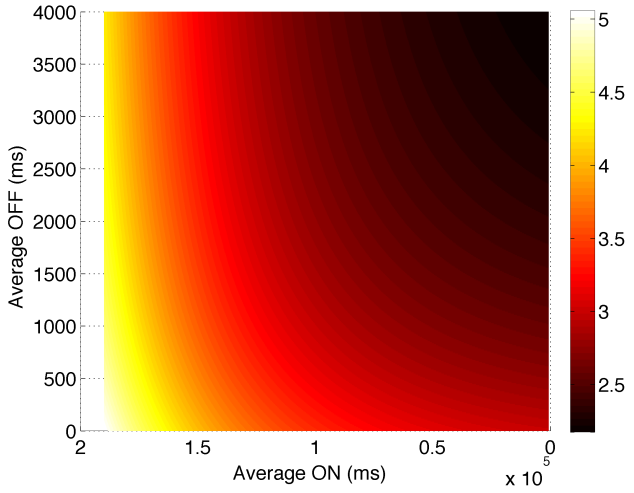
The presented energy measurements are based on average power consumption values obtained from many iterations on particular smartphones and phone settings. It is necessary to state that we do not provide generic energy consumption values, but instead we compare a set of scenarios with the same experimental setting in order to identify differences in between different scenarios and to recommend directions for energy saving.

7.9 Summary

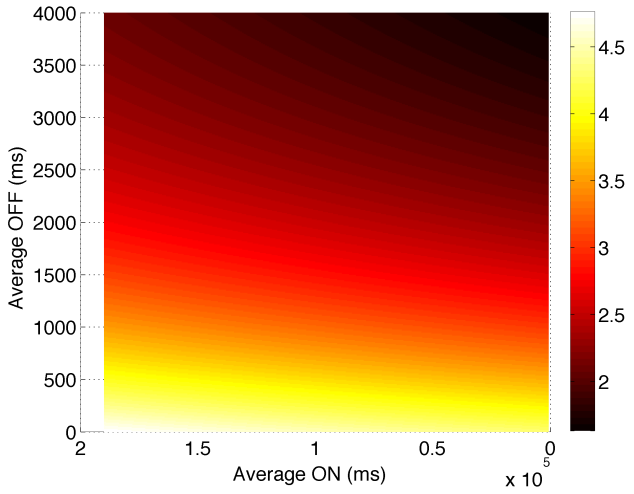
In this section, we have presented extensive energy measurements for video application on smartphone, and then addressed the relationship between the energy consumption and QoE. We compare a set of scenarios such as real time video streaming and local streaming (*i.e.*, streaming a pre-downloaded video through the local storage). Next, we presented that the anomalies of smartphone power consumption can reveal anomalies (*e.g.*, freezes) at the user interface for a steady-state video stream. This enables the future enhancement of QoE tools that assess the QoE in a non-intrusive and energy efficient manner. Consequently, we present outcome of further download duration and energy measurements during file downloading, and suggest energy-efficient download mechanisms and power models via the 3G interface of smartphone. We stated that downloading multiple files (*e.g.*, multimedia podcasts) independently on smartphone is not energy efficient, and thus we recommend downloading them in a scheduled manner. However,

the choice of scheduling must also be chosen carefully to avoid congestion or under utilization of the available link throughput. Based on our results on a use case, if there are multiple large-size files (*e.g.*, 10 MB each), scheduling the downloads in parallel is not recommended as both the average processing time and the total energy consumption are statistically significantly higher as compared to the sequential downloading scenario. In contrast, if there are multiple rather small-size files (*e.g.*, 10KB each), then scheduling the downloads in serial is not recommended as both the average processing time and the energy consumption are statistically significantly higher as compared to the concurrent downloading scenario. Lastly, we discuss the freezes during real-time video stream with respect to the energy and the QoE. In between the energy consumption and the QoE, we evaluated the tradeoff (when there is a picture jump) and the win-win (when there is no picture jump) cases. Based on the presented use case, we concluded that 5 J energy saving per a 3 minutes long video session can be achieved during real-time video streaming in a with-jump scenario while keeping the MOS ratings greater than or equal to 3. Although the saving is low per a short video clip, small savings might be critical in "low battery" level context, and minimize battery consumption during an important video stream. In addition, the saving would scale up to gigajoules in overall as the mobile-based video streaming has exponential growth over the years. Additionally, we showed the energy waste if the same video is streamed via the without-jump approach, which in parallel influences negatively the MOS values. Thus, during a without-jump video streaming as in TCP-based stream, the more the freezes the higher the energy waste and the less the MOS level. Thus, the ultimate aim is to avoid freezes, which can be accomplished via careful adaptation of the video resource consumption to the capabilities of the communication systems.

Yet, we have studied the energy consumption of an interactive smartphone application, video streaming. In the next chapter, we study the energy consumption when the smartphone is not actively used by user, *e.g.*, when the smartphone screen is OFF.



(a) $MOS = 0.9 \cdot \exp(-0.0006 \cdot \overline{OFF}) + 0.1 \cdot \exp(1.62e\text{-}05 \cdot \overline{ON}) + 2$, $R^2 = 0.71$. Right legend: MOS



(b) $MOS = 3.0 \cdot \exp(-0.0005 \cdot \overline{OFF}) + 1.34 \cdot \exp(1.57e\text{-}06 \cdot \overline{ON}) - 0.04$, $R^2 = 0.74$. Right legend: MOS

Figure 7.25: Average ON and OFF states vs MOS

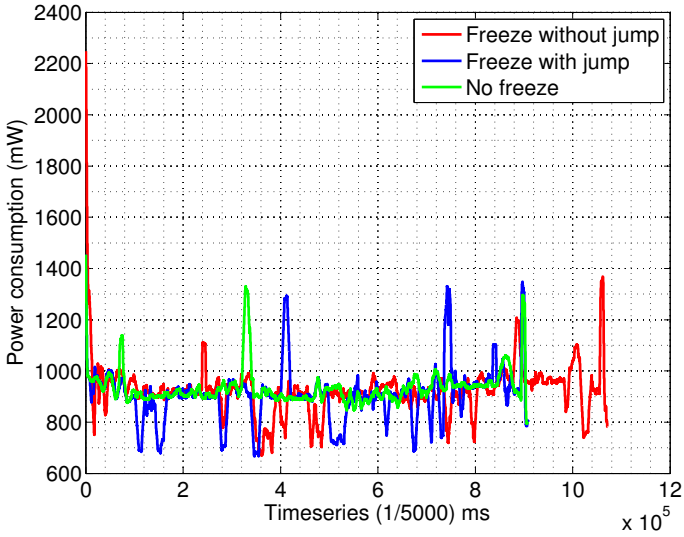


Figure 7.26: Snapshot of a power consumption data for the three scenarios ($W = 7500$).

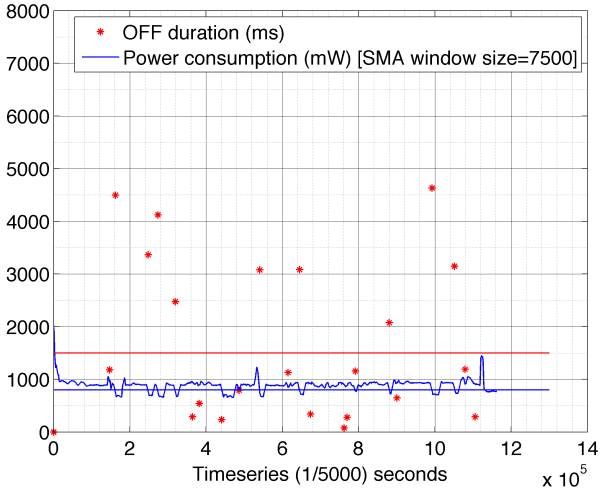
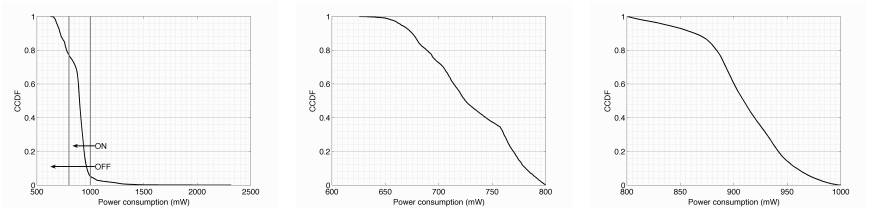


Figure 7.27: Snapshot of a power consumption data with the corresponding OFF events triggered by the tool.



(a) Power consumption ($W = 7500$ mW) (b) Power consumption ($P < 800$mW) (c) Power consumption ($P > 800$mW and $P < 1000$mW)

Figure 7.28: CCDF of power consumption datasets.

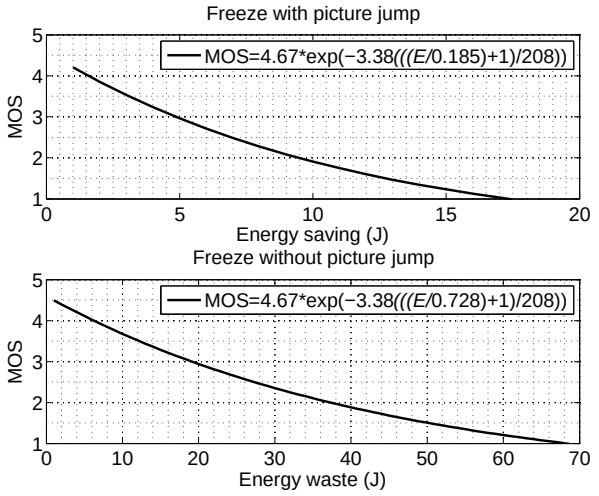


Figure 7.29: MOS per energy saving while video streaming on the smartphone.

Chapter 8

Energy Saving When the User is Not Interactive

"I have no special talent. I am only passionately curious." –Albert Einstein.

8.1 Introduction

Mobile network traffic grows exponentially, and this increases the overall energy consumption including the radio access networks and the mobile terminals such as the smartphones [189]. In Chapter 3, the battery life of smartphones is concluded to be one of the most influential factors on the end-user perceived quality. Radio Network Controller (RNC) controls the network resources to efficiently manage the spectrum and to extend battery life, in such a way that while the smartphone is connected to Internet via the cellular data interface, power consumption varies depending on RRC power states such as IDLE, PCH, FACH, or DCH [95]. However, the frequent and asynchronous network activities from different apps cause 3G data module of smartphone to oscillate in-between different power-consuming states. During the oscillations, there is demotion latency, *e.g.*, caused by state timers, and also promotion latency, *e.g.*, due to channel re-allocation. In parallel, there is excessive energy being consumed, which is considered as wasted. Therefore, it is important to minimize the number of transitions in-between RRC power states, and to perform the data traffic in bursts whenever applicable. One way to minimize state transitions is by prolonging RRC's IDLE state. Controlling RRC power states on smartphones is not feasible as the promotions and de-

motions depend on the application's network characteristics, device configuration, network settings, and particular operator configuration at Radio Access Network (RAN). However, it is practical to control the state of the 3G cellular data module. When the 3G module is switched OFF for long enough and then when data module is switched ON later point in time, network-based applications fetch the updates and present them to the user in bursts (all at once). Therefore, the 'cost' of bursty traffic is the latency in the data transmission (*e.g.*, incoming messages and notifications), and it might eventually impact the end-user perceived quality, *i.e.*, QoE. Thus, the produced drawback of energy saving approaches should be evaluated throughout user studies involving real users.

There has been numerous approaches to prevent power waste caused by the characteristic of the RRC of the 3G cellular module on terminals. A comprehensive literature study has been presented in Chapter 2, Sections 2.10.6-2.10.9. Most of them are either sophisticated or hard to deploy on the smartphones found in the market as they might require special permission or specific configuration. We propose a rather simple "sledgehammer" algorithm to improve energy efficiency on smartphones when the screen is at OFF state. The strength of our algorithm is that it is highly user-centric, and we aim to maximize the smartphone energy saving when the screen is OFF without degrading user perceived quality.

The contribution of this chapter can then be presented in three parts. Firstly, we conduct three user studies with 43 users first in order to understand the user behaviour on smartphone. This is done through the *logging phase*, while users are using their own smartphones in daily life, while we log the screen events, network usage, charging attempts, as well as the running applications. We obtain and compare different mathematical models for the duration of the smartphone's screen states. Secondly, we present the NyxEnergySaver tool, conduct energy measurements, and confer the amount of energy saving when the screen is OFF (screenOFF) with different data state durations (*e.g.*, data module is ON or OFF). Toggling the data module might impact the user-perceived quality of the smartphone. Thus, thirdly, we conduct further user studies during the *experimental phase* (also called intervention phase), with the same participants in the logging phase, and study the impact of NyxEnergySaver tool on the user perceived quality.

This chapter is organized as follows. The identified metrics and the data collection are presented in Section 8.2. The method and the results for the logging phase, are given in Section 8.3. The details of NyxEnergySaver tool together with its validation are described in Section 8.4. The method and the results for the experimental phase, which users install the NyxEnergySaver on their smartphones, are given in Section 8.5. The main contributions are summarized in Section 8.6. The discussion on the results are presented together with a set of limitations in Section 8.7. The paper is concluded

and future directions are outlined in Section 8.8.

8.2 Metrics and Data Collection

The measured metrics and the data collection methods employed in three user studies are presented in this section. User consent is obtained from each volunteered user before the studies. The studies comprise automatic data logging on the user smartphones via a logger tool and user interviews. We collected the user demographics such as user occupation, age, gender, and the phone type. The logger tool autonomously collected data from the user's smartphones during daily life usage. The logger tool was kept running continuously as a background application on the user's smartphones during the study period. The collected metrics via the logger tool are as follows.

8.2.1 Screen State Metrics

The screen state comprises of two states, either screen is ON (screenON) or screen is OFF (screenOFF). During the user studies, the screen state events are recorded to guide on how to reduce energy consumption with the awareness of the user interaction with the screen. Predicting the screenOFF duration, may help to reduce the energy consumption without degrading the user experience.

The timestamp of *i*'th screen state, *s*, is represented by T_i^s, where $s \in \{\text{screenON}, \text{screenOFF}\}$. The screenOFF duration is calculated in Eq. 8.1 as the time difference between the timestamp of the OFF-to-ON transition, T_i^{screenON}, and the timestamp of the previous ON-to-OFF transition, ${T_{i-1}}^{\text{screenOFF}}$. The screenON duration is calculated in Eq. 8.2 as the time difference between the timestamp of the ON-to-OFF transition, $T_i^{\text{screenOFF}}$, and the timestamp of the previous OFF-to-ON transition, ${T_{i-1}}^{\text{screenON}}$.

$$\Delta T_i^{\text{screenOFF}} = T_i^{\text{screenON}} - {T_{i-1}}^{\text{screenOFF}} \qquad [\text{ms}] \tag{8.1}$$

$$\Delta T_i^{\text{screenON}} = T_i^{\text{screenOFF}} - {T_{i-1}}^{\text{screenON}} \qquad [\text{ms}] \tag{8.2}$$

We calculated the *hour of the day* (HOD) of each screen-based user interaction. The timestamp is floored to the corresponding HOD, *i.e.*, *hh*:*mm* is recorded as *hh*, where *h* stands for the hour, and *m* stands for the minute. In addition, the *number of interactions per hour* (IPH) is calculated for each user as the total number of detected screenON events divided by the logging phase duration.

8.2.2 Network Metrics

We recorded the wireless technology, which the smartphone is connected to the Internet, upon a detection of screen state change event, amongst a widespread cellular access technologies and WiFi. This way, we collected information regarding which wireless technology is active during the interaction. It should also be noted that, in all measurements, only one network interface is active at a time. For simplicity, based on the background knowledge, we categorized EDGE and GPRS as 2G; UMTS, HSPA, High Speed Downlink Packet Access (HSDPA), High Speed Uplink Packet Access (HSUPA) as 3G; and LTE as 4G. In addition, we recorded the cellular data state, *i.e.*, ON or OFF, upon screen change during the experimental phase.

8.2.3 Other Collected Data

During the logging phase, we have also collected data regarding the charging events on users' smartphones. The charging events are collected when smartphone is plugged or unplugged to/from the power source (AC or USB).

We studied the background applications(based on names and keywords) during the logging phase of Study 2 and 3. We filtered out the dominant names, which do not identify applications such as the wireless network interface name, clock, server, Android, and the phone brand name.

8.2.4 In-lab Energy Measurements

In order to quantify the energy saving achieved via NyxEnergySaver, we conducted benchmark energy measurements in the lab. The Monsoon power monitoring tool [170] is well accepted by the research community as it can provide accurate high resolution (5 kHz) power measurements, *i.e.*, one power measurement (P_t) sample every 200μs on the smartphone. We conducted energy measurements, while the Monsoon power monitoring device is intercepting the battery terminal of the smartphone. We installed NyxEnergySaver with various dataON and dataOFF settings, and then conducted energy measurements for each setting in order to study the energy savings provided by NyxEnergySaver. $1\,\mathrm{s}/200\mu\,\mathrm{s} = 5000$ samples, are averaged within every second, and the average $\bar{P}_t$ values are obtained. Next, energy consumption E_T, during screenOFF duration T, is calculated as the sum of per-second mean power values $\bar{P}_t$, multiplied by the unit second as shown in Eq. 8.3.

$$E_T\,[\mathrm{J}] = \sum_{t=1}^{T} \bar{P}_t\,[\mathrm{W}] \cdot 1\,[\mathrm{s}] \tag{8.3}$$

8.2.5 User Studies

Phases	Study 1	Study 2	Study 3
Logging	✓	✓	✓
Experimental	×	✓	✓
NyxEnergySaver Configuration	×	1 min dataON in every 30 min	Personalized

Table 8.1: User study phases and NyxEnergySaver configuration

We performed three different user studies in order to collect information on user behaviour on smartphones. All three studies comprise the initial *logging phase*, in which we collected data autonomously, as described in Section 8.2, on the users' smartphones during their daily life. Study 2 and Study 3, in addition to the logging phase, consist of an extra *experimental phase*, in which NyxEnergySaver is installed on users' smartphones and user feedback is obtained. Table 8.1 summarizes the phases in all user studies. The last row depicts the configuration during screenOFF in each study. There was no experimental phase in Study 1.

First, the data obtained from three user studies is presented in this section. The subject ID, the logging phase duration, and experimental phase duration (for Study 2 and 3) are given in columns $1-3$ in Table 8.2. Demographics (gender, age, occupation, phone type) of each participant in three studies are given in columns $4-7$. In total, there were 43 subjects (12 female, 31 males), with various occupations, at different ages (within the range $18-45$), and using various smartphone brands. The average logging phase durations were 27 days, 4 days, and 22 days, for Study 1, 2, and 3, respectively. The average experimental phase durations were 2 days, 13 days, for Study 2 and 3, respectively.

8.3 Logging Phase

8.3.1 Method

The first user study (Study 1) is conducted in 2011 with the participation of 29 users located in the US [67]. Study 2 is conducted in May 2013 with the participation of 5 users located in two countries such as Sweden and Turkey [196]. Study 3 is the extended version of Study 2, and is conducted in Fall 2013 with the participation of 9 users, located in three countries, *i.e.*, Canada, Sweden, and Turkey. The collected data

consists of the screen state, network state, battery charging events, and the running applications. In addition to the autonomous logging on user smartphones, demographics on each user is collected via online survey, and qualitative data is collected via user interviews.

8.3.2 Overview of the Results

We first present the results regarding the screen events that were collected during the logging phase. Then, we present and compare different mathematical models on the user smartphone interaction. We discuss the wireless access technologies being used, and how the usage has been influenced by the data plan. We present the used applications on the smartphone during the logging phase of Study 2 and 3. In addition, we present the battery charging patterns of the users in Study 2 and 3.

Modeling Screen States

We first extract each user's smartphone screen durations for screenON and screenOFF states that were collected in the logging phase. We have identified a rather high divergence in the screen-based user smartphone interactions as presented in a boxplot in Fig. 8.1. The rectangular box is located in between the first (25 %) and the third (75 %) quartiles of the data, while the horizontal line inside the box depicts the median. The median number of interactions during night hours (*i.e.*, 1 AM−5 AM) is less than the ones during the day. There are users with a high number of interactions around midnight, *e.g.*, *S*39. We know from the interview that this user keeps chatting with friends during night hours. Fig. 8.2 depicts the number of interactions per hour (IPH) for each user. IPH is varying from one user to another (*S*35 was the least interactive with 0.4 interactions per hour, *S*9 was the most interactive with 7.2 interactions per hour). *S*9 has occasional interactions during the day and also in the evening (*i.e.*, between hours 6 AM−12 PM). The mean screenON and screenOFF durations are calculated for each user, as given in columns $8-9$ in Table 8.2. Overall, the mean screenON and screenOFF durations are 4 ($\pm$1.4) min and 20 ($\pm$6.9) min, respectively (with the 95 % confidence level). The total number of screen change events (sum of screenON and screenOFF) are given in the last column of Table 8.2.

We have spotted a few extreme values with the subjects and confirmed these values with the users via interviews. *S*22 and *S*35 have mean screenOFF durations, 40.6 min and 158.6 min, respectively as upper outlier users. *S*22 switches OFF the smartphone during night, and this applies to all weekdays. The phone was turned ON in-between hours 8 AM and 1 PM during the day. For *S*35, 4.38 % of the screenOFF data is greater

S	Log. Phase (days)	Exp. Phase (days)	Gender (M/F)	Age	Occupation	Phone Type	Mean screenON (minute)	Mean screenOFF (minute)	# of Screen Events
					STUDY 1				
1	32	-	M	18-24	Cust. Service	Samsung Captivate	4.1	11.5	3847
2	28	-	M	25-35	Company	Motorola Droid	2.5	11.4	5762
3	28	-	M	25-35	Driver	MyTouch 4G	4.1	12.6	4014
4	33	-	F	18-24	Researcher	HTC Incredible	0.8	17.1	4548
5	28	-	F	25-35	Edu. Admin	G2	1.0	8.3	8374
6	28	-	F	25-35	ICT Consultant	Motorola Droid X	1.8	19.1	3866
7	24	-	M	25-35	Web developer	Motorola Droid	7.5	12.8	3777
8	28	-	F	25-35	Medical Admin	Motorola Droid	1.7	23.4	2983
9	28	-	F	25-35	Nanny	HTC incredible	1.6	5.8	9253
10	8	-	F	25-35	Unemployed	Samsung G. S	3.4	14.1	1249
11	28	-	M	36-45	Unemployed	HTC Evo	14.2	25.8	2012
12	28	-	M	25-35	Univ. Mngt.	Motorola Droid	2.4	21.2	2875
13	28	-	M	25-35	Contractor	Motorola Droid X	2.3	11.2	3485
14	28	-	M	25-35	Acc. Coord.	Motorola Droid	1.9	18.9	3323
15	28	-	F	25-35	Oper. Analyst	Motorola Droid X	11.5	13.7	2970
16	23	-	M	36-45	System Analyst	Motorola Droid	9.5	15.7	2098
17	28	-	M	25-35	ICT consultant	HTC EVO	3.0	14.0	4456
18	28	-	M	25-35	Teacher	Motorola Droid	26.8	10.2	1962
19	29	-	F	25-35	Admin assistant	HTC Evo	0.6	10.1	7140
20	28	-	M	25-35	Univ. student	Motorola Droid	1.2	11.3	5803
21	28	-	M	25-35	Grant admin	HTC Incredible	3.2	9.4	7673
22	28	-	M	25-35	Univ. student	Motorola Droid 2	3.7	40.6	1683
23	28	-	M	25-35	Systems analyst	Htc Evo (WiMAX)	1.6	28.7	2622
24	28	-	F	18-24	Univ. student	Motorola Droid 2	1.0	12.6	5363
25	28	-	M	25-35	Senior admin	Motorola Droid	3.8	17.0	3339
26	29	-	M	25-35	Univ. student	Motorola Droid X	1.9	9.3	6209
27	27	-	M	36-45	Paramedic	Motorola Droid	11.3	13.0	2593
28	28	-	F	36-45	Housewife	Motorola Droid X	2.3	22.7	3028
29	21	-	M	25-35	Nurse	Samsung Captivate	4.3	12.0	3661
					STUDY 2				
30	5	3	M	25-35	Researcher	Samsung Note 2	2.7	22	558
31	5	3	M	25-35	Unemployed	Samsung G. S3	3.0	7.7	1311
32	2	2	M	25-35	Univ. Student	Sony Xperia Droid	2.3	10.5	468
33	5	1	M	25-35	Univ. Student	Samsung G. S3	5.1	32	353
34	3	1	F	25-35	Researcher	Samsung G. S3	1.5	7.9	782
					STUDY 3				
35	32	11	M	25-35	Researcher	Sony Xperia	3.4	158.6	548
36	24	6	M	25-35	Unemployed	Samsung G. S2	3.8	26.2	1691
37	36	8	M	25-35	Univ. student	Samsung G. S3	3.0	37.0	3317
38	33	24	M	25-35	Engineer	Samsung G. Ace 2	1.0	16.1	5386
39	14	22	M	25-35	Engineer	Samsung G. S4	1.25	19.7	1925
40	27	-	M	25-35	Univ. student	Sony Xperia	2.0	17.9	3725
41	13	6	M	25-35	Engineer	LG Optimus GPro3	1.9	6.9	4115
42	15	19	M	25-35	Researcher	Samsung G. Note	2.1	34.8	1002
43	6	-	F	18-24	Univ. student	Vivo Y19t	1.2	25.2	657

Table 8.2: Study 1, 2, 3: User profiles and mean screen state durations (in the logging phase) are depicted.

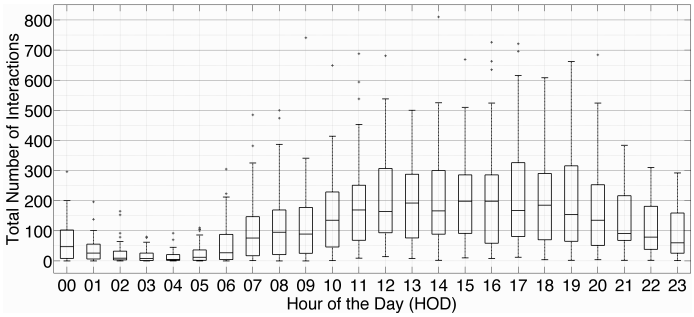


Figure 8.1: Study 1, 2, 3: Number of screenON events based on hour of the day, for all users.

than 15 hours. From the interview, we confirm that $S35$ used the phone only for calling related to work. This means that the smartphone screen was ON only during the phone calls, therefore the number of interactions is limited to the incoming or outgoing calls. $S7$, $S11$, $S15$, $S16$, $S18$, and $S27$ are spotted as outlier users based on the mean screenON durations. In $S18$'s screenON data, 2.30 % of the screenON durations are greater than 7 hours. Also, this user has the highest mean screenON duration, 26.8 min. From the interview, we know that $S18$ is a sports trainer, and switches OFF the smartphone during the trainings. We studied the reasons for this further; $S18$'s smartphone was switched OFF during weekends. We have observed from the collected data that the smartphone was switched ON at noon and this applies for different weekdays, too. Improper shutdowns caused the screenOFF timestamps not being recorded by our logger tool, which reduced the granularity of the data for that particular user. We did not identify any error in the remaining users' data. Nevertheless, overall, we interpret that the outliers are due to diverse user behaviour, which represents the realistic environment. Thus, in our analysis, we include all the collected data throughout the user studies.

For each user study, we study three interaction models, *i.e.*, fit the screen durations into Weibull, Exponential, and Pareto distributions, as previously given by Eq. 2.8-2.10 in Chapter 2. We calculated the corresponding parameters along with the coefficient of determination (R^2) values for each model. The models for the screenOFF durations for the three user studies are given in Fig. 8.3(a)-8.3(c), respectively. Similarly, the models for the screenON durations are given in Fig. 8.3(d)-8.3(f). Just by looking at the CCDF plots that are presented with the $\log(x)$ and $\log(y)$ axes, high tails are visible, which can also be modeled with Pareto. In order to compare the overall fit amongst the models,

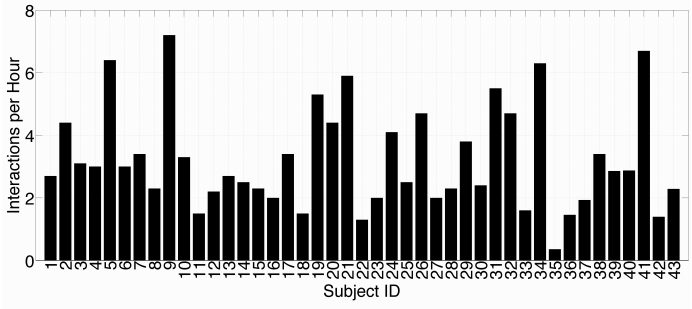
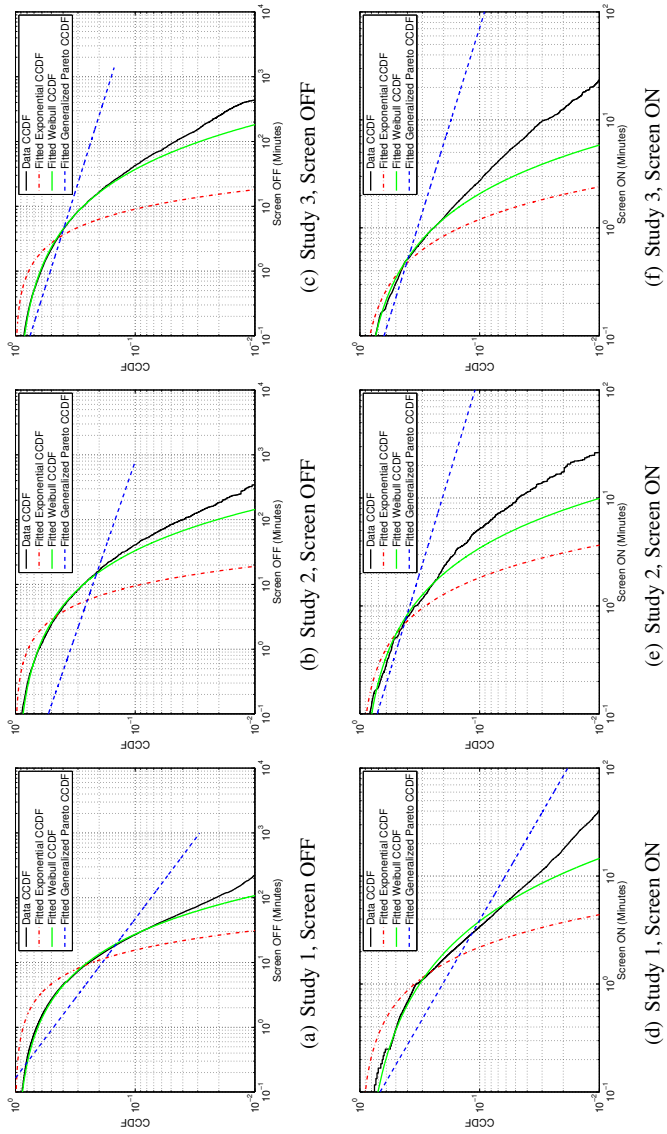


Figure 8.2: Study 1, 2, 3: Number of screenON events per hour (per subject)

the corresponding R^2 values are given together with the parameters and number of data points in Table 8.3. The rows 3 to 6 correspond to screenOFF modeling in Study 1, Study 2, Study 3, and for all studies, respectively. The last four rows are the models for the screenON duration. Amongst the three models, the best-fitted one is the Weibull distribution with an overall R^2 value of 0.99 with a values less than 1. This means that the longer the screen has stayed OFF, the more probable that it will stay OFF. The result is similar to the ones presented in [276]. The R^2 values for the exponential distribution are considerably high (minimum 0.90), but it does not represent the long screen state durations as good as Weibull, *i.e.*, it decays faster than Weibull. However, we consider it suitable enough to model the screen interaction as a two-state exponential model with mean screenON and screenOFF durations of 4 ($\pm$1.4) min and 20 ($\pm$6.9) min, respectively.



(a) Study 1, Screen OFF (b) Study 2, Screen OFF (c) Study 3, Screen OFF

(d) Study 1, Screen ON (e) Study 2, Screen ON (f) Study 3, Screen ON

Figure 8.3: Study 1, 2, 3: Models for screen state durations (all subjects)

State	Study	Samples	Pareto			Weibull			Exponential	
			a	b	R^2	a	b	R^2	b	R^2
screenOFF	Study 1	59770	0.17	0.41	0.77	0.51	0.19	0.99	0.15	0.79
screenOFF	Study 2	3219	0.03	0.19	0.84	0.47	0.18	0.99	0.24	0.82
screenOFF	Study 3	9702	0.02	0.17	0.84	0.44	0.18	0.99	0.25	0.78
screenOFF	All Studies	72691	0.03	0.20	0.80	0.51	0.18	0.99	0.22	0.91
screenON	Study 1	60199	0.05	0.52	0.83	0.51	1.31	0.98	1.05	0.85
screenON	Study 2	1735	0.03	0.27	0.87	0.65	1.05	0.99	1.26	0.93
screenON	Study 3	11176	0.02	0.28	0.86	0.67	1.7	0.99	1.92	0.94
screenON	All Studies	73110	0.02	0.3	0.88	0.64	1.32	0.99	1.6	0.94

Table 8.3: Model coefficients based on screen state durations for all studies

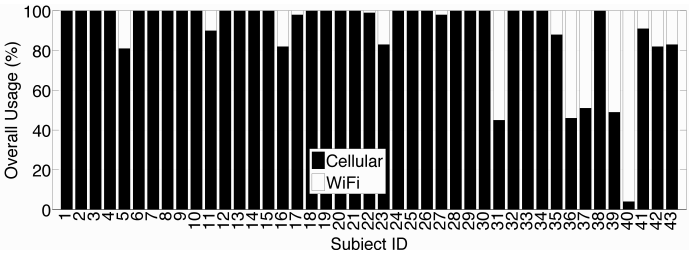
Network Data Usage v.s. Cellular Data Plan

The study on the network data usage for each user is important in order to choose the target users for the potential energy saving solution, NyxEnergySaver. First, the data usage with respect to WiFi and cellular is presented in Fig. 8.4(a). *S*31, *S*36, *S*39, and *S*40 used WiFi more than 50 % of the time. *S*38 never used WiFi, and he had unlimited cellular data access. *S*39 stated the reason for switching to WiFi as to prevent using up from his monthly data limit. *S*40 had a limited 3G data plan in the beginning of the study, then he reported that he cancelled it as he thinks that it is "worthless". *S*41 and *S*42 used WiFi very seldom (less than 20 % of the time). This is due to the fact that the cellular data is paid by the employer, and there is no restriction in the amount of data usage. *S*43 did not have 3G data plan, as the user thought that it was expensive. WiFi usage was also studied with respect to the users' cellular data plan. CCDF plots of the WiFi usage for limited and unlimited cellular data plans for all users in three studies are presented in Fig. 8.4(b). The users with limited cellular data plan had a tendency to use WiFi access, as expected. Limited cellular data usage helps to decrease the network traffic in 3G networks [88]. As WiFi is less power-consuming than any cellular data interface, the energy consumption can also be reduced in this way [185].

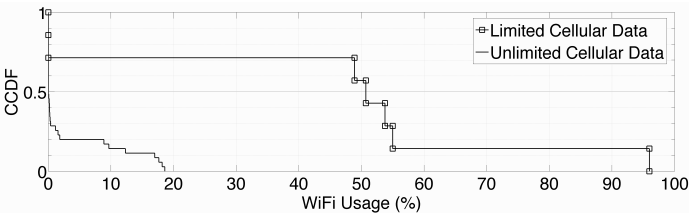
In addition, for Study 2 and 3, the percentage of the data usage based on different cellular wireless access technologies are presented in Fig. 8.4(c). Amongst the cellular wireless access technologies, except for *S*40 and *S*43, more than 90 % of the time all users are connected to Internet via 3G or 4G. *S*39 was the only subject using 4G LTE (located in Canada) 94 % of the time. *S*40 had a 3G data plan during the logging phase, but he claimed that he was not using it much. *S*43 only used 2G amongst the cellular data access technologies, as this user did not have 3G or 4G data plan.

Charging Patterns

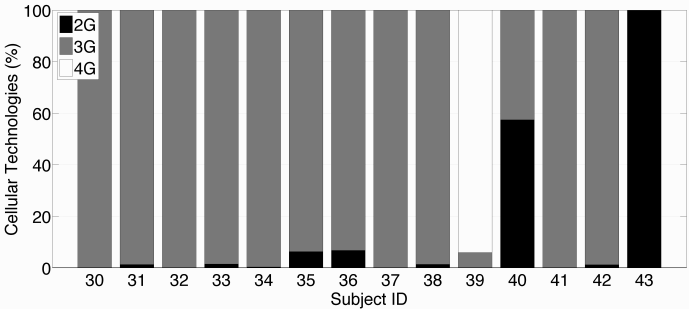
We studied the charging patterns of the users in Study 2 and Study 3. CCDF plot of the HOD of detected charging events for each user are given in Fig. 8.5. *S*32 and *S*34



(a) Study 1, 2, 3: WiFi and cellular data connectivity



(b) Study 1, 2, 3: WiFi usage vs data plan



(c) Study 2, 3: Used wireless access network

Figure 8.4: Used wireless network access technologies

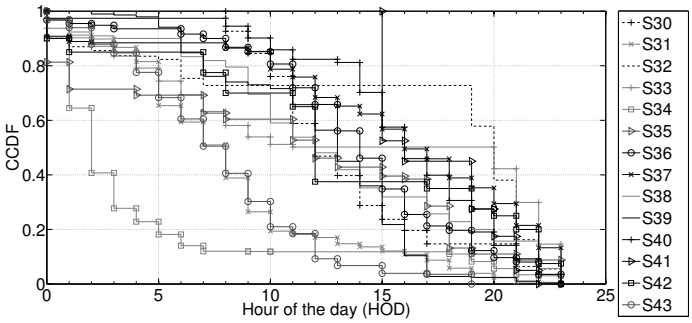


Figure 8.5: Study 2, 3: Charging attempts based on HOD

were the users who did not charge the phone much during the day (7AM−7PM). *S*41 charged the phone during afternoon and evening, *i.e.*, in between 3PM−11PM. It is observed that the remaining 11 users charged their smartphones occasionally, and the charging hours are spread over the whole day, *i.e.*, 24 hours. The results of the charging study indicates that the majority of the smartphone users in this study were accustomed to the limitation of battery life, craving for charging opportunities during the day, and thus they can be classified as opportunistic chargers.

Running Applications

The number of occurrences of the applications (represented by keywords) are given in Fig. 8.6. Some mostly used popular applications were Facebook, Maps, Email, Whatsapp, Talk, Viber, Twitter, and Dropbox in a descending order. Amongst the users, only one user, *S*30, occasionally used Internet radio. This app can also be used, *e.g.*, streaming audio packets, when the screen is OFF. Overall, the top three application categories, used during interaction, were communications (Email, Whatsapp, Talk, Viber, Tencent, Skype, Handcent), social networking (Facebook, Instagram, Twitter, Linkedin, Socialhub, Socialbook, Smartshare), and travel (Maps, Google, location services) with 37 %, 36 %, 11 %, respectively.

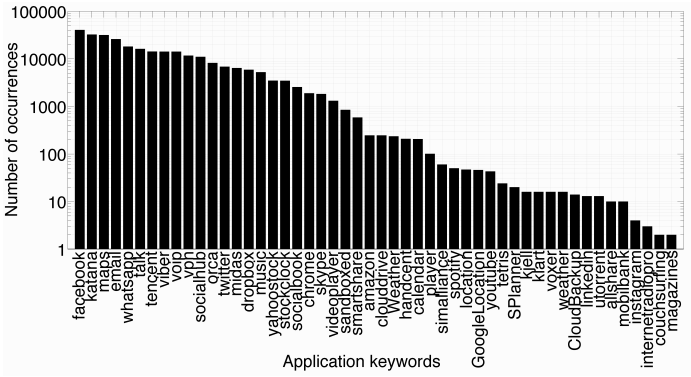


Figure 8.6: Study 2, 3: Applications used in the logging phase

8.4 NyxEnergySaver Tool

8.4.1 Tool Description

The goal of this research is to minimize the energy consumption during screenOFF without impacting user-perceived quality. Therefore, we developed an Android OS based smartphone tool, NyxEnergySaver, which controls the cellular data module of the smartphone. It switches OFF the cellular data module when the smartphone screen is switched OFF, and switches it ON when the screen is switched ON. During screenOFF, the data module is switched ON and OFF based on the cellular data module's configured ON and OFF durations. This way, the data transmission to and from smartphone is delayed during the dataOFF interval, and is performed in bursts when the data module is ON later. If the WiFi is the active interface and Internet-reachable, the tool does not affect the connectivity; it only affects the available 3G or 4G connectivity if the WiFi is unavailable or disabled. The reason is the following. If a smartphone is in the reach of a known accessible WiFi hotspot, *e.g.*, at home, it should not switch on cellular data module, upon screenON, as the preferred connectivity is WiFi. The functional flow of NyxEnergySaver is given in Alg. 3.

8.4.2 Validation of NyxEnergySaver

The validation of the tool is done with three different smartphones at a stationary position during screenOFF, *i.e.*, without user interaction. Components such as vibration and audio are switched OFF in order to minimize the influence on the power mea-

Algorithm 3: Functional Flow of NyxEnergySaver

```
Assuming that the tool is installed while screen is ON.
while SmartphoneIsRunning and WiFiIsDisabled. do
    Listen to screen change trigger and update screenONTriggered;
    Listen to 3G/4G data change trigger and update dataOFFTriggered;
    if screenOFFTriggered then
        screenOFF = TRUE;
        Switch OFF 3G/4G data module;
    if screenONTriggered then
        screenOFF = FALSE;
        Switch ON 3G/4G data module;
    if dataOFFTriggered then
        if screenOFF then
            Schedule to switch ON 3G/4G data module after T_OFF minutes;
    if dataONTriggered then
        if screenOFF then
            Schedule to switch OFF 3G/4G data module after T_ON minutes;
```

surements. The experiments on different smartphones are conducted consecutively, because we used the same SIM card. The Samsung Galaxy S is rooted (1 GHz CPU running Android v.4.2.2, kernel v.3.0) that operated in 3G, the Samsung Galaxy S4 is non-rooted (1.9 GHz CPU running Android v.4.3, kernel v.3.4) that operated on 3G or 4G, based on the manual configuration. Another non-rooted smartphone ZTE T40 (1.2 GHz CPU running Android v.4.1.2, kernel v.3.4) is used in the experiments as the third phone type that operated on 3G. During the experiments, 3G data was HSDPA, and the 4G data was LTE. Popular network-based applications such as Twitter, Facebook, GMail, GTalk, and WhatsApp were installed on the smartphones with the same user accounts in all phones. We used one smartphone at a time in each experiment. The amount of network usage highly depends on the number of connections with other accounts, *e.g.*, number of followed accounts on Twitter, number of friends on Facebook, and the activity level of the accounts. During the experiments, a Twitter account was following 400 other accounts, and a Facebook account had 520 friends, which can be considered as realistic.

Phone Type	Mean [min]		Minimum [min]		Maximum [min]		Std. Deviation [min]		Std. Error ($\bar{SE}$)	
	T_{OFF}	T_{ON}	T_{OFF}	T_{ON}	T_{OFF}	T_{ON}	T_{OFF}	T_{ON}	T_{OFF}	T_{ON}
ZTE T40 (3G)	5.105	1.091	5.100	1.082	5.111	1.120	0.003	0.006	0.001	0.001
Samsung S (3G)	5.098	1.102	5.094	1.087	5.108	1.132	0.003	0.009	0.001	0.002
Samsung S4 (3G)	5.070	1.035	5.061	1.030	5.097	1.046	0.011	0.003	0.002	0.001
Samsung S4 (4G)	5.029	1.024	5.023	1.115	5.036	1.016	0.003	0.016	0.001	0.003

Table 8.4: Validation of NyxEnergySaver mean dataON and dataOFF durations for nominal 5 min OFF, and 1 min ON. More than 30 data points (more than 3 hours) are collected for each experiment.

Data State Durations

The delays during state transitions in-between dataON to dataOFF states, *i.e.*, between the DCH and the IDLE modes of the cellular data module, can increase the nominal dataON and dataOFF durations. Since the transition delays might vary with the device hardware, we validated the functional flow of NyxEnergySaver on the aforementioned three different phone types. The results are presented in Table 8.4. For testing purposes in the lab, we have set the nominal dataON and dataOFF durations to $T_{\text{ON}} = 1$ min and $T_{\text{OFF}} = 5$ min, respectively. The experiment is iterated multiple times to minimize the error in-between the mean of the measured values and the nominal values, *i.e.*, less than or equal to 2.1 %. The Standard Error (SE) of the sample mean is calculated as the standard deviation of the sample data divided by the square root of the number of the samples. With validation, we also make sure that the standard error ($\bar{SE}$) of the measured mean, *i.e.*, percentage of the standard error w.r.t. to the mean, dataON and mean dataOFF durations do not exceed 1 % as given in the last two columns of Table 8.4.

Data Traffic Patterns

We studied the data activities on the Samsung Galaxy S4 smartphone in multiple scenarios: (i) the received data traffic while the NyxEnergySaver is not running in multiple time scales (1 min, 2 min, 4 min, 10 min, 15 min), and (ii) the received data traffic while the NyxEnergySaver is running with 1 minute dataON in every 16 minutes, respectively. We developed a separate smartphone logger app in order to record the amount of received data and installed it on the Samsung Galaxy S4. We collected 2370 data points, and studied the received data at pragmatically selected timescales, *i.e.*, 1 min, 2 min, 5 min, 10 min, and 15 min. For example, for 1 minute timescale, we calculated the received data (Bytes) during every 1 minute time interval. As a baseline, in Fig. 8.7, the data volumes are presented with CCDF curves, from bottom to second from top, within 1 min, 2 min, 5 min, 10 min, and 15 min timescales. Obviously, there exists a

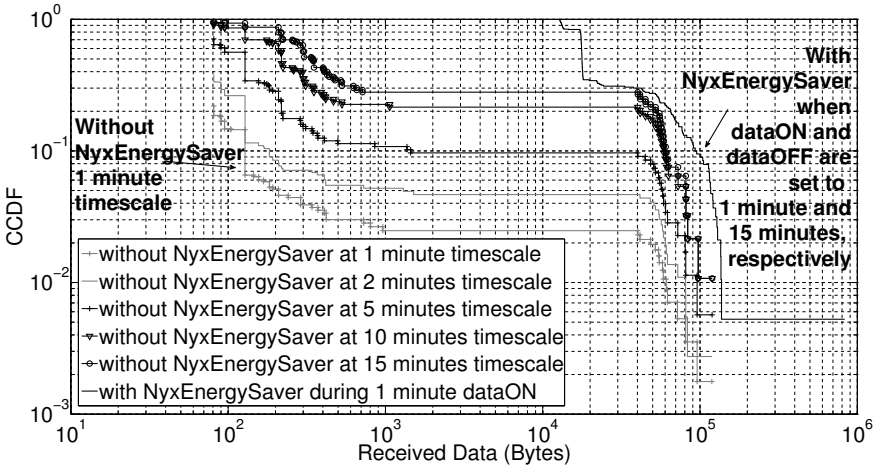


Figure 8.7: Received data at different timescales (Samsung Galaxy S4)

large volume of data activities when NyxEnergySaver is not running. Within 1 min timescale (see bottom-most curve), the mean is 1.6 KB with a relative standard error of the mean of 13 %. For the use case when NyxEnergySaver is running, we recorded the received data via the logger at two points in the process: one measurement immediately after the data module is switched ON, and one measurement just before the data module is switched OFF. When NyxEnergySaver is running with 15 min dataOFF and 1 min dataON, the data volume during the 1 min dataON (see topmost curve) is given in Fig. 8.7. In this case, there exists a rather large-sized data volume, *i.e.*, minimum 10 KB, during the 1 min dataON durations. The mean is 40.6 KB with a relative standard error of the mean 11 %. We confer that the volume of data activities within one minute time intervals is increased (such as the topmost curve) via NyxEnerySaver, by delaying the more frequent small sized data activities (such as the bottom-most curve). The choice of 15 min dataOFF is motivated further in Section 8.4.2.

Preliminary Power Measurements

We investigate the energy consumption during *web page download* for a set of website URL's. The test software is instructed to send GET messages to popular website URL's and then downloads the contents of the websites: cnn.com (news); (ii) bbc.com (news); (iii) slashdot.org (RSS feed); (iv) cvs.com (retail). The test software records the total

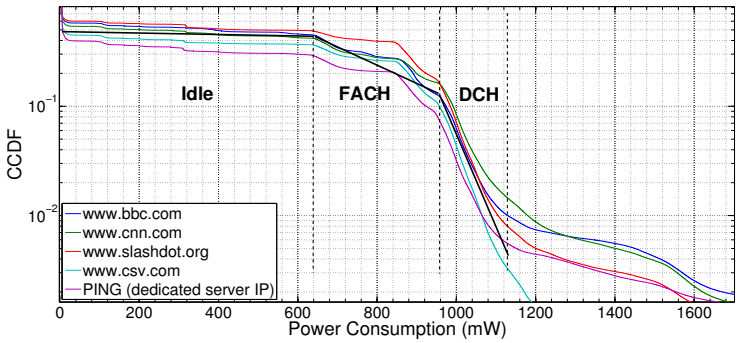


Figure 8.8: Power consumption patterns for a set of applications in Samsung Galaxy S. The black solid line shows the approximate trend [196].

amount of download time and MPMT records the power consumption similar to the ping experiments. We repeated these experiments in the lab every 45 seconds, *i.e.*, a time period so that the radio module is switched completely to the *IDLE* state, before the next network activity starts. In total, a minimum of 35 iterations with 45 seconds time gap were conducted and each iteration consisted of more than eight million power data points as the power metric is recorded at a sample rate of 5 kHz. The CCDF plots for the power consumption for the mentioned experiments are depicted in Fig. 8.8. We classified the three energy states intuitively and showed the corresponding approximate trend per state with a black solid line.

Baseline Energy Saving

The baseline energy saving is evaluated for two different phones: Samsung Galaxy S and Samsung Galaxy S4.

Samsung Galaxy S: We launched a set of cloud-based applications such as Facebook, Twitter, LinkedIn, Instagram, GMail, GTalk, Google Services, WhatsApp from a dedicated user application account. We also launched one highly popular Swedish news application (AftonBladet [226]) that also has polling functionality. During the execution of the app, the NyxEnergySaver's extra power consumption characteristics are investigated as well. For the verification of power consumption of NyxEnergySaver, we conducted a set of ground truth energy measurements in the lab with different *scenarios* via MPMT during the screen-OFF state: (1) $T_{\text{DataOFF}} = 3$ min; (2) $T_{\text{DataOFF}} =$

9 min; (3) $T_{\text{DataOFF}} = 14$ min; (4) $T_{\text{DataOFF}} = 19$ min; (5) $T_{\text{DataOFF}} = 29$ min; (6) OFF, $T_{\text{DataOFF}} = T_{\text{ScreenOFF}}$, *i.e.*, data-disabled period is equal to the screen-OFF period as parameterized by user; (7) *NoToggling*, *i.e.*, with NyxEnergySaver but without toggling cellular data interface; (8) *w/o NyxEnergySaver*, *i.e.*, without listening on sensors, without writing metrics to the smartphone's local storage, and without toggling cellular data interface to obtain the base-line power consumption. We collected samples for the eight different scenarios to such extent that each scenario has a Standard Error (SE) of the mean values less than 1 %. The CCDF plots are presented together with the Standard Errors for each scenario in Fig. 8.9. In addition, we calculate the gain factors for each

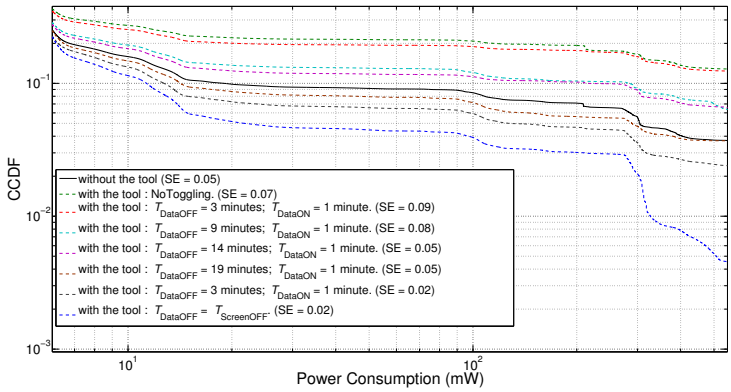


Figure 8.9: Power consumption patterns for different available cellular data durations in screenOFF state in Samsung Galaxy S [196].

scenario (over 10 million samples each), while taking the *w/o NyxEnergySaver* scenario as base-line and is calculated in Eq. 8.4. $\bar{P}_{\text{w/o NyxEnergySaver}}$ is the average power consumption *without NyxEnergySaver*; and $\bar{P}_{\text{Scenario}}$ is the average power consumption of any of the eight scenarios. Therefore, the larger the gain factor, the higher the saved energy.

$$gainFactor = \frac{\bar{P}_{\text{w/o NyxEnergySaver}}}{\bar{P}_{\text{Scenario}}} \tag{8.4}$$

The gain factors, *gainFactor*, obtained through in-lab measurements for different DataOFF durations is presented in Table 8.5. The gain factor is less than 1 for the scenarios where the T_{DataOFF} duration is set to less than 29 min. One reason is that NyxEnergySaver is itself consuming some amount of energy, which is likely caused by the

Table 8.5: The gain factors for different scenarios in Samsung Galaxy S.

w/o NyxEnergySaver	NoToggling	3 min OFF	9 min OFF	14 min OFF	19 min OFF	29 min OFF	Completely OFF
1	0.34	0.35	0.54	0.63	0.83	1.34	2.57

logging (listening and writing) process. Without the latter, the energy consumption is expected to decrease even further. Based on the in-lab experiments, $T_{\text{DataOFF}} = 29$ min and $T_{\text{DataON}} = 1$ min combination has a gain factor of 1.34, and expected to be even more when the energy consuming logging process is disabled. By taking the results obtained from this part as basis, we conducted a user study to identify the influence of the (29 min DataOFF, 1 min DataON) combination on the end-user perceived quality during the smartphone's screen-OFF state.

Samsung Galaxy S4: We studied and compared the energy consumption in an other smartphone (and indeed more representable Samsung Galaxy S4) when NyxEnergySaver is not running, or when it is running with pre-configured dataOFF and dataON durations. The energy consumption of the smartphone is depicted in Fig. 8.10 when NyxEnergySaver is not running, or when NyxEnergySaver is running with different dataOFF durations. The curves (from top to bottom) present the energy consumption of the smartphones when 5 min dataOFF, without NyxEnergySaver (black solid line), 10 min dataOFF, and 15 min dataOFF, respectively. The bottommost curve presents the energy consumption when NyxEnergySaver never switches ON the data module during screenOFF. We performed 100 min long experiments for each configuration. We collected indicative energy measurements that visualize the differences in energy savings amongst different dataOFF durations. The amount of energy saving of the NyxEnergySaver tool increases with the duration of the dataOFF period, as expected. On the other hand, the particular setting in Samsung Galaxy *S*4 of 5 min dataOFF followed by 1 min dataON duration does not save energy (see top curve in Fig. 8.10), but in fact it increases the energy consumption.

Based on Fig. 8.10, we observe that amongst the indicative energy saving, 15 min dataOFF, and 1 min dataON pair has the minimum dataOFF duration amongst the others that likely save energy as compared to the scenario without NyxEnergySaver. The energy gain increases with the dataOFF time, but we would like to keep it at minimum level as it might influence the QoE of the apps.

Furthermore, we performed six iterative measurements for the two scenarios in order to obtain statistically significant results. The two scenarios were (i) without running NyxEnergySaver, (ii) with NyxEnergySaver configured to 15 min dataOFF and 1 min

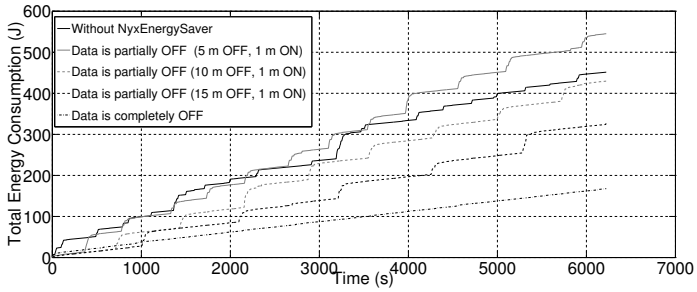


Figure 8.10: Total energy consumption on Samsung Galaxy S4 for five scenarios

dataON on the smartphone. In total, we collected more than 38 thousand energy measurement data points (throughout a 10 hours long experiment) for each scenario during the screenOFF. We obtained the linear energy consumption models as given in Eq. 8.5 ($R^2 = 0.97$) and Eq. 8.6 ($R^2 = 0.92$) for with and without NyxEnergySaver scenarios, respectively. The confidence interval of the linear models were 95 %. The baseline energy gain (18 %) is calculated in Eq. 8.7 as the percentage of energy saving for scenarios without and with use of NyxEnergySaver.

$$E_{\text{with}}\,[\text{J}] = 0.05554\,[\text{J/s}] \cdot T\,[\text{s}] \tag{8.5}$$

$$E_{\text{without}}\,[\text{J}] = 0.06747\,[\text{J/s}] \cdot T\,[\text{s}] \tag{8.6}$$

$$E_{\text{gain}}\,[\%] = 100 \cdot \left| \frac{E_{\text{with}}\,[\text{J}] - E_{\text{without}}\,[\text{J}]}{E_{\text{without}}\,[\text{J}]} \right| \tag{8.7}$$

8.5 Experimental (Intervention) Phase

8.5.1 Method

In this phase, we installed the NyxEnergySaver tool on the users' smartphones, and kept the tool running as a background application. Thus, configuring the NyxEnergySaver for all users based on the hour of the day (*e.g.*, switching OFF 3G between 1 AM−6 AM), is not feasible, as the interaction might still occur during the night (see Fig. 8.1). The configuration of NyxEnergySaver applied in Study 2 and Study 3 is given in the last row of Table 8.1 in Section 8.2.5. In Study 2 (for subjects $S30 - S34$),

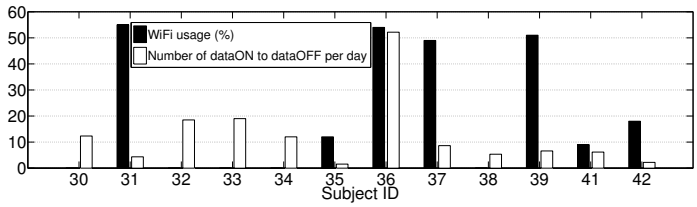


Figure 8.11: Study 2, 3: WiFi Usage and Number of dataON-to-dataOFF

the tool was configured such that the data module was ON for 1 minute ($T_{\text{ON}} = 1\,\text{min}$) in every 30 min while the screen is OFF during $T_{\text{OFF}} = 29\,\text{min}$. In Study 3, the tool (*i.e.*, T_{ON} and T_{OFF}) was personalized for each user. In other words, we maximized the dataOFF durations during the screenOFF period as much as possible for each user. The longer the user is not interacting with the smartphone, the longer the dataOFF duration can be extended during screenOFF without degrading QoE. This is done by adapting the dataOFF intervals to the screenOFF durations collected in the logging phase.

The users were provided an interface on the tool to submit their feedback and QoE via five-level (from 1 bad to 5 excellent) ACR scale, *i.e.*, MOS, whenever. However, the users chose to provide only qualitative feedback. In addition, at the end of this phase, we conducted user interviews and asked them two questions: (i) "Have you realized any change in the overall performance, *e.g.*, connectivity, received messages latency, of your smartphone during the experimental phase?", and (ii) "Have you perceived improvement in battery life during the experimental phase as compared to the initial logging phase?".

8.5.2 Quantitative Results

As already explained, we conducted an additional experimental phase in Study 2 and 3. In Study 2, NyxEnergySaver is configured such that dataOFF and dataON duration during screenOFF are set to $T_{\text{OFF}} = 29\,\text{min}$ and $T_{\text{ON}} = 1\,\text{min}$, respectively. In Study 3, the configuration is performed based on the user-smartphone interaction, *i.e.*, the screenOFF durations, during the logging phase.

In the second column of Table 8.6, the number of dataOFF events, *i.e.*, the number of state transitions from dataON to dataOFF, during the experimental phase are given. Configured and applied dataON and dataOFF durations on users' smartphones are given in the third column. The dataOFF durations are set such that the 87.5 % (*i.e.*, 7 out of 8) of the user's screenOFF duration in logging phase is less than or equal to the applied dataOFF duration. This way, there is a high probability that the user will

S	# dataON to dataOFF	Applied dataON, dataOFF	Mean screenOFF
30*	37	1 min, 29 min	22.0 min
31	13	1 min, 29 min	7.7 min
32	37	1 min , 29 min	10.5 min
33	19	1 min, 29 min	32.0 min
34*	12	1 min, 29 min	7.9 min
35	17	1 min, 474 min	158.6 min
36	313	1 min, 84 min	26.2 min
37	69	1 min, 84 min	37.0 min
38	128	1 min, 39 min	16.1 min
39	145	1 min, 39 min	19.7 min
41	61	1 min, 39 min	6.9 min
42	65	1 min, 84 min	34.8 min

Users marked with '*' noticed the behaviour of NyxEnergySaver.

Table 8.6: Study 2, 3: Parameters in the experimental phase

realize the change (as they switch ON the screen when data is OFF). They would also realize that the notifications and the content were delivered late as compared to the logging phase. Thus, the users might find this behaviour as annoying. The dataOFF durations were set to 474 min for *S*35; 84 min for *S*36, *S*37, and *S*42; 39 min for *S*38, *S*39, and *S*41. *S*40 and *S*43 did not have 3G data plans in the experimental phase, thus no cellular data module change events were detected for those particular users.

Amongst the users, the ones with low WiFi usage (as they use more cellular data) are expected to be influenced more by the tool. Thus, we present the WiFi usage and the number of dataOFF events per day in Fig. 8.11. Accordingly, *S*30, *S*32, *S*33, and *S*34 never used WiFi, and the number of dataOFF events per day were higher than 10.

8.5.3 Qualitative User Feedback

The results presented in this section are based on the qualitative feedback on NyxEnergySaver, which were obtained from the participants in Study 2 and 3 via interviews after the experimental phase. *S*40 did not have data plan during the experimental phase, so no data was received from this participant. *S*43 realized that the 3G data module was switched ON when the screen was switched ON, and found this behaviour as annoying. The reason was that she did not have a 3G cellular data plan and she had to pay for

the extra bytes used if the 3G data module was activated without permission (a phone specific issue). She eventually quitted the study. Amongst the remaining 12 users, only two subjects (*S*30 and *S*34, marked with '*' in Table 8.6), noticed the behaviour of NyxEnergySaver, *i.e.*, that the data module was switching ON and OFF based on the screen state. *S*30 used Internet radio, thus he realized that the data module was switching OFF when the screen was OFF. He requested to keep the radio ON when the screen is OFF. We added the additional feature particularly for this user, but we asked the user to disable the feature when he was not using the Internet radio. *S*34 perceived that the data was switching OFF when the screen was switched OFF. She did not find this behaviour annoying. She provided additional feedback: *"I do not want to get notifications. Notifications for example for Facebook would probably stress the life out of me and are absolutely switched off on my phone"*. *S*35 used the phone only for talking (did not use 3G data at all), thus no network-based applications were installed on the device although he had limited 3G data access on the smartphone provided by his employer. We were surprised that *S*32 and *S*33 did not realize the dataOFF events, although their smartphone connectivity were mainly based on 3G. *S*38 stated: *"Should there be a performance degradation?"*; this feedback indicated that the change was not perceived by the user. Indeed, for this subject, the number of detected dataOFF events per day was rather low; less than 10 per day.

We asked users whether they have perceived any improvement in the smartphone battery life. The feedback from the participants is varying. *S*35 replied: *"...the battery life has improved a lot, the battery was dying all the time before"*. We interpret the reason for this as follows. During the logging phase, the 3G cellular interface of the user was ON more than 90 % of the time, although he was not an active network-based app user, some built-in apps might have caused the signaling traffic, and extra energy consumption. In the experimental phase, this traffic has been prevented by NyxEnergySaver when the screen was OFF, thus energy saving is achieved. *S*39: *"I charge a lot, probably that's the reason I did not notice any change in the battery... both in work and office, my phone was charging..."*. Another reason for this user for not perceiving a change in battery life could be that he used WiFi more than 50 % of the time (*i.e.*, when NyxEnergySaver is not used). *S*41 did not perceive any improvement in the battery life. He commented: *"I use my phone a lot... I do not start the day always with* 100% *charge, and sometimes half full... I charge when I'm in the car, at office, at home..."*. This shows that he was an opportunistic charger. Another reason could be due to the Weber-Fechner law [72] that relates the human perception to the relative change of the stimulus. In other words, the improvement in the battery life may not be just-noticable-enough to be perceived by the user. Overall, the feedback from the participants is encouraging.

8.6 Contribution Summary

The main contributions presented in this chapter are summarized as follows:

Weibull and exponential screen state models: We compared Pareto, Weibull, and Exponential distributions for the screenON and screenOFF durations. We concluded that Weibull fits the best with R^2 value 0.99 with $a < 1$. Thus, it can be said that the longer the smartphone screen is OFF, the higher the probability it will stay OFF. Alternatively, an exponential distribution can also be suggested as the overall R^2 values are higher than 90 %, thus the screen interaction can be modeled with a two-state ON/OFF exponential model with screenON and screenOFF durations, 4 ($\pm$1.4) min and 20 ($\pm$6.9) min, (with the 95 % confidence level), respectively,

Sledgehammer-based energy saving is achieved without degrading user percieved quality: At least 18 % energy saving is achieved with Samsung Galaxy S4 during screenOFF, with 1 min dataON every 16 min, as compared to the scenario without NyxEnergySaver (in 10 hours). The percentage of energy saving increases with the dataOFF duration.

DataOFF durations during screenOFF are set such that in 87.5 % of the cases, the user switches ON the screen during the cellular 3G data module is OFF. Amongst 12 users (in the experimental phase), only two users (who were connected to Internet via 3G and never used WiFi), perceived a change in cellular data connectivity on the smartphone. Although, the two users realized that the data module was switched OFF, they have not reported this behaviour as annoying.

8.7 Discussion and Further Recommendations

We are aware of the following limitations in this work. The presented energy measurements are highly specific to the lab settings. Modeling user interaction is a hard task, and we need more user data, collected throughout longer studies, and for different smartphones. However, we obtained similar results highly support the results obtained in related work. According to Ericsson Report [188], the total screen time was measured as 37 hours per week. This corresponds to 22 % of the whole week. If we normalize our result from a two state exponential model with 4 minutes screenOFF time in every 24 minutes time inteval, we obtain 28 hours of mean screen ON time per week. This corresponds to 17 % of the whole week, which are 5 % less than the results provided by Ericsson Report.

The session length, *i.e.*, eventually screenON duration, depends on the application type, *e.g.*, maps and games might have rather high session lengths. However in this work, we rather focus on the screenOFF state durations as NyxEnergySaver saves energy during this state. Lessons learned from the user study answers to the question: "What can be improved in NyxEnergySaver?". The tool is not suitable for the users whose primary communication depends on VoIP, *i.e.*, which requires constant connectivity even if screen is OFF. Such applications are widely used in the US and Japan, but in Europe, still the GSM-based voice calls are dominant. In our user study, although there were a few VoIP users, *e.g.*, Viber and Skype, they did not complain as these applications were used mostly for texting when the screen is ON. Lastly, the tool needs to be improved in such a way that switching to dataON state can be delayed for a few seconds based on user preferences. The reason is that some users may use their smartphones as their primary clock, and switch ON the smartphone screen for this purpose only for a few seconds. In this case, switching ON the data module is not necessary. The charging patterns signify an opportunistic behaviour for the majority of the users, thus this might have prevented users to perceive the energy saving of NyxEnergySaver during the experimental phase.

We strongly recommend that the screen events need to coordinate and synchronize with the application API's. This way the functionality of applications are adapted to the user context. This can be done if application developers can register screen events to their Android applications and manage the network resource usages accordingly. We also recommend the application developers to enable their apps notifying other apps when the application uses network resources and in parallel listen to other apps for similar events. This way, the synchronization of application with respect to the network usage can be enabled towards energy saving.

8.8 Summary

In this chapter, we present a model of user interactions with the smartphone screen, and present a potential energy saving approach NyxEnergySaver without influencing the user experience. We state that the durations when the screen is ON and screen is OFF can be presented with two-state exponential models with a mean screenON and screenOFF duration of 4 ($\pm$1.4) min and 20 ($\pm$6.9) min, respectively. The screen state durations can also be modeled with Weibull. Different from the existing applications, first the user model through user studies are studied and then the energy saving tool, NyxEnergySaver is personalized based on the obtained user models.

The proposed software is highly simplistic, as one may call it a sledgehammer implementation. The proposed software does not require any major changes in the

network stack or in the OS, and can be easily installed to most of the Android OS based smartphones. The strength of our approach is its simplicity. The user perceived quality is considered with the mentioned simplistic approach, and we have obtained encouriging qualitative feedback from the participants. The idea here is that to release all network resources, and reduce the energy consumption to baseline (*e.g.*, completely disabled 3G data module). A clumping approach for only a particular application does not help as activity of even only a single application can make the RRC state to stay at the high power consuming state. Thus, all have to be deactivated simultaneously. The core of our approach is to enable an energy saving without impacting the user perceived quality. We also discuss the sociological aspects of this problem, emphasizing that the always-connected approach is often misinterpreted. Users need to be connected when they benefit from it (*e.g.*, instead of continuous connectivity), and this is only possible with context-aware smart mechanisms while also considering the big data. This would in parallel save time and eventually energy. We strongly recommend that the screen events need to coordinate and synchronize with the application API's enabling the apps to adapt to the varying device functionalities and user context.

Next, we presented the NyxEnergySaver tool that saves smartphone energy by prolonging the duration without any 3G-based data activity. This is done by switching OFF the 3G data module when the screen is switched OFF. In order not to impact the end-user perceived quality, the data module is switched ON and OFF for particular durations during the screenOFF periods. In Samsung Galaxy S4, with the defined specific settings (in ten hours), 18 % energy saving is achieved when there is 1 min dataON duration every 16 min. The energy gain increases with the data OFF duration. Finally, we present the results of the experimental "intervention" phase, *i.e.*, NyxEnergySaver is personalized and installed on users' smartphones. It has been observed from the qualitative data that the user-perceived quality has not changed. Therefore, it can be concluded that a sledgehammer-based energy saving is possible as long as the proposed approach is user-centric.

We plan to conduct and repeat the user study with more number of users, and extend the dataOFF durations to increase the energy savings further while maintaining user perceived quality. In addition, we plan to add a feature to adapt the NyxEnergySaver configuration, in real time, in such cases when the user behaviour (data plan, network usage, charging patterns) change, or are alerted by the user feedback.

Chapter 9

Conclusions

"Simplicity is the ultimate form of sophistication." –Leonardo da Vinci.

9.1 Thesis Summary

Smartphones have become portals to the outside world guiding users in diverse domains including health, navigation, and communication in daily life. In parallel, smartphones and the applications/services running on them, can quantify the user behaviors with the available sensors and enhanced API's, which can help to monitor the problems in real time and improve applications' perceived quality by the users. The QoE perceived especially from network-based applications on smartphones is highly vital and is critical to user acceptance. Network operators have a crucial role in enabling at least an acceptable application performance as they are often considered as applications' primary network interface to the Internet. Thus, a poor perceived quality by a user from a network-based smartphone application running on a smartphone might have negative consequences reducing the revenue of the corresponding network operator, and might also cause users to churn to other network operators. In order for a better management of network resources at the network operator, it is important to understand the thresholds and circumstances for an application to be perceived at least within the acceptable quality levels. Understanding the perceived quality of applications and improving it, in addition to its benefits such as enabling lean network management (reducing costs) in the core network by setting the quality threshold levels, would also enable new opportunities and services/products that might eventually increase revenues of applica-

tion/network providers.

This thesis addresses smartphone QoE issues by first identifying them throughout comprehensive user study, and then studying them in depth, applying qualitative and quantitative methods. We provide suggestions on how the QoE can be improved or at least maintained from both network-level, application-level, and energy perspectives. Thus, this thesis contributes to the smartphone-related QoE in many aspects particularly in applications such as video streaming with the focus on temporal impairments as these highly influence both the QoE and the energy consumption of smartphones. We provide corresponding energy saving recommendations for smartphone-based video streaming. We also study the energy consumption while there is no interaction with the smartphone. We suggest rather simple sledgehammer solution (which can be deployed with a minimum effort) in order to decrease the energy consumption with the focus on 3G-based network activities, while the smartphone screen is at OFF state. The novelty of the studies is that we consider the QoE aspects during energy saving.

9.2 Concluding Remarks

This thesis can be concluded with the following items. Each item is presented with the corresponding contribution, *i.e.*, C.*x*.

- We present a comprehensive user study to identify the most influential factors on smartphone-based QoE. The study methodology consists of both qualitative and quantitative procedures. ESM, DRM, and online user survey enables us to collect subjective qualitative data; and autonomous logging through our software on the smartphones enable us to objectively collect quantitative device performance data in users' natural environments. As a result of the study, we have found many factors influencing QoE, which go beyond the usability, usefulness, well-being, hedonic, and other user value factors as indicated in the literature. We have found that battery efficiency, application performance, application interface design, phone features, application and connectivity cost, user routines, and user lifestyle are a set of factors influencing smartphone QoE. The role of QoS on QoE is also studied; and the increased SRT and RTT reduced the MOS values. We also found that the users are often well-connected and using smartphones in a fixed position and when they are alone. We spotted the battery life as a highly emphasized item by the users that influences the user-perceived quality. The temporal impairments during video streaming, *i.e.*, freezes, are identified as very important factors on the overall perceived quality of smartphone. This phenomenon might influence user decisions while choosing a particular network

operator as users demand video streaming more and more. We addressed the research questions R.Q.1 and R.Q.2.

- Given that video streaming applications might highly influence the QoE (particularly in the case of temporal impairments on smartphones) and in parallel might be highly power consuming, we studied further the temporal impairments during 3G-based video streaming on the smartphone. We focused on packet-level metrics such as PDV measured at the kernel level. We matched the objectively collected metrics with the subjectively collected MOS values on the user interface of the video player. We presented that the packet-level metrics can indicate a low QoE and the relation in-between the MOS and the PDV can be represented with a power-law model. The collected data is studied from the human perception aspect; we found that EWMA technique (that reveals the previous user experiences) improves the goodness-of-fit of the data by 100 %. On-off-flushing behaviour during the 3G-based video stream and its impact on the correlation between the PDV and the MOS values are presented. We also recommended the Maximal Burst Size (MBS) metric, which reveals (in a power-law model) the bursty traffic in 3G, in order to relate the long time intervals without any throughput to the user-perceived quality, *i.e.*, MOS. We addressed the research questions R.Q.3.1 and R.Q.3.2.

- In another video QoE study, we recorded metrics such as inter-picture time objectively measured at the user interface together with the corresponding QoE, and presented a set of outcomes. One of the most important outcomes of the study is that, to the best of our knowledge, we developed for the first time a cross-layer smartphone-based video streaming tool, open-source VLQoE, enabling to assess video QoE, with the consideration of various aspects such as application, network, user interface, and data collected from other sensors of the smartphone. VLQoE tool is provided as open-source (thus can be enhanced further), and also can be used in smartphone-based video QoE studies with its current version. We presented that, based on the comprehensive measurements conducted at various locations within Karlskrona city and particular experiment settings presented in Chapter 5, the inter-picture time of a 3G-based video stream can be represented via two-state exponential ON/OFF model. According to our experiment settings, we measured a mean OFF value of around 600 ms and mean ON value of around 10 s. Based on the results obtained from the user study, the mean of the maximum inter picture time values were within the second order of magnitude (and monotonically increasing) for user rating 5 to 1 (in five-level MOS scale). The mean value increases slightly higher to the third order of magnitude for the case

of user freeze indications. We measured the mean minimum perceived inter-picture time roughly around 300 ms, and the increase of this value causes the user rating values less than 4. We have spotted tolerable and impatient users; *i.e.*, some users do not respond immediately despite the high number of freeze alarms raised by the VLQoE, while some users are rather picky and give a low user rating even when there is a low number of freezes. We addressed the research questions R.Q.3.3 and R.Q.3.4.

- Battery life highly influences smartphone QoE, thus as video streaming applications are power-hungry, we study the energy consumption during video streaming. In order to accurately measure and reveal the anomalies via high precision, we used the Monsoon power monitoring tool. We compare the tool for a set of scenarios with a software-based measurement tool. We conclude that although the software-based power monitoring tools are non-intrusive and can be used in user studies to obtain high-level view on energy patterns, the collected measurements might not be accurate as the hardware-based external tools. Thus, we highly encourage researchers who are interested in collecting accurate measurements and prepare demonstrators to use Monsoon-like hardware-based power monitoring tool. We presented that the anomalies in QoS and QoE can be revealed via the (instantaneous) power consumption as the power consumption fluctuates (more precisely drops), when there is a video stalling quantified by the inter-picture time measured at the user interface via the VLQoE tool. We addressed the research questions R.Q.3.5, R.Q.3.6, and R.Q.4.

- Throughout extensive energy measurements on the smartphone, we identified potential scenarios where energy saving might be possible and we recommended approaches to increase energy saving while maintaining QoE. The suggested approaches can be categorized in two parts: (i) while user is *interacting* with an application, (ii) while the user is *not interacting* with an application. For a user interactive scenario, we evaluate a video streaming application from both the QoE and the energy consumption perspectives. We present that energy can be wasted or saved during temporal impairments, *i.e.*, freezes, depending on the network streaming protocol. In the case of a freeze during a video stream, if there is a packet retransmission (*e.g.*, initiated by TCP), then all the video content is shown to the user. During the freeze time, the energy is kept consuming due to a set of factors including CPU, AMOLED display, low power state of 3G data module. Thus, the energy consumed during the stall duration is considered as wasted. Therefore, a smoother video playout causes a lower total energy consumption and a higher QoE. In contrast, in UDP-based video stream, there is a

freeze jump, which causes some content to be skipped (*i.e.*, no packet retransmission). Thus, the total video streaming duration is unchanged (as compared to the original video). In that case, there is a tradeoff between the QoE and the energy consumption, *i.e.*, the stalling events both decrease QoE and the energy consumption. Thus, we present non-linear QoE models that relate to energy saving and the energy waste depending on the streaming protocol. This way, we calculate the maximum energy saving during a UDP-based video streaming on the smartphone while keeping the MOS at an acceptable level. Similarly, we quantify the amount of energy waste during a TCP-based video stream, and show that the energy waste is increasing exponentially with the decrease in MOS. We addressed the research question R.Q.5.1.

- Other video streaming scenarios on smartphone include amongst others the download-and-watch-later approach, *e.g.*, in the case of podcasts. Thus, we also study different scenarios while downloading files with different sizes. This way, we compare the scenarios such as downloading and local streaming versus the direct network streaming. Median power consumption during local streaming is 750 mW less as compared to the 3G-based video streaming. We measured the power consumption during only file downloading as 170 mW (median) less than direct 3G-based streaming. In overall, we present that the total energy consumption with the download-and-watch-later scenario yields less energy consumption as compared to direct network-based real-time video streaming. We conclude the reason as the video player limits the download rate based on the bit rate of the streamed video, however in the case of only-download, the file is being downloaded with the maximum available throughput. This reduces the download duration and thus minimizes the active duration of the 3G cellular data module. Another advantage is that the playout does not contain freezes as it is pre-downloaded. On the other hand, the energy saving during the download-and-watch-later approach comes with a tradeoff and one should sacrifice a possible QoE degradation caused by the initial waiting time until the video starts. Eventually, this also causes the total duration (initial download plus the playout time) to be higher than the direct streaming duration. We show that the file size highly influences the download duration and eventually the energy consumption. Scheduling of downloads for multiple files save energy. Based on the size of the files, the appropriate scheduling can be chosen. If small file sizes are being downloaded, then downloading them in series yields statistically significantly higher average processing time and total download time as compared to the hybrid (*i.e.*, sequence of parallel downloads) or parallel approach. Thus, sequential downloading is not suggested for the small size files. For large size files, sequential or

hybrid downloading is recommended (as compared to the concurrent downloading) due to the reduced download time and the reduced energy consumption. We addressed the research questions R.Q.5.2 and R.Q.5.3.

- We identified also the energy waste on the smartphone while a user is not interacting with a smartphone, *i.e.*, while the smartphone screen is OFF. The network-based applications create a signaling storm during screen OFF state, thus forcing the cellular data module to be always in an active (high power consuming) state. In order to address this issue, we first run an extensive user study and collect information from user smartphones on the application behaviour of users, *e.g.*, how often is the smartphone screen switched ON and OFF? We found out that the user interaction with the smartphone screen can be represented by a two state exponential model with mean screen ON duration of 20 minutes and mean screen OFF duration of 4 minutes. Then, based on per-user smartphone screen state pattern, the 3G cellular data module of smartphone is toggled in between ON and OFF states when screen is OFF by considering the following tradeoff. If the cellular data module is kept OFF too long, it will impact the QoE as the user might expect to receive information and notification (*e.g.*, email, instant message services) within a particular screen OFF period. On the other hand, we aim to extend the data OFF duration during screen OFF as much as possible to increase the energy saving. We applied per-user configuration, *i.e.*, by setting the appropriate dataON and dataOFF intervals, based on the personal interaction model with the smartphone. We measured potential energy savings with Monsoon power monitoring tool. We have achieved 34 % power gain with a 29 minutes dataON and 1 minute dataOFF cycle while maintaining the QoE on user smartphones. The novelty of the tool is it being both user-centric and also being a so-called sledgehammer implementation that enables easy deployment on any Android OS-based smartphone. We addressed the research question R.Q.6.

9.3 Future Work

A set of potential future directions can be drawn based on our conclusions and experiences obtained with the presented thesis work. First, we strongly encourage that new assessment methods and tools need to be developed for assessing the QoE while the users are using their mobile devices in the wild. Yet, we have developed assessment methods and tools, employed them with the focus on Android operating system as it is open-source and the availability of its extensive API's. The next step would be to develop similar tools for other operating systems including Windows OS and iOS. The

QoE concept has been increasing in popularity over the past years, due to the operators' increasing demands to use the existing network and energy resources as efficiently as possible. This in parallel causes operators to obtain (along with core QoS metrics including data rate, latency, or packet loss) reliable user feedback with respect to the services being provided. This necessitates the infrastructure industry to use robust QoE products and methods. Therefore, another future work includes QoE assessment by integrating the "Big Data" (detailed user behaviour and interaction with the applications on the smartphone, application specific metrics, as well as potential network KPI (Key Performance Indicators)), obtained from the the device sensors, application reports, as well as the network operator. In addition, the metrics can be complemented via a easy-to-use non-obtrusive in-situ feedback assessment interface, enabling the user to indicate the level of delight or annoyance.

Other future work includes further QoE and energy studies by considering the whole end-to-end system. Measurements and corresponding approaches to improve a quality of a service or to reduce the energy consumption at particular point within the network should consider a holistic evaluation. For example, improving the energy efficiency on the access network (*i.e.*, via separating the control plane from the signaling plane), *e.g.*, Radio Access Network (RAN), might on the other hand increase the energy consumption of mobile terminal as it might necessitate two separate simultaneously active wireless interfaces on the smartphone. There are other approaches regarding the migration of high power-hungry computations to the cloud. On the one hand, moving the computing work (*e.g.*, transcoding during video streaming) to the cloud to minimizes the processing power on energy-constrained mobile devices. This means that smartphone then need to download the uncompressed data from the cloud with a rather higher data volume. Thus, this might highly increase the wireless network based data activity on smartphone, which is highly energy consuming. Thus, extensive trade-off analysis and measurements need to be considered for the aim of the desired study before giving critical decisions for the design of a service or a product for mobile devices. We, as engineers, need to evaluate the sociological and the energy aspects of our proposed solutions. As the first item of IEEE code of ethics states [285]: "to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment".

Chapter 10

Bibliography

10.1 References

[1] G. E. Moore, "Cramming More Components Onto Integrated Circuits", *Proceedings of the IEEE*, vol.86, no.1, pp.82,85, Jan. 1998.

[2] Internet.org Project, [Online], Available at: http://internet.orgbyFacebook, Accessed: 2014.

[3] Ericsson Lab Report: Ericsson Traffic and Market Report (2012), [Online], Available at: http://www.ericsson.com/bd/res/docs/2012/traffic_and_market_report_june_2012.pdf, Accessed: 2014.

[4] IDC Worldwide Mobile Phone Tracker, [Online], Available at: http://www.idc.com/prodserv/smartphone-os-market-share.jsp, Accessed: 2014.

[5] Trends on Tuesday: Global Mobile Data Traffic Nearly Doubles in 2013, [Online], Available at: http://www.digitalgov.gov/2014/03/04/trends-on-tuesday-global-mobile-data-traffic-nearly-doubles-in-2013/, Accessed: 2014.

[6] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018, [Online], Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html, Accessed: 2014.

[7] S. Venugopal, "Scheduling Distributed Data-Intensive Applications on Global Grids", *Doctor of Philosophy Thesis*, Department of Computer Science and Software Engineering, The University of Melbourne, Australia, July 2006.

[8] Smartphone Battery Life and SoC Power Benchmarking, [Online], Available at: http://www.qualcomm.com/media/blog/2013/09/23/smartphone-battery-life-and-soc-power-benchmarking, Accessed: 2014.

[9] R. Bolla, R. Bruschi, F. Davoli, F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures", *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 223-244, 2011.

[10] A. Baumgartner, T. Bauschert. "Small Cells in UMTS Radio Access Networks: Implications on Coverage and Energy Efficiency", *22nd ITC Specialist Seminar on Energy Efficient and Green Networking (SSEEGN)*, 2013.

[11] FP7 EU STREP PERIMETER Project (2008 - 2011), [Online], Available at: http://www.ict-perimeter.eu, Accessed: 2014.

[12] A. Baumgartner, T. Bauschert, "Greening cellular radio access networks: A numerical method for the selection of detachable base stations in low traffic scenarios", *24th Tyrrhenian International Workshop on Digital Communications - Green ICT (TIWDC)*, pp. 1-6, 2013.

[13] A mobile application development primer, Thought leadership White Paper, IBM, 2012.

[14] A. K. Dey, "Understanding and Using Context", *Personal Ubiquitous Computing*. Vol.5(1), pp. 4-7, 2001.

[15] Qualinet White Paper on Definitions of Quality of Experience, *Dagstuhl seminar 12181*, Version 1.1, Dagstuhl, June 2012.

[16] ITU-T Report, Definition of Quality of Experience (QoE), 2007.

[17] A. Albasir, K. Naik, B. Plourde, N. Goel, "Experimental study of energy and bandwidth costs of web advertisements on smartphones", *6th International Conference on Mobile Computing, Applications and Services (MobiCASE)*, pp. 90-97, Nov. 2014

[18] N. V. Rodriguez, J. Crowcroft. “Energy Management Techniques in Modern Mobile Handsets”, *IEEE Communications Surveys and Tutorials*, 2012.

[19] R. B’Far, Mobile computing principles: designing and developing mobile applications with UML and XML. Cambridge, England: Cambridge University Press, 2004.

[20] S. Consolvo, M. Walker, “Using the Experience Sampling Method to Evaluate Ubicomp Applications”, *IEEE Pervasive Computing*, pp. 24-31, 2003.

[21] K. Kilkki, “Quality of Experience in communications ecosystem”, *Journal of Universal Computer Science*, vol.14, pp. 615-624, 2008.

[22] Q. Dai, “A survey of quality of experience”, *Energy-Aware Communications*, Lecture Notes in Computer Science, Springer, vol. 6955, pp. 146-156, Berlin, 2011.

[23] ETSI STF 354, [Online], Available at: `http://portal.etsi.org/stfs/STF_HomePages/STF354/STF354.asp`, Accessed: December, 2012.

[24] ITU-T Rec. E.800, Terms and definitions related to the quality of telecommunication services, 1995.

[25] K. Laghari, K. Connelly, “Toward total Quality of Experience: A QoE model in a communication ecosystem”, *IEEE Communications Magazine*, vol. 50, no. 4, pp. 58-65, 2012.

[26] M. Fiedler, S. Moller, and P. Reichl, “Quality of Experience: From user perception to instrumental metrics (Dagstuhl Seminar 12181)”, *Dagstuhl Reports*, 2(5), pp. 1-25, 2012.

[27] S. Afshari, N. Movahhedinia. “Non-Intrusive Online Quality of Experience Assessment for Voice Communications”, *Wireless Personal Communications*. Springer US. pp. 1-6. 2014.

[28] P. Reichl, D. Hausheer, B. Stiller, “The cumulus pricing model as an adaptive framework for feasible, efficient, and user-friendly tariffing of Internet services”, *Computer Networks*, vol. 43, Issue 1, pp. 3-24, 2003.

[29] ITU-T Rec. P.862, Perceptual evaluation of speech Quality(PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs, 2001.

[30] ITU-T Rec. P.800, Methods for subjective determination of transmission quality, 1996.

[31] ITU-T Rec. G.107, The E-model, a computational model for use in transmission planning, 2000.

[32] A. Yazdani, E. Kroupi, J. Vesin, T. Ebrahimi, "Electroencephalogram alterations during perception of pleasant and unpleasant odors", *4th International Workshop on Quality of Multimedia Experience (QoMEX2012)*, pp. 272-277, 2012.

[33] S. Arndt, J. Antons, R. Schleicher, S. Moller, G. Curio, "Perception of low-quality videos analyzed by means of electroencephalography", *4th International Workshop on Quality of Multimedia Experience (QoMEX2012)*, pp. 284-289, 2012.

[34] J. M. Gottman *et al.* , "The relationship between heart rate reactivity, emotionally aggressive behaviour, and general violence in batterers", *Journal of Family Psychology*, vol. 9, no. 3, pp. 227-248, 1995.

[35] W. Chen, B. Parrein, P. L. Callet, "A new H.264/AVC error resilience model based on Regions of Interest", *17th International Packet Video Workshop*, pp. 1-9, 2009.

[36] Y. Chen, S. Zhang, S. Xu, G. Y. Li, "Fundamental Trade-Offs on Green Wireless Networks". *IEEE Communications Magazine*, vol. 49. no. 6. pp. 30-37, 2011

[37] X. Zhang, J. Zhang, Y. Huang, W. Wang "On The Study Of Fundamental Trade-Offs Between QoE And Energy Efficiency In Wireless Networks", *Transactions on Emerging Telecommunications Technologies*, vol. 24. no. 3. pp. 259-265, 2013.

[38] The Skin Conductivity Response, [Online], Available at: `http://vismod.media.mit.edu/tech-reports/TR-468/node4.html`, Accessed: December, 2012.

[39] R. Schatz, T. Hossfeld, L. Janowski, S. Egger. "From Packets to People: Quality of Experience as a New Measurement Challenge", *Data Traffic Monitoring and Analysis*, vol. 7754. Springer Berlin Heidelberg, pp. 219-263, 2013.

[40] S. Chikkerur *et al.* , "Objective video quality assessment methods: A classification, review, and performance comparison", *IEEE Transactions on Broadcasting*, vol. 57, no. 2, pp. 165-182, 2011.

[41] S. Winkler, "Video quality measurement standards - Current status and trends", *7th International Conference on Information, Communications and Signal Processing (ICICS 2009)*, pp. 1-5, 2009.

[42] A. Khan, S. Lingfen, E. Ifeachor, "QoE Prediction Model and its Application in Video Quality Adaptation Over UMTS Networks", *IEEE Transactions on Multimedia*, vol.14, no.2, pp. 431-442, 2012.

[43] VQEG, [Online], Available at: http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx, Accessed: November, 2012.

[44] "Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment, Phase II", Video Quality Experts Group (VQEG), 2003.

[45] MOtion-based Video Integrity Evaluation (MOVIE) Index, Motion-based Perceptual Quality Assessment of Video. [Online], Available at: http://live.ece.utexas.edu/research/quality/movie.html, Accessed: November, 2012.

[46] H. E. Ross, "On the possible relations between discriminability and apparent magnitude", BritishJournalofMathematicalandStatisticalPsychology, vol.50, no. 2, Blackwell Publishing Ltd,pp. 187-203, 1997.

[47] J. Christian, V. D. B. Lambrecht, "Color moving pictures quality metric", *International Conference on Image Processing*, vol. 1, pp. 885-888, 1996.

[48] A. M. Eskicioglu and P. S. Fisher. "Image quality measures and their performance", *IEEE Trans. Communications*, vol. 43, pp. 2959-2965, 1995.

[49] A. Richards, G. Rogers, M. Antoniades, V. Witana. "Mapping User Level QoS from a Single Parameter," *MMNS 98*, Versailles, France, 1998.

[50] Ericsson Mobility Report (2014), [Online], Available at: http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf, Accessed: 2014.

[51] H. R. Sheikh and A. C. Bovik, "A Visual Information Fidelity Approach to Video Quality Assessment", *First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2005.

[52] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement", *Signal Processing: Image Communication, Special Issue on Objective Video Quality Metrics*, vol. 19, no. 2, pp. 121-132, 2004.

[53] ITU-T J.247, Objective perceptual multimedia video quality measurement in the presence of a full reference, 2008.

[54] ITU-T G.1030. Estimating end-to-end performance in IP networks for data applications. 2005.

[55] ITU-R BT.500-11, Methodology for the subjective assessment of the quality of television pictures, 2002.

[56] M. Hirth, T. Hossfeld, P. Tran-Gia, "Cost-optimal validation mechanisms and cheat-detection for crowdsourcing platforms", *5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2011)*, pp. 316-321, 2011.

[57] C. Keimel, J. Habigt, K. Diepold, "Challenges in crowd-based video quality assessment", *4th International Workshop on Quality of Multimedia Experience (QoMEX2012)*, pp. 13-18, 2012.

[58] B. L. Jones, P. R. McManus, Graphic Scaling of Qualitative Terms, 95:(11), pp. 1166-1171, 1986.

[59] M. T. Virtanen, N. Gleiss, M. Goldstein, "On the use of evaluative category scales on telecommunications", *Human Factors in Telecommunications*, 1995.

[60] K. Teunissen, "The validity of CCIR quality indicators along a graphical scale", *SMPTE Journal*, pp. 144-149, 1996.

[61] A. Watson, M. A. Sasse, "Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications", *ACM Multimedia Conference*, pp. 55-60, 1998.

[62] W. E. Howden, Functional Program Testing and Analysis. New York: McGraw-Hill, 1987.

[63] B. Beizer, Software Testing Techniques. New York: Van Nostrand, 1983.

[64] T. Hossfeld, R. Schatz, and S. Egger, "SOS: The MOS is not enough!", *3rd International Workshop on Quality of Multimedia Experience (QoMEX2011)*, pp. 131-136, 2011.

[65] S. Intille, C. Kukla, X. Ma. "Eliciting user preferences using image-based experience sampling and reflection", *Human Factors in Computing Systems*, CHI, USA, 2002.

[66] J. M. Hektner *et al.*, Experience Sampling Method: Measuring the Quality of everyday life, Sage Publications Inc., 2006.

[67] S. Ickin, K. Wac, M. Fiedler, L. Janowski, J. H. Hong, and A. K. Dey. "Factors influencing Quality of Experience of commonly-used mobile applications", *IEEE Communications Magazine*, vol. 50, no. 4, pp. 48-56, 2012.

[68] QoE of Mobile Services: Can it be measured and improved?, Nokia Corporation Networks, *White paper* , 2004.

[69] M. Fiedler, T. Hossfeld, P. Tran-Gia, "A generic quantitative relationship between Quality of Experience and Quality of Service", *Network Manamagement of Global Internetworking*, 24, 2, pp. 36-41, 2010.

[70] Fisher's Exact Test, [Online], Available at: `http://mathworld.wolfram.com/FishersExactTest.html`, Accessed: 2014.

[71] T. Hossfeld, D. Hock, P. Tran-Gia, K. Tutschku, M. Fiedler, "Testing the IQX hypothesis for exponential interdependency between QoS and QoE of voice codecs iLBC and G.711", *University of Wurzburg Institute of Computer Science Research Report Series-442*, 2008.

[72] P. Reichl, S. Egger, R. Schatz, A. D'Alconzo: "The Logarithmic Nature of QoE and the Role of the Weber-Fechner Law in QoE Assessment", *IEEE International Conference on Communications*, pp. 1-5, 2010.

[73] A. Raake, "Speech Quality of VoIP: Assessment and Prediction", *John Wiley & Sons*, 2006.

[74] S. Daly, "The visible differences predictor: an algorithm for the assessment of image fidelity", *Digital images and Human Vision*, pp. 179-206. MIT Press, Cambridge, 1993.

[75] E. Diener, L. Tay. "Review of the Day Reconstruction Method (DRM)", *Social Indicators Research*, Springer Netherlands, pp. 1-13, 2013.

[76] Oxford Dictionaries. [Online], Available at: `http://www.oxforddictionaries.com/definition/english/energy`, Accessed: 2013.

[77] N. V. Rodriguez, J. Crowcroft, "The Case for Context-Aware Resources Management in Mobile Operating Systems", *Mobile Context Awareness*, Springer London. pp. 97-113, 2012.

[78] L. Wang, J. Manner, "Energy Consumption Analysis of WLAN, 2G and 3G interfaces", *GreenCom'10 and CPSCom'10*, vol., no., pp. 300 - 307, 2010.

[79] J. Huang *et al.* , "Screen-off traffic characterization and optimization in 3G/4G networks", *IMC '12*, ACM, NY, USA, 357-364, 2012.

[80] X. Fengyuan *et al.* , "Optimizing Background Email Sync on Smartphones", *11th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 55-68, ACM, 2013.

[81] N. Balasubramanian, A. Balasubramanian, A.Venkataramani, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications", *9th ACM SIGCOMM Conference on Internet Measurement Conference*, pp. 280-293, 2009.

[82] JuiceDefender, [Online], Available at: `https://play.google.com/store/apps/details?id=com.latedroid.juicedefender`, Accessed: 2014.

[83] Auto3GSaver, [Online], Available at: `https://play.google.com/store/apps/details?id=com.slv3r.auto3gPro`, Accessed: 2014.

[84] EnergySaver, [Online], Available at: `https://play.google.com/store/apps/details?id=com.uhopro.energysaver`, Accessed: 2014.

[85] Battery Saver, [Online], Available at: `https://play.google.com/store/apps/details?id=com.comodo.batterysaver`, Accessed: 2014.

[86] BlueFi, [Online], Available at: `https://play.google.com/store/apps/details?id=com.okean.btcom&hl=en`, Accessed: 2014.

[87] K. P. Athivarapu *et al.* , "RadioJockey: Mining Program Execution to Optimize Cellular Radio Usage", *18th Annual International Conference on Mobile Computing and Networking*, ACM, pp. 101-112, 2012.

[88] Ericsson Mobility Report (2013), [Online], Available at: `http://www.ericsson.com/res/docs/2013/ericsson-mobility-report-november-2013.pdf`, Accessed: 2014.

[89] S. Baskaran, P. Thambidurai, "A Dynamic Slack Management Technique for Real-Time System with Precedence and Resource Constraints", *Advances in Computing and Information Technology*, Springer Berlin Heiderberg, vol.198, pp. 365-374, 2011.

[90] Social Media's Affect on Human Interaction, [Online], Available at: `https://www.hastac.org/blogs/`, Accessed: 2014.

[91] Tap Project, [Online], Available at: http://tap.unicefusa.org/, Accessed: 2014.

[92] Your Social Life Is Not Your Social Media, [Online], Available at: http://www.psychologytoday.com/blog/our-gender-ourselves/, Accessed: 2014.

[93] G. P. Perrucci *et al.* , "On the impact of 2G and 3G network usage for mobile phones' battery life", EW. European Wireless Conference, vol., no., pp.255, 259, 2009.

[94] Birth, life and death of an app, [Online], Available at: https://www.adjust.com/assets/downloads/AppleAppStore_Report2014.pdf, Accessed:2012.

[95] I. Grigorik, "High Performance Browser Networking: What every web developer should know about networking and web performance", O'Reilly Media, Inc, 2013.

[96] H. Holma, A. Toskala, "WCDMA for UMTS : HSPA evolution and LTE. Chichester", Wiley InterScience (Online service) Fifth Edition, 2010.

[97] "Smartphone Battery Life has Become a Significant Drain on Customer Satisfaction and Loyalty", J. D. Power and Associates Reports, 2012.

[98] A. Shye *et al.* , "Into the wild: studying real user activity patterns to guide power optimizations for mobile architectures", *Micro-42*, ACM, USA, 168-178, 2009.

[99] A. Carroll, G. Heiser, "An analysis of power consumption in a smartphone", USENIXATC'10. USENIX Association, Berkeley, CA, USA, 21-21, 2010.

[100] A. Dix *et al.* , *Human Computer Interaction*. Prentice Hall, 2004.

[101] R. Duan, B. Mingsong, C. Gniady, "Exploring memory energy optimizations in smartphones", *International Green Computing Conference and Workshops (IGCC '11)*, IEEE Computer Society, pp. 1-8, USA, 2011.

[102] K. Salamatian and S. Fdida, "Measurement based modelling of Quality of Service in the Internet: A methodological approach", *Workshop on Digital Communications*, pp. 158-174, 2001.

[103] Z. Jaroucheh, X. Liu, S. Smith, "Recognize contextual situation in pervasive environments using process mining techniques", *Journal of Ambient Intelligence and Humanized Computing, Springer-Verlag*, vol. 2, no. 1, pp. 53-61, 2010.

[104] H. Korhonen *et al.* , "Analysing user experience of personal mobile products through contextual factors", *ACM ICMUM*, pp. 1-10, 2010.

[105] M. Hassenzahl, N. Tractinsky, "User experience-A research agenda", *Behaviour and Information Technology, 2006.*

[106] J. Park *et al.* , "Developing elements of user experience for mobile phones and services", *Human Factors and Ergonomics in Manufacturing & Service Industries*, 2011.

[107] Y. Shin *et al.* , "Examining influencing factors of post-adoption usage of mobile Internet", *Information Systems Frontiers*, vol. 12, no. 5, pp. 595-606, 2011.

[108] ITU-R. BS.1534-1, Method for the subjective assessment of intermediate quality level of coding systems, 2003.

[109] P. Reichl *et al.* , "The Liliput Prototype: A wearable test environment for mobile telecommunication applications", *Proc. CHI 2007*, pp. 1833-1838, 2007.

[110] K. Wac *et al.* , "Studying the experience of mobile applications used in different context of daily life", *ACM SIGCOMM Workshop W-MUST*, pp. 7-12, 2011.

[111] D. Kahneman *et al.* , "A survey method for characterizing daily life experience: The Day Reconstruction Method", *Science*, vol. 306 no. 5702 pp. 1776-1780, December 2004.

[112] S. Barakovic, L. S.-Kapov, "Survey and Challenges of QoE Management Issues in Wireless Networks", Journal of Computer Networks and Communications, vol. 2013, Article ID 165146, 28 pages, 2013.

[113] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2012-2017. [Online], Available at: `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html`, Accessed: September, 2013.

[114] S. Ickin, M. Fiedler, K. Wac, P. Arlos, C. Temiz, K. Mkocha, "VLQoE: Video Quality of Experience Instrumentation on the Smartphone", *Multimedia Tools and Applications. Special Issue on Advances in Tools, Techniques and Practices for Multimedia QoE*, Springer US, vol.74/2, pp.381-411, 2014.

[115] T. Hossfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia. "Quantification of YouTube QoE via Crowdsourcing", *IEEE International Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions (MQoE 2011)*, USA, 2011.

[116] T. Hossfeld, R. Schatz, E. Biersack, L. Plissonneau, “Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience”, Data Traffic Monitoring and Analysis Lecture Notes in Computer Science Volume 7754, Springer Berlin Heidelberg, pp 264-301, 2013.

[117] N. Staelens *et al.* , “Assessing Quality of Experience of IPTV and Video on Demand Services in Real-Life Environments”, *IEEE Transactions on Broadcasting*, vol.56, no.4, pp. 458-466, 2010.

[118] Quality of Service, [Online], Available at: `http://www.etsi.org/technologies-clusters/technologies/quality-of-service`, Accessed: 2014.

[119] M. Fiedler *et al.* , “QoE-based Cross-Layer Design of Mobile Video Systems: Challenges and Concepts”, *International Conference on Computing and Communication Technologies, RIVF ’09*, Vietnam, 2009.

[120] M. Fiedler. “Traffic Models for Quality of Experience Assessment”, *Tutorial: 23rd International Teletraffic Congress*, USA, 2011.

[121] VLC Media Player for Android. [Online], Available at: `http://www.videolan.org/vlc/download-android.html`, Accessed: March, 2013.

[122] RFC2326: Real Time Streaming Protocol (RTSP). [Online], Available at: `http://www.ietf.org/rfc/rfc2326.txt`, Accessed: September, 2013.

[123] HTTP Live Streaming draft-pantos-http-live-streaming-02. [Online], Available at: `http://tools.ietf.org/pdf/draft-pantos-http-live-streaming-02.pdf`, Accessed: September, 2013

[124] T. Bonald and J. W. Roberts. “Internet and the Erlang formula”, *SIGCOMM Comput. Commun. Rev.* 42, 1, pp. 23-30, 2012.

[125] D. Katabi, M. Handley, and C. Rohrs. “Congestion control for high bandwidth-delay product networks”, *Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM ’02)*, ACM, USA, 89-102, 2002.

[126] T. Hossfeld, R. Schatz, E. Biersack, L. Plissonneau. “Internet Video Delivery in YouTube: From Traffic Measurements to Quality of Experience”, *Data Traffic Monitoring and Analysis, Lecture Notes in Computer Science*, Volume 7754, pp. 264-301, 2013.

[127] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl. "Anatomizing application performance differences on smartphones", *8th international conference on Mobile systems, applications, and services (MobiSys '10)*, ACM, USA, pp. 165-178, 2010.

[128] Dialogic Corporation Quality of Experience for Mobile Video Users, White Paper, Canada, 2009.

[129] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. "Multimedia streaming via TCP: An analytic performance study", *ACM Trans. Multimedia Comput. Commun. Appl.* 4, 2, Article 16, 22 pages, 2008.

[130] Y. Wang and M. Claypool. "RealTracer - tools for measuring the performance of RealVideo on the Internet", *Kluwer Multimedia Tools and Applications*, vol. 27, no. 3, 2005.

[131] M. Li, F. Li, M. Claypool, R. Kinicki. "Weather forecasting - predicting performance for streaming video over wireless LANs", *NOSSDAV*, Stevenson, Washington, 2005.

[132] K Yang, C. C. Guest, K. El-Maleh, P. K. Das, "Perceptual Temporal Quality Metric for Compressed Video", *IEEE Transactions on Multimedia*, vol.9, no.7, pp. 1528,1535, 2007.

[133] H. T. Quan, M. Ghanbari. "Temporal aspect of perceived quality of mobile video broadcasting",*IEEE Trans. Broadcasting*, 54 (3), pp. 641 - 651, 2008.

[134] Top Project Listings, [Online], Available at: https://www.videolan.org/vlc/stats/downloads.html, Accessed: 2013.

[135] S. Van Kester, T. Xiao, R. E. Kooij, K. Brunnstrm, O. K. Ahmed. "Estimating the Impact of Single and Multiple Freezes on Video Quality", *SPIE'11*, vol. 7865, 2011.

[136] S. Egger, T. Hossfeld, R. Schatz, M. Fiedler. "Waiting times for Quality of Experience for web based services", *4th International Workshop on Quality of Multimedia Experience*, 2012.

[137] W. Cherif, A. Ksentini, D. Negru, M. Sidibe. "APSQA: Efficient Real - Time Video Streaming QoE Tool in a Future Media Internet Context", *IEEE Conference in Multimedia and Expo (ICME)*, pp. 1 - 6, 2011.

[138] V. Menkovski, G. Exarchakos, and A. Liotta, "The value of relative quality in video delivery", *Journal of Mobile Multimedia*, 7, 3, pp. 151-162, September 2011.

[139] T. Zinner, O. Abboud, O. Hohlfeld, T. Hossfeld, P. Tran Gia, "Towards QoE Management for Scalable Video Streaming", *21st ITC Specialist Seminar on Multimedia Applications, Traffic, Performance and QoE*, Japan, 2010.

[140] T. Hosfeld, M. Fiedler, T. Zinner, "The QoE provisioning-delivery-hysteresis and its importance for service provisioning in the Future Internet", *7th EURO-NGI Conference on Next Generation Internet (NGI)* , vol., no., pp.1,8, 2011.

[141] A. Vuppala, L. N. Sriram, "Measurement of user related performance problems of live video streaming in user interface", *Masters Thesis*, Blekinge Institute of Technology, 2011.

[142] B. Staehle, M. Hirth, R. Pries, F. Wamser, D. Staehle. "YoMo: A Youtube Application Comfort Monitoring Tool", *EuroITV Workshop QoE for Multimedia Content Sharing (QoEMCS'10)*, Tampere, Finland, 2010.

[143] P. Juluri, L. Plissonneau, D. Medhi. "Pytomo: A tool for analyzing playback quality of YouTube videos", *23rd International Teletraffic Congress (ITC)*, vol., no., pp.304,305, 6-9, 2011.

[144] A. Vishwanath, P. Dutta, M. Chetlur, P. Gupta, S. Kalyanaraman, A. Ghosh. "Perspectives on Quality of Experience for Video Streaming over WiMAX", *ACM SIGMOBILE Mobile Computing and Communications Review*, Volume 13, Number 4, pp. 15-25, 2010.

[145] P. Ameigeiras, and Juan J. Ramos-Munoz, J. Navarro-Ortiz, J.M. Lopez-Soler, "Analysis and modelling of YouTube traffic". *Transactions on Emerging Telecommunications Technologies*, vol.23(4). pp. 360–377. John Wiley & Sons, Ltd. 2012.

[146] PEVQ Perceptual Evaluation of Video Quality. [Online], Available at: `http://www.pevq.org/`, Accessed: March, 2013.

[147] D. Migliorini, E. Mingozzi, C. Vallati. "QoE - Oriented Performance Evaluation of Video Streaming over WiMAX", *8th International Conference on Wired/Wireless Internet Communications*, Sweden, 2010.

[148] T. Minhas. Network impact on Quality of Experience of mobile video, *Licentiate Dissertation*, Blekinge Institute of Technology, Sweden, 2012.

[149] V. Menkovski. “Intelligent control for adaptive video streaming”, *International Conference on Consumer Electronics*, USA, 2013.

[150] S. Latre and F. De Turck. “Autonomic quality of experience management of multimedia networks”, *IEEE IFIP Network Operations and Management Symposium*, p.872-879, 2012.

[151] I. J. Myung, “Tutorial on maximum likelihood estimation”, *Journal of Math. Psychol*, 47, 1, pp. 90-100, 2003.

[152] FreeBSD Man Pages (Procfs), [Online], Available at: `https://www.freebsd.org/cgi/man.cgi?query=procfs&sektion=5`, Accessed: 2014.

[153] Android Developers (BatteryManager), [Online], Available at: `http://developer.android.com/reference/android/os/BatteryManager.html`, Accessed: 2014.

[154] ArchLinux, (upower 0.99.2-2), [Online], Available at: `https://www.archlinux.org/packages/extra/x86_64/upower/`, Accessed: 2014.

[155] S. Winkler and P. Mohandas. “The evaluation of Video Quality Measurement: From PSNR to Hybrid Metrics”, *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660-668, 2008.

[156] ITU-T Recommendation P.910. Subjective video quality assessment methods for multimedia applications, Recommendations of the ITU (Telecommunication Standardization Sector), [Online], Available at: `http://www.itu.int/rec/T-REC-P.910-200804-I/en`, Accessed: September, 2013

[157] S. C. Seow. “Designing and engineering time: The psychology of time perception in software”, *Addison-Wesley Professional*, 2008.

[158] Voice quality and MOS. [Online], Available at: `http://searchunifiedcommunications.techtarget.com/tip/Voice-quality-and-MOS`, Accessed: April, 2013.

[159] R. P. Vidal, J. Gicquel, C. Colomes, and H. Cherifi. Sporadic frame dropping impact on quality perception, Human Vis. Electron. Imaging IX, vol. 5292, 2004.

[160] S. Ickin, K. Wac, M. Fiedler. “Demonstrating the Stalling Events with Instantaneous Total Power Consumption in Smartphone-based Live Video Streaming”, *2nd IFIP Conference on Sustainable Internet and ICT for Sustainability*, Italy, 2012.

[161] S. Ickin, L. Janowski, K. Wac, M. Fiedler. “Studying the challenges in assessing the perceived quality of mobile-phone based video”, *4th International Workshop on Quality of Multimedia Experience (QoMEX)*, pp.164,169, 2012.

[162] G. Musser. Time on the Brain: How You Are Always Living In the Past, and Other Quirks of Perception. [Online], Available at: `http://blogs.scientificamerican.com/observations/2011/09/15/`, Accessed: April, 2013.

[163] S. Grondin. “Timing and time perception: a review of recent behavioral and neuroscience findings and theoretical directions”, *Attention perception psychophysics*, vol. 72, no. 3, pp. 561 - 582, 2010.

[164] VLC Media Player for Android. [Online], Available at: `http://www.videolan.org/vlc/download-android.html`, Accessed: March, 2013.

[165] R. Schatz, T. Hossfeld, L. Janowski, S. Egger. From Packets to People: Quality of Experience as a New Measurement Challenge, Data Traffic Monitoring and Analysis”, Lecture Notes in Computer Science, Volume 7754, 219-263, 2013.

[166] C. Fenimore, J. Libert. Perceptual Effects of Noise in Digital Video Compression. [Online], Available at: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.196.1791&rep=rep1&type=pdf`, Accessed: September, 2013.

[167] P. De la C. Ramos,F. G. Vidal, R. P. Leal. “Perceived Video Quality Estimation from Spatial and Temporal Information Contents and Network Performance Parameters in IPTV”, *5th International Conference onDigital Telecommunications (ICDT)*, 2010.

[168] E. Oliver. “The challenges in large-scale smartphone user studies”, *2nd ACM International Workshop on Hot Topics in Planet-scale Measurement*, ACM, USA, Article 5, 5 pages, 2010.

[169] T. Hossfeld *et al.* , “Quantification of YouTube QoE via crowdsourcing”, *Multimedia (ISM), 2011 IEEE International Symposium*, pp. 494-499, 2011.

[170] Power Monitor, [Online], Available at `http://www.msoon.com/LabEquipment/PowerMonitor/`, Accessed: December, 2010.

[171] S. Ickin, K. D. Vogeleer, M. Fiedler, D. Erman, “The effects of Packet Delay Variation on the perceptual quality of video”, *35th IEEE Conference on Local Computer Networks* (LCN2010), pp. 663-668, 2010.

[172] PowerTutor. [Online]. Available at: http://powertutor.org. Accessed: May, 2012.

[173] A. A. Nacci *et al.* , Adaptive And Flexible Smartphone Power Modeling. Mobile Networks and Applications, Springer US, 2013.

[174] Darwin Streaming Server. [Online]. Available at: http://developer.apple.com/opensource/server/streaming, Accessed: May, 2012.

[175] A. Carroll, G. Heiser. "An analysis of power consumption in a smartphone", *USENIX conference on USENIX annual technical conference (USENIXATC)*, USENIX Association, USA, 2010.

[176] T. Manning-Dahan *et al.* , "Measuring the Power Consumption of Smartphones", Technical Report, March, 2012.

[177] Galaxy S Standard Battery. [Online]. Available at: http://www.samsung.com/us/mobile/cell-phones-accessories/EB575152VABSTD, Accessed: April, 2012.

[178] D. McMillan, A. Morrison, O. Brown, M. Hall, M. Chalmers. "Further into the wild: Running worldwide trials of mobile systems", Pervasive Computing, pg 210-227. 2010.

[179] H. Falaki *et al.* , "Diversity in smartphone usage," *8th international conference on Mobile systems, applications, and services (MobiSys '10)*, ACM, NY, USA, pp. 179-194, 2010.

[180] J. Manweiler, R. R. Choudhury, "Avoiding the rush hours: WiFi energy management via traffic Isolation", *IEEE Transactions on Mobile Computing*, vol.11, no.5, pp. 739, 752, 2012.

[181] S.Ickin, T. Zinner, K. Wac, M. Fiedler. "Catching the Download Train: Energy-efficient File Downloading On Smartphones", *26th IEEE International Teletraffic Congress (ITC)*, Sweden. 2014.

[182] A. Carroll, G. Heiser, "An analysis of Power Consumption in a Smartphone," USENIX Conference on USENIX Annual Technical Conference (USENIXATC'10), pp. 271–284, Berkeley, CA, USA, 2010.

[183] I. Grigorik, "High Performance Browser Networking: What every web developer should know about networking and web performance", *O'Reilly Media, Inc.*, 2013.

[184] Behavior Analysis of Smartphone. [Online], Available at: http://www.huawei.com/en/static/hw-001545.pdf, Accessed: March, 2014.

[185] J. Huang *et al.* , "A close examination of performance and power characteristics of 4G LTE networks", *10th international conference on mobile systems, applications, and services (MobiSys '12)*, ACM, USA, pp. 225-238, 2012.

[186] X. Chen *et al.* , "How is Energy Consumed in Smartphone Display Applications?", *14th Workshop on Mobile Computing Systems and Applications (HotMobile '13)*. ACM, New York, NY, USA, , Article 3 , 6 pages.

[187] GreenTouch, [Online], Available at: http://www.greentouch.org/, Accessed: 2014.

[188] Ericsson Consumer Lab, "Interactivity Beyond the Screen", [Online], Available at: http://www.slideshare.net/Ericsson/interactivity-beyondthescreen, Accessed: 2014.

[189] GreenTouch, "Green Meter Research Study: Reducing the Net Energy Consumption in Communications Networks by up to 90% by 2020", [Online], Available at: http://www.greentouch.org/uploads/documents/GreenTouch_Green_Meter_Research_Study_26_June_2013.pdf, Accessed: 2014.s

[190] Digit Precision Multimeters, [Online], Available at: http://us.flukecal.com/products/data-acquisition-and-test-equipment/bench-multimeters/8845a8846a-65-digit-precision-multime?quicktabs_product_details=3, Accessed: 2014.

[191] P. A. Frensch, "Composition during serial learning: a serial position effect", *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(2), 423-443, 1994.

[192] J. A. Gibbons, S. A. Lee, W. R. Walker, "The fading affect bias begins within 12 hours and persists for 3 months, *Applied Cognitive Psychology*, 25(4), 663-672, 2010.

[193] E. J. Vergara *et al.* , "Kernel level energy-efficient 3G background traffic shaper for android smartphones", *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, vol., no., pp.443, 449, 2013.

[194] A. Sharma *et al.* , "Cool-Tether: energy efficient on-the-fly wifi hot-spots using mobile phones", *5th international conference on Emerging networking experiments and technologies (CoNEXT '09)*, ACM, USA, pp. 109-120, 2009.

[195] A. Blenk *et al.* , "Dynamic HTTP download scheduling with respect to energy consumption," *24th Tyrrhenian International Workshop on Digital Communications (TIWDC)*, Italy, 2013.

[196] S. Ickin, K. Wac, M. Fiedler. "Quality of Experience-Based Energy Reduction by Controlling the 3G Cellular Data Traffic on the Smartphone", *22nd International Teletraffic Congress Specialist Seminar on Energy Efficient and Green Networking (ITC SSEEGN)*, New Zealand, 2013.

[197] C. Kirkpatrick, J. R. Dahlquist, "Technical Analysis: The Complete Resource for Financial Market Technicians", *Pearson Education*, Inc., pp. 276, 2011.

[198] ITU-T Rec. E.800, Definitions of terms related to Quality of Service, 2008.

[199] M. Hirth, T. Hossfeld, P. Tran-Gia, "Anatomy of a crowdsourcing platform - Using the example of microworkers.com", *5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS2011)*, pp. 322-329, 2011.

[200] J. L. Fleiss, "Measuring nominal scale agreement among many raters", *Psychological Bulletin*, vol. 76, no. 5, pp. 378-382, 1971.

[201] S. Egger, T. Hossfeld, R. Schatz and M. Fiedler, "Tutorial: waiting times in Quality of Experience for web based services", *QoMEX'12*, Australia, 2012.

[202] F. Qian *et al.* , "Characterizing radio resource allocation for 3G networks", *IMC'10*, USA, pp. 137-150, 2010.

[203] H. Holma, A. Toskala, "HSDPA/HSUPA for UMTS: High Speed Radio Access for Mobile Communications", *John Wiley & Sons*, 2006.

[204] A. Gerber, S. Sen, O. Spatscheck, "A Call for More Energy-Efficient Apps", AT&T Labs Research, April 2011.

[205] V. Bernardo, T. Braun, M. Curado, M. Fiedler, D. Hock, T. Hossmann, K. A. Hummel, P. Hurni, S. Ickin, A. Jamakovic, S. Nadjm-Tehrani, T. Trinh, E. J. Vergara, F. Wamser, T. Zinner, Green Wireless-energy efficiency in Wireless Networks, *Book Chapter, John Wiley & Sons*, 2015.

[206] S. Ickin, M. Fiedler, K. Wac, "Energy-based anomaly detection in quality of experience", *16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, vol., no., pp.1-6, USA. 2013.

[207] X. Wei, L. Gomez, I. Neamtiu, M. Faloutsos, "ProfileDroid: Multi-layer profiling of android applications", *18th annual international conference on Mobile computing and networking (Mobicom '12)*, ACM, New York, NY, USA, pp. 137-148, 2012.

[208] E. Oliver, "Diversity in smartphone energy consumption", S3'10. ACM, NY, USA, pp. 25-28, 2010.

[209] N. Banerjee *et al.* , "Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems", *UbiComp'07: Ubiquitous Computing Lecture Notes in Computer Science*, Springer Berlin Heidelberg, vol. 4717, pp. 217-234, 2007.

[210] The New Multi-screen World: Understanding Cross-platform Consumer Behavior, Google, 2012.

[211] C. D. Lai, D. N. Murthy, M. Xie, Weibull Distributions and Their Applications. *Springer Handbook of Engineering Statistics*, pp. 63-78, 2006.

[212] E. J. Vergara *et al.* , "Kernel level energy-efficient 3G background traffic shaper for android smartphones", *IWCMC'13*, vol., no., pp.443,449, 2013.

[213] F. Qian *et al.* , "TOP: Tail Optimization Protocol for cellular radio resource allocation," *ICNP'10*, Japan, pp. 285-294, 2010.

[214] Notification Center, Available at: `http://gigaom.com/2011/10/12/ios-5-notifications-and-notification-center`, Accessed: May, 2013.

[215] D. O'Reilly, "How to silence notifications on smartphones and tablets", [Online], Available at: `http://howto.cnet.com/8301-11310_39-57551600-285/how-to-silence-notifications-on-smartphones-and-tablets/`, Accessed: June, 2013.

[216] AwayFind, [Online], Available at: `http://www.awayfind.com/`, Accessed: June 2013.

[217] D. Chu *et al.* , "Mobile apps: it's time to move up to CondOS", *HotOS'13*, USENIX Association, USA, 16, 2013.

[218] R. N. Valina *et al.* , "ErdOS: achieving energy savings in mobile OS", MobiArch '11. ACM, USA, pp. 37-42, 2011.

[219] R. D. Edwards and J. Magee, Technical Analysis of Stock Trends, *8th edition AMACOM*, 2001.

[220] B. Shneiderman, C. Plaisant, Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition), 2004.

[221] W. H. Davidow, "Our tools are using us", *The Magazine of Technology Insiders IEEE Spectrum*, pp. 45-48, August 2012.

[222] J. M. Hektner *et al.* , "Experience Sampling Method: Measuring the quality of everyday life", Sage Publications Inc., 2006.

[223] By the Numbers: 104 Amazing Twitter Stats, [Online], Available at: http://expandedramblings.com/index.php/march-2013-by-the-numbers-a-few-amazing-twitter-stats/#.UzvGua2SzmY, Accessed: 2014.

[224] Shapira *et al.* , Problematic Internet Use: Proposed Classification and Diagnostic Criteria, Depression and Anxiety, 17:201-216, 2003.

[225] J. Ugander, The Anatomy of the Facebook Social Graph. CoRR, [Online], Available at: http://arxiv.org/pdf/1111.4503v1.pdf, Accessed: 2011.

[226] AftonBladet, [Online], Available at: https://play.google.com/store/apps/details?id=se.aftonbladet.start&hl=en, Accessed: June 2013.

[227] A. Carroll and G. Heiser, "An analysis of Power Consumption in a Smartphone," *USENIX Conference on USENIX Annual Technical Conference (USENIXATC'10)*, pp. 271–284, Berkeley, CA, USA, 2010.

[228] J. Huang *et al.* , "RadioProphet: Intelligent radio resource deallocation for cellular networks," *Passive and Active Measurement*, Springer International Publishing, vol. 8362, pp. 1-11, 2014.

[229] G. P. Perrucci, "On the impact of 2G and 3G network usage for mobile phones' battery life," *EW. European Wireless Conference*, pp. 255-259, 2009.

[230] E. J. Vergara *et al.* , "Kernel level energy-efficient 3G background traffic shaper for android smartphones," *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 443-449, 2013.

[231] SPeeDY, [Online], Available: https://developers.google.com/speed/spdy.

[232] A. Sharma *et al.* , "Cool-Tether: energy efficient on-the-fly wifi hot-spots using mobile phones," *5th international conference on Emerging networking experiments and technologies (CoNEXT '09)*, ACM, USA, pp. 109-120. 2009.

[233] A. Blenk *et al.* , "Dynamic HTTP download scheduling with respect to energy consumption", *24th Tyrrhenian International Workshop on Digital Communications (TIWDC)*, Italy, 2013.

[234] Apple. (Online: Verified May, 2012). Available at `http://developer.apple.com/opensource/server/streaming`.

[235] P. Arlos, On the quality of computer network measurements, *Doctoral Dissertation*, Blekinge Institute of Technology, 2005.

[236] Z. Chen and H. Nakazato, A model for time varying quality of speech services, *IEEE Globecom*, pp. 240-244, 2005.

[237] M. Claypool and J. Tanner, The effects of jitter on the perceptual quality of video, *ACM Multimedia Conference*, pp. 115-118, 1999.

[238] C. Demichelis and P. Chimento, IP Packet Delay Variation Metric for IP Performance Metrics (IPPM), RFC 3393, 2002.

[239] M. Fiedler, T. Hossfeld, and P. Tran-Gia, A generic quantitative relationship between quality of experience and quality of service, *Network Management of Global Internetworking,* 24, 2, pp. 36-41, March 2010.

[240] F. Guyard and S. Beker, Towards real-time anomalies monitoring for QoE indicators, *Annals of Telecommunications*, vol. 65, no. 1-2, 2010.

[241] B. Hestnes, P, Brooks, and S. Heiestad, "QoE measuring QoE for improving the usage of telecommunication services", *Technical Report R 21*, Telenor, 2008.

[242] M. I. Iqbal, H. Zepernick, and U. Engelke, "Perceptual-based quality assessment for error protection schemes for wireless JPEG2000", *6th International Conference on Symposium on Wireless Communication Systems, ISWCS'09*, pp. 348-352, USA, 2009.

[243] L. Isaksson, Seamless Communications, *PhD Dissertation*, Blekinge Institute of Technology, 2007.

[244] R. Jiuchun, M. Dilinm and W. ZhiWei, A neural network based model for VoIP speech quality prediction, *2nd International Conference on Interaction Sciences, ICI '09*, ACM, pp. 1244-1248, USA, 2009.

[245] Understanding the Basic Networking Functions, Components, and Signaling Protocols in VoIP Networks, White Paper, Juniper Networks Inc., 2007.

[246] J. Lubin, “A human vision system model for objective picture quality measurements”, *International Broadcasting Convention*, no. 447, pp. 498-503, 1997.

[247] N. Miller and D.T. Campbell, “Recency and primacy in persuasion as a function of the timing of speeches and measurements”, *Journal of Abnormal and Social Physchology*, vol. 59, no. 1, pp. 1-9, 1959.

[248] K. Minghao, K. Yun Chyang, and E. K. Karuppiah, “Performance analysis and optimization of user space versus kernel space Network application”, *5th Student Conference on Research and Development SCOReD*, pp. 1-6, 2007.

[249] H. Schulzrinne, A. Rao, and R. Lanphier, Real Time Streaming Protocol (RTSP), RFC 2326, April 1998.

[250] DSL Forum Technical Report TR-126. Triple-play Services Quality of Experience (QoE) Requirements, 2006.

[251] M. Gustarini, S. Ickin, K. Wac, Evaluation of challenges in human subject studies ”in-the-wild” using subjects’ personal smartphones. *ACM conference on Pervasive and ubiquitous computing adjunct publication (UbiComp ’13 Adjunct)*. ACM, USA, pp. 1447-1456, 2013.

[252] H. Zeng, C. S. Ellis, A. R. Lebeck, A. Vahdat. “ECOSystem: managing energy as a first class operating system resource”, *SIGOPS Oper. Syst.*, Rev. 36, 5, 123-132, 2002.

[253] E. J. Vergara and S. N. Tehrani, “EnergyBox: A Trace-driven Tool for Data Transmission Energy Consumption Studies”, *International Conference on Energy Efficiency in Large Scale Distributed Systems (EE-LSDS 20)*, Lecture Notes in Computer Science. Springer, 2013.

[254] A. Pathak, Y. C. Hu, M. Zhang. “Where is the energy spent inside my app?: fine grained energy accounting on smartphones with Eprof”, *7th European Conference on Computer Systems (EuroSys ’12)* ACM, USA, 29-42,2012.

[255] J. Flinn and M. Satyanarayanan, “PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications”, *2nd IEEE Workshop on Mobile Computer Systems and Applications (WMCSA ’99)*, IEEE Computer Society, USA, 1999.

[256] F. Bellosa, “The benefits of event: driven energy accounting in power-sensitive systems”, *9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system (EW 9)*, ACM, USA, pp. 37-42, 2000.

[257] M. Anand, E. B. Nightingale, J Flinn, “Self-tuning wireless network power management”, *9th annual international conference on Mobile computing and networking (MobiCom ’03)*, ACM, USA, pp. 176-189, 2003.

[258] D. C. Snowdon, E. L.Sueur, S. M. Petters, and G. Heiser, “Koala: a platform for OS-level power management”, *4th ACM European conference on Computer systems (EuroSys ’09)*, ACM, USA, pp. 289-302, 2009.

[259] N. Banerjee *et al.* , “Users and Batteries: Interactions and Adaptive Energy Management in Mobile Systems”, Lecture Notes in Computer Science, 4717, Springer Berlin Heidelberg. pp. 217-234. 2007.

[260] B. Dusza *et al.* , “CoPoMo: a context-aware power consumption model for LTE user equipment”, *Transactions on Emerging Telecommunications Technologies*, 24(6), pp. 615–632, 2013.

[261] A. J. Oliner *et al.* , “Carat: Collaborative energy diagnosis for mobile devices”, *11th ACM Conference on Embedded Networked Sensor Systems*, 2013.

[262] M. Dong, L. Zhong. “Self-constructive High-rate System Energy Modeling for Battery-powered Mobile Systems”, *MobiSys’11*, 2011.

[263] C. Yoon *et al.* , “AppScope: Application Energy Metering Framework for Android Smartphone Using Kernel Activity Monitoring”, *USENIX Annual Technical Conference (USENIX ATC 12)*, pp. 387-400, 2012.

[264] W. Jung *et al.* , “DevScope: a nonintrusive and online power analysis tool for smartphone hardware components”, *8th IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis (CODES+ISSS ’12)*. ACM, USA, 353-362. 2012.

[265] OpenMoko. Open Mobile Free. Available at: `http://wiki.openmoko.org/wiki/Main_Page`. Accessed: August, 2014.

[266] Nokia Energy Profiler. Available at: `http://nokia-energy-profiler.en.softonic.com/symbian`. Accessed: August, 2014.

[267] A. Hergenroeder, J. Wilke, D. Meier, “Distributed Energy Measurements in WSN Testbeds with a Sensor Node Management Device (SNMD),” *23rd International Conference on Architecture of Computing Systems (ARCS)*, vol., no., pp.1,7, 2010.

[268] H. Liu, Y. Zhang, Y. Zhou. "TailTheft: leveraging the wasted time for saving energy in cellular communications", *6th international workshop on MobiArch (MobiArch '11)*. ACM, USA, pp. 31-36, 2011.

[269] E. Tan *et al.* , "PSM-throttling: Minimizing Energy Consumption for Bulk Data Communications in WLANs", *IEEE International Conference on Network Protocols, ICNP 2007*, vol., no., pp.123,132, 16-19 Oct. 2007.

[270] Y. Agarwal *et al.* , "Wireless wakeups revisited: energy management for voip over wi-fi smartphones", *5th international conference on Mobile systems, applications and services (MobiSys '07)*, ACM, New York, NY, USA, 179-191, 2007.

[271] G. Ananthanarayanan and I. Stoica, "Blue-Fi: enhancing Wi-Fi performance using bluetooth signals", *7th international conference on Mobile systems, applications, and services (MobiSys '09)*, ACM, USA, 249-262, 2009.

[272] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, "ZiFi: wireless LAN discovery via ZigBee interference signatures", *16th annual international conference on Mobile computing and networking (MobiCom '10)*, ACM, USA, 49-60, 2010.

[273] A. Rahmati and L. Zhong. "Context-for-wireless: context-sensitive energy-efficient wireless data transfer", *5th international conference on Mobile systems, applications and services (MobiSys '07)*. ACM, USA, pp. 165-178, 2007.

[274] A. Roy, "Energy management in mobile devices with the cinder operating system", *6th conference on Computer systems (EuroSys '11)*, ACM, New York, NY, USA, 139-152, 2011.

[275] J. Flinn, M. Satyanarayanan. 1999. "Energy-aware adaptation for mobile applications", *17th ACM symposium on Operating systems principles (SOSP '99)*, ACM, New York, NY, USA, 48-63, 1999.

[276] H. Falaki *et al.* , "Diversity in smartphone usage", *MobiSys '10*, ACM, USA, pp. 179-194, 2010.

[277] H. ChanTerri, "Facebook and its Effects on Users' Empathic Social Skills and Life Satisfaction: A Double-Edged Sword Effect", *Cyberpsychol. Behav. Soc. Netw.*, 17(5): pp. 276-280, 2014.

[278] H. T. G. Chou, N. Edge, "They Are Happier and Having Better Lives than I Am: The Impact of Using Facebook on Perceptions of Others' Lives", *Cyberpsychol. Behav. Soc. Netw.*, 15(2): pp. 117-121, 2012.

[279] F. Dobrian *et al.* , "Understanding the impact of video quality on user engagement". *SIGCOMM'11*, 2011.

[280] S. S. Krishnan and R. K. Sitaraman, "Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs". *IMC'12*, 2012.

[281] M. Ra, "Energy-delay tradeoffs in smartphone applications", *8th international conference on Mobile systems, applications, and services (MobiSys '10)*, ACM, NY, USA, pp. 255-270, 2010.

[282] V. Rodriguez *et al.* , "RILAnalyzer: a Comprehensive 3G Monitor On Your Phone." *IMC 13, ACM SIGCOMM*, Spain, 2013.

[283] A. Takahashi, D. Hands, V. Barriac, "Standardization activities in the ITU for a QoE assessment of IPTV," *IEEE Communications Magazine*, vol.46, no.2, pp. 78-84, 2008.

[284] A. Balachandran *et al.* , "Developing a predictive model of quality of experience for internet video", *SIGCOMM Comput. Commun.*, Rev. 43, 4, pp. 339-350, 2013.

[285] IEEE Code of Ethics, [Online], Available at: `http://www.ieee.org/about/corporate/governance/p7-8.html`, Accessed: 2014.

Appendix A

Appendix

A.1 Formulae of the Obtained Models

Table A.1: List of the obtained formulae. "GOF": goodness-of-fit; "$\sim$" symbol: as a function of; exp.: exponential; pow.: power.

DESC.	CHAP.	FORMULAE	GOF	MODEL
user rating $\sim$ packet delay variation	4	$UR = 4.21 \cdot \exp(-0.002 \cdot PDV/\text{ms})$	0.40	exp.
user rating $\sim$ packet delay variation	4	$UR = -0.88 \cdot PDV/\text{ms}^{0.27} + 6.38$	0.51	pow.
user rating $\sim$ maximum burst size	4	$UR = 59.96 \cdot MBS^{-0.036} - 51.71$	0.78	pow.
CCDF of inter-picture time (for freeze indications)	5	$CCDF = 0.91 \cdot \exp(-0.001 \cdot D_\text{p}/\text{ms})$	0.98	exp.
CCDF of inter-picture time (for $UR = 1$)	5	$CCDF = 0.88 \cdot \exp(-0.002 \cdot D_\text{p}/\text{ms})$	0.98	exp.
CCDF of inter-picture time (for $UR = 2$)	5	$CCDF = 0.87 \cdot \exp(-0.002 \cdot D_\text{p}/\text{ms})$	0.94	exp.
CCDF of inter-picture time (for $UR = 3$)	5	$CCDF = 0.88 \cdot \exp(-0.004 \cdot D_\text{p}/\text{ms})$	0.84	exp.
CCDF of inter-picture time (for $UR = 4$)	5	$CCDF = 2.05 \cdot \exp(-0.020 \cdot D_\text{p}/\text{ms})$	0.90	exp.
CCDF of inter-picture time (for $UR = 5$)	5	$CCDF = 4.62 \cdot \exp(-0.039 \cdot D_\text{p}/\text{ms})$	0.94	exp.
CCDF of mean OFF duration	5	$CCDF = 1.08 \cdot \exp(-0.004 \cdot \overline{OFF}/\text{ms})$	0.81	exp.
CCDF of mean ON duration	5	$CCDF = 0.67 \cdot \exp(-0.00006 \cdot \overline{ON}/\text{ms})$	0.93	exp.
CCDF of user response time (long)	5	$CCDF = \exp(-0.0007 \cdot D_{\text{response}_\text{long}}/\text{ms})$	0.98	exp.
inter-picture time $\sim$ instant. power cons.	7	$D_\text{p}/\text{ms} = 1.26 \cdot 10^5 \cdot \exp(-0.003 \cdot \tilde{P}_n/\text{mW})$	0.60	exp.
$MOS \sim$ ON probability	7	$MOS = 0.15 \cdot \exp(3.38 \cdot P_\text{ON})$	0.74	exp.
$MOS \sim$ OFF probability	7	$MOS = 4.67 \cdot \exp(-3.38 \cdot P_\text{OFF})$	0.74	exp.
$MOS \sim$ mean OFF duration	7	$MOS = 4.67 \cdot \exp(-0.0003\overline{OFF}/\text{ms})$	0.72	exp.
$MOS \sim$ mean ON and mean OFF duration	7	$MOS = 0.9 \cdot \exp(-0.0006 \cdot \overline{OFF}/\text{ms}) + 0.1 \cdot \exp(0.0000162 \cdot \overline{ON}/\text{ms}) + 2$	0.71	exp.
$MOS \sim$ energy waste	7	$MOS = 4.67 \cdot \exp(-3.38 \cdot (((E_\text{waste}/\text{J}/0.728) + 1)/208))$	derived model	exp.
$MOS \sim$ energy saving	7	$MOS = 4.67 \cdot \exp(-3.38 \cdot (((E_\text{sav.}/\text{J}/0.185) + 1)/208))$	derived model	exp.
CCDF of screen OFF duration	8	$CCDF = \exp(-0.22 \cdot \overline{screenOFF}/\text{ms})$	0.91	exp.
CCDF of screen ON duration	8	$CCDF = \exp(-1.6 \cdot \overline{screenON}/\text{ms})$	0.94	exp.

List of Figures

List of Tables

ABSTRACT

Smartphones have become crucial enablers for users to exploit online services such as learning, leisure, communicating, and socializing. The user-perceived quality of applications and services is an important factor to consider, in order to achieve lean resource management, to prevent user churn and revenue depletion of service or network providers. This is often studied within the scope of Quality of Experience (QoE), which has attracted researchers both in academia and industry.

The objective of this thesis is to study the most important factors influencing QoE on smartphones and synthesize solutions for intervention. The temporal impairments during a real-time energy-hungry video streaming are studied. The aim is to quantify the influence of temporal impairments on the user-perceived video QoE at the network and application level together with energy measurements, and also to propose solutions to reduce smartphone energy consumption without degrading the user's QoE on the smartphone for both user-interactive, e.g., video, and non-interactive cases.

QoE measurements on smartphones are performed throughout in-the-wild user studies. A set of quantitative Quality of Experience (QoE) assessment tools are implemented and deployed for automatic data logging at the network- and application-level. Online momentary survey, Experience Sampling Method (ESM) software, and Day Reconstruction Method (DRM) along weekly face-to-face user interviews are employed. The subjective QoE is obtained through qualitative feed back including Mean Opinion Score (MOS) as well as in-situ indications of poor experiences by users. Additionally, energy measurements on smartphones are conducted in controlled-lab environment with the Monsoon device.

The QoE of smartphone applications and services perceived by users depends on many factors including anomalies in the network, application, and also the energy consumption. At the network-level, high packet delay variation causes long video freezes that eventually impact negatively the end-user perceived quality. The freezes can be quantified as large time gaps in-between the displayed pictures during a video stream at the application-level. We show that the inter-picture time in cellular-based video stream can be represented via two-state exponential ON/OFF models. We show models representing the non-linear relationship between the QoE and the mean inter-picture time. It is shown that energy measurements help to reveal the temporal impairments in video stream enabling energy consumption as a QoE indicator. Next, energy waste and saving during temporal impairments are identified. Additionally, other video streaming use cases, e.g., "download first and watch later", are studied and appropriate energy-saving download scheduling mechanisms are recommended. The possibility for decreasing energy consumption when the smartphone screen is OFF, while maintaining QoE, is revealed. We first show exponential models to represent user's interaction with smartphone, then propose a NyxEnergySaver software, to control the cellular network interface in a personalized manner to save smartphone energy. According to our findings, more than 30 % smartphone energy can be saved without impacting the user-perceived QoE.

ISSN 1653-2090

2015:06

ISBN: 978-91-7295-303-1