



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

ENERGY-EFFICIENCY AND PERFORMANCE  
*in*  
COMMUNICATION NETWORKS

Analyzing Energy – Performance Trade-offs in Communication Networks  
and their Implications on Future Network Structure and Management

Vom Fachbereich Elektrotechnik und Informationstechnik  
der Technischen Universität Darmstadt  
zur Erlangung des akademischen Grades eines  
Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Dissertation

von

DIPL.-ING. FABIAN KAUP

Geboren am 02.08.1984 in Lahnstein

Vorsitz: Prof. Dr.-Ing. Franko Küppers  
Referent: Prof. Dr. David Hausheer  
Korreferent: Prof. Dr. Jörg Widmer

Tag der Einreichung: 22. März 2017  
Tag der Disputation: 09. Mai 2017

Hochschulkennziffer D17  
Darmstadt 2017

Please use the following identifiers when referring to this thesis:

- <http://tuprints.ulb.tu-darmstadt.de/id/eprint/6295>
- [urn:nbn:de:tuda-tuprints-62956](https://nbn-resolving.org/urn:nbn:de:tuda-tuprints-62956)

Dipl.-Ing. Fabian Kaup: *Energy-efficiency and Performance in Communication Networks*,  
Analyzing Energy – Performance Trade-offs in Communication Networks  
and their Implications on Future Network Structure and Management. © 22. März 2017

## ABSTRACT

---

The demand on communication networks has increased over the past years and is predicted to continue for the foreseeable future [Cis16]. Cellular network access with a compound annual growth rate (CAGR) of 53% is the main area of growth [Cis16]. This affects the network quality, bringing current network technologies to their limits [Qua13]. Future network standards like 5G promise to satisfy this demand, providing a 1000-fold increase in data rates and latencies as low as 1 ms [Qua13].

With information and communications technology (ICT) causing 10% of the global energy consumption [Mil13], the increasing demand is also reflected in a growing energy consumption of communication networks [BBD+11]. The major contributor to the network power consumption are home gateways (HGWs) in the fixed access network, and mobile base stations in the cellular network [VHD+11]. This trend is predicted to continue [BBD+11].

To assess and optimize the power consumption of communication networks, power models of the involved devices are required. Using these, the efficiency of proposed optimization approaches can be assessed before deployment. A number of power models of conventional network equipment for different device classes can be derived from literature. Still, models of new device classes such as single-board computers (SBCs) and OpenFlow switches are not available. For each class, representative power models of several device types are presented. Further, the power consumption caused by new communication protocols such as MultiPath TCP (MPTCP) is not fully analyzed yet. This work is, to the best of the author's knowledge, the first to publish SBC and OpenFlow power models and contributes to the understanding of MPTCP power consumption during constant bit rate (CBR) streaming.

For the analysis of the power consumption, also the knowledge of network performance is required, as it defines relative costs and the maximum number of supported users. This is well known and comparatively simple in fixed networks, but more challenging in a wireless context. A number of approaches are described in literature and implemented as commercial software (e.g. [SSM13; OpS]), but the data required for analysis and optimization is not available. Hence, extensive measurements of the cellular network are conducted in this work. The location-based availability and performance of cellular and WiFi networks are assessed in a crowd-sensing study. Based on measurements on regional trains, the predictability of the cellular service quality based only on available network technology and latency is shown to be feasible. Anomalies observed within the crowd-sensing data are analyzed using dedicated, stationary measurements. The main observation is that network management decisions have significant effects on end-to-end performance. By allocating users to random points of presence (PoPs)/exit gateways of the mobile net-

work operator (MNO), the latency compared to the best observed allocation is increased by more than 58 % in over 80 % of the time.

Combining the energy models and network performance measurements as presented in this work, an energy evaluation environment is created to analyze the cost of mobile data communication. This combines the empirically determined performance of cellular and WiFi networks with the energy models of smartphones and traffic traces recorded by the participants of a crowd-sensing study. Thereby, the power consumption of the generated data patterns is established, and the effectiveness of network optimization approaches as presented in literature assessed. These prove to be less potent than originally claimed by the authors. This is expected considering the improvements in cellular networks and smartphones. Nonetheless, energy savings are observed. Considering the requirement of 5G networks to reduce latency to 1 ms, and improve capacity by a factor of 1000, while simultaneously reducing energy consumption, also changes in fixed access networks are required. A promising approach assuming further virtualization of networks using software defined networking (SDN) and network functions virtualization (NFV) is the placement of services closer to the end-users. Extrapolating the trend of increasing hardware capabilities of HGWs at almost constant cost, these may be used to provide additional services to local users. This may be achieved by e.g. using virtualized content distribution network (CDN) nodes running on HGWs, thus utilizing these often idle resources. This further equalizes the traffic within the core network by providing content locally and refreshing it during less traffic intensive periods. Simultaneously, the end-user perceived service quality is expected to increase. Thus, installed capacities can be used longer, resulting in better service quality at fixed energy cost.



## KURZFASSUNG

---

Seit Bestehen des Internets wird jährlich ein stark ansteigendes Datenvolumen verzeichnet und es wird erwartet, dass der beobachtete Trend weiter anhält [Cis16]. Gerade in Anbetracht des steigenden Bedarfs an hochauflösendem digitalen Videomaterial und dem zu erwartenden Anstieg durch interaktive Virtual Reality (VR) Anwendungen, wird der Bedarf an höheren Datenraten und niedrigeren Latenzen steigen. Besonders stark ist das Wachstum in Mobilfunknetzen, für die ein jährlicher Anstieg des Datenverkehrs von 53 % prognostiziert wird. Hier sollen Mobilfunknetze der fünften Generation (5G) Abhilfe schaffen [Qua13]. Das Ziel ist es, eine 1000-fach höhere Datenrate bei Latenzen von 1 ms zu erreichen [Qua13].

Gleichzeitig steigt allerdings auch der Energiebedarf stationärer und mobiler weltweiter Kommunikationsnetze [BBD+11]. Nach verschiedenen Schätzungen fallen aktuell bis zu 10 % des elektrischen Energieverbrauchs zu Lasten von Informations- und Kommunikationsinfrastruktur [Mil13]. In kabelgebundenen Netzen wird der Großteil des Energiebedarfs durch Heimrouter erzeugt, während in Mobilfunknetzen die Basisstationen die größten Energiekosten erzeugen [VHD+11]. Auf Grund dieser beiden Entwicklungen wird auch weiterhin ein steigender Energiekonsum erwartet.

Um die Kosten mobiler Kommunikation zu bestimmen und zu optimieren, werden entsprechend Energiemodelle der beteiligten Geräte benötigt. Einige Modelle konventioneller Netzwerkinfrastruktur sind bereits in entsprechender Fachliteratur veröffentlicht, allerdings bestehen Lücken bezüglich neuartiger Geräte (Einplatinencomputer (SBCs)) und Datenflusssteuerungsverfahren (OpenFlow/SDN und MPTCP). Um auch diese in zukünftigen Optimierungsverfahren berücksichtigen zu können, wurden exemplarisch Energiemodelle für mehrere Geräte jeder Klasse erstellt und veröffentlicht.

Da allerdings der Energieverbrauch speziell mobiler Endgeräte stark von Durchsatz und Latenz der verfügbaren Netzwerktechnologien abhängt, müssen auch diese zur Optimierung bekannt sein. Hier wurden bereits einige Ansätze vorgestellt (z.B. [SSM13; OpS]), allerdings sind die zur Optimierung benötigten Daten nicht verfügbar. Entsprechend wurde eine Android App veröffentlicht, die in mehreren Studien eingesetzt wurde um die Netzwerkqualität in verschiedenen Situationen zu ermitteln. Ziel hierbei war eine Karte der verfügbaren Mobilfunk- und WLAN-Netze, ihrer Latenz und des maximalen Datendurchsatzes zu erstellen. Zusätzlich wurden auch Messungen in Regionalbahnen durchgeführt, um einzig basierend auf Netzwerkverfügbarkeit und Paketverlustrate die Nutzbarkeit des Netzes für verschiedene Szenarien vorherzusagen. In den Studien beobachtete Unregelmäßigkeiten wurden mittels einer dedizierten, stationären Messstudie analysiert, woraus abgeleitet wurde, dass die zufällige, aber für 36 Stunden statische Zuordnung von Mobilgeräten zu bestimmten Standorten des Mobilfunkanbieter

eters signifikanten Einfluss auf die Ende-zu-Ende Latenz hat. Hier wurden zwischen dem mobilen Endgerät und einem dedizierten Messserver in über 80 % der Fälle um mehr als 58 % erhöhte Latenzen beobachtet. Ähnliche Ergebnisse wurden auch beim Zugriff auf die zehn in Deutschland beliebtesten Internetseiten bestätigt.

Zur Ermittlung der Energiekosten mobiler Datenkommunikation wurde eine Evaluationsumgebung erstellt, die basierend auf der gemessenen Netzverfügbarkeit und Qualität in Kombination mit Energiemodellen verschiedener Smartphones und Aufnahmen der Datennutzung mehrerer Smartphone-Nutzer den Energieverbrauch berechnet. Hiermit wurden zwei in der Literatur vorgestellte Optimierungsverfahren mit dem Energieverbrauch des unmodifizierten Systems verglichen. Die resultierenden Energieeinsparungen sind wesentlich niedriger als von den ursprünglichen Autoren angegeben, was allerdings auf Grund der Weiterentwicklung der Mobilfunknetze und Smartphones zu erwarten ist. Basierend auf weiteren gemessenen und in der Literatur veröffentlichten Energiemodellen in Kombination mit den Anforderungen von 5G Netzen wird das Potential der Nutzung von Heimroutern mit erweiterter Funktionalität (z.B. zum Vorhalten von Daten oder Optimierung des Datenverkehrs) bezüglich Datendurchsatz und Energieverbrauch hergeleitet. Da diese Geräte häufig nicht ausgelastet sind, und moderne Hardware kontinuierlich erhöhte Ressourcen bei ähnlichem Energieverbrauch bietet, werden durch diese Optimierungen Energieeinsparungen erwartet, da die Auslastung des dahinterliegenden Netzes ausgeglichen und somit verfügbare Kapazitäten länger genutzt werden können.

## DANKSAGUNG

---

Die vorliegende Dissertation wäre ohne die vielfältige Unterstützung meiner Kollegen, Studenten, Freunde und Familie nicht in dieser Form möglich gewesen.

Zuerst möchte ich mich bei David Hausheer bedanken, der es mir ermöglicht hat, diese Arbeit zu schreiben. Seine konstante Wissbegierde hat mich stets angespornt die verschiedensten sich stellenden Probleme anzugehen und die jeweils bestmögliche Lösung zu finden. Mein Dank gilt auch Jörg Widmer für die wertvollen fachlichen Diskussionen im Projekt eCOUSIN und während meines Aufenthalts bei IMDEA.

Bedanken möchte ich mich auch bei meinen Kollegen Julius Rückert, Jeremias Blendin, Björn Richerzhagen, Matthias Wichtlhuber, Leonhard Nobach, Christian Koch und allen anderen Kollegen bei KOM und PS für ihr konstruktives Feedback in Seminaren und verschiedenen Vorträgen, die gemeinsamen Kaffeepausen und den spontanen, informellen Austausch bei auftretenden Problemen. Auch die gelegentlichen gemeinsamen Abende haben zur nötigen Entspannung während anstrengender Phasen beigetragen.

Wesentliche Unterstützung wurde mir von meinen Studenten Sergej Melnikowitsch, Stefan Rado, Manuel Fuentes Sifuentes, Carsten Englert, Hendryk Herzog, Florian Fischer, Thomas Schnabel, Rohit Shah und Rainer Wahler zuteil. Ihr wissenschaftlichen Beitrag in diversen Bachelor- und Masterarbeiten lieferte einen wichtigen Beitrag zu dieser Arbeit. Bedanken möchte ich mich auch bei meinen HiWis Eike Mentzendorff, Stefan Hacker, Philip Gottschling, Ousama Esbel und Hamid Elmi, die mich bei der Implementierung verschiedener Komponenten unterstützt haben. Weiterhin möchte ich mich bei allen Studenten bedanken, die bei mir diverse Praktika und Seminare belegt haben und mit ihren Arbeiten verschiedene Thesen bestätigt oder widerlegt haben.

I would also like to thank Foivos Michelinakis, Valentin Burger, Michael Seufert, Nicola Bui, George Petropoulos, and all others from the projects SmartenIT and eCOUSIN, for their interest in my work and encouraging further joint projects. Without you, I would have missed a number of further aspects on our common research area, and the cultural highlights of various locations in Europe. Weiterhin möchte ich Christian Meurisch für die gemeinsamen Arbeiten und konstruktiven Diskussionen zu aktuellen und zukünftigen Forschungsarbeiten danken.

Besonders bedanken möchte ich mich für die seelische und moralische Unterstützung meiner Freunde und Familie. Ohne euch wäre das Unterfangen Promotion in dieser Form nicht möglich gewesen. Danke für die konstante Unterstützung und regelmäßigen Unterbrechungen, die mich immer wieder reflektiert und voller Energie in die weitere Arbeit starten ließen.



## ACKNOWLEDGMENTS

---

This work has been supported in parts by the European Union (FP7/#317846, SmartenIT and FP7/#318398, eCOUSIN), and the German Research Foundation (DFG) as part of project B03 within the Collaborative Research Center (CRC) 1053 – MAKI.



# CONTENTS

---

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	Research Questions . . . . .	2
1.2	Contributions . . . . .	6
1.3	Thesis Organization . . . . .	7
<b>2</b>	<b>BACKGROUND</b>	<b>9</b>
2.1	Structure and Properties of Communication Networks . . . . .	9
2.1.1	Types of Network Access . . . . .	12
2.1.2	Structure of Fixed Access Networks . . . . .	13
2.1.3	Structure of Cellular Networks . . . . .	14
2.1.4	Network Performance Metrics . . . . .	17
2.2	Emerging Networking Approaches . . . . .	19
2.2.1	Multipath TCP . . . . .	19
2.2.2	Software Defined Networks . . . . .	20
2.2.3	Network Functions Virtualization . . . . .	21
2.3	Energy in Communication Networks . . . . .	22
2.3.1	Energy Metrics in Networked Systems . . . . .	22
2.3.2	Power Measurements . . . . .	23
<b>3</b>	<b>RELATED WORK</b>	<b>25</b>
3.1	Power Consumption of Network Devices . . . . .	26
3.1.1	Single-Board Computers . . . . .	26
3.1.2	Routers and Switches . . . . .	29
3.1.3	Smartphones . . . . .	31
3.1.4	Comparison of Power Models and Approaches . . . . .	35
3.2	Network Performance Measurement . . . . .	37
3.2.1	Latency . . . . .	39
3.2.2	Throughput Measurements . . . . .	41
3.2.3	Network Coverage Maps . . . . .	43
3.2.4	Comparison of Measurement Approaches . . . . .	45
3.3	Comparison of Mobile Network Optimization Approaches . . . . .	46
3.4	Differentiation from Related Work . . . . .	50
<b>4</b>	<b>POWER CONSUMPTION OF NETWORK ENTITIES</b>	<b>53</b>
4.1	Power Consumption of Single-board Computers . . . . .	55
4.1.1	Measurement Setup . . . . .	56

4.1.2	Measurements . . . . .	62
4.1.3	Modeling of the Power Consumption . . . . .	64
4.1.4	Evaluation of the Power Models . . . . .	68
4.1.5	Summary and Conclusions . . . . .	74
4.2	Power Consumption of Wired Network Infrastructure . . . . .	75
4.2.1	Measurement Setup . . . . .	76
4.2.2	Measurements . . . . .	79
4.2.3	Power Consumption Model for OpenFlow Switches . . . . .	84
4.2.4	Evaluation of the Derived Power Models . . . . .	88
4.2.5	Summary and Conclusions . . . . .	92
4.3	Power Consumption of End-user Devices . . . . .	94
4.3.1	Measurement Setup . . . . .	94
4.3.2	Interface Reference Power Consumption . . . . .	98
4.3.3	MPTCP Power Consumption . . . . .	102
4.3.4	Summary and Conclusions . . . . .	107
4.4	Power Consumption of Communication Infrastructure . . . . .	109
<b>5</b>	<b>ANALYSIS OF NETWORK PERFORMANCE</b>	<b>113</b>
5.1	Crowd-sensed Cellular Network Performance Measurements . . . . .	115
5.1.1	Measurement Methodology . . . . .	115
5.1.2	Description of the Measurements . . . . .	119
5.1.3	Summary and Conclusion . . . . .	122
5.2	Cellular Network Performance on Trains . . . . .	122
5.2.1	Measurement Methodology . . . . .	124
5.2.2	Description of the Measurements . . . . .	126
5.2.3	Predicting the Cellular Service Quality on Trains . . . . .	130
5.2.4	Summary and Conclusion . . . . .	135
5.3	Influence of Traffic Management Decisions on Network Performance . . . . .	136
5.3.1	Measurement Methodology . . . . .	136
5.3.2	Measurements . . . . .	137
5.3.3	Summary and Conclusion . . . . .	143
5.4	Conclusion on Performance of Cellular Networks . . . . .	144
<b>6</b>	<b>INFLUENCE OF TRAFFIC MANAGEMENT ON ENERGY CONSUMPTION</b>	<b>147</b>
6.1	Analysis of the Power Savings of Mobile Scheduling Approaches . . . . .	148
6.1.1	Data Sets . . . . .	149
6.1.2	Evaluation Environment . . . . .	153
6.1.3	Energy Conservation Schemes . . . . .	157
6.1.4	Evaluation Results . . . . .	159
6.1.5	Summary and Conclusions . . . . .	162



6.2	Energy Efficiency in Future Networks . . . . .	164
6.2.1	Potential of Emerging Network Technologies . . . . .	165
6.2.2	Network Structure and Management . . . . .	168
6.2.3	Energy Efficiency . . . . .	169
6.3	Conclusion . . . . .	172
<b>7</b>	<b>CONCLUSIONS</b>	<b>175</b>
7.1	Summary of the Thesis . . . . .	175
7.1.1	Power Consumption of Network Entities . . . . .	175
7.1.2	Analysis of Network Performance . . . . .	177
7.1.3	Influence of Traffic Management on Energy Consumption . . . . .	178
7.1.4	Contributions . . . . .	179
7.1.5	Conclusions . . . . .	180
7.2	Outlook . . . . .	181
	<b>REFERENCES</b>	<b>183</b>
<b>A</b>	<b>APPENDIX</b>	<b>201</b>
A.1	List of Acronyms . . . . .	201
A.2	Architecture of the Network Coverage App . . . . .	207
A.3	Architecture of the OpenFlow Power Measurement Environment . . . . .	213
<b>B</b>	<b>AUTHOR'S PUBLICATIONS</b>	<b>215</b>
B.1	Main Publications . . . . .	215
B.2	Co-Authored Publications . . . . .	217
<b>C</b>	<b>CURRICULUM VITÆ</b>	<b>219</b>
<b>D</b>	<b>ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG</b>	<b>223</b>



## INTRODUCTION

---

THE increasing availability of broadband Internet connectivity has heavily changed the way we live. Data access has become ubiquitous. Almost unrestricted in time and location, knowledge, communication, and entertainment are available within arm's reach. This has changed the way we work, travel, and communicate. Consequently, the demand on both mobile and fixed networks has constantly increased over the past years, and is predicted to further rise in the future [Cis16]. Over the next 15 years, an overall yearly traffic growth of 21 % is predicted, increasing the monthly transferred traffic from currently 75 EB to approximately 195 EB per month in 2020. Out of this, currently 3.6 EB is caused by mobile devices in both cellular and WiFi networks, showing a compound annual growth rate (CAGR) of 53 %. Thus, the cellular traffic is expected to grow to over 30 EB per month in the year 2020. This growth is caused by the rising popularity of smartphones, changing usage patterns like the increasing use of bandwidth intensive applications (e.g. video streaming), and increasing data volumes at constant cost. Similar effects to the availability and price of artificial lighting [TW10] can be observed, in which utilization increases while the price drops, keeping the overall cost stable.

These developments put a large burden on both fixed-line and wireless networks. A possible solution is upgrading capacities, which is both costly and challenging. Upgrades on wireless networks conventionally require only the replacement of a single base station and perhaps an upgrade of the backhaul link. Nonetheless, throughput on wireless networks is strictly limited by spectrum availability. Reducing cell sizes improves frequency reuse, thus increasing capacity. Still, when deploying more and smaller cells, these network upgrades converge in cost and complexity with the upgrade of fixed-line networks, where usually construction works are required to install or upgrade links, thus also increasing the capital expenditure (CAPEX).

Increasing the number of base stations, and using higher modulation schemes is expected to increase the power consumption of the network due to additional hardware and higher power required to achieve low bit error rates. Already 72 % of the electricity demand of a cellular network is consumed by base stations [CM012]. Combined with the large increase in demand for mobile data services, the total energy cost each network operator has to bear is also expected to increase considerably. Furthermore, handling this traffic in the backbone requires additional capacities, thereby also increasing energy consumption and cooling requirements.

Similarly, on the mobile terminal side, energy issues arise. Smartphone capabilities like CPU frequency, storage, screen resolution, and wireless data rates have improved consid-

erably over the last years [CCD+14]. Although improvements in component power consumption are made, the overall power draw of mobile devices has increased [CCD+14]. A considerable fraction (31.2 %) is caused by the various network interfaces, followed by the screen and system on chip (SOC) [CDJ+15], resulting in average battery lifetimes of less than a day. As mobile communication is becoming more and more common, its power consumption is also expected to increase.

The rise of 5G with its increasing number of connection points, technologies, and communication schemes [NGNM15] will further increase the power consumption of participating network devices. Although small cells require less power on both base-stations and mobile terminals, their sheer number and deployment in support of the already existing infrastructure will increase the overall power consumption. Further, advanced modulation and coding schemes are expected to increase the power consumption of mobile devices and base stations [CGB05].

Besides adding additional cells and access points, possible optimizations of network energy efficiency include the use of software defined networking (SDN) and network functions virtualization (NFV), increasing the flexibility of the network, thus allowing for a more efficient use of available resources. Hence, traffic in the core network may be reduced by serving content from local, decentralized content distribution network (CDN) nodes, whose contents are refreshed during less traffic intensive periods. An example for the application of NFV is the placement of performance enhancing proxies (PEPs) directly on femto-cells or home gateways (HGWs), thus improving the performance of wireless links by reducing unnecessary delays and retransmissions.

Summarizing, the increasing traffic results in a rising demand on networks to support the requested data rates. This demand may be satisfied by upgrading capacities both at the edge and in the core network. But increasing data rates also increases the resulting power consumption [BBD+11]. In particular wireless network access causes the highest energy cost, both on WiFi and cellular networks [VHD+11]. Due to economical and ecological reasons, increases in power consumption should be reduced to a minimum, possibly even reducing the energy demand of the network while still supporting the required data rates. Hence, the focus of this work is the analysis of the performance and energy efficiency of communication networks, with a special emphasis on wireless networks. In particular, the goal is to analyze existing energy optimization approaches, and derive promising approaches for future optimizations based on the obtained results.

### 1.1 RESEARCH QUESTIONS

A number of research questions are derived from the requirement of finding and optimizing the relation between energy cost and network performance. These questions arise in three areas, corresponding to the estimation of power consumption of networked devices,

determining network performance required to accurately model the network, and the analysis of possible energy savings when applying different traffic management schemes.

When considering the energy consumption of fixed infrastructure networks, usually the nominal power consumption of connected devices is considered. Nevertheless, devices are often idle or only lightly used. Hence, the following work focuses on idle consumption, dynamic changes under load, and possible optimization approaches. A number of studies are already available describing the power consumption of conventional network infrastructure (e.g. [CTM+13]). Hence, the following work focuses on emerging network management approaches and their influence on energy consumption.

A relatively new class of devices are single-board computers (SBCs) like the Raspberry Pi and its successors. Due to their comparatively high processing power and easy programmability, these often serve as gateways in home automation and monitoring projects [PCK15] or are proposed to replace cloud servers [ZLH+16]. Research has been conducted concerning their energy consumption, but results are either limited to a specific deployment (e.g. [ZLH+16]), or focused on the processor power consumption alone (e.g. [LCB15]). Still, to estimate their power consumption under various conditions, general power models are required. These power models are derived in this thesis. By using ready to use system utilization values as basis, the power consumption of arbitrary services can later be estimated using these system monitoring values only. Thus, the cost of services as may later be deployed on HGWs can be derived.

In addition, core and access networks are becoming increasingly virtualized. Recent studies analyze the performance and possible load management approaches of SDN. Still, the consumption of these deployments, in particular when using advanced functionality, is not fully known. Hence, the power consumption of a hardware and software OpenFlow switch are analyzed in this thesis. Thereby, the cost of the additional functionality can be estimated.

On smartphones, the demand for increasing data rates and uninterrupted connectivity is rising. Here, the MultiPath TCP (MPTCP) serves as a means to ensure the continuity of transport control protocol (TCP) connections in dynamically changing network environments. Furthermore, by bundling connections on multiple interfaces, the link capacity can be increased. Related work analyzes the power consumption of MPTCP for regular downloads with maximum data rates (e.g. [LCN+14]). Contrary, this thesis focuses on the use case of constant bit rate (CBR) streaming which is common in video conferencing and live cloud gaming.

By filling these gaps, the power consumption of larger, virtualized network deployments can be analyzed, which was previously not possible using well-funded power models. Examples are the placement of virtualized CDN nodes on HGWs or the instantiation of PEPs close to the end user. Using these models, the influence of service placement on the energy consumption of smartphones and the network can be calculated, also includ-

ing the cost of HGWs and SDN based traffic redirection. From these observations, the following research questions are derived:

- RQ 1.1 What is the energy cost of decentralized caching and computational offloading using SBCs and how can it be determined?
- RQ 1.2 How does the energy efficiency of hardware and software OpenFlow switches compare and what are their respective benefits?
- RQ 1.3 What is the energy cost of increasing reliability and throughput of mobile communication using MPTCP for CBR streaming?

These research questions guide the analysis of the energy consumption of the respective device classes and traffic management approaches as described in Chapter 4.

The energy consumption of mobile devices depends, besides the available network interfaces and modems, on the cellular network performance. This influences the duration an interface needs to be active and thus its power states and their duration. Still, cellular network performance is often communicated in peak data rates. However, large differences in throughput, latency, and packet loss can be observed when accessing the cellular network from a mobile device.

A number of studies exist, analyzing the performance of cellular networks under various conditions. Still, their data is generally limited to specific locations or routes. Furthermore, the data sets required to analyze cellular performance are often not available. Hence, using a crowd-sensing study, the key performance indicators (KPIs) of both WiFi and cellular networks, annotated with locations and timestamp are measured. Thus, the performance of different network types can be compared at the same location, and the relations between different parameters can be evaluated.

The crowd-sensing measurements detailing the wireless network performance in an urban area allow the estimation of cellular network performance for pedestrians and mobile data access on buses and trams. Additionally, users desire access to mobile data on trains, in particular when commuting. There, network access is expected to differ considerably due to higher velocity, frequent handover, and less developed network infrastructure. Hence, a detailed analysis of cellular network access on trains is conducted, with the goal of deriving differences from urban data access. Of particular interest is the predictability of network access to communicate the estimated network performance to the user, and later automatically optimize network access based on this.

Due to the increasing performance of networks, also previously unknown effects become visible in the collected data. Examples are varying latencies for a fixed location, cell, and signal strength. Hence, a detailed analysis of these effects using dedicated measurements is conducted in this thesis.

The cellular service quality is thus determined for different scenarios. Knowing the latency and throughput of a network, also the power consumption of smartphones can

be derived using device-specific power models. The main focus of the conducted studies is the analysis of the real-world, end-to-end performance of deployed cellular networks for different network technologies, and their implications on the quality of service (QoS). The following research questions guide the subsequent analysis:

- RQ 2.1 What are the parameters affecting cellular service quality and user-perceived network performance when being mobile?
- RQ 2.2 How does cellular network access on trains differ from general mobile network access, and how can the network performance be predicted?
- RQ 2.3 What is the influence of network structure and management on end-user perceived network performance?

These research questions guide the analysis of the wireless network performance as discussed in Chapter 5.

By combining the energy models with the actual network performance, the power consumption under a given load can be estimated. Related work presents a number of energy optimization approaches focused on improving the battery life of smartphones (e.g. [HQM+12]). Still, these publications compare the achieved performance against an undisclosed 'status quo'. Hence, an energy evaluation environment is developed to compare their energy savings under comparable network conditions and devices. Thus, for the first time, a comparison of the energy saving potential of these optimization approaches is possible. The goal of this approach is to derive promising approaches to increase energy savings in modern wireless networks.

Similarly, the estimation of power consumption and energy savings of fixed communication networks is possible. Based on general observations derived from the power and network performance measurements, recommendations on further promising research in the direction of 5G networks are derived and discussed. The guiding research questions in this context are:

- RQ 3.1 What is the energy cost of mobile communication for a regular smartphone user, and how is this affected by smartphone-based energy conservation approaches?
- RQ 3.2 What is the potential of emerging network technologies on network infrastructure and mobile devices considering performance and energy consumption?

These questions guide the analysis of the combined network performance and energy consumption in Chapter 6.

Knowing both the performance and energy consumption of network devices participating in data transfers, the overall power consumption of each interaction can be determined.

Using this combined knowledge, existing and future network optimization approaches can be analyzed and their impact on end-user perceived network performance and energy consumption in different network domains can be derived. Thus, the increasing traffic demand in communication networks on overall power consumption can be assessed, and possibly reduced.

## 1.2 CONTRIBUTIONS

The contributions of this work are threefold. First, energy models for exemplary devices in different network domains are generated. Secondly, realistic measurements and models estimating the performance of cellular networks are derived. Thirdly, the power models and network performance measurements are combined to assess the energy savings of optimization approaches as described in literature. Furthermore, general recommendations on promising areas for future research reducing the energy consumption of communication networks are derived.

First, energy models for exemplary devices representing different device classes are generated. The analyzed classes are smartphones, SBCs as example of an augmented HGW, and OpenFlow switches. For each of these classes two or more exemplary devices are selected, for which the power measurement and modeling procedure is detailed and the resulting models are presented. Since one goal of this work is to show the feasibility of fine-granular power model generation, suitable models of the power consumption depending on system utilization are derived. The resulting SBC power models are the first ready to use power models of these devices, working on readily available system monitoring values only. Similarly, the published OpenFlow power models are the first to focus on the impact of OpenFlow rules on energy consumption. The MPTCP CBR streaming measurements analyze the effect of load-balancing on the power consumption of bandwidth intensive real-time applications like video streaming or live cloud gaming.

To properly model the power consumption of smartphones under real-world conditions, the knowledge of network latency and throughput is required, as it considerably influences the active time of the wireless interfaces. Hence, the second main contribution of this thesis is the analysis of cellular and WiFi network performance. Detailed measurements of the mobile network performance are conducted, using a range of different measurement techniques. From these, models of the network availability and performance are derived. User-focused studies are run on Android smartphones and are augmented with dedicated studies focusing on particular effects detected in the crowd-sensing studies. The obtained results show the implications of the previously unknown influence of network management in the surveyed network on end-to-end latency. Increases of 58% are observed in 80% of the time, which would be mitigated by relatively simple changes in network management.



Combining the power models of network entities with the detailed knowledge of mobile network performance, the power consumption of mobile devices is modeled for realistic usage scenarios. Based on these, the effectiveness of different smartphone-based energy optimization approaches derived from literature is compared. The accuracy and validity of these studies is maximized by using recorded, geo-tagged traffic traces. The analyzed approaches show considerably smaller energy savings than originally claimed by the authors, which is expected considering the improvements in network infrastructure and smartphones. To the best of the author's knowledge, this thesis is the first work comparing different smartphone-based energy optimization approaches based on a unified data set. Based on the general observations of network performance and energy consumption, suggestions on improving the energy efficiency in future networks are derived. Due to the currently mainly static power consumption of network infrastructure, the promising approaches derived are improving network utilization by using SDN and NFV features. Therefore, extended HGWs may be used to serve content to end-users, thus simultaneously improving performance.

### 1.3 THESIS ORGANIZATION

Based on the motivation, identified research questions, and outlined contributions, the remainder of this thesis is structured as follows.

CHAPTER 2 introduces background knowledge required to understand the network performance and power measurements. Namely, this chapter defines the terminology used throughout the work, summarizes network structures, introduces the basics of network virtualization, and recaps the foundations of power measurements.

CHAPTER 3 gives an overview of related work in the areas of power modeling and performance measurement of communication infrastructure. Furthermore, approaches optimizing the energy efficiency of mobile network access are surveyed, with the goal of identifying promising approaches for comparison and further optimization.

CHAPTER 4 presents the work conducted in the area of network power measurement and modeling. This chapter describes and discusses power measurements for a variety of SBCs, different smartphones, and OpenFlow switches. For each of these, power models describing the energy consumption depending on device state and utilization are presented.

CHAPTER 5 discusses the measurement and modeling of the cellular network performance and its implications on mobile clients. Measurements cover passive signal strength monitoring as well as active network probing in a crowd-sensing based manner. Round-trip time (RTT) and throughput measurements are conducted, thus deriving maps of location dependent network performance. Based on stationary reference

measurements, the influence of network management on the end-to-end QoS is determined.

CHAPTER 6 combines the power models discussed in Chapter 4 with the observations of network performance in Chapter 5. These are combined with recorded user traces containing geo-tagged network utilization. Thus, realistic power consumption values of mobile devices are derived. Different power saving approaches as discussed in literature are compared to derive implications on optimal traffic management strategies within the network and on mobile devices. Furthermore, the implications of energy consumption and network management approaches on future networks are discussed.

CHAPTER 7 summarizes the derived results concerning network power consumption, performance of communication networks, and possible optimizations combining the observations from both areas. Based on these results, conclusions on improving the energy efficiency of communication networks are drawn and promising areas for future research identified.

## BACKGROUND

---

**A**CCURATE measurements of the cellular network performance and power consumption require knowledge of the underlying network structure and a general understanding of the applied network management approaches. Hence, the following chapter introduces these concepts, thus building a common foundation on which the subsequent studies of network performance are built. Besides describing network management approaches also the power measurement of information and communications technology (ICT) is discussed, and the respective requirements for the power measurements reported in this work derived.

This chapter gives an overview of common network topologies of cellular and fixed communication networks, and their interconnection in Section 2.1. The goal is to provide a common understanding of these topologies to discuss their implications on network performance in Chapter 5 and Chapter 6. Section 2.2 introduces emerging traffic management techniques as analyzed in Section 4.2, Section 4.3, and Section 6.2. Finally, Section 2.3 describes the fundamentals of power measurements with a focus on computing and network devices, thus giving further background on Chapter 4. The scope of this chapter purposely limited to the level of detail required to assess the network performance and its energy consumption, thus omitting non-essential information to improve clarity.

### 2.1 STRUCTURE AND PROPERTIES OF COMMUNICATION NETWORKS

Commonly, contemporary communication network consist of an access and aggregation layer, the core network, and the interconnection to Internet exchange points (IXPs) or Tier 1 Internet service providers (ISPs). The general structure for mobile network access is shown in Figure 2.1 for general mobile users. These may connect to a cellular network only, be roaming between cellular and WiFi networks, or use WiFi networks only. The ISPs then connect to either IXPs or Tier 1 ISPs to provide connectivity to other networks. The cellular network providers are also often called mobile network operator (MNO). When a MNO sells its services to a virtual network provider it is called a mobile virtual network operator (MVNO).

The general structure of these networks forms a tree in the access domain. Only within the backbone of the ISP networks and in the core, redundancy is present, thus resulting in a mesh structure. This improves reliability and increases bandwidth by providing multiple paths to route traffic.

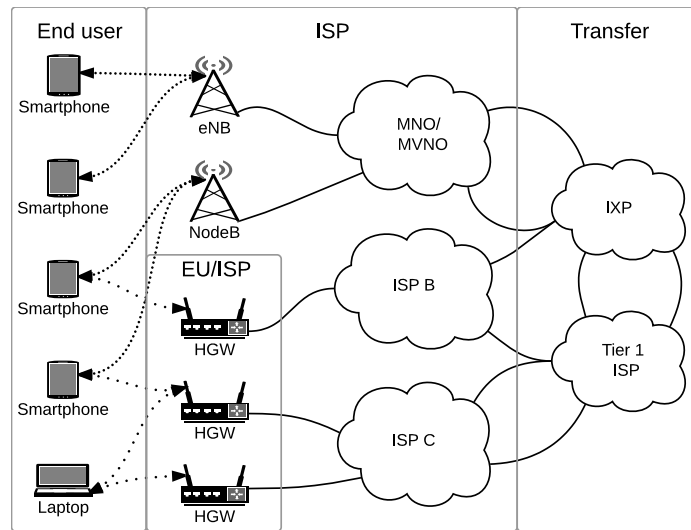


Figure 2.1: Structure of combined fixed/wireless and cellular network access

The exact definition of participating parties, their dependencies and differentiation are thus defined for the remainder of this work as follows:

**INTERNET SERVICE PROVIDER (ISP)** describes a company providing fixed/wired or wireless network access. This may include serving end users, companies, or other ISPs in the case of Tier 1 ISPs.

**MOBILE NETWORK OPERATOR (MNO)** describes the company operating the cellular network. Usually the MNO has its own customers, but often also sells spare capacity to MVNOs.

**MOBILE VIRTUAL NETWORK OPERATOR (MVNO)** describes a re-seller of a MNO's services. Usually MVNOs do not operate their own network. In rare cases they operate their own network management systems, but generally rely on services provided by the MNO [VSK15].

**TIER 1 ISP** describes an ISP providing worldwide connectivity to a large number of networks. Per definition, all networks must be reachable via a Tier 1 ISP.

**INTERNET EXCHANGE POINT (IXP)** provides OSI layer 2 connectivity to ISPs to exchange traffic with others at the same location. Thus, traffic may be exchanged at terms negotiated between peers, but eliminating the requirement of building dedicated links to the other party.

**TRANSIT PROVIDER** defines a service provider connecting different independent networks. This includes both IXPs and Tier 1 ISPs.

Each network can be divided into different network domains, showing different properties and functionality. The individual network domains, their definition and demarcation are defined in the following.

**RADIO ACCESS NETWORK (RAN)** describes the wireless domain of the cellular network between user equipment (UE), also called terminal or mobile device, and the 3G base station (NodeB) or eNodeB (eNB) in 4G networks.

**CELLULAR BACKBONE** describes the part of the MNO's wired network connecting base stations, NodeB, or eNBs to the external networks. This also includes any network functions (NFs) used to manage traffic within the network.

**CELLULAR NETWORK** describes a wireless network using dedicated licensed spectrum managed solely by a single entity within a geographical region. Cellular network as used throughout this work comprises both the wireless domain (radio access network (RAN)) as well as the cellular backbone, if not stated otherwise.

**INTERNET BACKBONE** describes the part of the network connecting different operator's networks. In the simplest case this might be an IXP, while in other cases multiple IXPs and transit providers are involved.

**WIRELESS NETWORK** describes any wireless technology used to receive or transmit data to and from the Internet. Wireless networks comprise of WiFi and cellular networks, but exclude Bluetooth or near field communication (NFC), as these are mainly used for local, device-to-device communication as opposed to worldwide connectivity.

To simplify referring to specific types of network access, these are defined in the following. Network technology is used to refer to the aggregation of different network generations as used in the cellular network, while network types distinguish between cellular and WiFi access. Hence, these are defined as:

**NETWORK TECHNOLOGY** describes the different network generations as defined by the 3rd Generation Partnership Project (3GPP) for cellular data access. Generally, only the differences between network generations are considered, as these include changes in network management and structure, while differences within a network generation are mainly restricted to bandwidth upgrades using higher modulation schemes.

**NETWORK TYPE** distinguishes between cellular network access and wireless network access using WiFi networks.

Details on the structure of these networks are discussed in the following. Section 2.1.2 discusses the structure of fixed network access, while Section 2.1.3 describes the structure of a cellular network and their implications on network measurements.

### 2.1.1.1 *Types of Network Access*

In the following, possible methods of network access as used throughout this work are defined. The different options depend on the used device, but also the context. For example, using a laptop from a café, network access would be classified as nomadic. Contrary, using a smartphone on public transport would be classified mobile.

**MOBILE NETWORK ACCESS** describes Internet access from mobile end-user devices attached to a packet based communication network using wireless network technologies like WiFi or cellular networks. Mobile network access particularly includes moving devices, thus also handover within and between networks are considered.

**NOMADIC NETWORK ACCESS** describes quasi stationary network access using devices frequently connected to different networks using various technologies. These range from Ethernet over WiFi to cellular networks. Contrary to mobile network access, nomadic network access describes the use of online resources from a stationary location.

For mobile or nomadic data access, different network types may be used. These range from wired network access as used for laptops and HGWs to wireless network access including WiFi and cellular networks as is common on smartphones. The resulting definitions are:

**FIXED/WIRED NETWORK ACCESS** describes Internet access of networked devices using digital subscriber line (DSL) or any other wire-based or optical methods to access remote servers. Devices may be conventional PCs, laptops using the Ethernet interface, or Internet of things (IoT) devices connected via Ethernet.

**CELLULAR NETWORK ACCESS** describes the access to data networks using the cellular network via a MNO or MVNO. This includes all available network technologies as defined by the 3GPP and excludes WiFi access or any other decentrally organized wireless network infrastructure.

**WiFi ACCESS** denotes access to the Internet using WiFi networks only. These include public and private access points (APs).

**WIRELESS NETWORK ACCESS** describes access to the Internet using any wireless network technology. This includes both WiFi and the cellular network. Applicable scenarios are mobile and nomadic network access.

Heterogeneous networks cause a number of challenges when using smartphones with multiple network interfaces. These are mainly vertical handover between different network types and multihoming when using two different connections simultaneously. These are defined as:

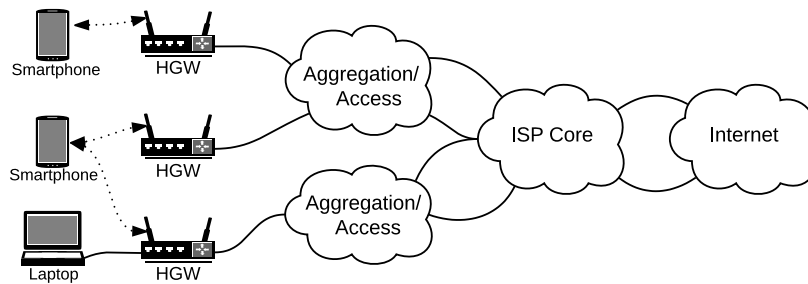


Figure 2.2: Structure of a fixed network provider

VERTICAL HANDOVER describes the handover of a mobile connection between different wireless technologies. A vertical handover may occur between a cellular network and a WiFi network when a user enters the coverage of the home WiFi network or vice versa.

MULTIHOMING describes mobile clients simultaneously connected to multiple networks. In the case of mobile handsets these commonly are different network types like WiFi and cellular networks. In other cases multiple cellular networks or WiFi networks may be used to provide uninterrupted network access while being mobile.

A vertical handover currently involves breaking of all existing connections, attaching to a completely separate network, and reestablishing connections after being reattached. Multihoming involves the same problem that networks are completely separate. This can be mitigated using MPTCP, keeping TCP connections between endpoints active while renewing the underlying TCP subflows. Additional details on MPTCP are given in Section 2.2.1.

### 2.1.2 Structure of Fixed Access Networks

Fixed access networks are the technology bearing the highest traffic [Cis16]. Different access technologies exist, with their own strength and weaknesses. Figure 2.2 gives an overview of the structure of a fixed network ISP. Generally, the access domain forms a star topology with only one link between the HGW and the aggregation switch or broadband remote access server (BRAS) in case of DSL access. The core network is usually built in a redundant manner. Also the uplink to IXPs or Tier 1 ISPs is usually redundant.

DSL, VDSL, and Vectoring provide a dedicated link to each endpoint. Data rates up to 100 Mbps are available, depending on location and distance to the closest DSL access multiplexer (DSLAM). Thus, no influence of neighboring users on the link bandwidth is expected. In this work, the general term DSL refers to the different available standards using twisted pair copper links to the end-user premises.

Cable-based network access provides theoretically higher bandwidths (e.g. 300 Mbps), but is a shared medium. Thus, high load on another link within the same shared domain is expected to reduce available throughput. Hence, these links show a high variability of available bandwidth over the course of the day.

An emerging technology, providing high link capacities to end users are passive optical networks (PONs). Here, an optical fiber is provided up to the end-user premises. Thus, dedicated bandwidth is available. Rates of  $>100$  Mbps are offered. These are generally limited by the hardware deployed on end-user premises and at the ISP, but may also be throttled by the ISP for service differentiation and to not overload the backbone. Mutual influence on available bandwidth is minimized, also compared to DSL, where electromagnetic interference may limit the capacity of neighboring links.

This work considers the fixed access network as a data pipe for end-to-end connections. Hence, the peculiarities of fixed access networks are generally not of importance as long as the network performance is not affected. The main aspect of interest is the connection point to other networks, thus defining the possible shortest routes to the remote server. In the case a particular effect is of importance, it is mentioned in the respective section.

### 2.1.3 *Structure of Cellular Networks*

As more and more mobile content is requested through the cellular network, more detailed knowledge of their infrastructure is required. Figure 2.3 shows the structure of combined 3G/4G networks using an evolved packet core (EPC). The figure limits the shown elements to packet based connectivity, thus omitting the circuit switched part of the network required for telephone calls within the 3G network. Furthermore, the network management functionality is abstracted into functional blocks regarding user management, accounting, and billing.

Generally, the network consists of the 4G RAN comprising of eNBs. These provide cellular network coverage to the end users. The eNBs connect via the backhaul network to the serving gateway (SGW), serving as the mobility anchor for mobile devices.

The 4G RAN is provided by the eNBs. These manage power allocation and scheduling of the long term evolution (LTE) resource blocks (RBs), thus defining which device may send and receive traffic in which time-frequency slots. Further, RB allocation between adjacent cells and between the macro and pico/femto cells is coordinated in the case the same frequency is used. For each mobile device, the physical and radio link layer parameters are defined, thus minimizing interference within the cell. Also handover between neighboring cells are handled by direct communication between eNBs via a direct interface. Thus the state of the mobile device, but also traffic in transmission queues can directly be transferred to the next base station, thus reducing overhead in the backhaul network.

The NodeBs, as the base stations in the 3G network are called, extend the functionality of the base transceiver station (BTS) as found in 2G networks. Their main improvement are



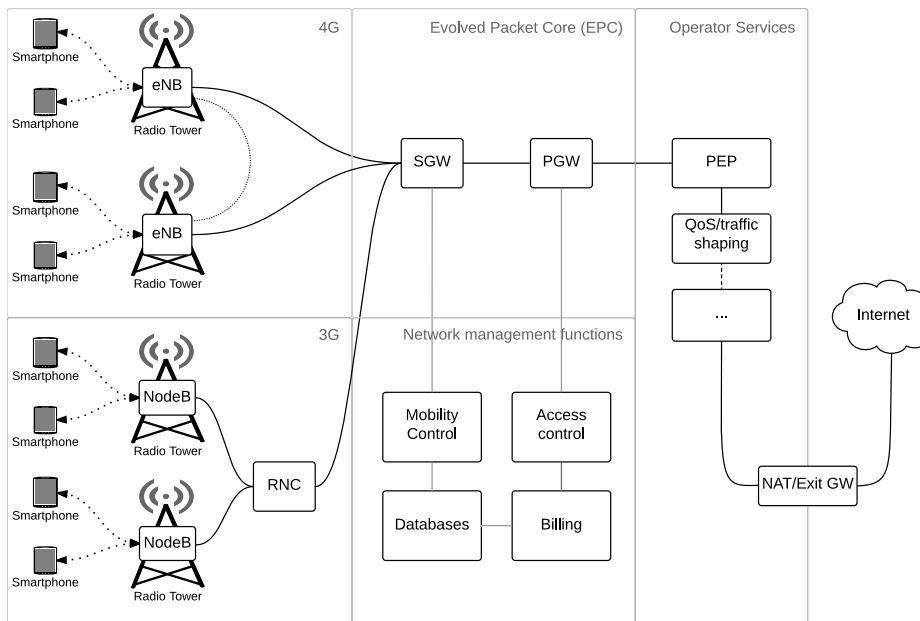


Figure 2.3: Structure of a packet switched 3G/4G network

the support of packet data transmissions, where in 2G only dedicated circuits are allocated to mobile devices requesting to transfer data. Still, the network control is conducted by the radio network controller (RNC), a centralized instance. This adds an additional hop into the connection.

The eNBs and RNCs connect to the SGW as part of the EPC, providing the cellular network specific functionality and control connectivity of mobile devices and manage traffic accordingly. The SGW connects to the packet gateway (PGW). The PGW enforces network access restrictions and serves as gateway to other IP-based services. The PGW is the first IP hop in the cellular network as seen from the mobile device. It connects the cellular network domain to the general IP services provided by the operator.

These operator services consist of NFs providing traffic shaping, Proxies, NAT, firewall, and any other functionality required to operate the network. For example performance enhancing proxies (PEPs) splits TCP connections between the mobile device and the remote server. Main functionality is to provide a cache of unacknowledged packets, thus reducing retransmission delays. This is particularly important in the cellular network, as latencies are comparably high. Thus, packets lost within the cellular network are retransmitted by the PEP, reducing the latency between request and response. Furthermore, the PEPs tune the TCP characteristics according to the underlying network parameters, e.g. adapting the number of packets in flight. Thereby the different retransmission methods in cellular networks can optimally be balanced. The problem here is that both the wireless domain and TCP provide reliability mechanisms. For example LTE automatically retransmits unack-

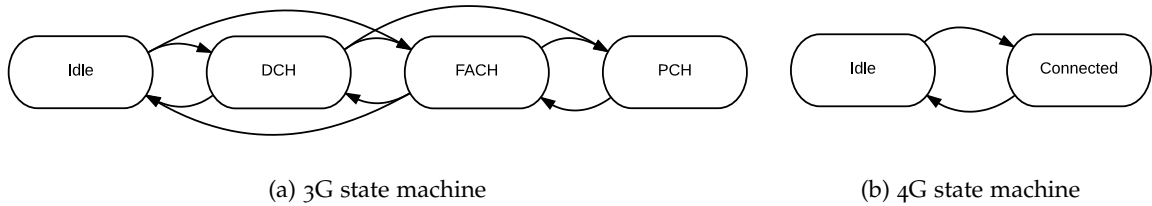


Figure 2.4: Comparison of 3G and 4G connection state machines (derived from 3GPP TS 25.331 [3GPP04])

nowledged packets after 8 ms (hybrid automatic repeat request (HARQ) retransmission). This may cause out-of-order delivery or duplicate reception of packets at the remote side. This reduces the receive window size as announced by TCP to the remote, thus reducing maximum throughput. If a packet is lost, the TCP connection switches to recovery mode, also reducing the bandwidth. By splitting the TCP connection, the PEP can avoid these negative impacts. Still, if the remote server is not available, the client still receives an *ACK* message, but the handshake fails, thus causing a timeout on the local device.

Further functionality inserted are traffic shaping and QoS management. This ensures that fairness between users is guaranteed, and high priority services are handled accordingly. Additionally, hypertext transfer protocol (HTTP) proxies and caches may be placed here, serving content from within the network, thus increasing throughput while reducing latency and external traffic cost.

Finally, the network address translation (NAT) server or exit gateway translates the internal, private IP addresses to public addresses. This serves both to protect the user from unwanted external traffic, and reduces the number of public IPv4 addresses required to support the network. The blocking of external traffic is beneficial to the mobile user, as these cannot be billed for externally initiated data transfers. Furthermore, the energy consumption of the mobile terminal is reduced by not waking the device up unnecessarily. For the operator, this is advantageous, as it prevents reconnaissance of the internal network. Still, this introduces additional complexity for mobile devices to be notified by cloud servers. Therefore, these must initiate a TCP connection with long TCP timeouts to a remote server. The NAT is furthermore still required, as Internet protocol version 6 (IPv6) is not fully rolled out yet.

Important for network performance, but also energy efficiency are the supported data rates by each network technology. These vary also within network technologies. Generally, the wireless interface is defined by the 3GPP RAN technical specification group (TSG). Each standard defines traffic classes which may be supported by hardware vendors. On connection, and based on received signal strengths, the appropriate modulation and possible channel aggregation are selected.

Besides the available data rates, also the connection states and timeouts specified for each technology are important. The connection states are defined by a state machine on both client and network side, while the timeouts are operator configurable. Hence, the messaging overhead within the core network, and thus the load on the management entities can be balanced with the energy consumption of end-user devices. The respective state machines are shown in Figure 2.4. The connection states in a 3G network consist of the idle state, dedicated channel (DCH), fast access channel (FACH), and paging channel (PCH). When data is to be transmitted by the mobile device, a DCH is opened. After no data is transferred for a configurable duration, the device may enter FACH. From this state, the DCH can be opened with minimal delay. Still, also the device power consumption is high, as the device may need to wake up within a short time to receive additional incoming data. A more power saving option is the PCH, which may be reached either via the FACH or directly from the DCH depending on operator configuration. This is used to be notified of further data by the network in an energy efficient manner. Changing back to idle mode is only possible via the FACH.

The 4G connection states are simplified compared to 3G. When the mobile device is in idle mode, only the PCH is monitored. The detailed behavior is defined in [3GPP14; 3GPP16]. This reduces the energy consumption by waking the modem only up at specific times as defined by the MNO. By omitting these idle states as used in 3G networks, the messaging overhead in the backbone is reduced. Further, keeping track of the various device states and synchronizing the state machines is simplified. This further improves agility between active and idle periods. Only thus the latency requirements of 4G networks are possible.

#### 2.1.4 Network Performance Metrics

The performance of communication networks can be measured using a number of different metrics. These KPIs of cellular network performance are defined in the following. Beginning at the lowest layer of wireless networks, the signal strength is the first indicator for its availability.

If the network is available, latency and round-trip time (RTT) are the simplest metrics to describe network quality. The latency describes the one-way delay between sending a packet and its reception at the destination. As this is impractical to measure due to different clocks, their jitter and drift, usually the RTT is measured. The RTT describes the time it takes for the packet to reach the destination, be processed, a response is created, and received by the source of the original packet. The RTT always describes end-to-end measurements. Depending on the service being tested, the RTT includes different network layers. An Internet control message protocol (ICMP) *ping* message is directly processed in the remote network stack, while for example HTTP RTTs are generally served from an application residing in user space.

Using ICMP packets with limited lifetime, ICMP time exceeded errors are returned to the originating host. By evaluating the sender IP and the RTT, routes through the network can be mapped. This is implemented in the various *traceroute* measurement tools. Often, these also resolve the IP address, thus simplifying network mapping.

The throughput in the following defines the maximum data transfer capacity between two hosts in one direction. This may be measured using TCP or UDP transfers. A number of different tools for the throughput measurement are described in [PMD+03]. These are also compared in detail in Section 3.2.2.

These objective network metrics or KPI are summarized under the term QoS, describing the requirements of a given service. These may define thresholds above which a service is expected to work. Quality of experience (QoE) extends QoS to also include the end user by analyzing the subjective service quality. This is usually done in crowd-sourcing studies, polling a large number of participants to rate the quality of a service on a scale of one to five. These ratings are then used to optimize the delivered quality and map the requirements back to QoS classes. The cellular service quality as used in this work refers to the overall quality of cellular service including signal strength, available network technologies, RTT, packet loss, and throughput, thus being similar to QoS, but not limited to a particular service.

The definitions of the derived network performance metrics are given in the following:

**SIGNAL STRENGTH** describes the received signal strength of wireless network beacons at a mobile device. These are in the case of 3G networks given as received signal strength indicator (RSSI) in arbitrary strength unit (ASU), which can be converted to received signal power as defined in [3GPP08]. In the case of LTE this is given in reference signal received power (RSRP), indicating the received reference signal power relative to the overall received power in the respective band, which may also be converted to ASU.

**LATENCY** describes the time required to a packet sent by the source to reach its destination. Depending on the type of packet (e.g. ICMP/TCP) the network layer triggering the packet may differ, and thus also the measured duration.

**ROUND TRIP TIME (RTT)** describes the time the response for a packet sent by the source and triggering a reply on the destination to be received by the source. The RTT includes the latency on both directions plus the processing time of the destination.

**THROUGHPUT** in this work describes the one-way application layer goodput of an end-to-end connection, if not otherwise stated.

**QUALITY OF SERVICE (QoS)** describes the nominal service parameters as can be measured from an end-user device. These include network availability, available network technology, signal strength, RTT, and throughput.

QUALITY OF EXPERIENCE (QoE) describes the subjective service quality as experienced by the end-user. As the systematic assessment of the user-perceived service quality was not focus of the thesis, quality of experience (QoE) is used as an abstract concept, highlighting the impact of network behavior and management decisions on the end user.

CELLULAR SERVICE QUALITY describes the end-to-end network performance as experienced by the end-user. In the context of this thesis it is used as an abstract concept, including QoS and to a certain extent also QoE by respecting the energy consumption of the device. Furthermore, the cellular network related parameters signal strength, available network technology and packet loss are included.

## 2.2 EMERGING NETWORKING APPROACHES

Over the past years, a number of network virtualization approaches have emerged. These range from optimizations of end-to-end performance (MPTCP) to virtualization of networks. The main goal of the virtualization solutions is increasing the flexibility of network functionality by abstracting network functionality from control logic. Thus, the control of the network can be centralized, while the underlying network substrate may be kept proprietary, with the limitation of implementing a standardized application programming interface (API) for the controller to modify functionality. The main areas of development are SDN, separating the data and control plane, and NFV, virtualizing extended network functionality like proxies which were previously deployed as hardware devices.

In the following, the fundamental operation of MPTCP is discussed in Section 2.2.1, while SDN and NFV are discussed in Section 2.2.2 and Section 2.2.3 respectively.

### 2.2.1 *Multipath TCP*

Multihoming is made possible using MPTCP. The concept is to split TCP connections as provided by the kernel to user space applications to multiple underlying subflows using different interfaces. This was first introduced by Barré et al. [BPB11] to improve throughput between two servers using multiple parallel network links. MPTCP also improves the reliability of connections by automatically reestablishing lost connections.

Figure 2.5 shows an exemplary application of MPTCP between two hosts using multiple interfaces. The application connects to a TCP socket provided by the kernel. The MPTCP control logic coordinates communication with the remote via TCP subflows. First, the feasibility of using MPTCP is tested during the handshake by adding the MPTCP options in the TCP options fields. If the handshake is successful, the first TCP subflow is used to establish additional flows. These subflows may either be established between the same interfaces (e.g. same IP, different port), or other interfaces at the same host. The current default behavior is to wait for additional configurations. An option is to establish a full

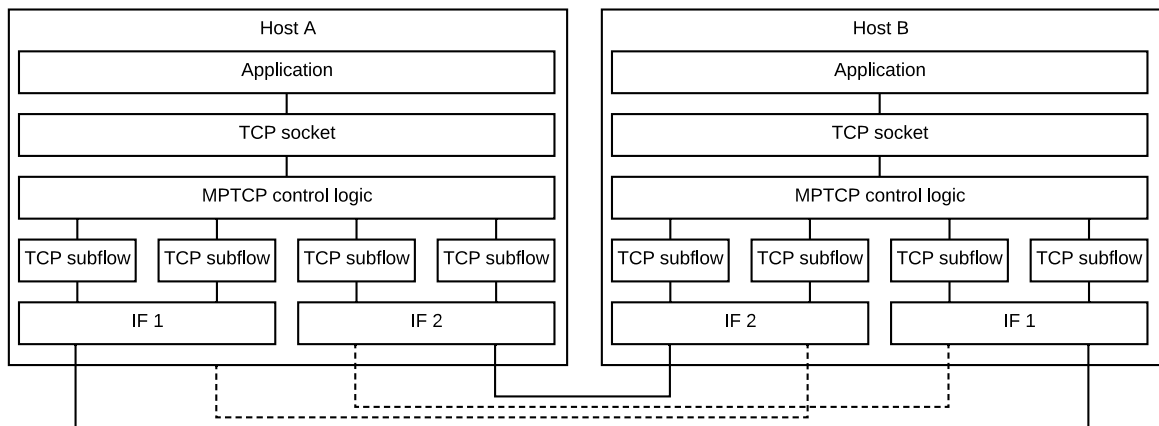


Figure 2.5: Overview of the MPTCP between two hosts

mesh between all available interfaces, thus maximizing the available throughput. This is visualized in Figure 2.5 when including the dashed links.

The scheduling of packets between both interfaces is managed by the scheduler. The default option is to use the interface with lower RTT until the congestion window is full, then adding interfaces in order of increasing RTT. Other options are the *round-robin* scheduler, randomly distributing packets on the available interfaces, and the *redundant* scheduler, sending all packets on all interfaces, thus achieving the minimum RTT while preventing outages caused by failed links.

### 2.2.2 Software Defined Networks

Software defined networking describes the approach of separating the control and data planes of communication networks. Where in the past both functions were integrated in one device (e.g. switches/routers being configured by the administrator to work independently), SDN defines switches as simple devices for execution of the network management decisions configured in a central controller. A general comparison of both approaches is shown in Figure 2.6.

One of the most prominent implementations of SDN is OpenFlow. The overall concept and a first implementation are introduced in a paper by McKeown et al. in 2008 [MAB+08]. Based on this concept, the OpenFlow switch specification [HPT+09] was published. This specification defines the required interface between switch and controller, as well as the matches, counters, and actions that must be supported by the switch.

Packet matching is based on the packet header of each packet. Depending on the protocol, the fields applicable for matching a packet may consist of Ethernet addresses and protocol only, in case of IP packets, source and destination IP addresses, and in the case of

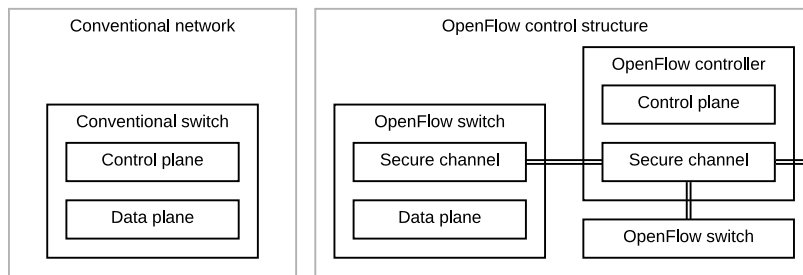


Figure 2.6: Comparison of conventional and OpenFlow architecture

TCP, UDP, or ICMP, also the source and destination ports. Further matches may include the ingress port, as well as the protocol types for higher layer matches.

For each rule, counter fields are defined, later used by the controller to collect traffic statistics. These are kept on port, rule, and flow-table level, thus allowing to later assess the amount of packets handled by each rule.

Finally, the OpenFlow protocol specifies actions to be applied to each packet. The actions to be applied may be empty, a single action, or a list of actions. These may be sending the packet out at a specific egress port, rewriting header data, or discarding the packet. In case no rule matching the incoming packet is installed, the respective packet is forwarded to the controller. This then examines the packet to handle it accordingly. Thus, the controller may install additional flow rules on the respective switches to handle further packets of the affected flow.

The switches are configured to connect to a central OpenFlow controller to receive their configuration. This is based on a transport layer security (TLS) connection, thus ensuring integrity of the messages and authenticity of the remote machine. The exchanged messages consist of periodic heartbeats, and the required configuration and data collection messages. Their details are described in the respective OpenFlow switch specifications as may be found on the Open Networking Foundations website<sup>1</sup>.

### 2.2.3 Network Functions Virtualization

NFV describes the virtualization of formerly hard-wired NFs like proxies or firewalls. These are usually inserted into a link, configured once, and left there to handle the incoming traffic accordingly. This comparatively static behavior causes a number of problems. First, their location in a chain of NFs is static, and cannot easily be changed. Secondly, their capacity is defined by the hardware of the respective device. Thirdly, their functionality and configuration interfaces are defined by the vendor of the device, and thus

<sup>1</sup> <https://www.opennetworking.org/> accessed 2017-03-09

may be limited and different between similar devices available from other vendors. These problems are to be solved by NFV.

By running virtual network functions (VNFs) on commercial off-the-shelf (COTS) hardware, capacities may easily be upgraded, and adapted to the current demand. Further, the 'wiring' becomes more flexible, by allowing the network operator to create virtual chains of VNFs depending on the traffic requirements. Specific functions can be overridden, or adapted according to the requirements of a given flow. Thus, the efficiency of the deployment can be further improved. Finally, VNFs of different vendors may be used as long as their interface is standardized. Thus, the implementation optimally fitting the available hardware may be selected, maximizing performance while minimizing overhead.

NFV can be seen complementary to SDN, where SDN defines the path of the traffic through a network, while NFV defines more complex operations to be performed on the processed traffic. These functions may include PEPs as used to improve TCP performance in wireless networks, load balancers, firewalls, or similar.

The concept of NFV was introduced by a number of network operators in [NFV12]. The standardization of NFV is promoted by the ETSI Industry Specification group for Network Functions Virtualization (ETSI ISG NFV)<sup>2</sup>.

### 2.3 ENERGY IN COMMUNICATION NETWORKS

The energy cost of communication networks is an increasingly important topic. Hence, the following sections give an overview of metrics involved in describing the network energy cost and background on measuring power consumption.

First, the different energy metrics in networked systems are summarized in Section 2.3.1. The fundamentals of power measurements are shortly summarized in Section 2.3.2, thus providing the required background for understanding the power measurements as described in Chapter 4.

#### 2.3.1 *Energy Metrics in Networked Systems*

The overall energy required by ICT may be classified in the energy contained within devices, the *emergy* and the energy transformed to heat during operation, the *energy*. The *emergy* is the energy required to produce the given product, and contains the full energy consumed during manufacturing and transport of the respective device. Contrary, the *energy* is consumed during operation, and depends on durability of the device, operating cycles, and load. Hence, when discussing energy requirements of networks, a clear definition on the scope is required.

Raghavan et al. [RM11] analyze the energy and emergy in ICT. Particularly power hungry, compared to the embedded energy, is communication infrastructure like switches,

<sup>2</sup> <https://www.sdxcentral.com/nfv/definitions/etsi-isg-nfv/> accessed 2017-03-09



routers, cellular base stations, and the cloud. Comparatively frugal in daily use are smartphones. Still, their production requires 30 times the energy they consume during operation [RM11]. Here, optimal trade-offs must be found, considering that ICT is estimated to already use 10% of the global energy consumption [Mil13].

The energy consumption of networked systems can be determined in different ways. By estimating the number of deployed devices, and representative power consumption values for each, the overall power consumption of a deployment may be estimated in a bottom-up approach. The opposite, the top-down approach, bases the estimates on the overall produced electricity and uses the power consumption of industrial branches to estimate the network power consumption. Each approach has different challenges. The bottom-up approach is inaccurate as the deployed number of devices can only be estimated. Furthermore, the energy consumption for each device class can only roughly be estimated. The top-down approach must find ways to exclude other energy consumed in different branches, thus is also restricted to estimates.

For all these approaches, the demarcation of system boundaries is important. For example, the power consumption of end-user devices may be included in the network power consumption in some cases, while being excluded in other studies, thus complicating comparisons. Hence, a clear demarcation, and communication of this is required.

### 2.3.2 Power Measurements

The power consumption of devices can be measured at different points. This depends on the purpose of the measurements. Determining the power consumption in specific states, or building a regression based model, the AC power consumption may be sufficient. If the power consumption of internal components is of interest, or a high time resolution is required, internal measurements may be used. Still, these measurements exclude the losses in the power supply, often showing an efficiency of between 85% and 95%. Hence, for each measurement tasks, the optimal measurement procedure must be found.

When measuring the AC power, also the power factor (i.e. the shift between AC voltage and current) and possible deformations of the consumed power waveform must be considered. Hence, meters measuring the root mean square (RMS) of the waveform are required to also include these effects.

When measuring the secondary power consumption in the device, the sampling rate must be sufficient to also include effects of short peaks in power consumption. This is particularly important when measuring electronics, as their power consumption may change considerably within short time frames. Hence, the behavior of the device under test (DUT) must be considered carefully before planning the measurements, possibly initially using meters with a higher accuracy to determine feasible measurement approaches.

Power measurements in general require the knowledge of supply voltage and the current drawn by the DUT. In case a stabilized power supply is used, the voltage can be

assumed to be fixed. Still, when high currents are drawn, the wiring and inductances may affect the measurement accuracy. Hence, it is generally recommended to measure both supply voltage and current at the point of interest.

Generally when measuring, care must be taken to not affect the measured quantity with the measurement. Hence, the measurement with the highest influence on the measured quantity should be placed closest to the DUT. These are affected by the measurement procedure and equipment.

For voltage measurements, the input impedance of the meter affects the measured quantity by connecting a measurement resistor parallel to the DUT. Still, in modern meters using high quality amplifiers, resistors with a high impedance (e.g. 1 M $\Omega$ ) are used. Thus, the additional reduction of measured voltage by increasing the current is minimal.

Current measurements are comparatively challenging. Possible approaches are direct measurements inserting a shunt into the supply line, or indirectly measuring the magnetic field caused by the current. The direct measurements have the advantage that the voltage drop over the shunt can directly be measured and is proportional to the current. Still, correctly sizing the resistor is challenging, in particular if the variation of observed currents is high. Considering that the voltage drop is proportional to the current, high currents may reduce the supply voltage as seen by the DUT significantly if the shunt is too large. Still, to maximize measurement accuracy, the voltage drop should be maximized. Hence, the maximum size of the shunt is limited by the voltage drop letting the DUT also work under full load. The inaccuracies caused by a small resistor can partially be minimized by using a voltmeter with a high resolution.

Using a small shunt, also the noise on the measurements is augmented. This is particularly severe, when using unshielded cables close to switched circuits. Hence, a measurement amplifier should be placed as close to the measurement shunt as possible. For this, integrated circuits (ICs) are developed, providing an amplified signal, or a measured voltage via a digital interface. Using these requires designing a circuit board and interfacing it with the measurement equipment to collect and synchronize the desired data.

Indirect measurements using Hall sensors can be used to measure higher currents with minimal effects on the measured system. Still, the measured effect is small, also requiring the use of measurement amplifiers close to the DUT. Their advantage is that it is not required to interrupt the power supply to insert a measurement shunt. These Hall measurements use split iron core placed around the power line of interest around which a coil is placed transforming the current to a proportional voltage.

Summarizing, when measuring the power consumption, the current measurements are most critical. Selecting a measurement shunt, the voltage drop should be limited to allow the device to operate, while still causing a signal large enough to be measured by the available meters. Furthermore, the sampling frequency must be high enough to include all effects caused by short term variations of the power consumption.

## RELATED WORK

---

RESEARCH in communication networks has resulted in a large variety of different approaches, first proposing and implementing mechanisms, then improving their performance. Two aspects of networking are relevant for this work, the power consumption and the determination of network performance. Based on a combination of both, advanced optimization algorithms as described in literature are compared.

The power consumption of communication networks is of growing importance, as constantly increasing data volumes are transferred [Cis16]. Already 10 % of the world energy consumption are attributed to ICT in the widest sense [Mil13]. Simultaneously, the traffic demand is predicted to further rise significantly [Cis16]. Major contributor to the network energy consumption are the access networks [VHD+11]. For fixed network access, the main contributor are HGWs, while in cellular networks the base stations are the main power consumer [VHD+11]. Considering the case of a cellular network provider, the base stations alone consume 72 % of the overall power [CMo12].

For conventional network equipment, detailed studies on their power consumption are available [ODL14]. Still, the energy cost of emerging network technologies like MPTCP, SDN, and edge computing using SBCs are not well analyzed yet. Hence, the related work covering these technologies is described and compared in Section 3.1. Based on this literature review, the measurement and modeling procedures applied to the devices analyzed in Chapter 4 are derived.

In particular in mobile scenarios, the power consumption is one of the restricting factors of mobile data access. Currently, 31.2 % of the energy of a smartphone is consumed by the different wireless modems [CDJ+15]. Still, the consumed energy depends, besides the network technology, also on network performance parameters like RTT and throughput. Hence, Section 3.2 analyzes related work covering the measurements of communication networks. After a general overview of tools, the methods developed for the analysis of mobile networks are presented, from which the measurement methodology used in Chapter 5 is derived.

Optimizing the energy consumption of mobile data access is an active field of research [RDC+15]. A large number of proposals improving mobile network performance have been published, focusing on different types of traffic and networks. These are compared in Section 3.3 with respect to possible energy savings. Out of these, the two with the highest energy saving potential are selected for a comparison and evaluation of their effectiveness, which is described in Chapter 6.

### 3.1 POWER CONSUMPTION OF NETWORK DEVICES

The power consumption of communication networks is an increasingly important topic. Consequently, a large number of publications cover different aspects of network energy consumption. A general overview of the energy consumption of different network access technologies based on numbers as published in literature is given in [VHD+11]. Combining the power consumption of the devices and their maximum data rates, their relative energy cost is derived. Caused by the increasing data rates requested by end users, also the share of the core network is predicted to grow [BBD+11].

In the following, different aspects of the network energy consumption are analyzed, relating to the area covered by the power models as described in Chapter 4. First, the related work analyzing the power consumption of ARM processors and their application are discussed in Section 3.1.1. An active research area is their use in high-performance computing, promising to reduce the overall energy consumption compared to conventional high-performance computing (HPC) using x86 processors. This is also expected to relieve cooling requirements (e.g. [MOF15; SRS16]). An increasing number of publications also analyzes the energy efficiency of SBCs for the processing of big data workloads, showing promising results (e.g. [MH15; ZLH+16]).

The second area of interest is the power consumption of switches. Section 3.1.2 discusses related work considering the energy consumption of conventional networking hardware. Here, the focus is on establishing the behavior of currently deployed hardware, and getting an insight into promising measurement approaches. Based on these, the differences when analyzing OpenFlow switches are discussed.

Section 3.1.3 covers publications focusing on the energy consumption of smartphones. These analyze and establish power models to develop optimization approaches based on these. Most of the related work focuses on 3G networks and improvements thereof. Considering the progressive evolution of both smartphones and the cellular networks a reevaluation of their potential is recommended.

Finally, Section 3.1.4 compares the presented approaches based on common parameters. Thus, their strength, weaknesses, and suitability for modeling the power consumption can easily be derived.

#### 3.1.1 *Single-Board Computers*

The power consumption of SBCs was not yet focus of comprehensive analysis. Still, the performance and energy-efficiency of ARM processors is analyzed in a number of publications (e.g. [SRS16; ZLH+16]). Their main focus is the application of low-power ARM processors in HPC, thus reducing the power consumption of computing nodes. These direct energy savings result in successive gains in power supply and air conditioning systems, as these then can be designed smaller, thus requiring less power themselves.

Jarus et al. [JVO+13] compare the power consumption of ARM, AMD and Intel processors using a set of standard benchmarks. They conclude that for time-critical computations the x86 platform is most suitable, while for energy efficient computation the ARM Cortex-A9 achieves favorable results with a significantly lower power consumption while executing the same tasks. Still, the overall processing time is increased compared to the Intel processor. Similar results are confirmed by Maqbool et al. [MOF15].

A similar study is conducted by Selinger et al. [SRS16], also including memory bandwidth into their analysis. Besides matrix multiplications as in [JVO+13], also the performance of floating point operations is analyzed. They conclude that neither acquisition cost nor energy efficiency are currently superior to x86/64 hardware. Lorenzon et al. [LCB15] similarly compare the power consumption of an Intel ATOM and an ARM Cortex-A9 for core utilization and memory access.

Malik et al. [MH15] analyze the power consumption of an extensive set of big data workloads on Intel ATOM and Intel XEON setups. They monitor the execution time and power consumption for different workload sizes on both setups. Based on this, the energy delay products (EDPs) for all workloads are calculated. They conclude that for most applications the deployment of ATOM processors consumes the least energy. Zhao et al. [ZLH+16] run similar tests comparing Intel Edison boards with a Dell R620 server. The analysis compares a cluster of 16 SBCs with a single x86 server, consuming a similar amount of power. The workload consists of web service and MapReduce applications. They state that the Intel Edison cluster serves 3.5x the load consuming the same energy at linear scalability, thus agree with results derived by Malik et al [MH15].

Nunez-Yanez et al. [NL13] analyze the low-level performance of the ARM Cortex-A9 for realistic workloads. They record the CPU and memory power consumption under different scenarios as included in the SPEC CPU2000/H.264 benchmark. The system performance is recorded using system performance counters. Thus, a fine-granular model of the CPU power consumption is generated. This can then be used to verify the correct working of processors, as well as input for hardware and software design decisions.

A more theoretical work is conducted by Tudor et al. [TT13]. Based on the flow of a program, traces describing the execution flow within the ARM Cortex-A9 are derived. These are then used to estimate the power consumption of the processor. This is achieved by running small benchmarks putting the CPU into particular states (i.e. work/stalling cycles) by executing computation or memory heavy tests. Knowing the power consumption of these states for the possible combinations of processor frequency and active processor count, the power consumption of the workload is derived. Applying these models to specific workloads, the optimal configuration of the system (e.g. processor frequency and count) minimizing the power consumption can be found.

Blem et al. [BMS13] analyze the differences between different instruction sets on four platforms. They select two reduced instruction set computing (RISC) and two complex instruction set computing (CISC) processors for comparison. Over the course of the study

a large number of different benchmarks is run, and also the generated machine code analyzed. Based on these, the authors conclude that on recent systems no difference in power consumption and performance between both instruction sets is visible. Main differences observed are caused by micro architecture design decisions, and the implementations running on the machines.

Besides analyzing the performance under lab conditions, also the effects as observed in larger deployments are of interest, in particular when focusing on a complex system of WiFi APs. Hence, Gomez et al. [GRR+12] present their cost-efficient and accurate power meter *energino*. The power consumption of selected APs is determined based on the WiFi modulation and the corresponding bit rate for different packet sizes and rates. By correlating the packet rate for fixed size packets or the packet size for fixed transmission rates with the measured power consumption, a linear and a logarithmic power model are derived. Here, additional work is required, correlating the energy consumption with the actual network performance, resulting in a single power model depending on packet size and rate.

Depending on the study, different results are derived. Mostly, the energy efficiency is determined to be better on ATOM processors compared to conventional x86/64 processors. Still, the real benefit of ARM processors cannot conclusively be determined. Depending on the study, the results are mixed [SRS16], or the computation time is largely increased compared to conventional deployments [JVO+13]. Here, the workload seems to be the determining factor for both energy efficiency and performance. Considering the huge energy gains determined in [MH15] and [ZLH+16], where distributed big data workloads based on real-world use cases are run. Contrary, Jarus et al. [JVO+13] and Selinger et al. [SRS16], using conventional benchmarks, do not identify energy savings. Hence, the targeted application of the deployment should be considered when selecting the underlying hardware platform. Thus, it may be summarized that the use of low-power ARM processors for distributed big data workloads decreases energy consumption at similar performance, while the application of ARM processors in HPC is questionable.

Contrary to these studies, the power models generated and described in this work focus on deriving a general purpose power model of the tested platforms using readily available system monitoring values. Thus, the energy cost of any workload on the SBC can be estimated, given the system utilization is known. In contrast to the above power measurements, the presented power models are dynamic, meaning that the influence of system load on the power consumption can directly be derived. Thus, accurate estimates of the power consumption of a larger deployment, but also optimizations based on the known energy models are possible.

### 3.1.2 Routers and Switches

Switches and routers build the backbone of any communication network. Hence, their performance and energy consumption are the focus of a number of different studies. The following section summarized related work focusing on the power consumption of conventional switches, their performance evaluation, and approaches promising to increase network energy efficiency. Based on these, the requirement of analyzing the power consumption of OpenFlow switches is derived and a promising measurement setup deduced.

Hlavacs et al. [HDP09] analyze and model the power consumption of conventional residential and professional network switches. They show that the dynamic power consumption of residential switches is mostly static or slightly decreasing under load. The variability between idle and saturated power consumption is less than 5%. In the case of professional switches, also this difference is negligible, with a variability of the model of less than 1%. Still, the number of active ports has a considerable influence on the overall power consumption, in the case of one of the professional switches doubling the idle consumption.

The power consumption of core routers is analyzed by Chabarek et al. [CSB+08] on the example of Cisco's 7507 and the GSR 12008. The main influence on the power consumption of the devices is determined to be caused by the configuration of the chassis and built-in line cards. These may increase the idle consumption by a factor of four. For a given configuration, the dynamic part of the power consumption caused by traffic is again comparatively low. For the 7507 the increase in power consumption under load is lower than 4%. The authors conclude that choosing a configuration close to the maximum demand minimizes power consumption. Still, this approach is static, and thus no adaptation to the actual load is possible.

An automated approach determining the power consumption of various network switches is presented by Mahadevan et al. [MSB+09]. They test their approach on a hub, three edge switches, a core switch, and a WiFi AP. Based on the collected data, the derived power models are presented. Their main contribution is the measurement environment, automating the analysis of the device power consumption and power model generation to a large extent. The measurement and modeling of the power consumption of the OpenFlow switches as presented in Section 4.2 extends this setup by adding the OpenFlow functionality to dynamically configure the DUTs to measure the effects of different OpenFlow rules on power consumption. By increasing the problem space with these additional functions, the resulting setup is slightly more complex and requires more measurements to determine the influence of different combinations of parameters.

Based on publicly available power models, Orgerie et al. [OLG+11] build an energy simulation environment. After entering the topology of interest, the power consumption of the full network can be determined. The framework supports routers, switches and hosts, thus resulting in an accurate view of the network. Besides static cost, also the

influence of traffic is calculated. Thus, the full power consumption of a wired network deployment can be derived.

Caused by the large variety of power models, Bolla et al. [BBD+14] develop an abstraction layer, simplifying the high-level analysis and optimization of networks regarding energy consumption. Their abstraction layer works in a hierarchical way, combining models of underlying layers into a common model for the full network. This approach also includes certain control of the local device or domain, which is also abstracted and accessible to the upper layer network management. Thus, the power consumption can be controlled using centralized approaches, while leaving the implementation details (e.g. active links, link speeds, active devices) to the lower layers.

In conventional networks, the power consumption of routers and switches must be accepted as is, because dynamic reconfigurability is highly limited. Nonetheless, a number of approaches are proposed, reducing the power consumption of large scale wired networks [BCL+10; VNS+11; NSA+13; YWX+13; LAH+15]. Their focus generally is disabling interfaces, reconfiguring link speeds, and shutting down devices. These generally formulate optimization problems, calculate exact solutions, and derive heuristics to optimize the network in a computationally efficient way. Still, the feasibility of these on current hardware is often not given, as devices usually do not support sleep modes. Neither do they have common configuration interfaces, thus complicating energy savings.

A relatively new approach on network management is OpenFlow [PLH+11]. Hence, the majority of publications cover its implications on network management, and the measurement of KPI like RTT, throughput, and re-programmability of these devices. In the following, a number of approaches using and analyzing OpenFlow are presented, with the goal of deriving suitable measurement approaches in SDN environments.

A framework for the performance evaluation of OpenFlow devices is proposed by Rotsoos et al. [RSU+12]. They describe a solution assessing the performance for different applied rule sets as well as delay and packet loss. This is achieved by controlling the device via the OpenFlow protocol and simultaneously monitoring traffic on the data plane. Their main observations are that performance and delays heavily depend on the used implementation, and that an evaluation of these metrics is only possible using data plane measurements.

Heller et al. [HSM+10] propose an approach maximizing link utilization by controlling the links in a data center topology using OpenFlow. Thus, the capacity of the installed devices can be better utilized by overcoming the limitations of the spanning tree protocol, limiting the use of active links to a tree topology. By using all available links, and thus increasing available bandwidth, the same hardware may be used for a longer time. A similar approach is implemented and tested by Prete et al. [PFC+12].

Exploiting the increased flexibility of OpenFlow, a number of problems inherent in conventional networks can be solved. Still, re-reconfiguration of network interfaces is not part of the specification. Another remaining challenge in the energy optimization of wired net-



work infrastructure is the comparatively high idle consumption and missing sleep modes. Here, only device vendors can improve the energy efficiency of the available hardware. Only then, the results derived using theoretical models and simulations developed under the term *energy proportional networking* [AMW+10] can be transferred to actual network deployments.

As currently the cost of the additional functionality provided by OpenFlow is not well known, Section 4.2 exemplarily analyzes the power consumption of a hardware and a software OpenFlow switch. The focus of this analysis is to determine the cost of specific functionality and possible implications on switching performance, thus giving additional insight into the feasibility of implementing the proposed approaches on currently available hardware.

### 3.1.3 Smartphones

According to a UK study, smartphones have become the most important device to access the Internet [Ofc15]. Similarly, the consumed data volume is increasing significantly [Cis16]. This increased demand is also visible in the network energy consumption [BBD+11]. Still, also on smartphones, the modems consume a significant fraction of the available energy [CDJ+15], hence is a major impact on the battery life time of the mobile devices. Therefore, a number of studies analyze the impact of mobile data communication on power consumption of smartphones, and based on their observations propose optimization approaches. In the following, first the studies analyzing the energy consumption of smartphones are introduced. Based on these observations, energy optimization approaches are discussed.

One of the earlier studies analyzing the power consumption of smartphones is conducted by Carroll et al. [CH10]. They analyze the component power consumption of an Openmoko Neo Freerunner by using circuit compartmentalization as introduced by the hardware developers. Instead of just connecting these, measurement shunts are inserted. Thus, the power consumption of each component is measured independently. For reference, also the power consumption of the HTC Dream and the Google Nexus One are recorded for different use cases. As their hardware layout does not provide these measurement points just the overall consumption and resulting models for different use cases are presented.

Similar measurements are repeated by the same authors on the Samsung Galaxy S3 with the focus of identifying trends in energy efficiency over time [CH13]. The power consumption of the individual components is identified, but no concluding power model derived. Still, some conclusions are drawn, namely that the power consumption during certain usage scenarios increases only marginally, while the maximum possible power consumption of the device increases.

Chen et al. [CCD+14] analyze the energy efficiency development of the Galaxy S lineup. The analyzed devices are the Samsung Galaxy S1, S2, S3, and S4. Over the range of four generations, the power consumption caused by a selection of representative Apps has increased by 38%, while the interface power consumption for the same network technology in each subsequent generation is reduced. Still, the newly introduced network technologies have a higher relative cost than the established ones.

These approaches describe the power consumption of the selected smartphones under the given load, but do not provide means to analyze the power consumption of other than the presented scenarios. The first, readily usable power model of a smartphone was published by Zhang et al. [ZTD+10]. The analyzed device is the HTC Dream. The authors publish detailed models for the different components, depending on their utilization. The measurements are validated by comparing the derived parameters between different devices of the same type. Based on these system utilization values and measured power, the device power consumption is modeled. Comparing the power consumption derived using the model, and the one measured using an external power meter, average errors of less than 10% are achieved.

Ding et al. [DWC+13] analyze the power consumption of wireless data transmissions on both WiFi and 3G networks depending on signal strength. They show that users are often connected to networks with unfavorable signal strength, and delaying data transfers improves the energy efficiency of communications. The power consumption of WiFi transmissions is up to 34% lower when the signal quality is high, while up to 68% of savings were visible on 3G networks. This is both caused by a higher power consumption of the power amplifier (PA), but also by the lower data rates as observed on the different networks.

Huang et al. [HQG+12] are one of the first to report measurements and models of the performance and power consumption of 4G networks in direct comparison to 3G and WiFi networks. Their power model includes, besides the idle cost of different components, the transmission power for different up- and download rates. Furthermore, they derive timings of the discovered cellular state machines. Thus, modeling of the power consumption for any transmission is possible based on traffic traces only. Still, the device and network operator for which these are valid are not given in the paper. As this is one of the earlier measurements of 4G performance, large differences in both timing and power consumption are expected for more recent smartphones and cellular networks [CCD+14].

The general smartphone power models presented here are valid for the measured devices and networks. Large differences are observed between the performance and timing measured in the available networks and the one reported in literature [HQG+12]. Thus, the available energy models only roughly approximate the observed network behavior. The most useful model is the one presented by Huang et al. [HQM+12], detailing the power consumption of the Samsung Galaxy S3 on WiFi, 3G, and 4G networks.

Qian et al. [QWG+10] recognize the problem of extensive energy usage of modern smartphones, where contrary to earlier usage patterns, almost constant connectivity is required. In the past, for web browsing, downloads, or checking an e-mail account only a few network connections were made on non-smartphones. Thus, the connection was idle for longer durations, resulting in acceptable energy consumption of the network interface. But the rise of instant messaging applications, requiring frequent connections to the server and often transmitting only a few bytes, considerably increases the power consumption of the network interfaces. The problem here is the long duration between the last transmitted byte and the tear-down of the data channel. They identify this duration to be 15 s, thus causing considerable energy expenses. Their approach proposes to use *fast dormancy* by notifying the network of the end of its transmission and shutting the interface down afterwards. Thus, the tail duration of the connection is considerably reduced, resulting in energy savings of up to 60 %.

Based on their earlier observations [HQG+12], Huang et al. [HQM+12] propose optimizations significantly reducing the power consumption of the mobile device. Their approach is two-fold. First, the power consumption of any mobile data connection is reduced by limiting the tail duration of the 3G connections. Therefore, the connection is torn down after 4 s when the device is idle, or 8 s when the device is actively being used. The second optimization is limiting background communication to given intervals. The most efficient configuration determined by the authors is permitting a 5 s communication interval in each 100 s slot. The presented results show power savings of up to 60 % when combining both approaches.

A similar approach is chosen by Ickin et al. [IWF13] analyzing the energy consumption of smartphone as used by end users in the wild, with a major focus on evaluating the influence on QoE. Similar to Huang et al. [HQM+12], they show that by disabling background communication considerable energy savings are possible. Their approach limits itself to the cellular interface only, thus leaving the WiFi functionality unmodified. Contrary to Huang et al., they limit communication to 5 min in each 30 min interval. User surveys state that the maximum delay of 25 min was almost imperceptible to the users. Thus, Ickin et al. claim to achieve power savings of 25 % ('power gain of 34 %' [IWF13]) without affecting the QoE of the end users.

A further interesting aspect of mobile power consumption is the impact of the relatively new MPTCP. Originally developed to map a single TCP flow between two machines to multiple underlying network interfaces [BPB11], it quickly gained popularity for mobile devices, where also multiple interfaces exist. MPTCP supports different operating modes making it attractive in mobile scenarios showing intermittent connectivity. The available modes are *full MPTCP*, where all available interfaces are used to capacity, *single-path mode*, used to resume failed connections, and *backup mode* immediately opening a backup TCP sub flow used to establish a fast failover without requiring to establish a connection on demand.

Chen et al. [CLG+13] conduct measurements on operating networks. They measure the latency and throughput on different cellular networks. Their analysis shows that download times are always lower when using MPTCP on multiple interfaces, while the RTT is similar to the one of the interface with lower RTT.

Pluntke et al. [PEK11] are among the first to explore MPTCP in mobile scenarios. They propose to use Markov decision processes to schedule traffic on the available links depending on historical traffic characteristics. Compared to the omniscient oracle, the proposed approach is near optimal. Their approach includes 3G and WiFi state machines, configured with realistic power values. Still, the influence of MPTCP scheduling, differing delays on the available network technologies and lower layer re-transmissions are neglected.

Paasch et al. [PDD+12] show the feasibility of using MPTCP on a smartphone using both the WiFi and cellular interface in parallel. They compare the impact of losing WiFi connectivity for the different MPTCP modes on the TCP goodput and latency of data transmissions. Furthermore, exemplary measurements of the energy consumption caused by MPTCP on a Nokia N950 smartphone are described. However, no concluding power model is presented.

Le et al. [LHR+12] describe an approach reducing the power consumption of regular MPTCP scheduling, while simultaneously improving load-balancing. Fairness against other TCP flows on the link is not affected. The energy model is based on fixed costs per Byte on the different interfaces, but does not include idle cost. Their results are based on simulations, thus indicating the regions where energy payoffs may be expected. Furthermore, the influence of highly different throughput, RTT, jitter, and packet loss on different links is not analyzed. Here, additional measurements are required to determine probable gains under real-world conditions. Thus, network optimizations and simulations may be based on realistic assumptions.

There are two publications by Chen et al. [CYM13b; CYM13a], claiming to significantly (i.e. 10% to 23%) reduce the energy consumption of MPTCP on a mobile device when using both WiFi and 3G simultaneously. Their approach eMTCP is evaluated for CBR scenarios in [CYM13b] and bursty traffic in [CYM13a]. Using client side re-mapping of received data, the energy consumption is apparently reduced. Simultaneously, the throughput is claimed to be higher than the sum of both interfaces. These results are at least questionable. Here, further evaluation of the underlying concepts, assumptions and the used simulation environment is recommended.

Lim et al. [LCN+14] analyze the scheduling of MPTCP as published in [RPB+12]. Their focus is the systematic analysis of the MPTCP performance for general downloads. They use a Galaxy S3, install an MPTCP capable kernel, and measure the power consumption using an external power meter. Their analysis shows that the use of MPTCP in the used implementation shows favorable energy consumption compared to cellular networks only in a small region of bandwidth distributions. Further, this effect is only visible for downloads larger than 1 MB. Based on their observations, the authors propose an energy aware

scheduling approach *eMPTCP*, reducing the energy consumption by up to 15% in comparison to regular MPTCP.

The work presented in Section 4.3 differs, as the power consumption of MPTCP for different load distributions is measured empirically. Further, the analysis is conducted for CBR traffic of given rates emulating CBR streaming as is used in live video conferencing or cloud gaming. During the experiment, the load distribution on the different interfaces is changed. Based on these measurements, the energy savings of MPTCP are modeled based on empirical measurements.

Furthermore, smartphone energy models are required describing the power consumption of WiFi and the cellular interface for data rates as available in modern networks. Hence, the power model introduced by Huang et al. [HQG+12] is not applicable, as network performance has increased considerably since publication of these results. Furthermore, current network configurations tune parameters to minimize power consumption on the mobile device, which was not the case in the analyzed network. Hence, the newly generated models provide the required means to analyze the power consumption of arbitrary data transfers in recent mobile networks.

#### 3.1.4 Comparison of Power Models and Approaches

The power models and optimization approaches presented in the previous sections are summarized in Table 3.1. The approaches are grouped into device classes in accordance to the sections as described before.

For each study, the corresponding reference is given in the first column. The first main group on the horizontal axis, *Components*, indicates the hardware components considered by the approach. The second main group, *Device Class*, indicates the device class for which these models or optimizations are valid. The group *Type of Work* indicates how, and what kind of results are derived. Finally, the column *Results* indicates, whether a power model is described, letting other researchers build upon their work.

The components identified in the surveyed studies differ depending on device class. A *static* power model indicates the power consumption of the device without considering any dynamic aspects (e.g. system utilization). The remaining columns indicate, whether the indicated component was modeled in the respective publication. Internal components are CPU and memory. Still, these are not always included as separate components in the derived models. Often, the effects of CPU and memory utilization are part of the interface utilization. This may be caused by the non-availability of low level monitoring values, in other cases by the requirements of the targeted model, abstracting the device power consumption to fewer metrics (e.g. idle and interface utilization only). The interfaces identified in the different studies are the cellular interface, here abbreviated *Cell*, *WiFi*, and *fixed* for any other wire based or optical network connections. Further interesting factors, in particular on smartphones, are the inclusion of *GPS* power consumption or the *display*

in the final power model. As these considerably alter the power draw of the device, and thus affect battery life time, these metrics may be of high interest for some evaluations. Furthermore, the consideration of *MPTCP* in the power models is of considerable importance, as the interface power consumption, and the overall composition of the resulting power models is affected by its use. Finally, virtualization is included to indicate whether SDN techniques are used, indicating a higher flexibility and extended functionality of the underlying network.

The device classes as identified in Table 3.1 reflect the applicability of the power models and optimization approaches derived from related work. Here, *Srv/PC* indicates the applicability to servers or PCs based on x86/64 processors. The column *Switch* indicates that the model is generated for hardware switches. Contrary to software switches, these employ application specific integrated circuits (ASICs) to support high throughput and low latencies, and generally have a larger number of ports. Indicated with *Router* are models and approaches including devices with OSI Layer 3 functionality, which is not present in conventional switches. The column *BS* indicates approaches including the power consumption of base stations in the cellular network. *SBC* here denotes either the use of single-board computers, or the evaluation of low-power ARM processors. These can be seen as complementary to the group *Srv/PC*, where high power general purpose processors are used. The studies where both columns are marked compare the performance and energy efficiency of both. *HGW* indicates approaches including home gateways into their models. Devices fitting this category have at least one WiFi and one wired interface, with their main responsibility being the forwarding of traffic or providing services to mobile users. Finally the column *SP* indicates the approaches using smartphones for either power modeling or optimization of their energy consumption.

The group *Type of Work* is structured into *Methodology*, indicating whether a study is based on directly measured device behavior (*meas*) or simulated based on values derived from literature (*sim*). Here a special case is the analysis of the full network power consumption by Vereecken et al. [VHD+11], whose analysis is based on a literature survey, and thus indicated *lit*. The column *Impr.* shows whether improvements of the status quo are proposed. If nothing is indicated here, the current behavior of the surveyed system is analyzed and models are derived.

Finally, the column *Pwr. mod.* in the *Results* group indicates, whether ready to use power models are published. This is more often the case for the pure measurement and modeling studies, as usually their goal is presenting a model for others to build upon. Still, there are also some studies describing the behavior of a system under a given load, without giving sufficient information for others to build upon this. In particular simulation based optimization approaches have not been observed to publish resulting power models.

Generally in Table 3.1, '~' indicates fields, where some results are presented, but are incomplete, inconclusive, or otherwise not ready to use. Examples are the analysis of *MPTCP* by Chen et al. [CLG+13; CYM13a], where assumptions are not fully clear, or

Gomez et al. [GRR+12] and Carroll et al. [CH13], where no interface power models for different rates are given.

Summarizing, the contributions concerning the power consumption as presented in Chapter 4 are three-fold. First, power models for SBCs are presented, which were formerly not available to the public. These models use readily available system monitoring values read from the `/proc` file system to derive the power consumption of the SBC. Modeled components include the CPU, Ethernet, and WiFi interface. Secondly, power models for smartphones for the use case of CBR streaming for real-time applications using MPTCP are presented. Related work in this area focuses on regular downloads or optimization approaches based on superficial power models. These presented results further include models describing the power consumption of the devices for regular cellular and WiFi network access. Thirdly, power models for a hardware and software OpenFlow switch are generated. Related work presents approaches automatically generating power models for regular switches, and performance evaluation of OpenFlow switches. Based on these, optimizations of the energy consumption of networks are proposed. Contrary to related work, the influence of the extended functionality provided by OpenFlow on the device energy consumption was not analyzed before.

### 3.2 NETWORK PERFORMANCE MEASUREMENT

The performance of wireless network access affects the energy consumption of mobile devices by influencing the active time of the modems. Important metrics are, besides the signal strength, RTT and raw throughput. Based on these, the actual energy consumption of the device can be derived using power models. Here, large differences exist between network technologies, affecting the power consumption to a considerable extent.

To accurately determine the cellular network quality, the measurement methodology must be well known and analyzed. Hence, this section discusses measurement tools, techniques, and approaches determining the latency, throughput, and coverage of communication networks

Section 3.2.1 discusses tools and approaches measuring the latency in communication networks. These also include traceroute measurements and their derivatives providing additional RTT estimates for each hop on the route taken through the network. Finally, approaches applying different tools to analyze networks are discussed.

Section 3.2.2 presents and discusses tools and mechanisms measuring the throughput of communication networks. First, the different metrics as can be measured are discussed, based on which available measurement techniques are assessed. Their applicability for different network domains are discussed in the following. Finally, network measurement approaches focusing on cellular networks are described and compared.

Section 3.2.3 discusses approaches including, besides network performance metrics, also the location of the mobile device. These approaches either generate geographical

Table 3.1: Comparison of power models and energy optimization approaches

Publication Reference	Component										Device Class					Type of Work		Results Pwr. mod.		
	Static	CPU	Mem	Cell	WiFi	Fixed	GPS	Disp.	MPTCP	Virt.	Srv/PC	Switch	Router	BS	SBC	HGW	SP		Math.	Impr.
SBC	[WV+13]	x																meas		x
	[NL+3]		x	x														meas		x
	[T1+3]		x	x							x							meas		x
	[SRS+6]	x																meas		x
	[ZLH+16]	x																meas		x
	Our	x	x				x			x									meas	
Smartphone	[GH10]	~	x															meas		x
	[QWG+10]					x												x		x
	[ZTD+10]		x			x	x		x		x							meas		x
	[PEK11]					x	x		x									meas		x
	[PDD+12]					x	x		x									meas		x
	[HQM+12]					x	x		x									meas		x
	[HQC+12]					x	x		x									meas		x
	[CYM13b]					~	~		~									sim		~
	[CYM13a]					~	~		~									sim		~
	[GH13]		x			x												meas		x
	[DWC+13]					x			x									meas		x
	[WFR13]					x			x									meas		x
	[CCD+14]		x						x									meas		x
	[LCN+14]					x			x									meas		x
Our	x				x			x									meas		x	
Switches	[CSB+08]	x				x			x									meas		x
	[HDP+08]	x							x									meas		x
	[MSB+09]	x							x									meas		x
	[HSM+10]	x							x									meas		x
	[BCL+10]	x							x									sim		x
	[OLC+11]	x							x									sim		x
	[VNS+11]	x							x									sim		x
	[YWX+13]	x							x									meas		x
	[NSA+13]	x							x									sim		x
	[BBD+14]	x							x									sim		x
[LAH+15]	x							x									sim		x	
Our	x							x									meas		x	
Other	[VHD+11]	x							x									hit		x
	[GRR+12]	x							x									meas		~



maps, or store the network performance based on routes taken for further analysis. Furthermore, the availability of data sets is discussed, and an overview of commercial and academic cellular network mapping approaches given.

Finally, Section 3.2.4 compares the presented approaches, so their focus, strengths, and weaknesses can easily be identified. Thus, the optimal tools and measurement techniques for a given measurement task can be selected, while simultaneously checking which publications cover similar measurements.

### 3.2.1 Latency

The latency in communication networks defines the time packet needs to travel from the source to destination. As due to different clocks, jitter, and drift these can never be fully synchronized, the more common metric in communication networks is the RTT, consisting of the latency between sender and receiver, the processing time on the receiver, and the latency back to the sender. Thus, the latency of the network can be estimated, assuming that the processing time on the remote is small compared to the propagation time. The latency is an important measure directly influencing the response time of real-time service, but also the achievable throughput of TCP connections by limiting how fast a steady transfer rate is achieved, and how fast packet losses can be mitigated.

Conventionally, the network latency is measured using the *ping* command, as described in the RFC defining the ICMP protocol [Pos81]. Thus, the end-to-end latency can be estimated between any two IP addressable hosts. Still, also RTT measurements show peculiarities which must be considered when interpreting the measured results. When using the *ping* command, service differentiation within the network may cause ICMP messages not be forwarded with the same priority as other traffic by intermediate routers. Furthermore, routing may differ between ICMP and other packets, thus possibly resulting in inaccurate results. Hence, alternative configurations change the type of the probe packet. Thus, the influence on the sender to receiver path is comparable to the type of the probed traffic. Still, ICMP responses are always of the same type. Thus from a possible difference between these, conclusions on the network behavior can be drawn.

The ICMP protocol also permits identifying the route taken through the network. Therefore, each subsequent packet is assigned an increasing time to live (TTL). This is reduced by one when forwarded by a router. If the TTL reaches zero, a *Time Exceeded Message* is created by the next router and sent to the source address of the incoming packet. By incrementing the TTL by one for each subsequent packet, the full route to the destination can be mapped. This is implemented in the classical *traceroute* [Tra] or *tracert* command on any modern operating system (OS). In the current traceroute implementations, also the type of traffic used for probing the network is configurable. Thus, differences in the response times and routing of these can be identified.

The measured RTT for each hop consists of the latency between the local machine and the respective hop with the chosen data type, the response time of the remote machine, and the latency back to the local machine. The response time of most routers is variable and higher as pure data path latency, as ICMP messages must be generated by the CPU of the router, which is comparatively slow, usually not well connected, and may also be busy. The response to the local machine is always an ICMP message. Hence, this message may be routed using different paths on both ways. This effect is analyzed by Dall'Asta et al. [DAB+06].

Still, due to the high complexity of modern networks, multiple routes between two endpoints are likely. Load balancing on these may influence the path taken depending on different characteristics of the probe packet. This causes several problems in the measurements. First, as packets may take different routes, and these may include a different number of hops, incongruent results on the measured path can be returned. Hence, a variant of traceroute, *paris-traceroute*, is developed with the focus of determining all possible paths [ACO+06]. Their approach forges packets with the same checksum by modifying identifier and sequence number fields. This increases the probability of using the same path. Thus, the artifacts as observed with classical traceroute can be reduced. A comparison of different traceroute implementations and configuration options is given in [LHH08].

When measuring delays from smartphones, the influence of the measurement methodology on the resulting metrics is important. Li et al. [LMW+15] analyze the device internal delays for different measurement methodologies over a range of latency classes. The authors measure the latency between an Android phone and a server with configurable response time by monitoring the WiFi packets using WiFi sniffers and compare these to the recordings taken on the smartphone. They conclude that the measured latency significantly depends on the chosen methodology, where RTT measurements via the OS provided commands (e.g. *ping*) cause the lowest overhead, while HTTP pings take slightly longer. Worst are the RTT measurements run by Android applications via the Java API. Hence, in the following only native implementations for *ping*, but also *traceroute* are used.

Ricchiato et al. [RHR08] analyze the one-way delay within an operational 3G network by placing monitoring nodes within the core network. These measure the latency in different parts of the network, while simultaneously recording the load on the respective links. The authors observe frequent network scanning, during which the latency in the network increases. From this it can be derived, that the latency and throughput of the mobile backbone correlate. Laner et al. [LSR+12] extend a similar study to 4G networks and compare the performance of both. As expected, the overall RTT in the measured LTE network is lower compared to a high speed packet access (HSPA) network. Further, the variability in the LTE core network is lower. Still, the wireless uplink in the LTE network shows a higher latency than the HSPA uplink.

These studies show that the latency within the cellular network are largely predictable and defined by the access technology and the corresponding backbone infrastructure. For the measurements conducted in Chapter 5, the regular *ping* command is used. This is advantageous, as it is natively available on all devices. For both directions the ICMP messages are used, thus resulting in a symmetric latency. The traceroute measurements record both the IP and resolved hostname of the intermediate hosts, thus giving a fine-granular view of the routes taken through the network. Using *paris-traceroute* on the smartphones of the participating users is not possible, as for the packet forging root access is required. Contrary to related work, the routes through the cellular network are included in the analysis. The resulting measurements are discussed in Section 5.3.

### 3.2.2 Throughput Measurements

Besides the latency of wireless network access, also the throughput is important. A number of different measurement tools and techniques are developed, profiling different metrics. Some definitions are required beforehand: First, it must be distinguished between *link capacity*, *available bandwidth*, and *bulk transfer capacity (BTC)* [PMD+03]. The former describes the installed capacity, which defines the maximum theoretical throughput on the physical layer. *Available bandwidth* in contrast includes link utilization into the calculation. Hence, it defines the free fraction of a link. BTC is the actual transferred payload usually defined for TCP connections. Where link capacity and available bandwidth are defined on a per-link basis, the BTC is an end-to-end metric. Link capacity and available bandwidth are measures of high interest when planning and upgrading networks. As the performance of a full network is to be determined, the different end-to-end BTC measurement techniques and tools are introduced in the following.

Prasad et al. [PMD+03] give an overview of the different available bandwidth measurement tools. These are classified into link loading and timing based approaches. The former establish a data connection (TCP) or testing the link with configurable UDP data rates, thus directly measuring the achieved goodput. An example implementation using this approach is *iperf* [Ipf10]. The latter methods use timing differences between single packets, within packet trains, or within chirps of packets to estimate the available bandwidth. Implementations of these approaches are *train of packet pairs (TOPP)* [MBG00], *Pathload* [JDo2], or *PathChirp* [RRB+03]. Compared to the link loading variants, these approaches are designed to minimize interference to the productive traffic within the network. By using timing differences and dispersion, only a small number of packets is required, and tests may be spread over a longer duration. Depending in the chosen measurement methodology, multiple repetitions with different probed bandwidths are required, thus approximating the available bandwidth in an iterative manner.

For available bandwidth tests loading the link to capacity, *iperf* is a ready to use solution. Still, for the TCP connection to saturate the link, the slow start must have finished. As

this is not easily visible from outside *iperf*, and no functionality is included stopping a test when a sufficient accuracy is achieved, Tirumala et al. [TCD03] extend *iperf* by monitoring the OS kernels TCP status. This is achieved by running a kernel patched with the *web100* instruments [MHR03]. By reading the additional information on the TCP connection, the measurement duration can be reduced to 1 s, saving over 90 % of the traffic caused by regular *iperf*. Considering the measurement of the cellular network, where traffic is expensive, *iperf* with the *web100* instrumentation is a promising approach. Still, as also crowd-sensed measurements of the cellular network performance are conducted, the installation of a modified kernel on the participating devices is not possible.

The timing based bandwidth measurement approaches are not suitable for measurements in the cellular network for two reasons. First, for these approaches to be applicable, the network conditions must be constant over the duration of a few minutes. Assuming high mobility scenarios, the measurement duration considerably exceeds the time a device can be assumed to be stationary. Already in a pedestrian mobile scenario, the measurement is thus stretched over 60 m to 120 m per minute. If during this interval the available bandwidth changes, the measurement cannot complete, or the determined bandwidth is incorrect. Secondly, this approach is sensitive to jitter, as is caused by LTE scheduling on the physical layer, causing measurements to become unreliable.

Contrary to measurements on end-user devices, Gerber et al. [GPS+10] measure the throughput of operational 3G cellular networks using in-network probes. They determine the maximum throughput by analyzing packet headers observed within the network. The collected data is thoroughly filtered, from which characteristics of reliable flows are derived. Thus, the throughput of individual users is determined. The drawback of in-network measurements is that only approximate locations of the observed network performance can be derived based on the currently used cell ID. Huang et al. [HQG+13] extend similar measurements to 4G networks and analyze their accuracy using end-to-end measurements. They conclude that the measured TCP implementations cannot fully utilize the bandwidth provided by LTE, on average using less than 50 % of the theoretical available bandwidth. This behavior is caused by limited receive windows sizes. The bandwidth under-utilization causes a lower QoS for the user, while simultaneously increasing power consumption. Generally, these measurements show a good fit between the in-network throughput estimation and end-to-end measurements conducted on the mobile device.

Michelinakis et al. [MBF+15; MBF+16] apply a similar methodology to packet traces collected on mobile phones. By sampling these, the used bandwidth of the local device can be determined based on passive measurements only. This is possible regardless of LTE scheduling and local device type. The resulting relative error using as few as 10 % of the available samples is 15 %. This measurement technique is thus feasible for implementation in the mobile OS, but cannot be used in a crowd-sensing study, as these traces of the

network behavior are commonly not available. By using a passive approach, also the number of resulting measurements cannot be controlled.

The location-based 4G performance is first systematically analyzed by Wylie-Green et al. [WS10]. The authors analyze RTT, throughput and handover performance in a freshly deployed LTE network. They show that the capacity of the network is well saturated in different end-to-end measurement scenarios, and the average RTT is below 35 ms. These measurements are carried out in an early testing deployment without load, thus only limited conclusions on the performance of real-world deployments can be drawn.

Becker et al [BRF14] analyze RTT and throughput in 4G networks by running end-to-end measurements. Their main contribution is the assessment of the performance implications of PEP in the cellular core network. For reference, also the performance via a fixed broadband access link is analyzed. They show that the use of a PEP significantly improves connection establishment and retransmission delays. Still, in the case the remote service is not available, additional timeouts occur.

Vallina-Rodriguez et al. present a more systematic approach on higher layer influences in [VSK15]. Their focus is on the analysis of service variations caused by middleboxes and network traffic management. They develop an Android application running a series of tests against measurement servers, focusing on the detection of modifications of the transferred traffic. Their focus is on changes in the domain name system (DNS) system, the insertion of PEPs, or any other unwanted modifications. The Android application presented by the authors is used in this work to determine eventual network service impediments.

These approaches show successful approaches on characterization of cellular network performance and promising measurement techniques. Still, the published data is often not sufficient for further network analysis, the techniques not applicable to the measurements required for this work, or the data does not reflect the performance as experienced in modern cellular networks. Furthermore, for an analysis of handover mechanisms performance measurements and network availability of cellular and WiFi performance within the same region are required.

### 3.2.3 *Network Coverage Maps*

Network coverage maps are mainly published by cellular network operators to show the reach of their networks to attract new customers. As one would expect, these show the optimum coverage of the available network technologies based on propagation models. Still, network performance also includes latency and throughput. Commonly, these are considered to be confidential information by the network providers. Thus, no reliable data of these KPI of cellular networks are available to end users and researchers.

Hence, a number of tools are developed to analyze the location-based cellular network performance. Yao et al. [YKH08] are among the first to systematically measure the influ-

ence of time and location on the throughput of 3G networks with the goal of predicting cellular service quality in a vehicular context. They conclude that the influence of location on network throughput is larger than the influence of time. Thus, by knowing the historical network performance at a given location, the probability of achieving similar results is high. Pögel et al. [PW12] strengthen these observations with additional measurements on both rural and urban routes. They extend the observations to different network technologies and cell IDs. Further, the authors analyze handover durations and behavior between different cells and technologies as well as RTTs in different connection states (i.e. DCH, FACH). Both studies exemplarily show the location dependency of cellular network performance, and the general feasibility of a mapping approach.

Also the cellular network performance on trains is of growing interest [FS16]. Yao et al. [YKH11] analyze RTT and throughput of 3G networks on several regional train routes near Sydney (AU). They conclude that the network performance mainly depends on location, while time of day and speed have no significant influence.

Sonntag et al. [SSM13] are among the first to publish a crowd-sensing App for multiple smartphone OSs focusing on throughput measurements. Based on an extensive collection of signal strength, RTT and throughput samples, a general map of the cellular network performance is generated. The majority of the measurements are located in the region of Helsinki (FI). They conclude that signal strength alone is not a sufficient indicator of network throughput. Neither does including the time of the day improve the correlation. Therefore, throughput measurements are required to estimate and predict the cellular network performance. Similar measurements are conducted by Huang et al. [HQG+12]. Neither study publishes location-based network performance data for further analysis.

Besides these scholarly approaches, also commercial services exist, measuring the wireless network performance and creating maps. Examples are OpenSignal [OpS] or Sensorly [Sen]. They use a similar approach to [SSM13]. Similarly, information is gathered by publishing smartphone applications and letting users measure the network performance. Thus, network coverage maps are created to guide the users in cellular operator selection and possibly also in finding locations with better network quality in case the current performance is insufficient.

From the above studies it is derived that for accurate modeling and prediction of cellular service quality, extensive measurements of the actual parameters of interest are required, as these cannot reliably be derived from others. Further, as the data of interest is commonly not available, these must be collected in a crowd-sensing study. Other approaches (e.g. systematic, location-based measurements) appear infeasible due to the high number of required measurements. Based on the collected data, models of the cellular network performance can be derived to later optimize traffic scheduling and network selection based on predicted network performance.

### 3.2.4 Comparison of Measurement Approaches

The above described tools, measurement techniques and approaches are summarized and compared in Table 3.2. These are grouped into *Tools*, *Techniques*, *Comparison* of techniques and tools, and the *Analysis* of operational networks. The network mapping approaches are grouped into *Scientific* works and *Commercial* products for comparison.

For each described approach the used network technology for development end evaluation is given in the column *Tech*. This may be any, if no specific network technology was in focus when being developed, while others are limited to fixed networks or the local device only. The classification may include cellular networks in general (*cellular*), or the respective network technology (e.g. 3G/4G). The main identified categories are *parameters* and the *classification* of the respective approach. The identified parameters are environmental (i.e. location (*Loc*), *time*), and network related (i.e. signal strength (*SigStr*), *RTT*, throughput (*TP*)). Further, higher layer network management functionality is taken into account by considering routing and user allocation within the cellular backbone (Column *Route*). The *classification* is split into parameters describing the measurement approach, *cost*, accuracy (*Acc.*), the requirement of a remote server (*Remote*), and the applicable network domain (*Domain*). Here, the most important is the measurement technique (Column *Techn.*), identifying whether an approach actively causes additional traffic (*act*), or monitors the network performance or traffic consumed by the device passively (*psv*). The placement of required measurement components is given in the column *Meas. loc.*, where either only the end-user device is required (*dev*), the metrics are recorded on both local device and server (*dev/srv*) to be later merged, or in-network probes are used (*n/w*). Besides the deployment, also the cost, duration and accuracy are of interest when selecting a network measurement approach. The measurement duration is classified into short, long or variable, depending on the tests to be run. In the case of network studies as given in the fourth group of rows, no data is available, thus *n/a* is given. Similarly, the cost and accuracy are classified into low, medium, high and variable, where applicable. Additionally, the requirement of a remote measurement endpoint is indicated in the *Remote* column. Here, '*~*' indicates the requirement of a measurements server providing content without measurement functionality. Finally, the applicable network domain is given. This categorizes measurement approaches depending on their applicability to fixed, mobile or any network domain.

The last column indicates, whether a model of the observed behavior is created. Approaches indicated with '*~*' provide data to model the behavior, but do not explicitly create a model.

The network measurements as described in Chapter 5 make use of the established measurement approaches. Besides measuring the latency and throughput of the cellular network, also the availability and performance of WiFi networks is measured. Furthermore, also traceroute measurements are integrated into the measurement application to

determine the routes taken through the network. Based on the collected data, models of the network performance are generated for further processing in the energy evaluation environment.

### 3.3 COMPARISON OF MOBILE NETWORK OPTIMIZATION APPROACHES

Detailed knowledge of the cellular network performance is the basis for further improvements and optimizations thereof. Here, numerous approaches have been presented. Still, as the focus of this work is the analysis and optimization of energy efficiency in cellular communication networks, the approaches presented in the following are limited to the ones considering both performance and energy efficiency.

The approaches also presenting power models of mobile device and optimizations based thereon [QWG+10; PEK11; HQM+12; CYM13b; CYM13a; IWF13; LCN+14] are already described in Section 3.1.3. Additional energy conserving approaches not providing a power model [LHR+12; GPN13; GK15] are summarized in the following.

Le et al. [LHR+12] analyze the performance and energy consumption of MPTCP. Based on their findings they propose the modification of MPTCPs load balancing algorithms to reduce the energy consumption and name it *ecMPTCP*. They build their optimization on simulations including a relatively crude energy model neglecting ramp and tail energies as are common in cellular networks. Their simulated network conditions show similar performance, and thus do not realistically reflect common network parameters. Extending the proposed model with more realistic energy and network models likely provides further interesting insight into the behavior of MPTCP.

Gautam et al. [GPN13] empirically analyze the energy consumption of video consumption on smartphones. Based on their analysis of the cost of different transmission modes (i.e. WiFi download, WiFi streaming, 3G streaming), they propose to download videos on WiFi before being requested by the end user. This approach is realized by implementing a background service periodically checking the user's video subscriptions and downloading the respective videos when connected to a WiFi network. Thus, energy savings up to 84 % compared to 3G streaming are possible. Still, their evaluation shows some flaws. The downloaded videos are indicated to the user, thus likely affecting the decision on which videos to watch. Further, notifications indicate the availability of new content, thus likely affecting the probability of further video consumption. Nonetheless, the presented work shows a significant potential of energy savings using WiFi offloading.

Gabale et al. [GDK+13] focus on the problem of frequent polling or App notifications on the energy consumption of smartphones. Based on traffic traces of real users, they analyze the frequency of network interactions caused by background traffic. The number of these requests is reduced by introducing a virtual App server or virtual notification server between App and cloud service, limiting connectivity at idle times. Thus, the power consumption of the device is significantly reduced (i.e. 50 %), which is determined using the



Table 3.2: Comparison of cellular network performance measurement tools, techniques, and approaches

	Publication Reference	Short	Tech.	Parameters				Classification				Model						
				Loc	Time	SigStr	RTT	TP	Route	Tech.	Meas. loc		Duration	Cost	Acc.	Remote	Domain	
Measurement	[Pos81]	Ping	any	-	-	-	x	-	-	-	act	dev	short	low	var	-	any	-
	[Tra]	Traceroute	any	-	-	-	x	-	x	-	act	dev/srv	long	low	var	x	any	-
	[lperf0]	Iperf	any	-	-	-	-	x	-	-	act	dev	long	high	high	x	any	-
	[MBC00]	TOPP	fixed	-	-	-	-	-	x	-	act	dev/srv	long	medium	high	x	fixed	-
	[JDoz]	Pathload	fixed	-	-	-	x	-	-	-	act	dev	long	high	high	x	fixed	-
	[RRB+03]	PathChirp	fixed	-	-	-	-	x	-	-	act	dev/srv	long	medium	high	x	fixed	-
	[TCD03]	Iperf+web100	any	-	-	-	x	-	-	-	act	dev	var	low	medium	-	fixed	-
	[ACO+06]	Paris-traceroute	any	-	-	-	x	-	-	x	act	dev/srv	long	low	high	-	any	-
	[RHR08]	In-netw. Meas	3G/4G	-	-	-	x	-	-	-	psv	n/w	var	low	high	x	any	-
	[GPS+10]	In-netw. Meas	3G	-	-	-	x	-	-	-	psv	n/w	var	none	var	x	mob	-
[HQG+13]	End-to-end	4G	-	-	-	-	x	-	-	act	dev+n/w	var	var	high	x	mob	-	
[MBF+16]	End-to-end	4G	-	-	-	-	x	-	-	psv	dev	long	none	var	-	mob	-	
[PMD+03]	TP meas cmp.	any	-	-	-	-	x	-	-	act	dev	long	var	n/a	~	fixed	-	
[DAB+06]	TR accuracy	any	-	-	-	-	x	-	-	act	dev/srv	long	low	n/a	-	any	-	
[LHH08]	TR cmp	any	-	-	-	-	-	-	-	act	dev/srv	long	low	n/a	-	any	-	
[LSR+12]	Latency cmp	3G/4G	-	-	-	-	x	-	-	act	n/w	var	medium	n/a	x	mob	-	
[BRF14]	TP+Latency	4G	-	-	-	-	x	-	-	act	dev+n/w	n/a	n/a	high	x	mob	-	
[LMW+15]	Latency	device	-	-	-	-	x	-	-	act	dev+n/w	n/a	n/a	high	x	mob	~	
[VSK15]	Higher layer effects	cellular	-	-	-	-	-	-	-	act	dev	n/a	n/a	high	x	mob	-	
[YKH08]	TP pred (geo)	3G	x	x	x	x	x	-	-	act	dev	long	high	high	x	mob	-	
[WS10]	TP analy. (new n/w)	4G	x	x	x	x	x	-	-	act	dev	long	high	high	x	mob	-	
[YKH11]	TP, RTT, on trains	3G	x	x	x	x	x	-	-	act	dev	long	high	high	x	mob	-	
[PW12]	RTT, TP (geo)	3G	x	x	x	x	x	-	-	act	dev	long	high	high	x	mob	-	
[SSM13]	RTT, TP (geo)	4G	x	x	x	x	x	-	-	act	dev	long	high	high	x	mob	-	
[OpS]	SigStr., RTT, TP (geo)	cellular	x	x	x	x	x	-	-	act	dev	long	high	high	x	mob	-	
[Sen]	SigStr., RTT, TP (geo)	cellular	x	x	x	x	x	-	-	act	dev	long	high	high	x	mob	-	
Our	TP, latency & routing	4G	x	x	x	x	x	x	x	act	dev/srv	long	high	high	x	mob	x	

Netw. Mapping

power model presented by Huang et al. in [HQG+12]. Similar approaches are already being implemented by mobile OS vendors, urging developers to use their cloud notification service, thus bundling notifications for a single device and minimizing wake-ups.

Other approaches focusing on the improvement of QoS and QoE while using energy efficiency as an additional metric are discussed in the following. Nicholson et al. [NN08] analyze the throughput of WiFi networks and develop an approach predicting the performance for mobile users based on historical measurements and mobility patterns. By knowing the current location and direction of movement, the performance of the network can be predicted with a relatively high accuracy (i.e. within 10 KB in 50 % of time). They evaluate their approach on WiFi networks within a campus scenario based on an extensive data set. Thus, the feasibility of location-based network performance prediction based on personalized mobility prediction is shown.

Balasubramanian et al. [BMV10] extend this approach by analyzing the possibility of offloading traffic from 3G networks to WiFi. They compare an adapted version of the above algorithm with their own approach and an oracle algorithm with perfect knowledge. Their results show that WiFi offloading is possible in 10 % of the time. By leveraging delay tolerance of applications, a 1 min scheduling delay reduces the 3G traffic by half. Rathnayake et al. [RPO+12] propose a similar approach using a central scheduler. The advantage of this is the shared network probing cost. Still, for short network requests, an additional call to the central server may defy its purpose. The authors of both publications mention improving energy efficiency, but no analysis of the proposed algorithms considering this aspect is included.

The prediction of cellular service quality is evaluated by Singh et al. [SOC12] based on a multimedia streaming approach. By pre-fetching videos when the network throughput is high, the QoE can be improved compared to conventional video streaming. Thus, short outages are mitigated without causing stalling or requiring to adapt the video quality. The proposed approach uses a centralized service predicting the network throughput based on historical measurements and the projected route of the user. This approach is also expected to be beneficial on energy consumption, but was not evaluated yet.

Orthogonal to these approaches, Ha et al. [HSJ+12] propose to dynamically price cellular data. They add an indicator of the current price to the smartphone, thus letting the user decide whether to use the cellular network or not. Thus, the network utilization during peak periods can be reduced by exploiting price-sensitivity of end users. The demand on the cellular network can thus be equalized, while simultaneously reducing the cost of mobile networking for end users. Due to the avoidance of congestion in the network, the energy consumption is expected to be reduced compared to conventional network access, but was not analyzed. Similarly, Gabale et al. [GDK+13] adjust the pricing of mobile data dynamically. Here, the prices of streaming video are adjusted based on the current network demand. Their approach improves QoE in heavily loaded networks while maintaining delivery guarantees. By exploiting delay tolerance of the end users,

the same content may be consumed at a later time for a fraction of the cost during peak hour. Similarly, Zhuo et al. [ZGC+13] adjust prices for general mobile data access, but using a reverse auction mechanism. Simultaneously, they consider WiFi offloading to reduce the load on the cellular network. The proposed approach is evaluated based on user traces. To be effective, all users of the network need to participate. However, the required user interactions may hinder acceptance. The above approaches may also conserve energy by reducing the demand on the network by shifting traffic to different times, locations, and networks. Only the approach proposed in [GDK+13] may increase the energy consumption of the mobile device by reducing the transmission rate to use the network to capacity, thus prolonging transmission times. In all cases, a detailed analysis of the energy consumption would be interesting.

Table 3.3 summarizes related work focusing on the reduction of energy consumption (group *Energy optimization*) or improving network performance (group *Performance*). For each publication also the name of the proposed approach, if available, is given in the second column. The *applicability*, *requirements*, *operation*, and derived *metrics* for each approach are indicated in the following columns. Finally, the type of work (i.e. implementation (*impl*), measurements (*meas*), simulation (*sim*), or theoretical work (*theor*)) is indicated. In the case multiple types are given a succession of these are used. For example in the case '*meas/mod/theor*', first the network performance is measured, modeled, and then a theoretical model of the proposed approach applied to derive the final conclusion. The main method used in the approach is summarized in the last column.

The *applicability* defines the network technologies (*Netw. Tech*) used in the analysis, implementation, and verification of the described approaches. The networks and combinations thereof are indicated as derived from the respective publication. The traffic type is given in the column *Traffic*. Possible options are that any traffic is handled by the approach, it focuses on background traffic, CBR traffic, bulk data transfers, or optimizes video delivery only.

The *requirements* are first identified by the software modifications required on the mobile device (*Mob s/w*). Possible are the implementation of a traffic management service (*Tm-svc*), the use of *MPTCP*, the inclusion of a *proxy*, or the use of an *App*, if interactions with the end user are required. The requirement of a server (column *Srv.*), the dependency on the current location (column *Loc.*), or user interactions (column *Int.*) are indicated in the respective columns.

The *operation* of the analyzed approaches summarizes how the traffic is modified or handled. Here, data transfers may be *scheduled*, *offloaded* to other networks (e.g. WiFi), or fast dormancy (column *Fast d.*) for interfaces with a long tail duration may be used. Often, also a combination of several methods is used, leading to the overall energy savings as reported in the respective works.

The *metrics* used in the evaluation are indicated in the respective columns. Approaches optimizing QoS, QoE, or cost of the data transfers are marked in the respective column.

Here, if a metric is only discussed or is a secondary optimization goal, the respective column is marked with a '~'.

The approaches discussed in this section analyze their energy and performance gains on given settings and assumptions. The most common claims are to improve the performance or reduce the energy consumption by a percentage number over a non-modified system. Still, the cellular network environment changes, new technologies become available, network parameters are changed. Thus the assumed reference for these studies cannot be reestablished. Further, the behavior of Apps, and user interactions, are highly varying and certain to change over time. Hence, the comparability of the different approaches and implementations as proposed in the respective publications is not given. Thus, also their claimed savings are omitted.

A possible solution to the above problem is a unified evaluation framework. Thus, implementations of the different approaches may be compared based on the same traffic patterns, network configuration and performance, and energy models. Comparing these approaches in this way allows network operators and mobile OS vendors to choose the most promising approaches before implementation.

The work presented in Section 6.1 compares the energy efficiency of selected approaches ([IWF13; HQM+12]) to analyze their performance using recorded user-traces. Thus, the comparability of approaches, which is conventionally not given, is established.

#### 3.4 DIFFERENTIATION FROM RELATED WORK

The publications presented in the previous sections summarize the related work in the areas covered by this thesis. The analysis and optimization of the energy consumption of communication networks requires knowledge of the related work concerning the power consumption of the involved devices. Hence, Section 3.1 summarizes related publications. On mobile devices, the energy consumption also depends on the throughput of the available networks, defining the time the modems must be active. Hence, also the related work covering the performance measurement of wireless networks, namely RTT and throughput is reviewed. These are described in Section 3.2. Finally, mobile energy optimization approaches are compared in Section 3.3 to derive promising directions for further research in current networks.

The analysis of related work regarding the power consumption of fixed network entities led to the conclusion that for emerging network management approaches like OpenFlow or NFV the required energy models are missing. Examples are the deployment of virtualized services on HGWs, the dynamic rerouting of traffic in the case of SDN, and uninterrupted live streaming on smartphones using MPTCP. Hence, exemplary power models for the different device classes are measured, thus allowing the estimation of the cost of different network optimization described in Chapter 4.

Table 3.3: Comparison of energy optimization approaches

Publication Reference	Name	Applicability		Requirements			Operation			Metric		Type	Method	
		Netw. tech	Traffic	Mob. s/w	Srv.	Loc	Int.	Sched.	Offl.	Fast d.	QoS			QoE
[QWG+10]	TOP	3G/4G	any	Tim-svc	-	-	-	-	-	x	-	-	impl/meas	Fast dormancy
[PEK11]	-	3G/WiFi	any	MPTCP	-	-	-	x	x	-	-	-	theoretical	Markov Decision Processes
[LHR+12]	ecMTCP	Wireless	any	MPTCP	-	-	-	-	x	-	-	-	impl/meas	Load balancing
[HQM+12]	-	3G/4G	bg	Tim-svc	-	-	-	x	-	x	-	-	impl/meas	Reduction of tail duration
[IWF13]	ExpCO2	3G/WiFi	bg	Tim-svc	-	-	-	-	-	-	x	-	impl/meas	Disabling IF in bg
[GPNr13]	IncomingTV	3G/WiFi	video	App	-	-	-	x	x	-	~	-	impl/meas	Pre-fetching
[CYM13b]	eMTCP	4G/WiFi	CBR	MPTCP	-	-	-	-	x	-	-	-	simulation	IF selection
[CYM13a]	eMTCP	4G/WiFi	bursty	MPTCP	-	-	-	-	x	-	-	-	simulation	IF selection
[LCN+14]	-	3G/4G/WiFi	bulk	MPTCP	-	-	-	-	x	-	-	-	meas/mod/theor	Load balancing
[GK15]	MobInsight	3G/WiFi	apps	proxy	-	-	x	x	-	-	x	-	meas/mod/theor	App proxy
[NNo8]	BreadCrumbs	wireless	any	Tim-svc	-	x	-	x	x	-	-	-	meas/mod/theor	TP prediction
[BMV10]	-	3G/WiFi	any	Tim-svc	-	-	-	x	x	-	x	-	impl/meas	Mobility
[RPO+12]	EMUNE	3G/WiFi	bulk	Tim-svc	x	-	x	x	x	-	x	-	theoretical	Dynamic pricing
[HSJ+12]	TUBE	3G	any	App	x	-	x	x	-	-	~	-	impl/meas	Dynamic pricing
[SOC12]	-	3G	video	App	x	-	-	x	-	-	-	-	meas/mod/theor	Predictive buffering
[GDK+13]	Async	3G	video	App	x	-	x	x	-	-	~	-	impl/meas	Dynamic pricing
[ZGC+13]	-	3G/WiFi	bulk	App	x	-	x	x	-	-	~	x	simulation	Dynamic pricing

Energy optimization

Performance

The power consumption of mobile devices depends besides the built in modems also on network throughput and latency. Hence, to analyze the effectiveness of proposed approaches and derive promising further optimizations, the quality of both WiFi and cellular networks at the same location is required. The literature review shows that the required data is not available. Still, suitable measurement approaches are presented. Hence, measurements of the cellular service quality based on these approaches are conducted and presented in Chapter 5. Besides providing the required data for the energy estimation, the measurements show the previously unknown influence of user association to point of presence (PoP) of the cellular operator on the measured end-to-end latency.

For the reduction of the energy consumption of mobile data access, a number of different approaches are presented in related works. Still, their energy savings are compared to an 'unmodified system', which is not clearly specified. Thus, currently no sound comparison of their effectiveness is available. Neither can thus their potential for further improvement reliably be determined. Consequently, the most promising of these approaches are compared in Section 6.1 based on the measured energy models and network performance to derive promising further optimizations. Also the structure and traffic management decisions in the fixed network affect the energy consumption of mobile devices. Hence, based on the measured power models and optimization approaches of fixed network structure as available in literature, the implications and potential energy savings are discussed in Section 6.2.

**P**ERFORMANCE is an important aspect of communication networks. Still, over the recent years, also their energy consumption of seen considerable interest. This is also termed *green networking*. For a cellular operator, already 72 % of the electricity are consumed by cellular base stations [CMo12]. Similarly, cloud computing already consumes more energy than, for example Germany or India [Mil13]. Routing, switching, or forwarding of traffic consumes 250 TWh to 400 TWh yearly [Mil13], which relates to approximately 20 % of the overall energy consumed by ICT. Thus, by reducing the power consumption of the required network equipment, considerable reductions of operational expenditure (OPEX) can be achieved.

One possibility of reducing OPEX while improving the QoS is the placement of services on augmented HGWs. These are currently only lightly used, but provide constantly increasing capabilities. Thus, content can be cached close to the end user to be served with low delay and high throughput [LPB+15]. Additionally, PEPs may be instantiated, reducing the retransmission delay of network requests by splitting TCP connections between the fixed and wireless network domains. Deploying these services is becoming possible by using the emergent technologies of SDN and NFV. Still, these functions require storage and processing capabilities at the respective node, which is not commonly available on HGWs. Hence, the energy cost is in the following exemplary determined based on SBCs, whose functionality may, in a similar form, be integrated into future HGWs. Thus, the estimation of capabilities and energy consumption of future augmented HGWs may already be analyzed based on empirical measurements using the presented power models.

Power savings in the fixed network may also be achieved using intelligent traffic and link control on intermediate switches and routers (e.g. [YWX+13]). By switching off unused or lightly used links during less demanding periods of network activity, energy can be saved within the core network [CTM+13]. As the power consumption of conventional switches and routers is already well known (e.g. [HDP09; GGS04]), the advanced capabilities of OpenFlow switches and their influence on energy consumption is analyzed in the following. By exploiting the increased capabilities of the network, dynamic load balancing, service placement, and in-network traffic modification becomes possible. Thus, additional energy savings are expected. Still, to optimize these services in respect to energy consumption, well-funded power models are required, which are presented in the following.

In the mobile sector, continuous connectivity is an ongoing requirement. MPTCP promises to provide uninterrupted connectivity using multiple wireless interfaces [PDD+12].

Still, using multiple interfaces in parallel increases the overall power consumption. At the time of writing, already a number of studies are available, analyzing the available MPTCP implementation [PDD+12; LCN+14] for limited scenarios, or proposing optimizations based on theoretical considerations [CYM13b]. Still, no conclusive power models describing the behavior for different load distributions were available. Hence, power models for different data rates, load distributions, and devices are generated, thus providing the means to model and optimize the power consumption for a wide range of usage scenarios.

Based on power models for conventional networking hardware as presented in literature, and the additional power models presented here, power optimizations of the full network become possible. Clearly, differences between individual devices are to be expected, but the general behavior of a device class can reasonably be assumed to be similar. The models presented here extend the previously published power models by a new device class (e.g. SBCs), or the analysis of new technologies (e.g. OpenFlow, MPTCP).

From these observations, the following the research questions are derived, targeting the power consumption of the above presented device classes:

- RQ 1.1 What is the energy cost of decentralized caching and computational offloading using SBCs and how can it be determined?
- RQ 1.2 How does the energy efficiency of hardware and software OpenFlow switches compare and what are their respective benefits?
- RQ 1.3 What is the cost of increasing reliability and throughput of mobile communication using MPTCP for constant bitrate streaming?

The power consumption of decentralized caching and computing using SBCs is analyzed on the Raspberry Pi and a number of its successors. Hence, first the power consumption of the platform stressing the individual components is analyzed, from which models of the power consumption under load are derived. These are verified by serving content from the local device and comparing the measured power consumption with the estimates derived using system monitoring on the devices in combination with the presented power models. The detailed measurement, modeling and verification procedure is described in Section 4.1.

The throughput and power consumption of OpenFlow switches is analyzed for different operating states as well as OpenFlow operations. Differences between hardware and software implementations are analyzed by comparing the power consumption of a commercial 48-port managed switch and an instance of Open vSwitch running on a bare metal server. Thus, the performance of these switches are measured for the full set of matches and actions defined by OpenFlow version 1.0, from which their power consumption is derived. The respective measurements and resulting power models are described in Section 4.2.



The focus when analyzing and modeling the power consumption of smartphones is put on the energy consumption of the available network interfaces (e.g. WiFi and cell interface) and connection technologies (e.g. 3G, 4G) for different data rates. Therefore, dedicated measurements are conducted on two devices under different traffic load and interface settings. Furthermore, the energy consumption of the emerging technology MPTCP is analyzed for the scenario of constant bit-rate streaming. The results of power measurement and modeling are described in Section 4.3.

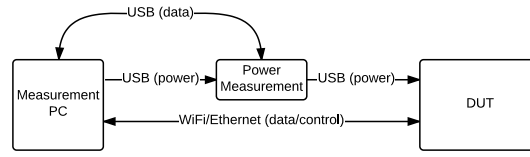
Finally, Section 4.4 compares the presented approaches, gives recommendations on their use, discusses further optimizations, and recommends promising future research directions further reducing the energy consumption of ICT.

#### 4.1 POWER CONSUMPTION OF SINGLE-BOARD COMPUTERS

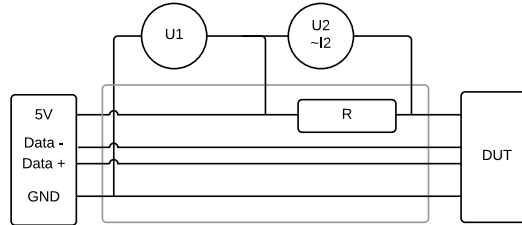
Single board computers like the Raspberry Pi and its many successors and derivatives show large popularity in the last years, owed to their high level OS, high computational capacity, and easy extensibility via general purpose input output (GPIO) pins. Although the original Raspberry Pi was originally developed for teaching, it is also widely enjoyed by hobbyists. Also in research, these devices show increased popularity, e.g. [VM14; TW]<sup>+13</sup>; WE15; PCK15]. Main reasons are, besides their versatility, the small form factor, low price (~US \$35), and low energy consumption.

The Raspberry Pi and its derivatives are already used in a variety of projects. The most immediate example is the Glasgow Raspberry Pi Cloud [TW]<sup>+13</sup>, creating a cloud environment for teaching. Papageorgiou et al. [PCK15] use the Raspberry Pi as a HGW for the IoT, aggregating sensor data. Vujovic et al. [VM14] propose to directly use it as a sensor node, expanding capabilities on the cost of energy consumption. Other usage examples are the extension of the cloud to the edge of the wired network [FLG<sup>+16</sup>; LPB<sup>+15</sup>], thus improving latency and throughput to (mobile) end users. Withnell et al. [WE15] use it in combination with a mobile power bank as mobile MPTCP node, thus avoiding the limitations posed by other mobile OSs. But also the application of SBCs in high performance computing are discussed [SRS16] due to their low power consumption.

Considering the large increase in computational capabilities, also the replacement of conventional x86/64 systems in power limited environments may be considered. Examples are the replacement of conventional servers at popular hotspots for task offloading as proposed by Satyanarayanan et al. [SBC<sup>+09</sup>]. Furthermore, the comparison of the energy consumption of task offloading not only for mobile devices, but the full deployment as proposed by Meurisch et al. [MSS<sup>+15</sup>] becomes increasingly interesting. Hence, the following section describes the modeling of the power consumption of a number of popular SBCs. Based on these models, further optimizations of existing and future deployments become possible.



(a) Measurement setup



(b) Electrical setup

Figure 4.1: Schematic and electrical setup of the SBC power measurements

The following section is based on [KGH14]. Section 4.1.1 describes the setup used to measure the power consumption under different usage scenarios. Section 4.1.2 describes the derived measurements and the behavior of the examined devices. Section 4.1.3 describes the modeling approach and the derived power models. Finally, Section 4.1.5 summarizes the findings, concludes the section and gives an outlook on future work.

#### 4.1.1 Measurement Setup

Generally, the power consumption of any electric equipment can be measured by inserting a measurement shunt in the power line and measuring the supply voltage and voltage drop over the shunt. In the case of the SBCs, this is done between power supply and device, thus eliminating the inefficiencies of the power supply from the measurements. The schematic setup is depicted in Figure 4.1a. For the electrical measurements a custom built board was used, inserting a measurement shunt into the 5 V line. The schematic is given in Figure 4.1b.

The power consumption is calculated by multiplying the supply voltage  $U_s$  and the current  $I_s$ . Hence, the derived power is

$$P = U_s \cdot I_s, \quad (4.1)$$

where the current drawn by the device  $I_s$  is defined by the voltage drop  $U_m$  over the measurement shunt  $R_m$

$$I_s = \frac{U_m}{R_m} \quad (4.2)$$

Hence, the accuracy of the measurements is defined by the resolution and quality of the two voltage measurements and the accuracy of the measurement shunt.

To maximize measurement accuracy, the optimal resistor must be chosen. The accuracy is maximized, by increasing the size of the resistor, thus increasing the voltage drop. Still, the maximum size of the resistor is limited. First, and most importantly, the power consumption of the DUT is not constant but highly variable. Hence, also the equivalent resistance of the device is highly varying. Fitting a measurement shunt to one of the possible states prevents usage in others. If, for example, the measurement shunt is too large, the voltage drop reduces the voltage  $U_s$  available at the DUT below the minimum threshold required for operation. Using a small resistor, these problems can be mitigated on the cost of accuracy. A smaller resistor causes a smaller voltage drop, thus increasing the error when sampling the resulting voltage drop  $U_m$ .

According to the USB specification the voltage of a USB high power port must be in the range (4.75 V to 5.25 V) [Usboo]. Considering a maximum power draw of 10 W, the maximum size of the resistor limiting the voltage drop to 200 mV can be calculated by

$$R_{m,max} = \frac{U_s}{I_s} - \frac{U_s}{P_{max}/U_s} = \frac{5V}{10W/5V} = 200m\Omega. \quad (4.3)$$

Here,  $U_d$  is the remaining voltage over the DUT while running the test. Hence, a precision resistor with a standard value of 100 m $\Omega$  and accuracy of 1 % was chosen.

The measurement hardware is a Measurement Computing USB-1608FS-Plus USB measurement card. It features a resolution of 16 bit on each channel and simultaneous sampling with a rate of 10 kHz on all channels. The voltage range of each channel can be configured independently to a range of  $\pm 1$  V,  $\pm 5$  V, or  $\pm 10$  V. The ranges of interest are  $\pm 10$  V for the monitoring of the supply voltage  $U_s$ , and  $\pm 1$  V to record the voltage over the shunt. Here, smaller ranges would be preferable, but are not available on affordable hardware. The errors in the respective ranges are given in the manual<sup>1</sup> as 0.68 mV in the 1 V range and 5.66 mV in the 10 V range.

More interesting is the minimum voltage drop caused by the DUTs while running. This is where the resolution of the measurement card results in the highest error. The minimum voltage drop caused by the current drawn by the DUTs while idle is 30 mV. Hence, the maximum relative error of the measurements can be calculated as

$$\max\left(\frac{\Delta P}{P}\right) = \max\left(\sqrt{\left(\frac{\Delta I_s}{I_s}\right)^2 + \left(\frac{\Delta U_s}{U_s}\right)^2}\right) \quad (4.4)$$

Here, the maximum error on the voltage channel is calculated by

$$\max\left(\frac{\Delta U_s}{U_s}\right) = \frac{\max(\Delta U_s)}{\min(U_s)} = \frac{5.66mV}{4.8V} = 0.12\% \quad (4.5)$$

<sup>1</sup> <http://www.mccdaq.com/PDFs/manuals/USB-1608FS-Plus.pdf> accessed 2017-01-18

and the error on the current channel by

$$\max\left(\frac{\Delta U_m}{U_m}\right) = \frac{\max(\Delta U_m)}{\min(U_m)} = \frac{0.68\text{mV}}{30\text{mV}} = 2.26\%. \quad (4.6)$$

Factoring in the accuracy of the measurement shunt, the resulting error on the current channel is

$$\max\left(\frac{\Delta I_s}{I_s}\right) = \max(\sqrt{0.0226^2 + 0.01^2}) = 2.47\% \quad (4.7)$$

leading to an overall error of

$$\max\left(\frac{\Delta P}{P}\right) = \max(\sqrt{0.0247^2 + 0.0012^2}) = 2.47\%. \quad (4.8)$$

The resulting error of 2.47% is the upper bound of errors and includes both systematic and stochastic errors. By averaging over a larger number of samples, the stochastic error is reduced. Still, this leaves the systematic error, which may be caused by linear offsets e.g. in the power meter. Hence, considering the maximum error to be below 2.5%, a sufficient accuracy for the selected measurement task is achieved.

Another possible error source is the sampling frequency of the measurement card. The maximum supported rate for a single channel is 10 kHz. This frequency is deemed to be sufficient for the measurement task, as it can reasonably be expected that the power conditioning on the DUTs includes filtering of the input voltage, smoothing the current draw over a few 100 ms. Hence, the chosen measurement approach is considered feasible, and measurements are recorded with the maximum supported sample rate.

The measurements are recorded on a PC using custom software based on the DAQFlex API<sup>2</sup>, for which Android libraries are available from the vendor. Based on these, a Java application is written, polling the measurement data from the measurement card and writing these into a CSV file. Thus, a versatile measurement solution is available, facilitating measurements with different OSs. Besides recording the raw samples in a file, the software also supports on-the-fly conversion and aggregation of the data, thus largely reducing the size of recorded data sets and post-processing efforts. Samples can be averaged over a configurable interval and the power consumption depending on the configured measurement shunt calculated.

The same personal computer (PC) is used for controlling the measurements and as remote endpoint for throughput measurements. Using the same machine is feasible, as the performance of the DUTs is generally lower. Thus, no negative influence on the measurement accuracy is expected. Still, during test runs the system utilization is periodically checked to ensure that no resource on the measurement PC is close to capacity.

As the goal of this study is to analyze the power consumption of the individual components of the SBCs, and the independent measurement of the individual components is not

<sup>2</sup> <http://www.mccdaq.com/solutions/DAQFlex-Solutions.aspx> accessed 2017-01-18

Listing 4.1: Source code to keep the CPU busy

```
int main(int argc, char* argv){
    volatile int x=0;
    volatile int y;
    while(1){
        y=x+x;
        x++;
    }
    return 0;
}
```

easily possible, a regression based approach is chosen. Hence, the DUT is put into different utilization and energy states, for each of which the state of the device and the power consumption are recorded simultaneously. Targeted components are CPU, random access memory (RAM), SD-card, and network I/O, both for Ethernet and WiFi. For each of these, the utilization is recorded and later correlated with the measured power consumption.

The utilization of the individual components is obtained by monitoring the `/proc` file system. This supports a maximum update frequency of 1 Hz. Compared to the power measurements, this is a comparatively low resolution. To collect a sufficient number of samples in each state, the DUT must be kept stable for a considerable duration. The decision to use the `/proc` file system is further influenced by the overhead of the measurement. As each measurement affects the accuracy of the recorded values, a low overhead is required. In particular on the SBCs, where the system monitoring can only run on the DUT, the influence of additional monitoring tasks must be minimized to not affect the quality of the measurements.

The influence of the CPU utilization on the power consumption is analyzed by stressing the CPU in incremental steps. For this, two different tools are used. The load on the CPU is generated by a simple C program, adding numbers. The source code is given in Listing 4.1. Clearly, this is not a comprehensive benchmark. Still, considering the variety of optimizations conducted when designing processors and their extensions, this serves as a baseline benchmark. More sophisticated benchmarks may give a more accurate view of the actual performance, but also increase the resulting model complexity due to dependencies on the load patterns.

The load on the CPU is limited using *cpulimit*<sup>3</sup>. Contrary to the published version of the tool limiting the load of a configured process to a given CPU percentage, here the overall CPU utilization needs to be limited. Still, as only one process on the DUT actively creates load, the adaptation is simple. The only modification required is changing the

<sup>3</sup> <http://cpulimit.sourceforge.net/> accessed 2017-01-19

load monitoring from a single process ID (PID) to the system utilization. The remaining functionality of the tool is left unchanged.

Besides knowing the approximate CPU utilization from the test setup, the actual utilization is also monitored using a custom load monitoring tool. This polls the `/proc` file system once every second to fetch the *user*, *nice*, *system*, and *idle* utilization of the CPU. To keep its interference to a minimum, the monitor stores the recorded values in a RAM disk, thus preventing any influence of SD-card writes on the power consumption. The system utilization is read from `/proc/stats`, containing different information on the CPU usage. The first line summarizes the CPU utilization over all cores. Then, for each logical CPU core one additional line follows. The utilization is accounted in a granularity of 1/100 s counting the time spent in the different categories *user*, *nice*, *system*, and *idle* and others as defined for the given architecture. Hence, the busy value  $c_{\text{busy}}$  is calculated as

$$c_{\text{busy}}[t] = c_{\text{user}}[t] + c_{\text{nice}}[t] + c_{\text{system}}[t] \quad (4.9)$$

over the summary line. As the CPU utilization values are counters, these are converted to a percentage by subtracting the values from the previous observation and normalizing by the overall number of cycles in a given interval. Hence, the relative CPU utilization is defined as

$$u[t] = \frac{c_{\text{busy}}[t] - c_{\text{busy}}[t-1]}{c_{\text{total}}[t] - c_{\text{total}}[t-1]}. \quad (4.10)$$

Still, to limit influence of the measurements on the performance of the DUT, only the raw data is collected and the relative CPU utilization is calculated during the evaluation only.

For the throughput tests, *iperf* is used. The upload is measured by running an *iperf* server on the measurement PC. For the download measurements the server is started on the DUT. The problem of TCP slow start is mitigated by using UDP for the throughput measurements. Further, target data rates can be configured, closely approximating the desired network utilization.

One drawback of *iperf* is that it generally uses random data for its benchmarks. Hence, the client creates a considerably high CPU load. This effect of the final power consumption is mitigated by also monitoring the CPU utilization during the throughput tests. Thus, the influence of any CPU load, also caused by other tasks, is eliminated in the regression step.

Similar to the CPU monitoring, also the network monitoring is conducted using a compiled binary reading the network utilization from `/proc/net/dev`. These values are stored in RAM and when the measurement has finished, written to the SD-card. Depending on the type of the test (e.g. Ethernet/WiFi), the utilization of the respective interface is monitored. Monitoring these values further ensures that the actual data rate on the interface is included in the analysis instead of the configured rate of the benchmark.

The WiFi measurements show some peculiarities in that one end of the connection has to provide the AP functionality, meaning that it coordinates the connection on medium

access control (MAC) layer and sends periodic broadcasts. As this is expected to cause additional overhead on the DUT, the AP functionality is run on the measurement PC. Further, the integrated WiFi chip set on the Raspberry Pi 3 does not support running a WiFi AP at all. Still, as the measurements show, no noteworthy power saving mechanism is visible on most devices when evaluating the collected data. Hence, running the AP functionality on the SBCs, no significant change in power consumption is expected, but for scientific rigor a detailed analysis would be required.

The accuracy of the tests is maximized by shutting down all non-essential processes running on the DUT. Essential processes required for the tests are *udev*, the dynamic host configuration protocol (DHCP) client, *dbus*, and the secure shell (SSH) daemon. By minimizing the number of running processes, also the ones only active periodically, the accuracy of the load limiting script can be maximized by minimizing external interference, thus minimizing the variance as observed in the measurements. Naturally, during the measurements, also system monitoring and load generation scripts are running.

Test measurements have shown that RAM utilization does not measurably affect the power consumption of the DUT. Tests of the SD-card power consumption were aborted, as the load on the cards was too high. Several cards failed due to the demand of these tests. Preliminary observations have shown that large parts of the performance seem to be related to caches, while sustainable write rates were comparatively low. The power consumption of the DUTs was not visibly affected by these.

As measurements are conducted on both the DUT and a measurement PC simultaneously also the alignment of the clocks must be considered. To assure maximum accuracy, the clock of the DUT is synchronized to a network time protocol (NTP) server running on the measurement PC. Still, requirements on the clock synchronization are less stringent considering the low sampling rate of the system monitoring on the DUT. Further, as the device is kept in a stable state, a slight misalignment between timestamps can be tolerated. To assure high measurement accuracy, the monitoring duration is configured to exceed the desired test duration by a few seconds, such guaranteeing that the device is in a stable state before samples to be used in the evaluation are recorded.

The devices measured and modeled over the course of the study are the Raspberry Pi, Raspberry Pi 2, Raspberry Pi 3, as well as the Odroid C1, Odroid C2, and the Cubieboard v3 for reference. The details of these devices are summarized in Table 4.1. The processors range from a 700 MHz ARM v6 on the original Raspberry Pi to a quad core ARM v7 with 1.5 GHz on the Odroid C1 and C2. Similarly, the RAM ranges from 512 MB on the Raspberry Pi to 2 GB on the Cubieboard v3 and Odroid C2. The network interfaces are fast Ethernet types on the Raspberry Pi family and Gigabit interfaces on the other devices. The Cubieboard v3 and the Raspberry Pi 3 have a built in WiFi chip, which is used for throughput measurements. On the remaining devices the TP-Link TL-WDN 3200 USB dongle is used, except on the Odroid C2, where no suitable driver for the 64-bit OS was available.

Table 4.1: Overview of the selected single board computers

	Raspberry Pi B	Cubieboard v3	Odroid C1	Raspberry Pi 2 B	Odroid C2	Raspberry Pi 3 B
<b>Release Year</b>	2/2012	10/2012	12/2014	2/2015	02/2016	02/2016
<b>CPU Type</b>	ARM1176JZF-S	ARM Cortex-A7	ARM Cortex-A5	ARM Cortex-A7	ARM Cortex-A53	ARM Cortex-A53
<b>Instruction Set</b>	ARMv6	ARMv7	ARMv7	ARMv7	ARMv8	ARMv8
<b>Number cores</b>	1	2	4	4	4	4
<b>Processor Freq.</b>	700 MHz	1 GHz	1.5 GHz	900 MHz	1.5 GHz	1.2 GHz
<b>RAM</b>	512 MB	2 GB	1 GB	1 GB	2 GB	1 GB
<b>Ethernet</b>	FE	GbE	GbE	FE	GbE	FE
<b>WiFi</b>	TL-WDN3200	BCM43362	TL-WDN3200	TL-WDN3200	—	BCM43438

Based on the initial idea for the Raspberry Pi of supporting teaching activities, comparatively powerful devices have emerged, still fulfilling the requirements of the original Raspberry Pi, but largely extending possibilities for other applications. Popular usage examples are media centers or home servers. Here, the different SBCs reveal their particular strengths. The Cubieboard v3 as one of the earlier SBCs provides a serial ATA (SATA) port. Combined with the Gbit network interface, it may serve as an energy efficient file server. Contrary the Odroid family, being largely based on mobile SOCs, provides extended graphical capabilities, making it a reasonable choice for a media center. All devices provide GPIO pins, following their initial idea of supporting easy and versatile extensibility to the end user.

#### 4.1.2 Measurements

The measurements run on the DUT consist of CPU and network performance tests. The influence on power consumption of the CPU is measured and later analyzed in a granularity of 10%. The network I/O related power consumption is measured for each interface in steps of 10 Mbps up to a rate of 100 Mbps. For the devices nominally supporting data rates up to 1 Gbps, additional measurements in steps of 100 Mbps up to the maximum rate of 1 Gbps are conducted. The data on the devices and the measurement PC is collected with the setup as described above.

Before the tests are started, the clocks are synchronized using NTP. Then, all non-essential services on the DUTs are shut down. After that, the desired tests are configured on the device. All this is coordinated from the measurement PC via SSH, thus allowing a high degree of automation. The tests are started with a configurable delay, allowing the experimenter to adjust environmental parameters as required. This is mainly the removal of the Ethernet cable in the case of CPU or WiFi measurements, thus eliminating the effect of background activity on the measured power consumption. After the tests are finished, another script is used to collect the system monitoring files from the DUT and store them



depending on the device model and measurement type on the local machine for later processing.

Due to the comparatively low sampling frequency of the system monitoring, a trade-off between measurement duration and accuracy needs to be found. To keep the overall measurement duration manageable, a test duration of 900 s (e.g. 15 min) was selected. Hence, for each operating state of the device 900 samples are available, permitting a meaningful statistical analysis. Considering the number of tests to be run (e.g. 10 CPU, 20 or 40 Ethernet, and 20 WiFi), the overall measurement duration of a single DUT is 12.5 h, or 17.5 h. This makes profiling a single device possible within a single day, also considering re-configuration delays between measurements.

As most of the measurements are automated, the required manual intervention is minimal. The Ethernet cable must be disconnected before the CPU measurements, then re-connected for the re-configuration of either WiFi or Ethernet test, and then another time before the remaining series of measurements.

When measuring WiFi some additional considerations are required. As the electromagnetic spectrum is a shared resource, and access to the 2.4 GHz band is unlimited, depending on time and location considerable interference can be observed. Particularly in urban environments the number of WiFi, *Bluetooth* or otherwise wireless devices using the 2.4 GHz band is high. Hence, the measurements were conducted in a sub-urban environment, where the density of devices in the respective spectrum was comparatively low. The WiFi channel was configured to use an otherwise unused channel. Further, the measurements were run during the night, such further reducing the chance of additional interference.

Figure 4.2 shows exemplary measurements of two different devices. Figure 4.2a depicts the measured values for the CPU-related power increase of the Odroid C2. The horizontal axis shows the observed CPU utilization, the vertical axis the resulting measured power. The color density indicates the number of samples observed in each bin of 1 Mbps width and 10 mW height. The plot contains all measured values. As is visible in the plot, the variance of both CPU utilization and resulting power consumption is quite low.

Figure 4.2b shows the Ethernet download tests on the Odroid C2. The horizontal axis indicates the data rate on the Ethernet interface. The vertical axis shows the power consumption caused by the network transmission. This is calculated by subtracting the power consumption as calculated based on the CPU power model according to the CPU utilization as monitored when recording the network measurement. The plot shows the high density of measurements for data rates below 100 Mbps, and then the lower measurement interval of 100 Mbps for the higher data rates. The measurements for each measurement fall in a quite narrow range. Still, compared to the configured rate, the observed rates are slightly higher. This is caused by the configuration defining the application level payload rate, while the system monitoring using the `/proc` file system returns the bytes transferred on the MAC layer, including IP and MAC overhead.

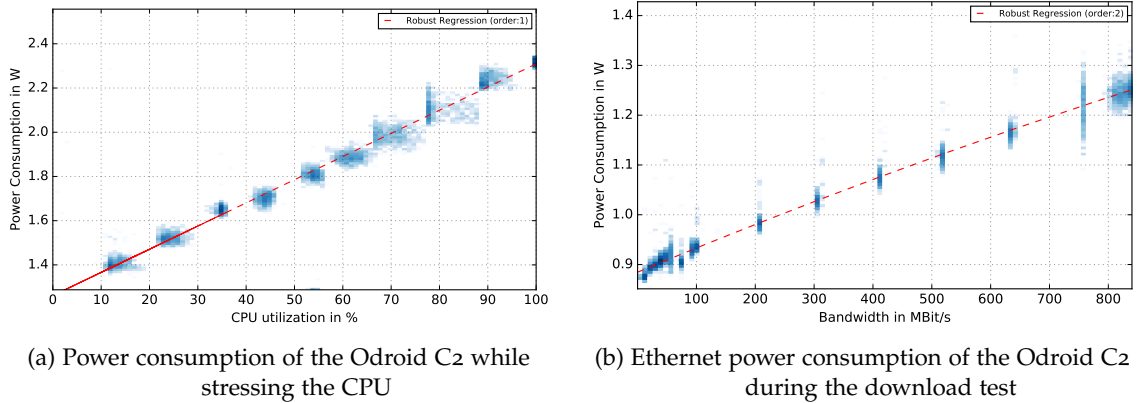


Figure 4.2: Exemplary power consumption and regression for CPU and Ethernet measurements

#### 4.1.3 Modeling of the Power Consumption

Based on the collected data, regression models are built to later estimate the power consumption of the device based on the current operating state. Using system monitoring values from the `/proc` file system is beneficial for two reasons. First, these values are readily available on any device, thus eliminating the need to install additional monitoring tools on the DUT. Secondly, as these values are collected by the system anyways, the additional overhead on the DUT is minimized. Thus, the derived models can be run permanently, with minimal effects on the performance of the SBC.

The energy consumption of the DUTs is modeled using polynomials with different degrees. The analysis shows that for the CPU a linear model is sufficient. The network performance in the case of Ethernet connections can often be modeled using a linear model, but some SBCs show nonlinear effects. Hence, for generality, second order models are selected. In the case of WiFi connections, generally second order models are required.

The derived power models for the selected DUTs are summarized in Table 4.2. The first column indicates the function described by the formula in the second column, while the third column gives the root mean square error (RMSE) of the estimate. Generally, the error is lower for the CPU and Ethernet models compared to the WiFi models. This is intuitive, as the potential error sources are more restricted as for CPU or Ethernet measurements. Generally, the error is in the tens of mW, up to 178 mW for WiFi uploads on the Cubieboard.

Figure 4.3 compares the derived influence of CPU power consumption on the different SBCs. Here, it must be considered that the devices show largely different processor frequencies, number of processor cores, and hence number of possible computations per

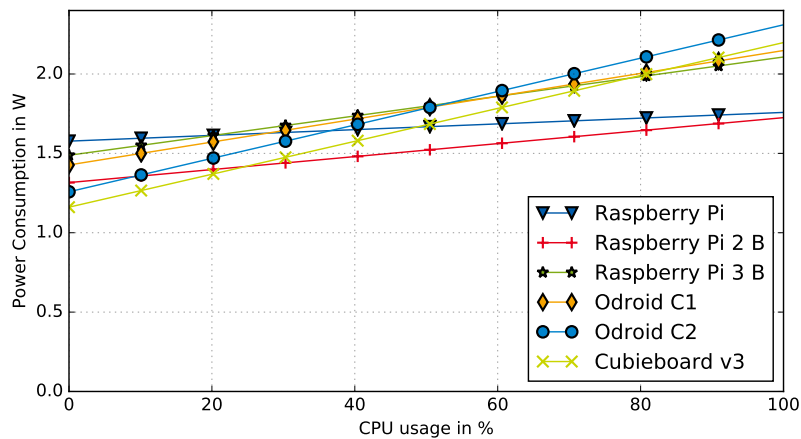


Figure 4.3: Comparison of the CPU power consumption of the analyzed platforms

unit time. This is caused by the selected measurement procedure, monitoring the relative CPU utilization. Other approaches were also considered, but CPU performance comparison itself is a complex area. Further, the decision on using relative CPU utilization as read from `/proc/stat` increases the versatility of the generated models.

The lowest idle consumption is visible on the Cubieboard with 1.16 W. Under load, the original Raspberry Pi and the Raspberry Pi 2 show the lowest energy consumption. The highest idle consumption is observed on the original Raspberry Pi, while under load, the Odroid C2 consumes the most energy, followed by the Cubieboard, Odroid C1, and Raspberry Pi 3. The original Raspberry Pi shows the lowest increase in power consumption under load, while the Odroid C2 shows the largest increase. When considering a highly dynamic load, thus clearly the Odroid C2 should be chosen. Also regarding the higher processor count and frequency, more tasks can be run on a single machine. Still, if only less demanding tasks are to be run, and an original Raspberry Pi is sufficient, clearly this or the Raspberry Pi 2 should be chosen, consuming approximately 800 mW less than the next energy efficient SBCs.

The power consumption of data up- and downloads on the available interfaces is compared in Figure 4.4. For each interface, the model fitted to the measured data is plotted up to the maximum observed rate on the respective interface. Thus, these plots, besides indicating the power consumption of a specific rate, also indicate the limits of the generated models.

Figure 4.4a compares the cost of downloads on the Ethernet interface of the respective DUT. An interesting observation is the negative influence of the Ethernet traffic on the power consumption of the Raspberry Pi 3. This is caused by frequent higher power estimates on the Pi 3 during CPU measurements. These cause the model, in particular for lower CPU utilization to overestimate the idle consumption. As this higher power consumption is not visible in the generated load, it must be caused by some low-level

functionality not visible to the CPU monitoring. This may be periodic probing of available interfaces, for example Ethernet and WiFi, although configured to be off. The cause for this was researched, but no reason was found. Still, as the RMSE of the Ethernet measurements shows, the resulting model is comparatively accurate.

The cost of connecting an Ethernet cable to the different SBCs covers a range from a few mW to almost 1 W. This is caused by the different chip sets, their capabilities, and configuration. On some devices (Raspberry Pi 2, Raspberry Pi 3, Cubieboard) these seem to be always active, while on others (Raspberry Pi, Odroid C1, Odroid C2) these seem to be activated on demand, consequently reducing the power consumption when idle.

The Ethernet throughput of the Cubieboard clearly does not satisfy expectations of a Gigabit interface by barely supporting data rates of 100 Mbps. Contrary, the Raspberry Pi family reasonably fulfills specifications. Interesting are the high data rates of the Odroids with achieved rates of 800 Mbps to 900 Mbps.

The Ethernet upload measurements (cf. Figure 4.4b) generally show a similar power consumption to the download measurements. Still, in some cases the influence of the CPU load seems to affect the resulting power models. This may be caused by the comparatively high demand of random number generation on the DUT, thus influencing the resulting range of the model. The lower upload rates on the Raspberry Pi line and the Cubieboard v3 are indicative for this. Here, further investigations are recommended. Interesting is the large difference in maximum data rates between the Odroid C1 and C2, nominally featuring SOCs with similar performance. Still, the 64-bit OS achieves considerably higher data rates. Here, an investigation on the influence of random number generators would be interesting.

The WiFi measurements show a similar picture. The downloads in Figure 4.4c show an idle consumption between 300 mW and 1.1 W. Under load, these increase up to 2.7 W in downlink direction. When uploading data (cf. Figure 4.4d), these increase to 3.5 W. Interesting here are the differences in observed packet rates. As download tests stress the SBCs only minimally, the differences here must be accounted to the network monitoring on the DUT, only registering successfully received packets, while set packets discarded by the modem or lost on the link are counted. Hence, the download rates as given in Figure 4.4c are definitely feasible, while the uplink measurements would also require accounting of the received packets on the remote PC.

Using these power models, the power consumption of one of the modeled SBC can be estimated by adding the power consumption of the individual components. Consequently, the system power consumption is given as

$$P = P_{\text{idle}} + P_{\text{cpu}}(u) + \sum_{\text{if}} (P_{\text{if,idle}} + P_{\text{if,up}}(r) + P_{\text{if,dn}}(r)). \quad (4.11)$$

The individual parameters and range of validity are taken from Table 4.2. Here,  $P_{\text{idle}}$  is the idle consumption of the device without load. CPU load is accounted by  $P_{\text{cpu}}(u)$ , where  $u$  is the CPU utilization in the range 0 to 1 as defined in Equation 4.10. For each

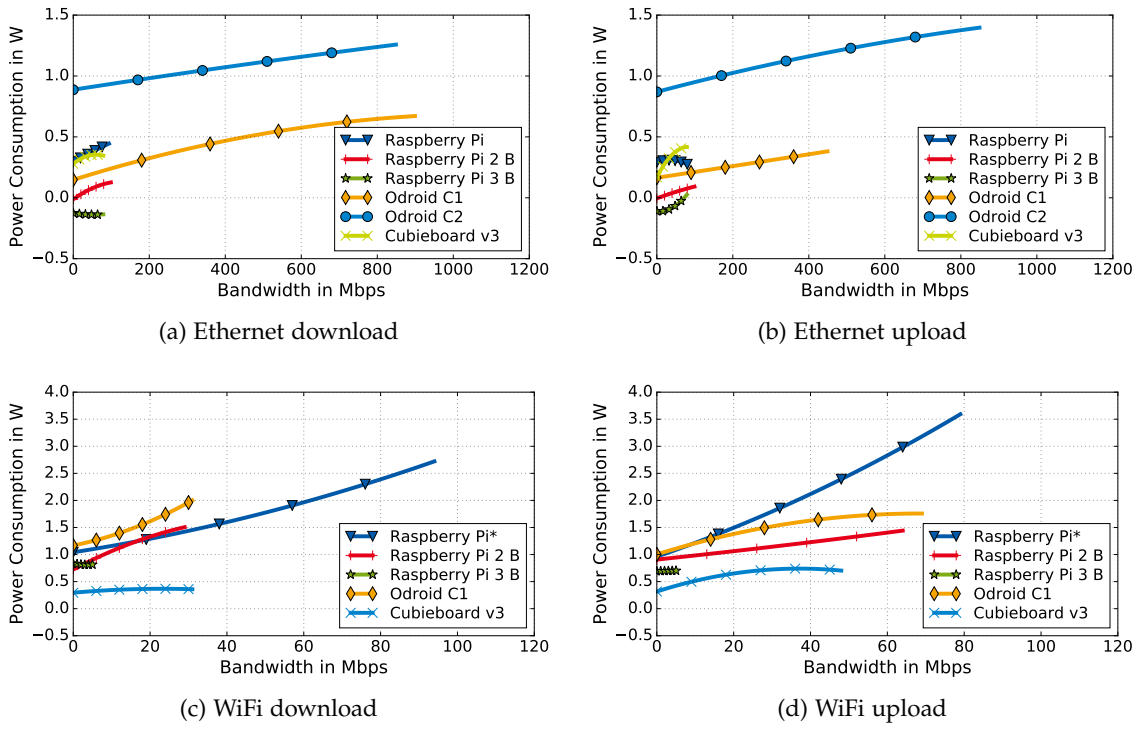


Figure 4.4: Comparison of the derived power models for Ethernet and WiFi up- and download on the different platforms (\*no encryption)

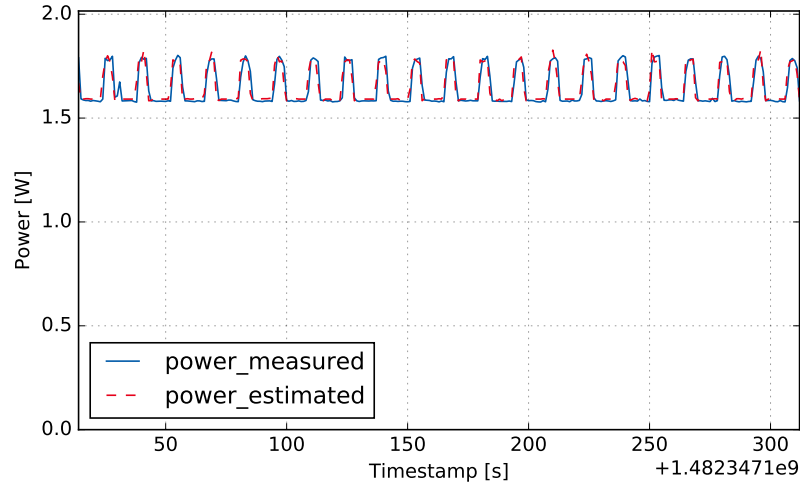


Figure 4.5: Reference measurement on the Odroid C1

active interface  $if$ , the idle power  $P_{if,idle}$  is added. Depending on the load on the respective interface, the power consumption caused by upload  $P_{if,up}(r)$  and download  $P_{if,dn}(r)$  is added. Here, the data rates as observed on the respective interface are inserted into the respective function from Table 4.2, where the rate  $r$  is measured in Mbps. The ranges of validity are given in the third column of Table 4.2.

#### 4.1.4 Evaluation of the Power Models

The validity of the presented models is verified by emulating a video-streaming scenario. As already described in the motivation, these SBCs may be used within future networks to extend cloud services or CDNs to the edge of the network. Hence, a web server is installed on the DUTs, serving a bursty data stream to a mobile client. This is modeled to emulate a HD video stream with an average bit rate of 1 Mbps. The burst interval is configured to 30 s. During the bursts, the maximum available bandwidth is used, thus closely approximating the behavior of conventional video streaming. The accuracy of the derived models is established by simultaneously recording the power consumption with the external power meter and comparing these values to the ones derived based on system utilization in combination with the presented power models. Figure 4.5 shows the results of a reference measurement on the Odroid C1. The continuous line shows the power as measured by the power meter, while the dashed line shows the power consumption as derived using the above described models. The peaks visible on the plot indicate the network activity. Generally, a good fit between model and measurement is visible.

The overview of the recorded and estimated power consumption over all reference measurements on the DUTs is given in Table 4.3. Generally, the maximum error  $P_{e,max}$  is

Table 4.2: Overview of the generated power models (\*measured without encryption)

Function	Model [W]	Range [Mbps]	RMSE [W]
Raspberry Pi B			
$P_{idle}$	= 1.577		
$P_{eth,idle}$	= 0.294		
$P_{wlan,idle}$	= 0.942		
$P_{cpu}(u)$	= 0.181u	[0, 1]	0.016
$P_{eth,up}(r)$	= $-15.2e^{-6}(r/Mbps)^2 + 1.005e^{-3}r/Mbps - 0.002$	[0, 82]	0.008
$P_{eth,dn}(r)$	= $-6.531e^{-6}(r/Mbps)^2 + 1.6344e^{-3}r/Mbps + 0.003$	[0, 95]	0.017
$P_{wlan,up}(r)*$	= $112.8e^{-6}(r/Mbps)^2 + 24.386e^{-3}r/Mbps + 0.020$	[0, 80]	0.071
$P_{wlan,dn}(r)*$	= $71.988e^{-6}(r/Mbps)^2 + 11.003e^{-3}r/Mbps + 0.010$	[0, 95]	0.026
Raspberry Pi 2 B			
$P_{idle}$	= 1.316		
$P_{eth,idle}$	= -0.019		
$P_{wlan,idle}$	= 0.899		
$P_{cpu}(u)$	= 0.409u	[0, 1]	0.038
$P_{eth,up}(r)$	= $1.95e^{-06}(r/Mbps)^2 + 1.17e^{-03}r/Mbps + 0.014$	[0, 100]	0.014
$P_{eth,dn}(r)$	= $-8.54e^{-06}(r/Mbps)^2 + 2.25e^{-03}r/Mbps + 0.006$	[0, 100]	0.012
$P_{wlan,up}(r)$	= $13.4e^{-03}(r/Mbps)^2 + 7.50e^{-03}r/Mbps + 0.008$	[0, 65]	0.043
$P_{wlan,dn}(r)$	= $-0.37e^{-03}(r/Mbps)^2 + 37.72e^{-03}r/Mbps - 0.176$	[0, 30]	0.081
Raspberry Pi 3 B			
$P_{idle}$	= 1.488		
$P_{eth,idle}$	= -0.1176		
$P_{wlan,idle}$	= 0.7645		
$P_{cpu}(u)$	= 0.6191u	[0, 1]	0.155
$P_{eth,up}(r)$	= $26.2^{-06}(r/Mbps)^2 + 0.357e^{-03}r/Mbps + 0.007$	[0, 80]	0.039
$P_{eth,dn}(r)$	= $-4.33e^{-06}(r/Mbps)^2 + 0.485e^{-03}r/Mbps - 0.007$	[0, 80]	0.007
$P_{wlan,up}(r)$	= $-0.25e^{-06}(r/Mbps)^2 + 1.99e^{-03}r/Mbps - 0.072$	[0, 6]	0.013
$P_{wlan,dn}(r)$	= $1.85e^{-03}(r/Mbps)^2 + -13.5e^{-03}r/Mbps + 0.072$	[0, 6]	0.022
Cubieboard v3			
$P_{idle}$	= 1.161		
$P_{eth,idle}$	= 0.0		
$P_{wlan,idle}$	= 0.306		
$P_{cpu}(u)$	= 1.037u	[0, 1]	0.038
$P_{eth,up}(r)$	= $12.2e^{-06}(r/Mbps)^2 + 1.99e^{-03}r/Mbps + 0.166$	[0, 80]	0.055
$P_{eth,dn}(r)$	= $-16.5e^{-06}(r/Mbps)^2 + 2.11e^{-03}r/Mbps - 0.031$	[0, 80]	0.024
$P_{wlan,up}(r)$	= $-0.307e^{-03}(r/Mbps)^2 + 22.8e^{-03}r/Mbps + 0.011$	[0, 49]	0.178
$P_{wlan,dn}(r)$	= $0.137e^{-03}(r/Mbps)^2 + 6.33e^{-03}r/Mbps - 0.011$	[0, 32]	0.064
Odroid C1			
$P_{idle}$	= 1.427		
$P_{eth,idle}$	= 0.156		
$P_{wlan,idle}$	= 1.086		
$P_{cpu}(u)$	= 0.721u	[0, 1]	0.023
$P_{eth,up}(r)$	= $20.1e^{-09}(r/Mbps)^2 + 0.47e^{-03}r/Mbps + 0.008$	[0, 450]	0.062
$P_{eth,dn}(r)$	= $-0.44e^{-06}(r/Mbps)^2 + 0.97e^{-03}r/Mbps - 0.008$	[0, 900]	0.019
$P_{wlan,up}(r)$	= $-0.16e^{-03}(r/Mbps)^2 + 22.0e^{-03}r/Mbps - 0.082$	[0, 70]	0.126
$P_{wlan,dn}(r)$	= $0.41e^{-06}(r/Mbps)^2 + 14.0e^{-03}r/Mbps + 0.082$	[0, 32]	0.086
Odroid C2			
$P_{idle}$	= 1.258		
$P_{eth,idle}$	= 0.878		
$P_{cpu}(u)$	= 1.052u	[0, 1]	0.051
$P_{eth,up}(r)$	= $-0.24e^{-06}(r/Mbps)^2 + 0.83e^{-03}r/Mbps - 0.009$	[0, 850]	0.012
$P_{eth,dn}(r)$	= $58.0e^{-09}(r/Mbps)^2 + 0.48e^{-03}r/Mbps + 0.009$	[0, 850]	0.010

Table 4.3: Evaluation of the accuracy of the power models

Device	$P_{e,max}$ [W]	$\overline{P_e}$ [W]	RMSE [W]	$\overline{P_{e,rel}}$ [W]
Pi 1	0.282	-0.137	0.141	6.5%
Pi 2	0.098	0.031	0.036	2.5%
Pi 3	0.718	-0.195	0.275	9.3%
CT	0.368	-0.013	0.120	6.3%
C1	0.121	-0.002	0.032	1.0%
C2	0.281	-0.016	0.076	1.7%

in the range of a few 100 mW. The worst error is observed on the Raspberry Pi 3, which is not unexpected considering the high variance in the CPU model. The mean error  $\overline{P_e}$  is mostly negative, with  $-0.055$  mW averaged over all tests, meaning that the estimated power is little too high. The RMSE of maximum 120 mW is expected considering the errors given in Table 4.2. The relative errors  $\overline{P_{e,rel}}$  are generally below 10%. The highest accuracy was visible on the Odroid C1 and C2 with errors of 1.0% and 1.7% respectively.

These power models can be used for different tasks. First, the power consumption of the devices can be estimated while operating, without the need to attach a power meter to each of these. Secondly, these models may be used to simulate the effect of load or traffic distribution changes on the power consumption of a larger deployment of these SBCs. By only relying on readily available system monitoring values, the overhead when estimating the power consumption is minimal. For the evaluation of the system power consumption, the system utilization values may either be collected on a central server, estimating the power consumption based on the device type. Thus, the monitoring of an existing deployment would not require changes, while still power estimates are available on the server. If no central system monitoring was available, it is also possible to estimate the system power consumption directly on the device, and only send this value to a central monitoring instance, such minimizing the network and monitoring overhead. Knowing the power consumption is particularly interesting in the case of either different hardware located at different places, or when devices may be shut off to save power, thus increasing the efficiency of the remaining devices, and eliminating the idle power of unused SBCs.

The analysis of the power consumption of a larger deployment of SBCs was analyzed in a large scale study conducted by the SmartenIT project<sup>4</sup>. The objective of the study was to analyze the effectiveness and potential of energy savings in a decentralized, social enhanced video caching and delivery scenario. For this, the RB-HORST system [LPB+15], implementing the required functionality, was run on 25 devices distributed over Europe. The participants of the study were mostly researchers at the project partner's locations.

<sup>4</sup> <http://www.smartnit.eu/> accessed 2017-01-22



The RB-HORST system is evaluated by each user deploying one SBC at their home premises and configuring a personal Facebook account on it. The software running on the devices was configured to periodically check the user's Facebook stream to determine videos to be downloaded to the local device. After the ID of the video is determined, the best source for a download is determined. This is identified based on the autonomous system (AS) number of the currently used ISP. Thus, the respective item can be loaded from the closest CDN node. To further reduce traffic cost of the ISP, other participating SBCs were checked for the availability of the respective item. This is conducted by looking up addresses and locations of the video in a distributed hash table (DHT). Thus, in case the video is available from within the same ISP no additional traffic is caused on the ISPs external links.

The locally cached items are made available to the end user via a WiFi AP. When the user requests a cached video from a connected device, the content is served by the SBC. In the case it is not yet available, it is streamed from the cheapest location as described above and simultaneously cached for an eventual further request. Thus, the QoE of the end user can be improved by providing high-quality content, which does not need to be loaded via a bandwidth limited link in real-time. Further, the ISP is unloaded by fetching content from within the same network. By pre-fetching content during idle periods of the network or CDN, e.g. during the night, the impact on the network infrastructure can further be reduced, leading in the long term to both monetary and energy efficiency benefits, as deployed hardware may be used longer to fulfill end-user requirements.

The described functionality is further extended by providing a *social* AP at the end-user's premises to friends and relatives. This is achieved by installing a custom built App on their smartphones exchanging the WiFi configuration with the SBC when a *friend* relation on Facebook is identified. Thus, the efficiency of the system can further be improved, by providing content with mutual interest to multiple end users at fixed cost.

For this scenario the power consumption of the full deployment is analyzed. On each of the deployed devices a system monitor is installed, periodically uploading usage metrics to a central server. Based on CPU consumption and network traffic on both Ethernet and WiFi interfaces the power consumption of the individual node and the full deployment is derived. The resulting power consumption is shown in Figure 4.6. The gray lines indicate the observed cumulative distribution function (CDF) of the individual system's power consumption. Here, it becomes apparent that some devices show a constantly higher load, while others are not as heavily used. The black line describes the mean power consumption of the full deployment. As is visible in the graphic, the power consumption varies only slightly. This is caused by the high idle power of both the SBCs, and the attached WiFi dongles. Further, the load on the devices is comparatively low, caused by the large capacity of the selected devices.

Considering the mean power consumption of these devices, the power consumption of the full deployment can be derived. The overview of the measurements is also given

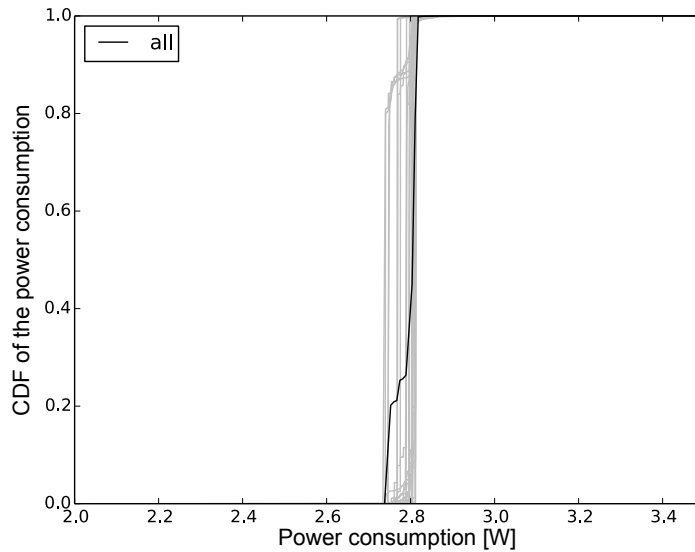


Figure 4.6: Evaluation of the power consumption of a large deployment of SBCs

in Table 4.4. The inner 90<sup>th</sup> percentile is in the 2.74 W to 2.81 W. The minimum value of 0.072 W is clearly a measurement error, as it is not possible to measure such low power with the model-based approach. This is likely caused by measurement outages, causing one extremely low estimate. For the 25 devices participating in the study, the resulting overall power consumption is 69.5 W. This compares to a conventional PC when lightly used. Considering that a CDN node usually serves a few hundred end users, the energy consumption is comparable. Still, by reducing the traffic in the network backbone, currently deployed hardware may be used for longer, reducing the increasing energy consumption in the core network. By keeping traffic local, also the ISP benefits from reduced traffic cost. Further, considering that the functionality currently implemented on SBCs may be integrated in future home routers, the additional energy cost for the end user becomes negligible. In case these are deployed by the ISP, these may also be offered as a decentralized cache to content providers, thus providing an alternative to current CDNs. Then, redirection to the respective node would be done via the DNS.

A further interesting point when working with SBCs is: How does the power consumption and computational efficiency of SBCs compare to conventional laptops and how does it develop over time? Koomey et al. [KBS+11] show that the energy efficiency of computation is doubling every 18 months. This energy efficiency is particularly important in areas where available energy is strictly limited, as is the case in mobile devices, or where heat dissipation is problematic or costly, as is common in high density computing environments like data centers and high-performance computing. Hence, a current topic in high performance computing is the use of low power ARM processors [SRS16; MOF15;

Table 4.4: Analysis of the power consumption of the devices participating in the RB-HORST study

Metric	Value
Minimum power	0.072 W
Mean power	2.783 W
Median power	2.804 W
Maximum power	3.167 W
5% - 95% percentile	2.740 W - 2.811 W
System power consumption (25 devices)	69.5 W

JVO+13]. A similar trend is visible when considering smartphones. Here, the available energy is strictly limited, forcing vendors to minimize device energy consumption.

Based on the collected data, the computational capability of the DUTs is related to the maximum power consumption as derived from the above power models. In addition to the above devices, a short measurement of the power consumption of the Odroid U3 under load was conducted, permitting its inclusion in the comparison. For this, the number of CPU cycles is multiplied by the number of cores available on the respective platform, thus ensuring comparability with the results as given by Koomey. Further, the power consumption is normalized to kWh. The resulting computational efficiency in ops/kWh is shown in Figure 4.7, plotted over the time of the first release of the DUT. As time reference, the release date of the original Raspberry Pi was chosen, as it marks the begin of this new device class.

From Figure 4.7, it becomes apparent, that generally the Odroids are more energy efficient compared to the Raspberry Pi. The Odroid C1, C2, and the Raspberry Pi are all available for a similar price. Considering the Cubieboard and the Odroid U3, these were more expensive, and thus provided higher CPU capacities earlier. Still, these are considered in this comparison, as these also belong to the same class of SBCs.

In the four years since the release of the original Raspberry Pi, the energy efficiency has increased almost 10-fold. Fitting an exponential function to the data, a growth rate of 0.4276 is determined. This is close to the rate determined by Koomey (0.456). The corresponding doubling time is 1.62 years. The coefficient of determination of this fit is quite small ( $R^2 = 0.747$ ). This is caused by the small number of devices. If a higher confidence was required, additional SBCs would need to be analyzed and included in the analysis.

Comparing the above curve with the one derived by Koomey, an offset between both is visible. Compared to x86/64 processors, the SBCs show a lower computational efficiency. This is expected, as the development goal of ARM and x64/86 architectures is slightly different. While x64/86 processors are developed to minimize power consumption while lightly used, e.g. in laptops, while still providing fast response times to inputs, one of the

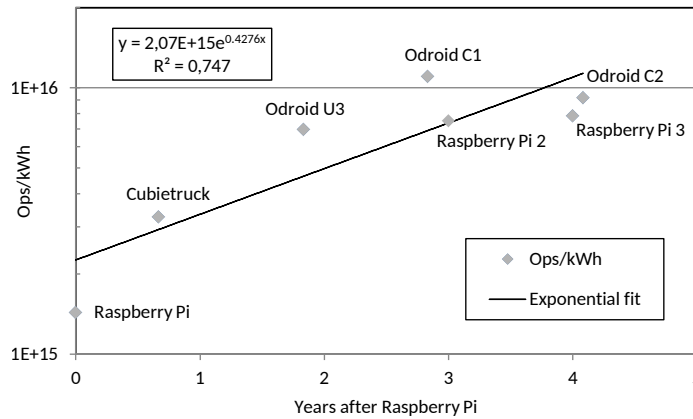


Figure 4.7: Progress of the energy efficiency per kWh since release of the original Raspberry Pi

current main areas of application of ARM processors are smartphones, requiring high performance and fast response times only while active. During inactive periods the energy consumption is minimized by implementing deep sleep modes. Further, on smartphones the available energy is more restricted, as battery capacity is limited. The reduced energy consumption of ARM processors while active but idle is minimized by switching off cores, or using lower power cores during low load periods as first included with the *big.LITTLE* architecture in the ARM Cortex-A7. Still, the main difference between both architectures is the absolute power consumption, prohibiting the use of x86/64 processors in highly mobile devices. Concluding, the above observations show that the computational efficiency of ARM processors under load is still inferior to x86/64 processors. Still, their absolute lower power consumption makes their use attractive in highly energy constricted environments like mobile devices.

#### 4.1.5 Summary and Conclusions

The previous section analyzes the power consumption of SBCs. Based on the idea of using low power ARM-based SBCs for extended network functionality at the network edge, their performance and energy consumption are analyzed. Focus of the presented work is the analysis of the influence of CPU and network load on the power consumption of the full system. First, the measurement setup is described, based on which the measurements of the SBCs are collected. The measurements comprise of power consumption measured using an external power meter and internal system monitoring metrics. Combining both, models for the load dependent power consumption of the DUTs are derived in a regression based approach. The analyzed devices are the Raspberry Pi line-up, the Odroid C line, and the Cubieboard v3. For each of these, the CPU and Ethernet model is generated. De-

pending on availability of an internal WiFi chip, either this, or the influence of an external WiFi dongle is modeled. The generated models generally show an error of less than 10%. Based on these models, the power consumption of the respective devices can be derived using system monitoring values only. Thus, the power consumption of large, distributed deployments can be monitored without requiring external measurement equipment. Alternatively, these models may be used to optimize traffic and workload distribution, e.g. in a Cloud computing scenario. Finally, the computational efficiency of the measured SBCs is analyzed and compared to x86/64-based systems as published by Koomey et al. [KBS+11], confirming the general trend of doubling computational efficiency approximately every 18 months. Still, the observed relative computational efficiency of the analyzed ARM-based SBCs is lower than on x86/64 systems. Nonetheless, the absolute energy consumption is lower, enabling use-cases not possible with x86/64-based processors.

#### 4.2 POWER CONSUMPTION OF WIRED NETWORK INFRASTRUCTURE

The power consumption in wired networks is analyzed in the following on the example access and aggregation switches. In particular, the extended functionality provided by SDN is in the focus of the following energy considerations. As this is a relatively new area, no related work is available regarding the power consumption of OpenFlow switches. Still, considerable theoretical work is available concerning the optimization of the energy efficiency of SDN-based infrastructure networks [VNS+11; PFC+12; VM16].

Exemplary, two different SDN switches are selected, representing the device classes of hardware and software switches. These switches operate on both OSI layer 2 and 3, thus may be considered as routers or switches in conventional network topologies. Analyzing their power consumption, in particular with respect to advanced operating modes like rewriting packets on the fly, opens a new perspective into networking in the backbone. Differences in performance and associated cost between hardware and software implementations of the same functions allow drawing conclusions on which operations should be executed by which hardware. Thus, network design decisions may be based on energy and performance characteristics of the most suitable devices for the required functionality.

The following section presents an OpenFlow-based performance and power measurement framework. The collected results show that large differences between the performance and power consumption of hardware switches and Open vSwitches exist. Naturally, not the full range of available devices can be measured. Hence, this section describes the derived measurement setup and design decisions in Section 4.2.1, presents the gathered measurements for the two selected devices in Section 4.2.2, and details the modeling procedure and generated models in Section 4.2.3. These are verified in Section 4.2.4 using independent measurements. Finally, Section 4.2.5 summarizes the results, derives conclusions, and gives an outlook on promising future research topics.

The results presented in this section are also published in [KMH14b; KMH14a], and are based on the Master's thesis [Mel14] by Sergej Melnikowitsch.

#### 4.2.1 Measurement Setup

Networking hardware is conventionally selected based on throughput performance of the individual ports. After appropriate hardware is chosen, the power consumption is checked and verified to suit the desired network deployment. In this setup, often only the maximum power consumption is of interest. Considering the growing concern of conserving energy, and at the same time reducing OPEX, the power consumption of individual devices during use is of growing interest. In particular in dense deployments, where air conditioning is required, surplus power consumption of any equipment causes direct costs for the consumed electricity, and secondary costs for cooling the equipment. Thus, knowledge of the actual energy consumption of any networking equipment during use is of growing interest. Other interesting aspects are power saving modes which are increasingly common in modern IT equipment.

The power consumption of OpenFlow switches depends on two different classes of operations, namely the processing and forwarding of productive traffic and the processing of configuration messages. For this, the device must be connected to a network, allowing the measurement environment to configure the device state and stress it with configurable traffic patterns. Then, the power consumption of the device must be measured for different operating states with sufficient accuracy in both time and power. The resulting measurement environment is depicted in Figure 4.8. This setup is similar to the one presented in [MSB+09], but adds the additional dimension of OpenFlow traffic control to the parameter space.

The measurement environment is structured as follows. The *SyncMaster* is the abstract test coordination entity. It runs the configured tests in a given or random order. The *SyncMaster* in its function as OpenFlow controller configures the DUT as required by the test and periodically requests traffic counters. Further, the *SyncMaster* configures traffic generation and power measurements on the *SyncClient*. After execution of a specific test, the *SyncMaster* collects the power measurements from the *SyncClient*.

The measurement setup reflects the requirements of measuring different operating modes of the devices. The tests are chosen to resemble different use cases and load scenarios. Traffic is generated using a traffic generator. This sends a stream of packets with the desired characteristics to the DUT, where it is processed according to the test requirements. The DUT sends the processed packets to a traffic receiver connected to a configured port. Both traffic generator and DUT are controlled by the *SyncMaster*. Besides being an SDN controller, it controls traffic generation at the traffic generator.

The power measurements are controlled by the *SyncMaster*. The accuracy of the measurements is maximized by running the power measurements on the same machine as

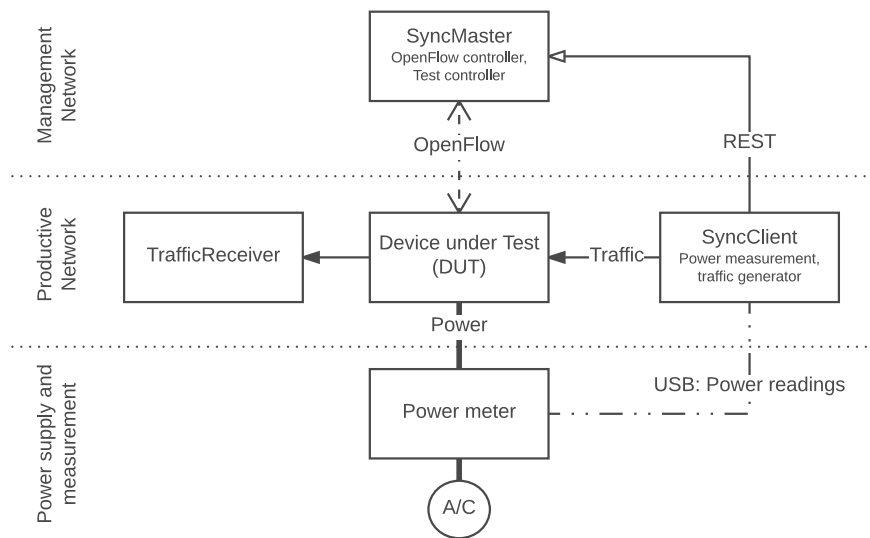


Figure 4.8: The measurement setup and wiring

the *traffic generator*. Thereby, the problem of clock synchronization between different machines is eliminated. Hence, the accuracy of the collected measurements is not affected by absolute time differences or clock drift. In the following, the combination of traffic generator and power measurements will be referred to as *SyncClient*.

For testing the full set of possible configurations, a measurement framework was developed. This is implemented for OpenFlow version 1.0 [PLH+11], but was designed with extensibility in mind, such making it simple to add additional tests. Considering the quick evolution of the OpenFlow standard, extensibility of the framework is a high priority.

Considering the physical dependencies between different tests, these can be structured in a tree. For example to determine the power consumption of a matching rule, the switch must be switched on and its power consumption known. Then, the port must be active, with a specific configured data rate. When a packet is received, the power consumption of the device in this specific configuration can be measured. Thus, the power consumption of a single test can be calculated by subtracting the cost of the dependent tests from the overall power consumption. The identified dependencies of these test cases are depicted in Figure 4.9. Beginning from the root of this tree, the power consumption of the components can be derived in a recursive manner by adding another test case and subtracting the power consumption of the preceding stage.

This approach requires individual tests to be independent of each other. In the case of hardware components, this is intuitive. On the software side, synergies between tests are identified, which is addressed in Section 4.2.3.

Knowing the power consumption of the individual tests, the overall power draw of the DUT in operation can be estimated by adding the cost of the respective configuration and

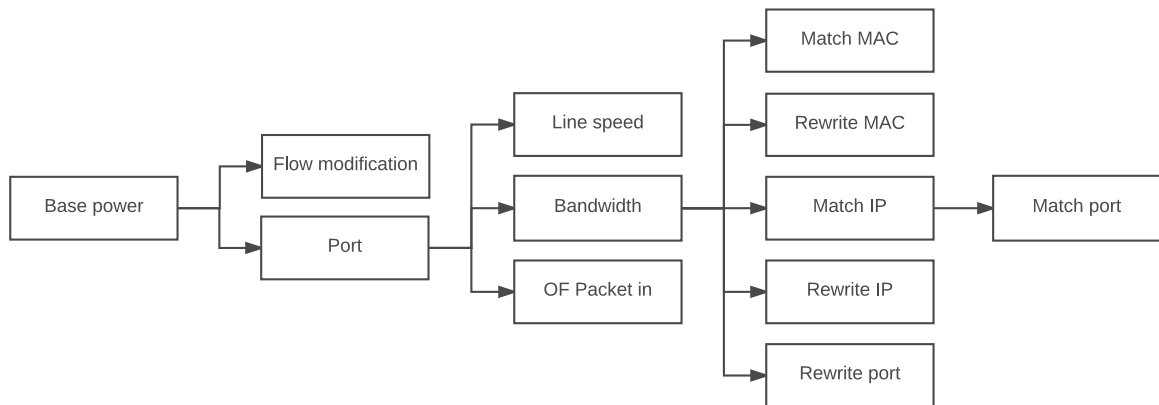


Figure 4.9: Dependencies between measurement tasks

utilization. In case multiple ports are active, the corresponding cost is simply added for the components.

The power consumption of the DUTs is measured using an external power meter. Requirements for AC measurements are a sufficiently high resolution to identify expected small changes in power consumption caused by load differences and a sufficiently high time resolution to identify short peaks. Further, also the measurement of distorted waveforms and phase shifts as may be expected in server rooms should be considered. Hence, a true RMS meter is required.

The Hameg HM8115-2 power meter is chosen for the energy consumption measurements. Available measurements ranges are 0 W to 80 W or 0 W to 800 W. The measurement resolution is 10 mW and 100 mW respectively. The error of the measurement is 0.8 % + 10 digits. With a nominal power consumption of 264 W for the hardware switch, the maximum error of the measurements is below 3 W. Still, the measured power consumption while active is around 120 W, leading to an error of 2 W. Similar values apply to the Open vSwitch with a base power consumption of 49 W, resulting in an error of 1.4 W in the 80 W range.

The maximum available sampling rate is 2 Hz. This is determined by polling the universal serial bus (USB) interface of the power meter as frequently as possible. As true RMS measurements are conducted, a number of AC cycles are required to guarantee accuracy of the result. This is already done within the power meter. Hence, the readings as read from the power meter are reasonably accurate and also include transient effects.

The accuracy of the final model is further improved by running individual tasks over considerable time, thus increasing the number of available samples for later post-processing. By aggregating these, the mean consumed power is calculated by averaging over the test duration, while also the confidence interval for each test can be determined.

The traffic processed by the switches is generated on the *traffic generator* as specified in the test requirements. Traffic monitoring as run on the DUT is used to confirm the config-



ured rate. For this, the OpenFlow counters are used, being updated by the switch OS once per second. This is also the minimal update interval. The OpenFlow controller polls these in the same interval and stores the retrieved values with the corresponding timestamp. The traffic metrics from the switch are stored on the *SyncMaster* and combined with the power measurements as collected by the *SyncClient* in the post-processing step. The accuracy is maximized by keeping the device in a stable state for 60 s, from which the inner 50 s are used in the analysis. Thereby, transient effects while starting or shortly before the end of the transmission are eliminated from the samples used in the evaluation. Each test is run several times at different times of the day and in a different order, thus eliminating influences of changing temperature, possible irregularities in the power supply, and changes of residual heat in the DUT caused by previous tests.

The selected hardware switch is a 48-port Gbit switch with 4x 10 Gbps uplink ports of a major switch vendor. Besides conventional switching the selected model was configured to run the vendor provided OpenFlow capable firmware. The software switch runs on a rack server (Dell PowerEdge R320, Intel Xeon E5-1410, 8GB DDR3 RAM, Intel I350T4 4x Gbit NIC). The operating system is Ubuntu 12.04 LTS, on which Open vSwitch version 1.10.2 is installed.

The architecture of the measurement environment is detailed in Section A.3. There, also the components added to the OpenFlow controller are described. Furthermore, details on the load generator and power measurement are given.

#### 4.2.2 Measurements

According to the requirements as derived in Section 4.2.1, the measurement setup was connected in a server room. The devices were deployed as depicted in Figure 4.8 and the software installed on the respective devices.

The dependencies between the individual tests are shown in Figure 4.9. These are derived from physical requirements (e.g. port must be active to measure throughput), and dependencies of OpenFlow matches as defined in the OpenFlow standard 1.1 [PLH+11]. These tests are run for each DUT multiple times in varying order, thus assuring a consistent outcome. For each test, the measured throughput and power consumption are recorded. In a post-processing step, the dependencies between these are resolved. Thus, the power consumption caused by the respective setting is calculated based on the power differences between this and the superior test.

The tests as described in the previous section are mostly automated. Still, measuring the differences in power consumption with changing number of active ports requires manual reconfiguration of the DUT. This is done by connecting one port after another. For each configuration a measurement is run and the measured values recorded together with the configured setting. This measurement is repeated for all available line rates (e.g. 10/100/1000 Mbps), as differences in the power consumption depending on link configu-

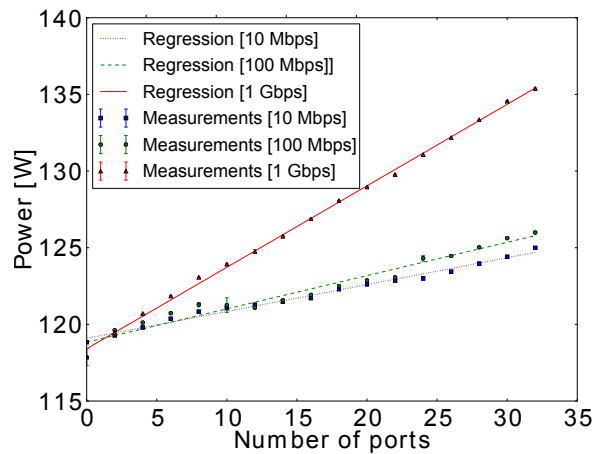


Figure 4.10: Comparison of the power consumption of the hardware switch for different port configurations (from [Mel14])

rations have also been observed in literature [GCN05]. The measurements of the hardware switch are depicted in Figure 4.10. The power consumption increases from a base power of 118 W to 135 W with 32 active ports when setting the line rate to 1 Gbps. The 95% confidence intervals are barely visible around the markers. The regression also shows a good fit ( $R^2 = 0.97$ ). Thus, a cost of 530 mW per active Gbit port is derived. Reducing the line rate of these ports also reduces the power consumption. The cost per 100 Mbps port is 336 mW, while reducing the line rate to 10 Mbps shows a cost of only 199 mW per port. This relates to a relative reduction of energy consumption of 55% and 62% respectively. Due to management issues and the limited number of ports no equivalent measurement was possible on the Open vSwitch.

The automated tests stress the forwarding performance of the DUT under various load conditions and also measure the energy impact of OpenFlow management messages. In case an OpenFlow switch does not have a rule to process a packet received on the data plane, the packet is forwarded to the controller to request a suitable rule. The influence of *packet in* messages on the power consumption is analyzed by sending packets for which no rule is installed to the DUT on one of the data plane ports.

The recorded power consumption of the *software switch* is shown in Figure 4.11a. The power consumption increases from a base power consumption of 49 W to 70 W while sending packets for which no rule is installed on the switch, causing it to create and send *packet in* messages to the controller. During the measurement interval, 1.7 M *packet in* messages were generated by the switch. This relates to a cost of 800  $\mu$ J per packet. The influence of incoming flow modification messages, used to pro-actively configure an OpenFlow switch, is evaluated by sending these with maximum rate over the duration of 60 s while monitoring the power consumption. The plot showing an idle time and the power consumption while flow modification messages are received by the Open vSwitch

is given in Figure 4.11b. The figure shows an increase to 66 W while receiving 727 k flow modification messages within the test interval. This relates to a cost of 1.445 mJ per *flow mod* message.

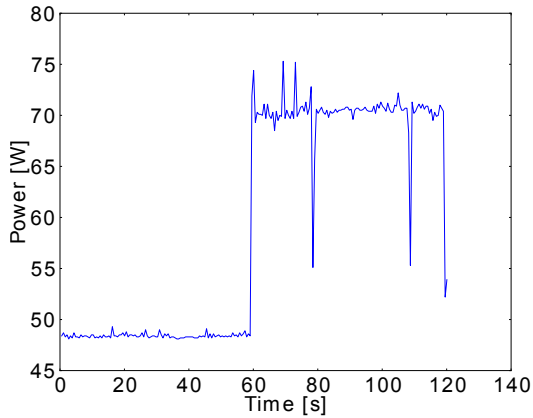
The cost per management message on the *hardware switch* is calculated by subtracting the idle power of the DUT from the active period. During these 60 s 720 k rules were inserted. This was only possible by periodically flushing the ternary content addressable memory (TCAM). As this relates to the normal usage of the switch, this is included in the measurements. While inserting flows, the power consumption of the hardware switch increases by 300 mW. The calculated cost for *flow modification* messages thus is 20.25  $\mu$ J per flow. Similarly, the hardware switch processes 17 k *packet in* messages during the 60 s measurement interval, increasing the power consumption by 200 mW. This relates to a cost of 771  $\mu$ J per *flow mod* message sent.

The measured values are comparable between both DUTs. The energy cost of the hardware switch, in particular when processing *flow modification* messages is lower than on the software switch. Still, comparing the cost of *flow modification* messages, the cost is by a factor of 70 higher on the software switch. The generated results are summarized in Table 4.5

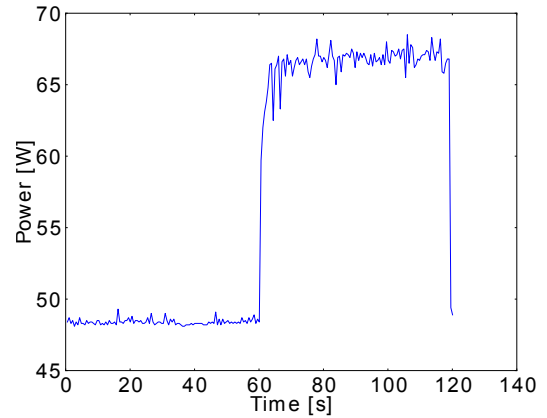
OpenFlow extends the functionality of conventional switches by intelligently changing behavior depending on different header fields. Therefore, the header of incoming packets is matched using a TCAM from which the desired action for the respective packet is derived. Here, the configured rules as described in the last section are applied. As described in Section 2.2.2, an OpenFlow rule consists of *Match Fields*, *Counters*, and *Actions*. In the following, the influence of different *matches* and *instructions* on the power consumption of both DUTs is analyzed.

The influence of matches and actions is identified by configuring the rules on the switch to match a single field or execute a single action. The traffic send through the device is created to be matched by the active rule. Hence, the influence of the individual actions on the overall power consumption of the DUT can be derived. For this, the packet rate is step-wise increased until the maximum supported rate by the DUT is reached. Thus, the influence of the respective rule or match on the power consumption is derived, and the cost per match or action is calculated.

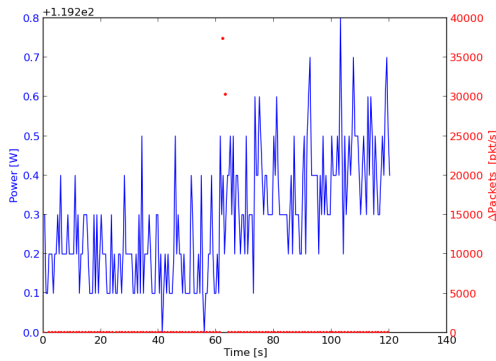
Measurements show, that on both DUTs the cost of OpenFlow matches and actions is comparatively low. Hence, the influence of matching the minimum and maximum number of fields and applying the minimum and maximum set of actions to the packet are compared. The minimum set of matches and actions applied consists of matching the input port and sending the packet to a configured output port. Compared to a conventional switch, this simplifies operations, as there the MAC address must be checked to determine the correct output port. The maximum set of matches and actions consists of matching the full header of the packet, rewriting all fields, and sending the packet out at a specified port. Figure 4.12 shows the results for the Open vSwitch. While the number of



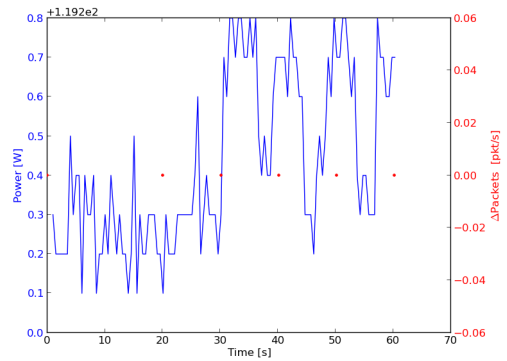
(a) Open vSwitch power consumption while processing management traffic



(b) Open vSwitch power consumption while receiving flow modification messages



(c) Hardware switch power consumption while processing management traffic



(d) Hardware switch power consumption while receiving flow modification messages

Figure 4.11: Comparison of the power consumption of the hardware switch and Open vSwitch under different management traffic load (from [Mel14])

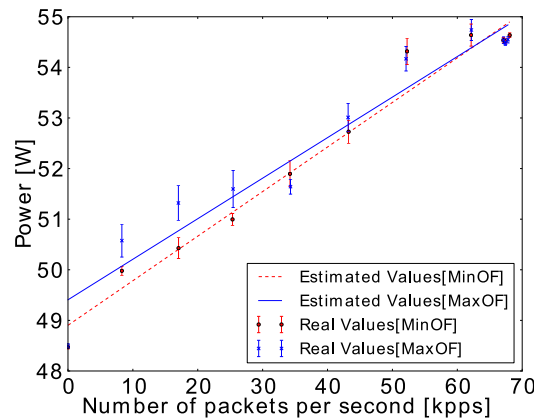


Figure 4.12: Open vSwitch power consumption for the minimum and maximum number of active OpenFlow features (from [Mel14])

packets processed per second is increased to approximately 70 000, the power consumption increases to 54 W. The difference between minimum and maximum set of active rules and matches is significant for low data rates, but differences decrease for higher data rates. The derived power consumption values of the individual matches and actions as derived using this procedure are listed in Table 4.5.

The same tests are also run on the hardware switch. Also there, the difference between the minimum and maximum set of OpenFlow matches and actions is marginal. One major difference though is the maximum packet rate observed for different settings. Differences in maximum throughput are in the range 300 pps to 70 000 pps. The large difference in throughput is explained by some operations being hardware supported, while others are processed by the CPU of the switch. The former can be processed on line rate, while the latter require several CPU cycles and must be transferred between CPU and ASIC using a non-optimized path. Figure 4.13a shows the energy impact of the hardware accelerated operations, while Figure 4.13b compares the energy consumption of the software supported OpenFlow features. Both figures show an only marginal increase in overall power consumption when OpenFlow features are activated. The increase in both cases is in the range of 100 mW to 200 mW. The cost of the individual matches and actions as derived from the individual measurements are detailed in Table 4.5.

As the above tests cannot identify the influence of individual matches and flows, the influence of these OpenFlow features is determined by running tests changing only one feature. Thus, their power consumption can be identified when compared to the setting with the set of minimum OpenFlow rules and actions being active. Each of these tests is run on both DUTs for 60 s. The results are recorded and later correlated with the measured network throughput performance. The calculated cost of each of these features is listed in Table 4.5.

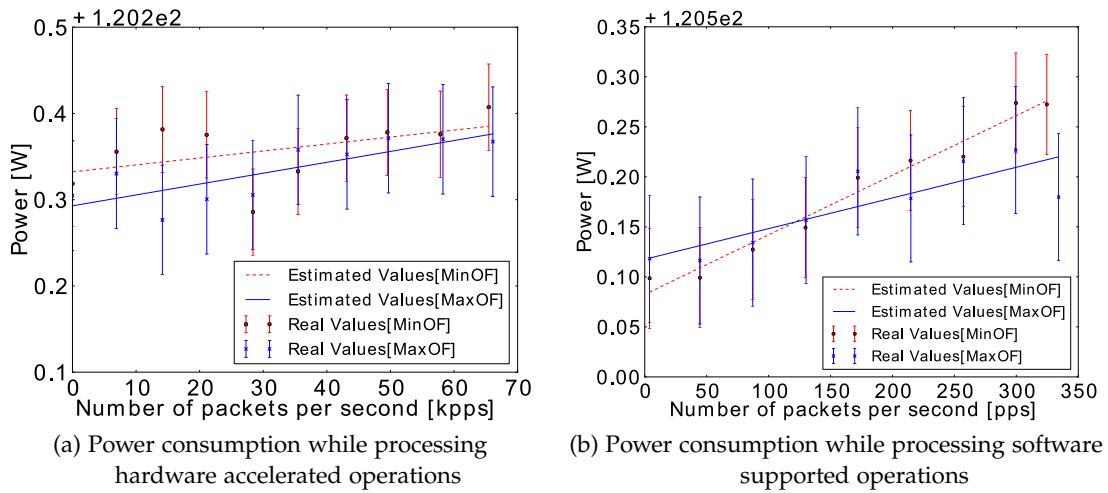


Figure 4.13: Comparison of the power consumption of the hardware switch while processing data plane traffic (from [Mel14])

Comparing the power consumption of the Open vSwitch and the hardware switch, a few observations are made. First, the idle consumption of the hardware switch is considerably higher compared to the software switch. Still, also the number of available ports is higher. Processing traffic has a negligible effect on the overall power consumption of the hardware switch, while the power consumption of the software switch increases by 10%. The power consumption when applying the minimum or maximum set of OpenFlow operations is comparable for both DUTs. Still, features not implemented in hardware heavily decrease throughput of the measured hardware switch. Operations implemented in software can also be spotted comparing the power consumption of these in Table 4.5, where their cost is 1000-fold higher compared to hardware features. Summarizing, it can be said that the energy efficiency of the hardware switch for hardware accelerated features is higher, in particular when a large number of devices is connected. Contrary, the Open vSwitch is more versatile, has predictable performance and power consumption, and is more energy efficient, when requiring only on a small number of links.

#### 4.2.3 Power Consumption Model for OpenFlow Switches

From the measured cost of the individual components and their configuration as given in Table 4.5, the power consumption of the DUTs for different workloads can be calculated. Therefore, an additive power model for both devices is proposed. Due to dependencies between matches and actions as derived from reference measurements applying the additive power model, alternative, simpler power models are proposed. These are based on the minimum and maximum cost of matches and actions being active in the selected

Table 4.5: Power and energy parameters of the switches and the corresponding OpenFlow features (from [Mel14])

	Hardware Switch	Open vSwitch
Base Power [W]		
Base	118.3300	48.7397
Port Configuration Power [W/port]		
Port	0.5295	n/a
Port Configuration Power Factor		
Line Speed [1Gbit]	1	1
Line Speed [100Mbit]	0.4455	n/a
Line Speed [10Mbit]	0.3761	n/a
Management Energy [ $\mu$ Ws/packet]		
Packet-In	711.3028	775.5246
Flow-Modification	20.2512	1,445.1309
Energy for Matching [ $\mu$ Ws/packet]		
Ingress Port	1.1013	47.1955
MAC Src	0.4234	51.4126
MAC Dst	0.7286	50.1219
IP Src	0.3982	50.4271
IP Dst	0.4390	48.3222
L4 Port Src	0.6725	0
L4 Port Dst	974.1202	0.5716
Energy for Actions [ $\mu$ Ws/packet]		
Output	0.4286	49.1954
MAC Src	0	0
MAC Dst	0.5672	0
IP Src	728.2008	0.5119
IP Dst	866.8180	0.8716
L4 Port Src	977.2340	5.1381
L4 Port Dst	846.8505	0.9332

rule set. In the following, the different power models are introduced and their accuracy is discussed.

The additive power model directly following from the test dependencies as shown in Figure 4.9 is described in the following. Caused by the dependencies, also the power consumption of the DUT is modeled by adding the power consumption of the individual tests.

$$P_{\text{DUT}} = P_{\text{base}} + P_{\text{config}} + P_{\text{control}} + P_{\text{OF}}. \quad (4.12)$$

Here,  $P_{\text{base}}$  is the idle power consumption of the DUT.  $P_{\text{config}}$  is the power consumption caused by the static device configuration. This includes the configuration of active ports as well as their configured line speeds.  $P_{\text{control}}$  describes the power consumption caused by OpenFlow management traffic. Hence, it includes *flow modification* and *packet in* events. Finally,  $P_{\text{OF}}$  adds the power consumption caused by productive traffic on the data plane to the overall power consumption. This includes the active flows with their respective load as well as their configuration.

The base or idle power consumption is defined as  $P_{\text{idle}}$  and measured directly for each device. This depends on the device type and installed software versions. Due to the immense parameter space, no further measurements varying these parameters are conducted. Hence, the parameters as given in Table 4.5 are directly used.

The energy cost of the switch configuration as caused by active ports and configured line speeds is defined as

$$P_{\text{config}} = \sum_{i=1}^{N_{\text{ports}}} s_i \cdot P_{\text{port}} \quad (4.13)$$

Hence, the configuration power  $P_{\text{config}}$  relates to the sum of the cost of all ports of the DUT, where  $N_{\text{ports}}$  is the number of available ports. For each port, the base power consumption of each port when active with highest line rate  $P_{\text{port}}$  is multiplied with the power factor  $s_i$  of the currently configured link rate, where  $s_i$  is in the range 0 to 1 as given in Table 4.5.

The power consumption caused by the control traffic is defined as:

$$P_{\text{control}} = r_{\text{packetIn}} \cdot E_{\text{packetIn}} + r_{\text{flowMod}} \cdot E_{\text{flowMod}} \quad (4.14)$$

Here,  $r_{\text{packetIn}}$  is the rate of *packet in* messages generated by the switch caused by missing rules on the local devices. As derived in Section 4.2.2, an energy cost is associated to each of these events. The energy required for sending this packet is defined as  $E_{\text{packetIn}}$ . Similarly,  $r_{\text{flowMod}}$  gives the rate of received *flow modification* messages with its associated energy cost  $E_{\text{flowMod}}$ . Thus, the energy consumption of the control traffic is modeled in an additive way, based on the energy consumption of the individual commands and their rates.

The power consumption of the productive traffic on the data plane is modeled in an additive manner according the test dependencies as given in Figure 4.9. Consequently, the



power consumption  $P_{\text{OF}}$  is modeled as sum of the number of flows with their respective traffic rates, active matches and actions.

$$P_{\text{OF}}(t) = \sum_{i=1}^{N_{\text{flows}}} r_{\text{packets}}(i, t) \left[ \sum_j^{N_{\text{matches}}} \mu_{\text{match}}(i, j) \cdot e_{\text{match}}(j) + \sum_k^{N_{\text{actions}}} \mu_{\text{action}}(i, k) \cdot e_{\text{action}}(k) \right] \quad (4.15)$$

Here,  $N_{\text{flows}}$  is the number of active flows. Each flow  $i$  has its specific packet rate  $r_{\text{packets}}(i, t)$ , which depends on the time. For each flow, the active matches and actions are defined as the sum over the energy cost of each match  $e_{\text{match}}(j)$  or action  $e_{\text{action}}(k)$  multiplied by  $\mu_{\text{match}}(i, j)$  and  $\mu_{\text{action}}(i, j)$  respectively, indicating whether it was active or not. These are added for each flow  $i$  over the number of possible matches  $j$  and actions  $k$ . The split between both is made for clarification. When applying the model, both sets can be appended and summed using one iterator. The size of the set of matches  $N_{\text{matches}}$  is limited by the number of header fields supported by the used OpenFlow standard. Contrary, the set of actions  $N_{\text{actions}}$  may differ in size, as multiple actions may be defined in one rule. An example is sending a packet out on multiple ports, possibly also with changed header fields.

For comparison, two simplified power consumption models are proposed in the following. As the influence of the managed traffic is small, in particular on the hardware switch, a power model is proposed modeling the power consumption only based on idle power, influence of the mostly static configuration, and control traffic. Further, a power model extending this minimal power model by a traffic model, where only the cost of the most energy-costly match and action are considered.

The minimal power model proposed adds the base power consumption of the DUT  $P_{\text{base}}$ , configuration  $P_{\text{config}}$ , and control traffic  $P_{\text{control}}$

$$P_{\text{switch,HW}} = P_{\text{base}} + P_{\text{config}} + P_{\text{control}}. \quad (4.16)$$

As this model not includes any time or load dependent terms, it is expected to perform worst in the following evaluations, and hence is used as baseline for comparison.

The intermediate model eliminates the inner sums from Equation 4.15, replacing each term with the maximum cost of matches and actions applied to the respective flow. Hence, the resulting model for the cost of the OpenFlow traffic is

$$P_{\text{OF,SW}} = \sum_i^{N_{\text{flows}}} r_{\text{packets}}(i) \cdot \left( \max_j [\mu_{\text{match}}(i, j) \cdot e_{\text{match}}(j)] + \max_k [\mu_{\text{action}}(i, k) \cdot e_{\text{action}}(k)] \right). \quad (4.17)$$

This simplifies computation on the controller side, but is expected to perform worse than the detailed model, but better than the static model.

#### 4.2.4 Evaluation of the Derived Power Models

The power models as proposed in Section 4.2.3 are assessed by analyzing the differences between measurements on the hardware and estimations using the proposed models using arbitrary traffic sent through the DUT. This traffic is generated by different traffic generators on different ports of the DUT, thus putting the model on a test, where not only the setting used to generate the model, but a more general setting is evaluated. Thus, the practical applicability of these models can be verified, and upper bounds for the errors under the given conditions be derived.

The reference measurements and error estimation are implemented by extending the OpenFlow controller to provide real-time estimates of the power consumption of the connected devices based on device type and selected power model. Simultaneously, the measured power consumption of the DUT is recorded and periodically requested from the *SyncClient*. Thus, packet rate, power measurement, estimation, and resulting error can be plotted in real-time. Due to technical reasons, the estimates are delayed by one second, as OpenFlow counters are only updated in this interval.

These additional components required for the model based power estimation are already included in Figure A.3. Hence, the derived OpenFlow framework can both be used to measure power models of connected DUTs when connected to a power meter, as well as used independently, estimating the power consumption of connected devices in case power models for the connected device are available. Furthermore, it is possible to include only the packages required for deployment, thereby reducing the complexity of the setup. Then, only the extended OpenFlow controller is required, eliminating the need of a *SyncClient*, a traffic generator, and power measurements. Additionally, the modules extending the OpenFlow controller for running the measurements and deriving the power models are not required. This leaves only the *EnergyMonitor* as additional Floodlight plugin, calculating the power consumption based on the monitored traffic.

The observed packet throughput, measured power consumption, estimated power consumption, and the resulting error are given in each of the following figures. The first line shows the observed packet rate in packets per second as returned by the OpenFlow counters. The second row shows the power consumption of the DUT as measured by the external power meter. This is given in  $W$ . The third row shows the estimated power consumption as derived using the respective power model as indicated in the caption of the figure in  $W$ . The last row calculates the difference between the measured and estimated power consumption. From these, general observations on the accuracy of the generated power models are drawn. The results are summarized in Table 4.6.

Figure 4.14 shows the results when applying the maximum power model to the measurements as taken on the Open vSwitch. In both scenarios the traffic rate is sequentially increased, while the metrics as defined before are recorded. Figure 4.14a shows the results while the minimum number of matches and actions are active. The traffic rate is increased

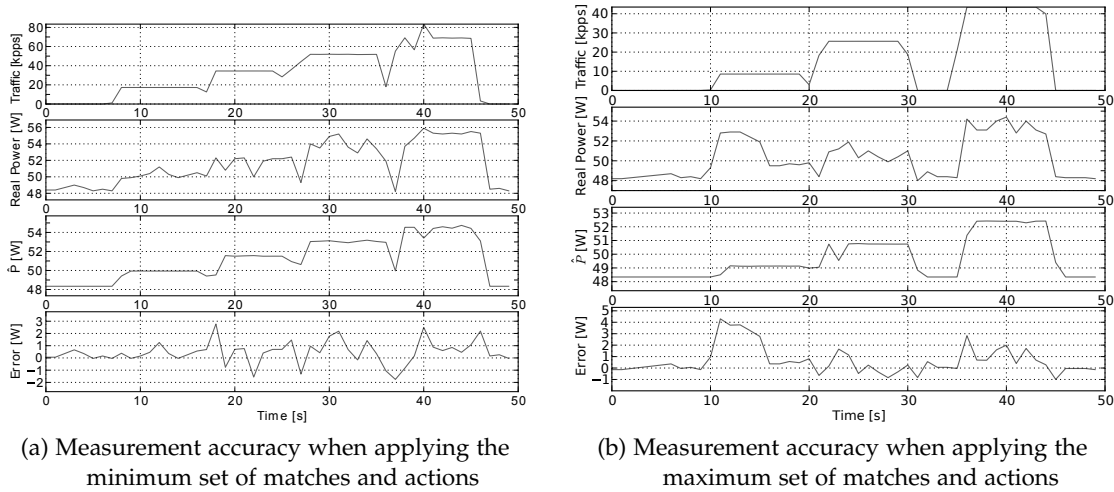


Figure 4.14: Power consumption of the Open vSwitch under varying load using the maximum model (from [Mel14])

in steps of 20 kpps to a maximum rate of 80 kpps. The second row shows the measured power consumption of the DUT increasing with increasing load. The power estimate in the third column shows as similar behavior. Both top out at a maximum of around 55 W. The resulting error as given in the last row stays in the range  $-2\text{ W}$  to  $3\text{ W}$ . A similar behavior can be observed in Figure 4.14b. Here, the measured rates are 10 kpps, 30 kpps, and 50 kpps. An interesting observation is that the measured power consumption as is given in the second row shows an increased power draw between seconds 12 and 14. Clearly, this is not visible in the power model, as the configuration was not modified and the traffic did not change, resulting in an increased error in the last row of the plot. This effect is expected to be caused by some system maintenance scripts (e.g. cron jobs) running on the Open vSwitch. Care was taken to eliminate these, but not all system background activity can be stopped while running a productive system. This unaccounted peak in power consumption is thus also visible in the resulting error. Disregarding this anomaly, the resulting error is in a similar range as before.

The anomaly visible in Figure 4.14b also clearly shows the limitations of the model based power estimation. Any effects not included in the model cause errors in the final estimation. In case these values are used for accounting, unexplained errors may remain. Here, a trade-off between model complexity, usability, and accuracy must be found. Caused by the requirements as defined before, only the OpenFlow protocol is available for monitoring the DUTs. Hence, no system utilization values from CPU, RAM, and mass storage of the underlying OS are available. Instead, the proposed modeling technique is applicable to any kind of OpenFlow capable devices, from virtual switches to highly integrated hardware switches due to the independence of low-level system monitoring.

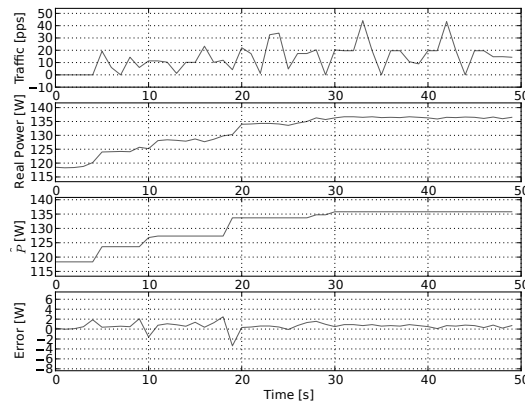


Figure 4.15: Power consumption of the hardware switch for varying number of active ports (from [Mel14])

Figure 4.15 shows the accuracy of the generated model for the hardware switch when incrementally connecting and configuring multiple ports. The data rate, as is expected from the chosen test setup, is almost zero. The observed packets are maintenance traffic like ARP request, replies, and similar. The second row shows an increasing power draw when connecting and configuring the ports. A similar behavior is visible in the third row for the estimated power. The derived error, as seen in the last row is in the range  $-3\text{ W}$  to  $2\text{ W}$ , and as such in the range of the measurement uncertainty.

Figure 4.16 compares the maximum and the minimum model for the hardware switch. Both show a similar traffic behavior, generated by the traffic generators connected to different ports of the DUT. Thus, a traffic rate of  $1.2\text{ Mpps}$  is achieved. Figure 4.16a shows the measured power consumption in the second row, while the power estimate is given in the third. The resulting error while idle is quite small, but increases to  $-3\text{ W}$  while under load. This result is already better than for the general model, where errors of up to  $8\text{ W}$  are observed (cf. [Mel14]). Figure 4.16b shows the same test, but here the static power model is applied, eliminating the influence of the observed traffic on the power consumption. The resulting error is smaller compared to the maximum model, but also increasing when traffic is observed. Still, the error remains in the range  $-0.5\text{ W}$  to  $0.5\text{ W}$ .

The above observations, plus additional measurements shown in [Mel14] are summarized in Table 4.6 for the two DUTs. For the hardware switch, the general, maximum, and static power models are evaluated, while for the Open vSwitch the general and maximum model are analyzed. Here, the static model is omitted, as the measurements in Section 4.2.3 clearly show a dependency between traffic and power consumption.  $P_{e,\max}[\text{W}]$  is the maximum error observed during the measurements. This is normalized to the maximum power consumption of the DUT,  $P_{\max}$ , as measured by the power meter, resulting in the maximum relative error  $P_{e,\text{rel},\max}$ .

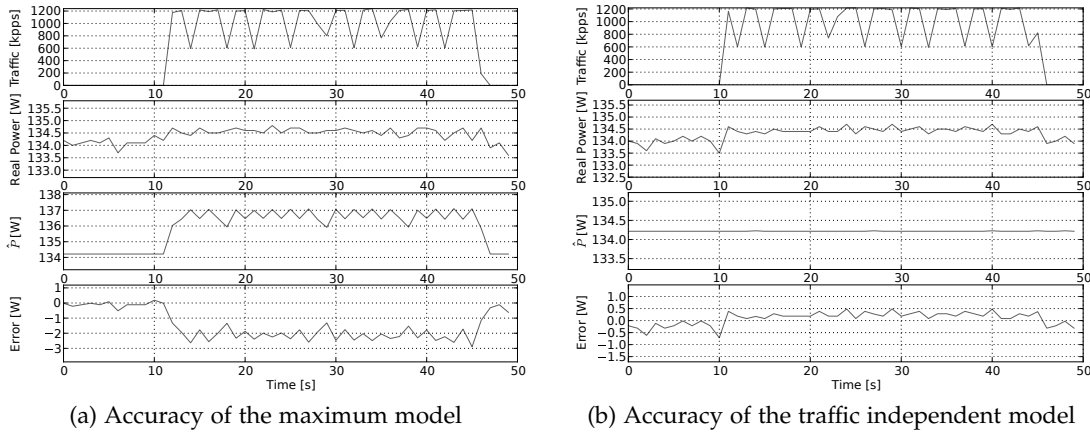


Figure 4.16: Comparison of accuracy of different power models for the hardware switch in the reference scenario using multiple traffic generators (from [Mel14])

Table 4.6: Error of the derived power models

Switch Model	Hardware switch			Open vSwitch	
	general	maximum	static	general	maximum
$P_{e,max}$ [W]	7.56	3	0.72	9	4.30
$P_{e,rel,max}$	5.6%	2.2%	0.5%	16.1%	7.7%

The errors for different power models of the hardware switch are generally low. This is mainly caused by the high idle consumption of the device, and the low variability under load. Still, differences between the models are visible. The worst performance is achieved by the general model with an error of 8 W in case of the hardware switch and 9 W for the software switch. This error can be reduced to less than 5 W for the Open vSwitch. On the hardware switch, the error can be further reduced by eliminating the influence of traffic on the power consumption to below 1 W, resulting in a static power model.

The relative error for the hardware switch is below 6% for the general model, reduced to 2.2% using the maximum model, and minimized to 0.5% with the static power model. The relative error of the Open vSwitch is comparatively high, which is caused by the lower idle consumption.

When applying the model, and the overall energy cost is required, the static model for hardware switches, and the maximum model for the Open vSwitch should be chosen. The expected errors are below 3 W, which is close to the accuracy of the power measurements. Still, in the case of the static power model of the hardware switch, the estimated power is generally too low while traffic is processed by the DUT. The errors of the software switch should cancel out, as both positive and negative errors are observed. If a higher

accuracy of the resulting models is required, additional measurements should be run with increased test durations. Thus, the noise on the measurements can be reduced, while increasing the accuracy of the derived mean. Still, this leaves systematic errors, which may be addressed by using measurement equipment with higher accuracy.

#### 4.2.5 *Summary and Conclusions*

The power models presented in this section are one of the building blocks to determine the power consumption of future communication networks. In particular the framework proposed within this section proves useful in this respect, as the power consumption of OpenFlow capable hardware can be determined in a highly automated manner. Still, changes of the hardware configuration like connected ports require manual intervention. One of the goals of the development of the measurement framework was simple extensibility. Thus, the presented OpenFlow power measurement framework is easily expandable to OpenFlow versions higher than 1.0.

The described power models serve as proof of concept of the measurement framework. Further, these can already be used to derive the power consumption of specific networks. For example in a large deployment, the derived power models may be used to derive the instantaneous power consumption of the full network.

Analyzing the differences between the models, a few general observations can be made. Simple, hardware-supported operations are best run on a hardware switch, as their power consumption increases only marginally under load. More complex operations, like packet duplication or header rewrites are better handled by an Open vSwitch. In particular operations requiring multiple changes show negligible additional power consumption compared to simple forwarding on the Open vSwitch. Still, the energy cost of processing a single packet on the Open vSwitch is 100x higher for these operations. Non-hardware accelerated operations on the hardware switch should be avoided at any cost, as these are processed by the internal CPU. This is often barely sufficient to run the OS and manage the built in ASIC of the switch. Further, the bandwidth between ASIC and CPU is a bottleneck. Thus, the packet rate of IP and L4 port rewrites as well as L4 port filtering reduces the processed packet rate to less than 350 pps. This is also reflected in the energy cost, where these operations increase the energy consumption per packet by a factor of 1000.

Applying the above observations to a productive network leads to the conclusion that hardware switches should be used as access switches, funneling the traffic to more central virtual switches handling OpenFlow rules with higher complexity. Examples may be software defined multicast (SDM) [RBH15], where a multicast service requires the in-network duplication of packets. Thus, the incoming packet stream can be processed at the second aggregation layer within Open vSwitches, while the aggregation switches just forward the modified packets.

The generated power models may also be used to minimize the power consumption of communication networks. This may be done in a practical manner, where based on observed traffic patterns the switches are reconfigured. The most promising example in this case is the reconfiguration of line speeds of the hardware switch. There, ports may be switched to lower rates during idle or low load periods. Compared to disabling ports, this allows maintenance functions like periodic heartbeats. Further, a re-configuration of the port is also possible by observing the activity of the remote machine, which would not be possible otherwise.

Alternatively, these power models may be used to optimize full networks based on statistical observations using mathematical models [VNS+11; VM16]. Knowing a large number of power models for various devices also allows the network operator to select the most energy efficient equipment considering the current and expected load of the target networks. This would require a central location or database, where these models can be published and retrieved, thus eliminating the repeated calibration procedure as described within this section.

Possible extensions of the described work include adapting the proposed measurement framework to more recent OpenFlow versions. Currently, only these two exemplary OpenFlow power models are available. Here, a larger number of models should be calibrated and made available to the public, thus increasing the comparability of OpenFlow switches and raising the awareness for energy efficiency in network infrastructure. Thus, purchasing decisions may include, besides cost and performance, also the OPEX in terms of energy consumption.

Sharing the generated power models in a machine readable form in a centralized repository would enable researchers to model the power consumption of complex network environments depending on network configuration and load. Hence, also optimizations of these networks can be simulated before deploying the respective traffic management approaches. Promising examples are disabling of links as is often proposed for conventional networks [CMN12], or reconfiguring link speeds [YWX+13] as is also shown to be promising on OpenFlow switches within this work.

Another interesting aspect for future work is the decoupling of the OpenFlow power model for the Open vSwitch or similar software from the system it is running on. Here, general power models for the underlying hardware may be combined with the performance requirements of different OpenFlow matches, actions, and reconfiguration messages. Thereby, the power consumption of virtual OpenFlow switches is determined in a building block based manner by first mapping the network load to system utilization, which may then be mapped to power consumption using general power models. This simplifies the modeling process by not requiring a calibration of each hardware-software combination.

### 4.3 POWER CONSUMPTION OF END-USER DEVICES

The power consumption of end-user devices is analyzed on the example of smartphones. These are chosen, as they are becoming the most important devices for mobile communication and media consumption for a large number of people [Eri16]. As these are strictly limited in available energy, the energy efficiency must be maximized to satisfy user expectations. Approximately 30 % of the power is consumed by network interface [CDJ+15]. Hence, these are the primary target for optimizations. Still, growing demands for continuous connectivity increase challenges in saving energy.

An interesting, emerging technology is MPTCP [BPB11]. It is a technology simultaneously using multiple network interfaces to optimize network performance and reliability. MPTCP works by modifying the underlying functionality of the TCP socket provided by the Linux kernel. Instead of establishing a single TCP connection to the remote server, multiple connections, called sub-flows, are established on different interfaces. This requires the remote server to also support MPTCP. Using multiple interfaces and connections thus provides higher bandwidth or improves reliability by dynamically handing connections over to another interface while one is being interrupted.

Related work at the time of writing covers the performance related aspect of MPTCP well [PDD+12; CLG+13; DNS+14], or simulates the energy consumption of various scenarios [CYM13b; CLG+13]. The power consumption, in particular on mobile devices, was measured for regular downloads only [LCN+14]. Thus, the power consumption of downloads and HTTP streaming can be evaluated.

An interesting aspect for interactive real-time systems is CBR streaming as is required for video-conferencing or live-cloud gaming. Here, the functionality of sustaining a TCP connection during a vertical handover to another network provides uninterrupted user experience. Hence, in the following the power consumption of MPTCP for CBR streaming is evaluated. The focus of the following section is analyzing the influence of load balancing on the power consumption of smartphones using MPTCP. The following section is based on work published in [KWR+15b] and [KWR+15a], which are based on a thesis by Stefan Rado [Rad14].

#### 4.3.1 *Measurement Setup*

In the following, the power consumption of MPTCP on mobile devices is analyzed. The chosen scenario is constant bit rate streaming. Therefore, a server providing a data stream with configurable and constant bit rate and a smartphone receiving this stream are required. Both need an MPTCP enabled kernel to support the test setup.

The proposed measurements target the influence of changing traffic distribution over the available interfaces considering the overall power consumption of the mobile device. As the scenario is constant bit rate streaming, corresponding traffic shaping needs to be



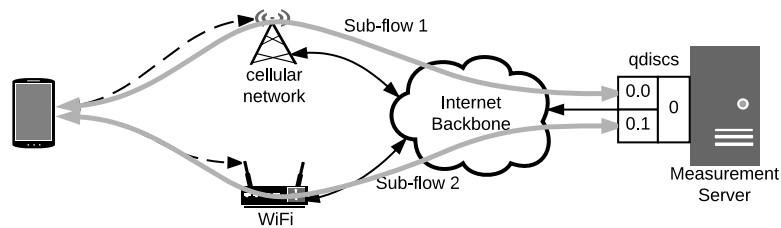


Figure 4.17: Schematic setup of the MPTCP measurements

implemented. Therefore, the server uses the kernel's built in traffic shaping, limiting the bandwidth on the respective flows as required.

The schematic setup is given in Figure 4.17. The mobile device connects to both WiFi and the cellular network. The data connection is then established using MPTCP. Thus, two different sub-flows between mobile device and server are created.

Two devices are selected as mobile end of the measurement. These are the Nexus 5 and the Nexus S. These are chosen because the full OS is available via Google's Android open source project (AOSP) developer program, thus providing free reconfigurability. Hence, a custom OS image for the devices is built. Secondly, patches for the used kernels enabling MPTCP are available. Thirdly, the power consumption of the DUT can be measured internally, as well as externally for the Nexus S using a custom built measurement setup.

Comparability of the implementations is assured by running the same OS on both devices. This is possible, as the Cyanogen project<sup>5</sup> was providing images based on Google's AOSP, also after Google suspended its support of the Nexus S. Therefore, both devices are equipped with CyanogenMod 11 OS. The Nexus S runs version 3.0.4 of the Linux kernel, the Nexus 5 version 3.4.0. These are patched with the latest available MPTCP patch (Nexus S: 0.86.6, Nexus 5: 0.86.7), thus adding the required functionality.

The network technologies to be evaluated are WiFi and the cellular network. The WiFi network used is an IEEE 802.11g compatible network, as was commonly found in home premises at the time of measurement. This is used with both smartphones. The cellular interfaces available on the Nexus S restrict the measurements to high speed downlink packet access (HSDPA) with a gross rate of maximum 7.2 Mbps. On the Nexus 5 LTE is available, supporting data rates of up to 150 Mbps in downlink and 50 Mbps uplink direction (LTE Cat 4).

As one goal of this work is the evaluation of the relative cost of using MPTCP on smartphones, first a reference needs to be established. Therefore, tests determining the power consumption of the network interfaces are run. This is conducted similar to Section 4.1.1 by first determining the idle consumption of the device, then adding additional load, to then derive the component power consumption in a regression based approach. In the

<sup>5</sup> <http://www.cyanogenmod.org/>, abandoned at the time of writing, available via archive.org: <https://web.archive.org/web/20161224194030/https://www.cyanogenmod.org/>

case of the mobile devices this means first identifying the idle consumption without any other components active, then determining the cost of having a single interface active, and finally profiling the respective interface under load. After the influence of the individual components is determined, the power consumption of the combination of interfaces is measured. The resulting power consumption is then compared to the cost of the individual interfaces.

The cost of using the different interfaces on the Nexus S is determined by directly measuring the power consumption using a modified 'charging' cradle. Instead of charging the battery, the cradle is only used to contact the battery. These are connected via a measurement shunt similar to Section 4.1 to a custom built battery dummy, which is inserted into the smartphone. The battery voltage and voltage drop over the measurement shunt are measured as also described in Section 4.1.1. Compared to the SBC measurements, the resulting error on the voltage channel is slightly lower (i.e. 3.66 mV instead of 5.66 mV) because the 5 V range of the power meter can be used. Still, as the main error source is the current channel and the same configuration is used, the resulting error remains at 2.47 %.

The power measurement on the Nexus 5 is more challenging, as no simple battery replacement is possible due to the built-in battery. Still, the Nexus 5 includes a power sensing chip. Its readings are made available via the kernel file system (/sys). The battery monitoring is conducted by the Maxim MAX17048 battery management chip. This provides accurate and reliable estimates of the remaining battery capacity in a granularity of 1/256 % steps with an error well below 5 %<sup>6</sup>. Based on this, the discharge rate is calculated. Both battery voltage and estimated discharge current are available via the files `voltage_now` and `current_now` in the folder `/sys/class/power_supply/battery`.

The monitoring frequency on the power meter is configured to 10 kHz. Tests on the Nexus 5 have shown that contrary to the /proc file system, calls to the /sys area return different values for each call. Hence, it can be concluded, that instantaneous values from the sensor are available. On the Nexus 5, the maximum feasible rate not overloading the system was determined to be 40 Hz, keeping the processor utilization below 20 %. To keep some additional capacity, and avoid possible implications on measured throughput, samples were recorded with a frequency of 20 Hz.

The maximum possible accuracy of the measurements is assured by switching off any non-essential components. This includes also the display. Still, as the tests must be configured and started via the touch screen an additional delay between starting a test and the begin of the measurement are introduced. This allows the system to settle, and thus results in more accurate measurements. The alternative of configuring and initiating measurements via the Android debug bridge (ADB) and USB or WiFi prohibits itself, due to the inherent influence on the measurements. Connecting USB would charge the battery, and thus leave no indication on how much energy is consumed by the device. Keeping

---

<sup>6</sup> <https://datasheets.maximintegrated.com/en/ds/MAX17048-MAX17049.pdf> accessed 2017-01-23

a TCP session open via WiFi also introduces errors, and is completely impossible when measuring the idle consumption, where all network interfaces need to be shut off.

The desired data rates for each test are configured from the mobile device using a command line interface on the server. Therefore, a script is running on the server accepting commands for shaping individual flows of the connecting device. For this, the kernel's queuing disciplines, short *qdiscs* are used. Still, determining the respective flows to control traffic for a given device and interface requires some additional effort. For the connecting IP the MPTCP mapping in the kernel file system is read, thus determining the corresponding sockets. These are then used to configure the *qdiscs* for the underlying TCP flows.

A number of different *qdiscs* are available (e.g. first in, first out (FIFO), round robin (RR), random early discard (RED), etc.). The measurement setup uses the hierarchical token bucket (HTB) method, guaranteeing a maximum stable rate as long as no interruptions are observed. The concept is that a 'bucket' is continually filled with tokens, which are taken out according to the size of the transmitted data. Only if no data was sent for some time, the rate may shortly be higher depending on the bucket size. Still, as TCP is used, its slow-start mechanism should prevent exceeding the configured rate.

On the server a website is available providing either traffic streams with configurable bandwidth or an unlimited data stream containing random data. As for all throughput tests, random data is used. This prevents on-the-fly compression of the data, and thus the measured result to become inaccurate. For the following tests, the unlimited data stream is used, as all traffic shaping is conducted within the kernel using *qdiscs*.

While testing the setup it was determined that MPTCP was not fully working within the cellular network. Measurements have shown that the MPTCP options in the TCP headers of the respective sub-flows are removed, but only when connecting to well-known HTTP ports (i.e. 80/443). This behavior is expected to be caused by a PEP within the core of the cellular network. PEPs are usually used to improve the performance of TCP connections, and thus are also called (transparent) TCP proxies. By responding with an ACK packet to any TCP connection request to a remote HTTP server, the TCP slow start is improved by eliminating one proxy-to-server RTT. But as this proxy is unaware of the MPTCP options, the connection attempt fails, due to the missing options in the response packet. Still, as MPTCP is comparatively young, and the PEP may not know of the remote server's capabilities, this behavior prevents starting an MPTCP connection, although everything is configured correctly. This problem was mitigated by using a different port on the server. Still, considering the long-term use of MPTCP, the PEPs also need to be adapted, or at least should not scrub unknown TCP options.

Configuration and measurement are both conducted from the mobile device. Therefore, a dedicated application was written, implementing the required functionality. Figure 4.18 shows the main configuration options and a log output. The *Background Service* is used to run measurements in the background while the App is not visible. This is required to allow the display to be switched off during the measurements, thus reducing the addi-

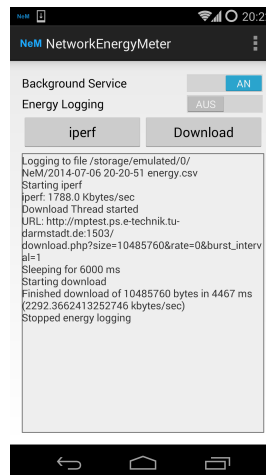


Figure 4.18: Screenshot of the main screen of the measurement application running on the mobile phones (from [Rad14])

tional error introduced by the screen backlight. The energy logging is only available on the Nexus 5. The iperf button starts an iperf download from the server as configured via the command line interface. The download button starts a regular download.

#### 4.3.2 Interface Reference Power Consumption

The measurements as described in the previous section are run in a radio-quiet environment, thus minimizing the error caused by interference of other devices. In particular, measurements were run during the night, where activity both on WiFi as well as within the cellular networks is expected to be low. Thus, the optimal performance is identified. Assuming a congested radio environment, the expected data rates are lower. The behavior of a congested network is similar to using lower data rates. This is intuitive, as in both cases the interface needs to be active for the full time. First, the reference model for each interface is established by running a number of tests with different data rates. Based on this, a general cost model for bulk data transmissions on the respective interfaces is derived. Only thus, the comparison with the energy consumption of MPTCP is possible.

The power models are measured for all available interfaces on both the Nexus 5 and Nexus S. On the Nexus S WiFi, 2G, and 3G are available. As the use case of interest is video streaming, 2G connections are neglected, as available data rates are insufficient to support even basic video rates. On the Nexus 5 WiFi, 2G, 3G, and 4G are available. Hence, on both devices WiFi and 3G are analyzed, with additional measurements on the Nexus 5 establishing the reference for LTE connections.

Table 4.7: Power consumption of the DUTs in different idle configurations  
(\*includes wake lock and measurement)

State	Nexus S		Nexus 5	
	Power	Std	Power	Std.
Idle	79 mW	3 mW	n/a	n/a
WakeLock	280 mW	27 mW	151 mW*	48 mW
Idle, WiFi	90 mW	72 mW	161 mW*	52 mW
Idle, 3G	96 mW	53 mW	171 mW*	58 mW
Idle, WiFi, 3G	97 mW	53 mW	170 mW*	61 mW
Idle, 4G	n/a	n/a	182 mW*	89 mW
Idle, WiFi, 4G	n/a	n/a	173 mW*	96 mW

Before the interface power consumption can be calculated, the influence of the different operating states on the overall power consumption is measured. Therefore, the devices are first placed in airplane mode to calibrate the minimum power consumption. Further references are established for idle modes with different interfaces active but idle. On the Nexus S, the measurements are run externally and require no additional considerations. Contrary on the Nexus 5, where only internal measurements are possible, measurements of the idle state are not possible. This is caused by the power measurement functionally constantly polling values from the /sys file system, thus requiring the device to be active. Hence, for the Nexus 5 all references are established with the power monitoring active. This does not influence the interface power models, because the respective reference measurement is used for calibration. Still, this prevents modeling of the full system consumption, as no idle power can be determined. However, considering that network interfaces are mostly used when the device is active, and idle consumption is small compared to actively using the device, some conclusions on the overall device behavior can be drawn.

The identified idle power of the respective devices and operating modes is given in Table 4.7. Generally, the differences between idle mode and additionally activating data interfaces on the Nexus S are the range of 20 mW. A similar increase is visible on the Nexus 5, when neglecting LTE. This increases the idle consumption by 31 mW. The observed standard deviations are comparably high. This is expected, because the interfaces are always active for only short intervals. Thus, the power consumption is mostly low, but shows spikes during periods of activity.

As cellular throughput measurements cause a considerable traffic and thus cost, care was taken to not exceed the traffic budget. For all tests, a monthly allowance of 10 GB was available. Tests were run according to these requirements. The available data contract

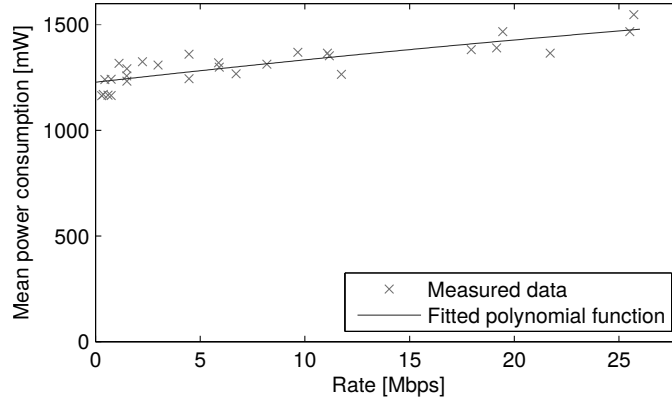


Figure 4.19: Reference power consumption of the Nexus 5 on LTE

allows the use of all available technologies without traffic shaping. Thus, the validity of the measurements should not be affected by artificial network limitations.

Figure 4.19 shows an exemplary measurement used to derive the power characteristics of the LTE interface. Here, the influence of the download rate on the power consumption of the LTE interface of the Nexus 5 is given. The figure shows an idle consumption of roughly 1.2 W while the interface is active. Increasing the data rate on the interface increases the power consumption to approximately 1.5 W.

Similar to the power measurement and modeling of the SBCs, also here a second order polynomial is fitted to the measurements. The resulting models are summarized in Table 4.8. For each interface as indicated in the first column, the respective polynomial is given in the second column. The ranges of validity are given in the third column, while the RMSE of the fit is given in the last column. Generally, the error is small compared to the values.

The model is used by selecting the respective operating state from Table 4.7. Then, the power consumption caused by the traffic on the active interface is added. Here, the index  $NxS$  denotes the model for the Nexus  $S$ ,  $Nx5$  the ones measured on the Nexus 5. The absolute power in a given operating state for the case of a single active interface is calculated by

$$P_{\text{dev, abs}} = P_{\text{Idle, \{if\}}} + \sum_0^{N_{\text{if}}} \delta_{\text{if}} \cdot P_{\text{dev, if}}(r), \quad (4.18)$$

where  $\delta_{\text{if}} = 1$ , if the respective interface is active. Further, only one interface may be active simultaneously:  $\sum_0^{N_{\text{if}}} \delta_{\text{if}} = 1$ .

From these models, the energy consumption of data transmissions on the available interfaces can be calculated. A more intuitive representation of this is the energy consumption per bit for a given data rate. The derived functions are given in Figure 4.20. From this,

Table 4.8: Power models of the different interfaces of the Nexus S and Nexus 5

Function	Model [W]	Range [Mbps]	RMSE [mW]
Nexus S			
$P_{N_{xS},WiFi}(r)$	$0.455 + 0.030 \cdot r/\text{Mbps} + 0.4e^{-3} \cdot (r/\text{Mbps})^2$	]0, 30]	31
$P_{N_{xS},3G}(r)$	$0.911 + 0.025 \cdot r/\text{Mbps} + 0.001 \cdot (r/\text{Mbps})^2$	]0, 6]	20
Nexus 5			
$P_{N_{x5},WiFi}(r)$	$0.332 + 0.057 \cdot r/\text{Mbps} - 0.001 \cdot (r/\text{Mbps})^2$	]0, 35]	83
$P_{N_{x5},3G}(r)$	$0.556 + 0.061 \cdot r/\text{Mbps} - 0.004 \cdot (r/\text{Mbps})^2$	]0, 11]	40
$P_{N_{x5},4G}(r)$	$1.046 + 0.011 \cdot r/\text{Mbps} - 0.006 \cdot (r/\text{Mbps})^2$	]0, 25]	52

some interesting observations can be drawn. First, the lowest cost of data transmissions is observed using WiFi on both devices, followed by 3G on the Nexus 5 and Nexus S. Most expensive over the given data rates is LTE on the Nexus 5. This confirms observations made by Huang et al. [HQG+12], showing that LTE requires more power per bit than 3G.

The observed data rates reflect the performance of the cellular network in a suburban environment during night. Assuming a more crowded network during the day, the expected data rates are lower. Hence, connection durations are longer, increasing the overall power consumption per transferred bit.

These measurements describe the power consumption of continuous, constant data rate streams. Often, a more relevant number is the energy consumed for a finite transfer. In WiFi networks, the steady state power consumption approximates the real power consumption well. Only small differences in power consumption are caused by opening and shutting down the connection. This is usually done within one beacon interval of 100 ms.

On cellular networks, connection establishment and teardown are more complex. The former is split into network admission, conducted when the device is switched on and registers on the network, and the establishment of a data channel. The crucial part for the mobile user is the actual connection establishment. This happens regularly, and is immediately noticeable by the user due to the initial delay. Additionally, this causes considerable energy consumption. Less noticeably by the user, but more prominent on the energy budget is the connection teardown, which is often delayed by a few seconds after the last data transmission is observed. This is done to reduce the signaling effort in the cellular network, but also to improve QoE by keeping the data channel open, and thus eliminating the connection establishment delay should a subsequent connection be required.

The delay before a connection is established and data may be exchanged between the mobile device and the network is called the *ramp* duration. Consequently, the energy spent while establishing the connection is called the ramp energy. Similarly, the energy spent after the last data has transmitted is called the *tail* energy. Both have a large influence on energy consumption of the mobile device, in particular for short data transfers. Figure 4.21

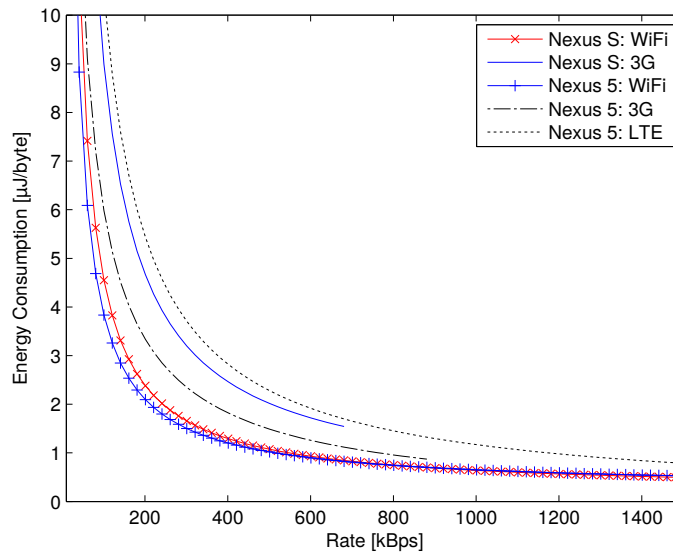


Figure 4.20: Comparison of the energy cost per byte on the different devices and interfaces

shows the ramp and tail energy on the example of the Nexus 5 on a 3G network. Ramp and tail energies are not considered in the following, as the goal is the analysis of the steady state energy consumption of streaming media. Should an accurate view of finite data transfers be desired, additional measurements are required.

#### 4.3.3 MPTCP Power Consumption

The power consumption of MPTCP is measured as described in Section 4.3.1. First, the network connections on the device are configured as required, meaning that on the Nexus S both WiFi and 3G are activated, while on the Nexus 5, two different measurements are run, first measuring WiFi in combination with 3G, then with LTE. On the remote side, the server is configured to throttle the download belonging to the DUT to the desired rate. Finally, the throughput and power measurement are started simultaneously.

The bandwidths evaluated are selected to be well within the limits of the network technology with the lowest available bit rate. Hence, on the Nexus S the combination WiFi and 3G was measured with a rate of 200 kBps, while the Nexus 5, supporting higher data rates on 3G was tested with a rate of 500 kBps. The energy consumption of WiFi in combination with LTE is measured on the Nexus 5 for data rates of 1 MBps and 2 MBps. For each setup, different measurements are run, incrementally increasing the load on the WiFi interface, while transmitting the remaining data on the cellular interface. Each test was run for 10 s to saturate the link and exit the slow start phase. Each test series was



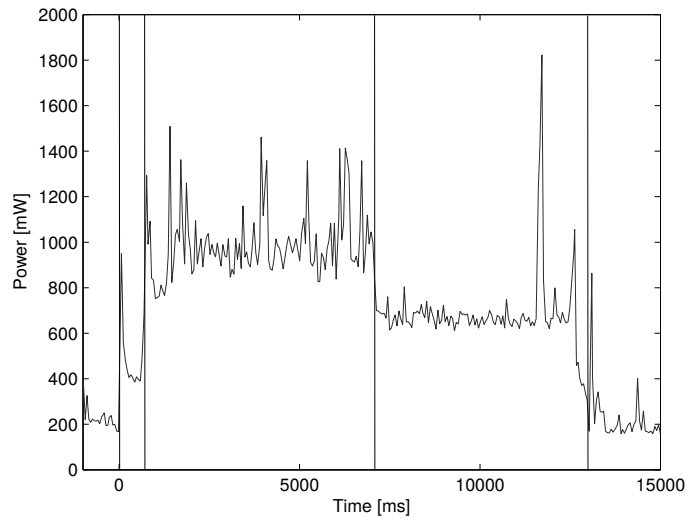


Figure 4.21: Measurement of ramp and tail durations and energy for a 3G download on the Nexus 5

repeated multiple times to collect sufficient data to gain significant results. Here, more repetitions were possible for the lower absolute rates due to limitations in data volume.

The measurements for the Nexus S with a rate of 200 kbps and the Nexus 5 on WiFi and LTE for a rate of 1 Mbps are shown in Figure 4.22. The blue crosses indicate the average energy consumed for transferring a byte in the given configuration. Clearly, for both cases the power consumption of WiFi alone is lowest. The second cheapest option receiving a constant bit rate stream is using the cellular interface alone. Only, if the cellular interface cannot supply the demand, the additional cost of using MPTCP should be considered.

It must be noted that caused by the throttling on the server, the data rate received on the client was lower than desired. This is caused by throttling the sockets belonging to the individual TCP subflows, neglecting the additional overhead introduced by the MAC encapsulation, TCP, and MPTCP, thus reducing the net rate. This is accounted for in the derived cost per byte by normalizing the consumed energy by the net rate received over the course of the measurement. A more detailed observation of the net rate showed that for a low fraction of traffic on the cellular interface in some configurations only half the configured rate is received. The cause for this lower rate is determined by checking a packet dump of one of these tests. The identified root cause was the variability in RTTs between interfaces. This problem is caused by MPTCP using the RTT to estimate the quality of the respective link. If it increases largely, the capacity is deemed to be insufficient. Hence, more packets are shifted to the second connection, and some also considered lost on the link, because packets not being acknowledged in time, are retransmitted on the

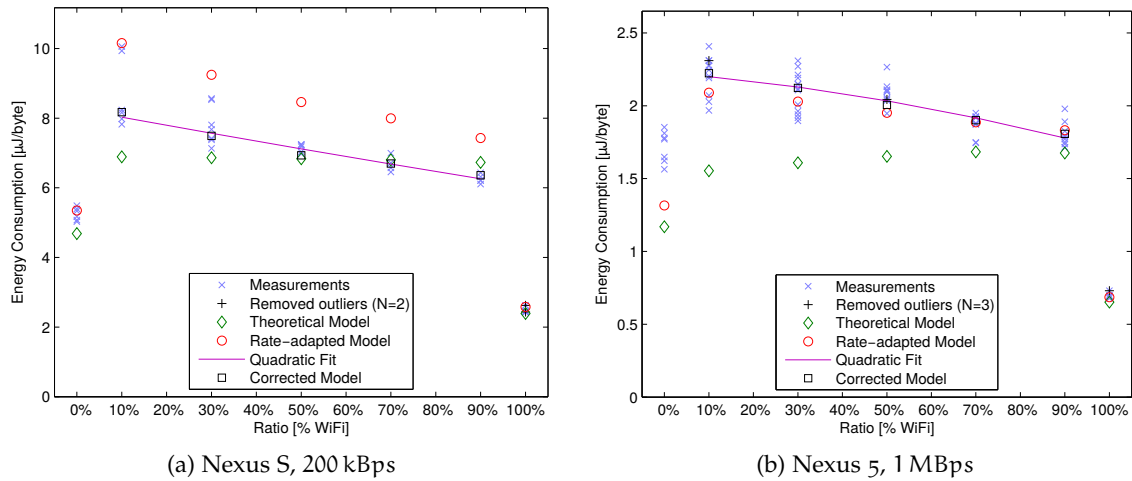


Figure 4.22: Energy consumption per transferred byte for different data rates

link with the lower RTT. Thus, the resulting net rate received on the remote device is reduced compared to two independent TCP connections, not knowing of the others state.

As stated in the beginning, the goal of this study is the comparison of the MPTCP power consumption with the alternative of using multiple TCP flows in parallel. Hence for each setting, the theoretical cost of receiving the requested net rate in the given configuration is shown in the form of green diamonds. Considering the actual rates as received via MPTCP, the rate-adapted model as given in red circles is achieved. Clearly, neither approximates the observed behavior well.

The first interesting observation is that, in particular for the pure cellular connection (e.g. 0% WiFi), the measured power consumption is considerably higher than the predicted one. Less visible, but showing similar behavior, this is also observed on connections using only WiFi. This is expected to be caused by the additional overhead of MPTCP. This includes both the increased computational complexity of handling both flows, as well as reduced packet size due to the overhead caused by MPTCP.

Before fitting models to the collected data, the data is cleaned to remove outliers. Therefore, Cooks distance [Coo77] is used. From the derived energy cost for each configuration, the samples showing a distance  $D_i$  of larger  $4/N_{\text{samples}}$  from the derived regression are removed from the data set. This is indicated in the plots by a black plus sign.

Generally, the power consumption of the MPTCP flows can be estimated based on the added power consumption of the active interfaces, or approximated by fitting a function to it. Both are sub-optimal: the modeling is based on the observed behavior, while the additive model shows large differences. Hence, a hybrid approach is chosen. The power models of the individual interfaces are added, and the difference between these and the

Table 4.9: Differences between the rate-adapted model and the quadratic fit and resulting correction functions

<b>Nexus S</b>		
<i>Setting</i>	<i>Mean Dev.</i>	<i>Function</i>
WiFi/3G (200 kB/s)	-21.2 %	$C_c(f_W)/\mu\text{J B}^{-1} = +1.144 \cdot f_W - 2.099$
<b>Nexus 5</b>		
<i>Setting</i>	<i>Dev.</i>	<i>Correction Function</i>
WiFi/LTE (1 MB/s)	+2.5 %	$C_c(f_W)/\mu\text{J B}^{-1} = -0.198 \cdot f_W - 0.153$
WiFi/LTE (2 MB/s)	-5.8 %	$C_c(f_W)/\mu\text{J B}^{-1} = +0.185 \cdot f_W - 0.152$

measurement approximated. Thereby the relative energy expense or savings are modeled and can thus directly be derived for the desired setting.

The correction function is derived by calculating the difference between the power estimated using the rate-adapted model based on the individual interfaces power consumption and comparing its outcome with a function fitted to the observed behavior. Thus, the changes in power consumption can be derived for the different operating states. This correction function is then used to calculate the power consumption based on the individual interfaces' power consumption.

The derived correction functions are given in Table 4.9. The first column identifies the analyzed setting, the second identifies the mean deviation between the rate-adapted model and the fit to the MPTCP power consumption. The derived correction function is given in last column. Here,  $f_w$  denotes the fraction of traffic on the WiFi interface. The mean deviation between the rate-adapted model on the Nexus S and the MPTCP power consumption with 21.2 % is comparatively high, thus indicating synergies between both interfaces. Concluding, MPTCP saves power compared to the use of two independent interfaces. Results on the Nexus 5 are mixed. Depending on interface and data rate, the cost of using MPTCP may also be larger compared to using two independent flows. Still, considering reliability of the established MPTCP connection, the additional overhead may be justified.

Table 4.10 compares the resulting errors of the presented power models. Compared to the naive MPTCP power model based on the nominal traffic distribution on the active interfaces (static) using the prepared single-interface power models, both the rate-adapted and corrected model show increased accuracy on the Nexus 5. On the Nexus S, the rate-adapted model results in less accurate results, while the corrected model cuts errors by almost 50 %. Still, also the absolute offset as given in Table 4.9 must be considered. Comparing the performance of the rate-adapted and the corrected model, large improvements are visible on the Nexus S, while the performance on the Nexus 5 is similar. This correlates with the mean deviation as given in Table 4.8.

Table 4.10: RMSE of the different models

<b>Nexus S</b>			
<i>Setting</i>	<i>Static</i>	<i>Rate-adapted</i>	<i>Corrected</i>
WiFi/3G (200 kB/s)	0.9972	1.2665	0.5263
<b>Nexus 5</b>			
<i>Setting</i>	<i>Static</i>	<i>Rate-adapted</i>	<i>Corrected</i>
WiFi/3G (500 kB/s)	1.3508	0.5148	0.5753
WiFi/LTE (1 MB/s)	0.4376	0.1834	0.1714
WiFi/LTE (2 MB/s)	0.3452	0.1365	0.1655

Considering the higher packet loss and power consumption when using MPTCP with the majority of traffic on the interface with lower RTT, modifications of the MPTCP functionality are recommended. As currently the interface with the lower RTT is used for control messages, longer delays on the secondary interface are often interpreted as packet losses, in particular when the RTT is variable. Although this interface may be cheaper in terms of energy consumption or monetary cost, from performance perspective most traffic should be transferred on the interface with lower RTT. Thus the number of retransmissions can be limited. The tested implementation by default balances the load equally on both interfaces, independent of the actual throughput and RTTs.

Based on these observations, two possible improvements become apparent. First, the scheduler balancing the load on available interfaces should be extended to put the majority of traffic on the interface with lower RTT. Thus, the performance can be improved while simultaneously reducing the probability of retransmissions caused by delayed packets on the interface with higher RTT. This often also reduces the monetary cost on the cellular interface, as it regularly shows higher RTTs compared to the WiFi interface. Secondly, the tolerance for delay variations should be increased to avoid retransmissions of packets still being delivered by the secondary interface. This is particularly problematic in cellular networks, where layered reliability functions (e.g. LTE HARQ and TCP retransmissions) delay and re-order packets delivered to the mobile device. This is a problem inherited from plain TCP, which also has difficulties adapting to mobile environments. Assuming the implementation of both, the overhead as observed in the measurements may be reduced by simultaneously improving end-to-end performance

Considering the architecture of MPTCP, rebuilding the functionality of TCP by keeping track of packets, sequence numbers, packet order, checksums, and retransmissions in combination with the problems caused by using TCP sub-flows (e.g. removal of TCP options, duplicate and often conflicting retransmission mechanisms), the underlying data flows should be migrated to UDP connections. Functionality wise, nothing would be lost, because MPTCP already implements the required functionality. Contrary, the efficiency

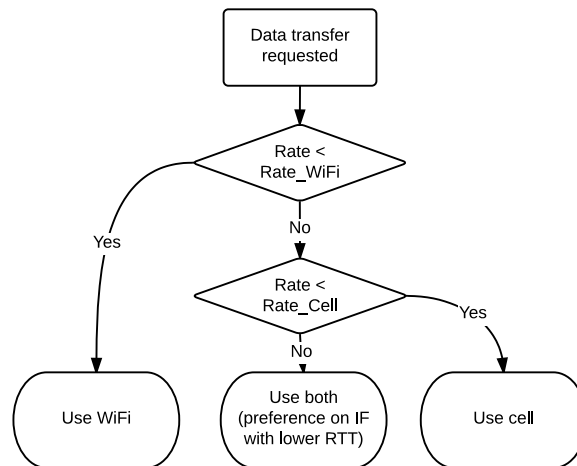


Figure 4.23: MPTCP decision tree

of the implementation is increased by avoiding the overhead and hassle caused by TCP sub-flows, in particular in wireless networks.

#### 4.3.4 Summary and Conclusions

Knowing the influence of MPTCP on the power consumption of the device allows a more comprehensive decision on when and when not to use MPTCP on the mobile device. Improving performance and reliability of mobile data connections for the end user is an honorable tasks, but the decision should include as many parameters affecting the user as possible. Clearly, on mobile devices this also includes energy consumption as one of the main restrictions.

Abstracting the above observations to practical recommendations for its use in CBR streaming leads, a few simple rules for using MPTCP in an energy efficient manner can be derived. The decision includes, besides the duration of the requested connection, the available network interfaces and the required bandwidth. According to the above observations, the following decisions should be made when deciding about the use of MPTCP: If WiFi is available, and the available bandwidth is sufficient, then it should be used. If WiFi is not available, or the available bandwidth is insufficient, and the cellular interface provides a sufficient bandwidth, the cellular interface should be used independently. Only if the available bandwidth on both interfaces individually is insufficient, the use of multiple interfaces is recommended. Still, the majority of traffic should be transferred on the interface with the lower RTT, thus minimizing the energy cost of the data transmission. The resulting decision flow is also shown in Figure 4.23.

The accuracy of the presented models may further be increased by monitoring the CPU and compensating for it, as was done in the SBC measurements. Thus, a more accurate

network power model can be generated and remaining, unexplained effects minimized. Still, this requires running load generators or benchmarks with configurable system utilization on the mobile devices. Here, the Android OS often interferes by terminating long running processes, and permission management complicating tasks compared to SBCs.

As already noted earlier, the presented data is valid for the estimation of steady state connections only. Further interesting areas of research are the estimation of the power consumption of finite data transmissions using MPTCP. For this, the ramp and tail energies of the different interfaces must be included. These depend on the cellular network configuration, and hence may change between different network operators and also over time. Here, the analysis of the energy cost of bursty video streaming as is commonly deployed by video portals is of interest.

A further interesting aspect of the power consumption of mobile communication is the idle power of recent devices. As these measurements are clearly not possible with the internal CPU-based measurement approach, external measurements as conducted for the Nexus S are recommended. Still, due to the built-in battery, a number of challenges arise. First, the device must be modified by inserting a measurement shunt between battery and device. Preliminary modification of one device and subsequent measurements show that the measurement approach is not as straightforward as expected. Compared to the SBCs and the Nexus S, a 100 m $\Omega$  resistor is already too large, preventing the successful boot sequence by causing a too large voltage drop. Similar observations were made when using an external power supply running at the maximum allowed battery voltage of 4.2 V. Reducing the resistance of the shunt further lets the device boot, but errors are too large to reliably measure the power consumption of the device with the available measurement equipment. Thus, another power measurement approach needs to be developed, either using a measurement card with lower voltage range, or devising a custom circuit board making use of operational amplifiers, a Coulomb counter and the respective circuitry, or a single chip measurement solution, directly interfacing with the measurement PC.

Particularly interesting for end users is the cost of background data transfers. Due to the ramp and tail states as observed in the measurements, considerable energy is wasted by different Apps frequently requesting updates from their servers. Recent Android versions already implement network and App activity scheduling and batching, but the precise quantification of its influence, in particular during daily use, on the resulting battery life time are not thoroughly evaluated yet. Android itself provides some power monitoring and estimation tools within their developer tool set. Still, their accuracy is not established yet. Here, the an improved power measurement approach would help quantifying their accuracy at least on a few exemplary devices, thus allowing the scientific community to base their work upon these results.

Based on the collected data, power models for long-running connections are derived. These are used to compare the power consumption of both DUTs for the case of constant bit rate streaming using MPTCP. From this it is derived that using MPTCP on single inter-

faces increases the power consumption compared to plain TCP. In the case of using both interfaces in parallel, the power consumption is reduced compared to the added power consumption of two individual TCP flows on both interfaces. Still, using both interfaces simultaneously is clearly more expensive than using a single interface alone. Hence, guidelines for using MPTCP in the given scenario in a most energy efficient way are derived: If possible, only a single interface should be used, where WiFi is more energy efficient than the cellular interface. For reliability purposes, MPTCP's backup mode may be used to re-establish connections on the secondary interface should the first link break. This increases the reliability of the circuit on the cost of energy consumption. If the capacity of a single interface is insufficient to satisfy the current demand, the second interface may be added, with the majority of the load on the interface with the lower RTT. This increase in performance is bought by a considerable increase in energy consumption. Here, individual decisions on its use should be made based on QoE estimates, preferably also including energy consumption.

#### 4.4 POWER CONSUMPTION OF COMMUNICATION INFRASTRUCTURE

The energy consumption of communication networks is established to be a major aspect in current and future networks. Particularly when considering developments in the direction of 5G, the sheer number of devices to be supplied with reliable and often high bandwidth links poses large demands on the communication infrastructure. In the previous sections, the power consumption of different emerging platforms and technologies in different network domains is established. At the edge of the wired network, the models for the energy cost of decentralized caching, offloading and computation using SBCs are developed. In the access and core network, the performance and energy consumption of both a hardware and software OpenFlow switch is analyzed and modeled, from which recommendations on their use in future networks are drawn. In the mobile access domain, the emerging technology of MPTCP providing bandwidth aggregation and seamless handover using multiple interfaces is analyzed for the case of constant bit rate streaming, from which recommendations for its use are derived.

The detailed results of these studies in the different network domains are summarized along the research questions posed in the beginning:

*RQ 1.1: What is the energy cost of decentralized caching and computational offloading using SBCs and how can it be determined?* The energy cost of decentralized caching and computation is analyzed based on power measurements of the Raspberry Pi line up, the Odroid C line up, and the Cubieboard. These measurements are correlated with the monitored system utilization (i.e. CPU and network utilization), from which power models of the DUTs are derived in a regression based approach. As these models are based on readily available system monitoring values, these may be used to estimate the power consumption of the respective device with minimal overhead. These models are exemplary used

to determine the power consumption of a larger deployment of Odroid C1s, showing the energy efficiency of using a decentralized caching and offloading approach. This shows the feasibility of using these high-level power models for the centralized monitoring of a decentralized deployment with minimum overhead, considering that system monitoring may already be implemented. Based on the system monitoring and power estimates, the distribution of content and services within a larger network can be optimized based on criteria like system resources, network cost and location, or overall energy consumption.

*RQ 1.2: How does the energy efficiency of hardware and software OpenFlow switches compare and what are their respective benefits?* The power consumption of both a hardware and a software OpenFlow switch is analyzed and modeled, serving as examples of the capabilities and related energy consumption of devices within the core network. Both show distinctive advantages and disadvantages both in performance and energy consumption. The hardware switch supports a large number of connected devices and executes simple OpenFlow rules in line speed. Still, advanced OpenFlow matches and actions not supported by the hardware drastically reduce observed performance and increase in the energy efficiency per packet. The idle power is comparatively high. The main influence on power consumption is observed when re-configuring the link speed of active ports. Traffic, also the one processed by the CPU does not significantly influence power consumption. The packet processing performance of the Open vSwitch was not significantly affected by matches or actions applied to the packets. Furthermore, the power consumption of the processed network traffic is proportional to the rate, and mostly independent of the matches and actions applied to it. Concluding, the software switch should be used for more complex traffic manipulation tasks, while the hardware switch is highly efficient with simpler network operations, only marginally increasing power consumption while still keeping the performance up.

*RQ 1.3: What is the cost of increasing reliability and throughput of mobile communication using MPTCP for constant bit rate streaming?* The power consumption of MPTCP is analyzed on the example of the Nexus S and Nexus 5 for different traffic rates on WiFi, 3G, and 4G networks. Similar to the SBCs, also here power models for transferring data on the respective interfaces are calibrated. Based on these, the comparison of the energy cost of MPTCP for the special case of CBR streaming is drawn. Compared to the added cost of two individual TCP connections with the configured data rates, MPTCP is more energy efficient, although its use on a single interface increases the power consumption on the Nexus S. From this, general recommendations for the use of MPTCP are drawn, namely that multiple parallel connections should only be used if the capacity of a single interface is insufficient. If reliability of a connection is desired, also the use of MPTCP on a single interface using backup flows may be considered, slightly increasing the cost of the overall data transfers.

Considering the availability of power models for the full network, conclusions on the optimal placement of content, services, and devices can be drawn. For example, popular



content as derived based on user preferences or global popularity may be moved close to the end user during idle times. Combining this with knowledge of the network infrastructure and routing paths, the most energy efficient option for each connection can be calculated, thus minimizing the energy footprint of the network. Considering the use of e.g. MPTCP, a mobile user may begin watching a video while at home, and be seamlessly handed over to the cellular network, routing traffic to the respective end-point. Further leveraging the flexibility of SDN, traffic to and from one location can be re-routed in the network backbone to the most energy efficient remote location. Adding the capabilities of NFV, also the remote service may be re-located to further optimize performance and reduce energy consumption.

Still, more needs to be done to further reduce the energy footprint of communication networks. The currently analyzed devices generally show a comparatively high idle consumption and do not implement any noteworthy energy saving techniques. Clearly most advanced are smartphones, in recent releases implementing batching of network requests and also App activity. But cellular network access is still a major contributor to mobile power consumption. Also WiFi achieves high data rates, but as soon as the network is congested, or reception is poor, mobile devices waste considerable amounts of energy trying to transmit and receive data. The SBCs show comparatively low idle power consumption, but do not implement further approaches like deep sleep to further reduce energy consumption when idle. Worst are the OpenFlow switches showing a dynamic of maximum 10% between idle power consumption and under full load. Here, clearly advanced power saving mechanisms should be developed, allowing devices to enter energy saving modes when only lightly used. Most prominent examples are access switches in office environments, being actively used only 8 h to 10 h per day.

Software-based optimizations have shown to be strictly limited due to the high idle consumption of the available devices and their mostly linear reaction to load. Considering the network load to be constant, and only flexible in time, the most promising network domain for optimizations is wireless, showing some nonlinearities in the power response. Here, having a better signal or generally higher throughput reduces the cost of data transmissions.



THE performance of communication networks is one of the major aspects of digital communication. Where in the past data rates of a few kbps were sufficient, modern communication requires constantly higher data rates [Cis16]. This is currently primarily caused by the increasing demand for video content, and will further rise due to consumers demanding higher video quality [Cis16]. Additional demand is predicted to be caused by the trend of virtual reality (VR) [Mas16]. There, considerably higher data rates are required, supplying a higher resolution for a larger field of view, required to create an immersive experience for the end user.

At the same time, data access is becoming more mobile [Cis16]. The data volume consumed by mobile users is predicted to grow by 53% over the following years. Also here, the majority of traffic (i.e. 75%) will be mobile video by 2030 [Cis16]. The increased load on the network infrastructure must be coped with by the mobile access network as well as the network backbone and CDNs.

Here, the power consumption of devices, in particular on handsets, becomes increasingly important. Currently, 30% of the power consumption of a smartphone is caused by network interfaces [CDJ+15]. Here, the power consumption mainly depends on the used device type, network technology, and transmission duration. Hence, the power consumption of an end-user device depends, besides the interface power consumption, on the available network performance.

Generally, the performance of cellular networks is by the network operators with maximum data rates and maximum coverage only. Hence, a number of approaches were developed and implemented measuring the performance of the cellular network (e.g. OpenSignal [OpS], Sensorly [Sen], or NetRadar [SSM13]). These collect network performance measurements in a crowd-sensing manner to create network coverage reports for cellular network operators. Generally, the collected data is not available for further analysis. Additionally, the focus of these studies is on the cellular network alone, preventing the comparison of different network types and interfaces.

Therefore, a crowd-sensing measurement study is set up to determine the KPIs of different cellular network technologies (e.g. 3G/4G), and WiFi. Based on these, first the location-based network quality is determined to create a network performance map. Furthermore, network performance models are derived. Combining these with the power models as derived in Chapter 4, the power consumption of data transmissions on different interfaces can be evaluated and compared for similar situations.

Pedestrian or nomadic mobile network access scenarios provide interesting research areas for mobility prediction [DG12], content offloading [RHR08], and scheduling [BKW+13]. Still, this is a relatively well researched area [GWC+15]. Another less prominent scenario is mobile data access on trains. Currently, 69% of travelers on UK trains use mobile data services during their journey, out of which 54% are dissatisfied with the available service [FS16]. A number of studies are available, proposing optimizations to improve network access on trains (e.g. [KLW12]), but no systematic study measuring mobile network access on trains is available to the best of the author's knowledge. Hence, a thorough analysis is conducted and described in the following, with the goal of guiding the user in its data consumption to improve the user experience.

Over the course of these studies, irregularities in the obtained measurements are observed. Hence, the measurement approach is systematically extended. Therefore, additional devices are used, keeping the maximum number of parameters fixed while running an extended set of tests. These target the analysis of the upper layer network functionality, finally determining the source of the previously observed variations to be caused by network management.

Based on these observations, the following research questions are derived, guiding the analysis of the cellular network performance for the above described scenarios:

- RQ 2.1 What are the parameters affecting cellular service quality and user-perceived network performance when mobile?
- RQ 2.2 How does cellular network access on trains differ from general mobile network access, and how can the network performance be predicted?
- RQ 2.3 What is the influence of network structure and management on the end-user perceived network performance?

RQ 2.1 is discussed in Section 5.1. There, crowd-sourced network measurements are described, which are used to derive the location-based network performance. Based on these measurements, the metrics and influences on network performance are discussed as well as conclusions on the performance and suitability of these for different tasks derived. The main observation derived from the crowd-sensed data is that signal strength and network performance are largely independent, while the major influencing factor is the available network technology.

RQ 2.2 is discussed in Section 5.2, presenting and discussing the particular challenges of cellular network access on trains. It discusses the challenges of localization and accurate estimation of the service quality. Based on this, the dependency between individual performance metrics and usability of the network for different tasks are analyzed. The resulting service classification is then used to develop a prediction algorithm, guiding the end user in his or her network consumption, thus increasing transparency of the network

performance, reducing disappointment, and possibly also guiding the network operator on improving network coverage.

RQ 2.3 is discussed in Section 5.3. Variations as observed in Section 5.1 are analyzed in detail and their root causes are identified. The influence of routing and gateway allocation in cellular networks on the QoS is derived. These observations are underpinned by a dedicated study analyzing the real-world impact of the observed variations on the page-load time of the 25 most popular domains, and concluding that network management decisions have considerable influence on the end-to-end performance.

Finally, Section 5.4 summarizes the findings derived from the location-based, train-based, and reference studies. Namely, the performance of the analyzed cellular networks mainly depends on the available network technology. If the technology is known, the performance of the network can be predicted with relatively high accuracy. Finally, within the analyzed network, the major implications on service quality are determined to be caused by network management decisions of the network provider.

## 5.1 CROWD-SENSED CELLULAR NETWORK PERFORMANCE MEASUREMENTS

Currently, different sources for information on the cellular network performance and availability are known. First, cellular operators publish maps showing the reach of their networks for different connection technologies. This data has considerable limitations: The covered area as shown on the maps is based on propagation models, which mostly include knowledge of the terrain, but fail in urban areas, where coverage is heavily affected by shadowing and multipath-fading. Further, these maps only show available technologies instead of useful performance metrics like RTT and throughput. Therefore, independent studies are set up, measuring KPIs of the cellular network in an end-to-end based manner. Examples are OpenSignal [OpS], Sensorly [Sen], or Netradar [SMS13] from academic side. These services are based on end-to-end measurements run from a mobile application to a remote server. However, the collected data is proprietary and thus not available for further analysis. Hence, an independent measurement solution was implemented and set up, with the focus on determining the mobile network performance based on end-to-end measurements.

The measurement setup, procedure, data collection, and visualization as presented here are also published in [KJH15; KMB+15; KMB+16].

### 5.1.1 *Measurement Methodology*

Measurements of mobile networks are conducted in an end-to-end based fashion, thus closely approximating the network quality as experienced by end users. For this, measurements are run from a smartphone to a remote server, dedicated to the network measurements. The always use the active network, which may either be the cellular, or the WiFi

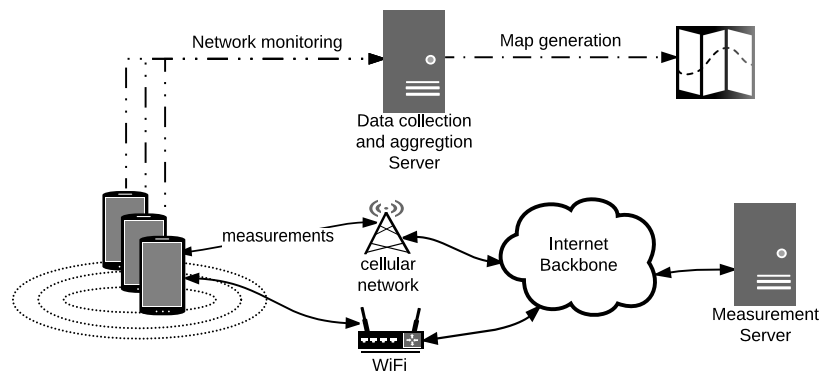


Figure 5.1: Schematic of the crowd-sensing measurement setup

network. Collected measurement data is then stored on a data collection server, from which the desired network performance maps are derived. The resulting measurement setup is depicted in Figure 5.1.

The general, and location based performance of the cellular network is measured in a crowd-sensing based approach. Therefore, a smartphone application is developed, passively recording system state, network environment, and location, while periodically triggering active measurements against a dedicated measurement server. This is published to attract a number of users interested in the performance of their networks. Furthermore, this is installed on smartphones dedicated to the measurement campaign.

The measurements can be divided into passive and active, depending on their requirement of actively using the available data interfaces. Passive measurements periodically record time, location, and signal strength of the network, while active measurements probe the network performance using the *ping* command to determine the RTT, or measure the network throughput by downloading a large file or running *iperf* measurements against a dedicated measurement server.

These measurements are recorded for the cellular network as well as the WiFi network. This is of particular interest, as no common data set is available, containing performance metrics of both networks within the same area. Thus, the performance of different network types (e.g. cellular/WiFi) can be compared at the same location, thus guiding network selection decisions.

The RTT measurements are least intrusive, requiring only one sent and received packet. However, the first data packet triggers the modem to activate the network connection and establish a dedicated data channel. This results in an extended RTT compared to subsequent requests. Hence, the first probe packet is discarded, as it mainly contains information on the connection establishment. To gain statistically significant results, a number of independent tests are required. Their number is limited by user mobility and possible changes in the network. The overall test duration is minimized by using *pings* -A option, sending the subsequent request as soon as a reply to the previous request has

been received. This reduces the measurement time while the connection is established, but does not work for unstable connections, where packets are lost, thus causing a time-out.

When measuring the RTT from mobile devices, also device internal delays must be considered. These are caused by OS internal processing, e.g. by the kernel, and upper layer processing, like the Java-VM. Hence, the resulting delay depends on the chosen tool and implementation. Li et al. [LMW+15] measure the device internal delay for different measurement techniques on WiFi. They compare the performance of native ping, HTTP ping, and a raw socket connection by monitoring the arrival of packets using wireless traffic sniffers. As expected, the best performance is achieved using the native ping command. Interestingly, they observe that the device internal processing delay also depends on the absolute RTT caused by the network and the remote machine. The ranges of interest for our measurements are below 50 ms, which is common on 4G networks. There, device internal delays for the used devices are in the range of 1.2 ms and 7.8 ms with a mean confidence interval of 0.3 ms and 0.9 ms respectively. Similar effects are expected on cellular networks, but cannot be verified due to lack of cellular packet sniffers. Concluding, the conducted RTT measurements first depend on the chosen measurement tools, but also on the device running the measurements. Hence, care was taken when evaluating the measurements to only compare measurements from the same device type.

Measuring the throughput of mobile data connections is a challenge itself. Throughput is simplest measured by saturating the connection for a considerable time, while simultaneously monitoring the network throughput. Particularly in mobile environments, this creates additional problems. First, cellular traffic is expensive. With currently available data rates, the monthly traffic budget may be used up in a few minutes. Secondly, the user may be mobile, thus affecting the potential network throughput, to which the measurement needs to adapt.

For fixed networks, a number of different techniques were developed, based on measuring delay differences of a pair or train of packets between sender and receiver. This allows probing whether the tested throughput can be processed on the full path between sender and receiver, or not. Thus, in an incremental approach, the available bandwidth can be narrowed down to a given range. This approach works well for fixed networks, but has two disadvantages prohibiting its use in mobile environments. First, the measurement duration is too long. Probing each targeted data rate requires a few to tens of seconds. As this approach narrows the range of possible bandwidth down in a binary manner, a large number of tests is required, resulting in measurement durations in the range of minutes. Running these in a mobile environment is clearly not feasible, where a pedestrian may move several hundred meters during a measurement. Secondly, due to traffic scheduling within the radio access layer, a large jitter is introduced. This highly affects the accuracy of the measurements. A possible countermeasure would be to increase measurement duration, thus eliminating the errors from the measurements. This is clearly not feasible given the above requirements.

Concluding, the only feasible option to determine the available bandwidth are link-loading measurements. Here, also a number of different techniques exist. From measurement perspective, UDP transfers are quite promising, as these are connection-less. Thus no slow start and backoff algorithms interfere with the measurement. Hence, the link can be saturated and the number of received packets at the remote side observed. Still, on cellular networks the traffic traversing the PGW is accounted for. Thus, a large number of incoming but later discarded packets still counts to the traffic budget. On the other side, TCP is currently the most common layer 3 network protocol. At the time of writing, more than 90 % of the Internet traffic is TCP traffic [FHL+14]. A number of different congestion control algorithms is available (e.g. Tahoe, Reno, Vegas, Cubic, ...) and more or less commonly observed on TCP end-points. As these measurements are focused on mobile devices running the Linux kernel, the most common congestion control algorithm is *cubic*. To not alter the behavior of the measurements in relation to other traffic present in the network, no modifications are made. Still, one needs to consider that the presented measurements are valid under the given assumptions only (e.g. TCP traffic using the *cubic* congestion control algorithm).

The measurement of the available bandwidth is conducted using *iperf*. Conventionally, it only measures the upload from the client to the server. Hence, it is problematic to measure the downlink in cellular networks, where heavy traffic engineering is conducted. Furthermore, the mobile device is behind a NAT, and thus not directly accessible from the Internet. Hence, a patched version of *iperf* is used, allowing the client to establish a connection to the server, which is then used to measure the 'uplink' from the server to the client. A version patched to include this functionality is publicly available<sup>1</sup>. This version was used for all throughput measurements as reported in the following.

The amount of traffic generated per measurement is minimized by reducing the measurement duration. In a number of test trials, the minimum duration was empirically identified to be 5 s. This results in reaching link saturation levels on all cellular technologies as well as on WiFi. The resulting error between the short and full duration bandwidth measurements was empirically determined to be below 10 %.

Limitations of the current setup are mainly caused by the high cost of cellular traffic, and the high energy consumption of accurate localization. Further, the frequency of signal strength updates depends on the status of the display. If the screen is on, updates are observed with a frequency of approximately 1 /s, while the interval is drastically reduced when the device is inactive. Additionally, active probing (e.g. RTT measurements) requires the device to be active to also get network coverage information. Further, it triggers a state change of the currently active network interface, which causes additional energy consumption. As already discussed in [KSB+13] the best trade-off between accuracy, battery consumption and latency must be found. Translated to the measurement application, the only way to improve energy efficiency is to reduce the amount of measurements. Hence, a

---

<sup>1</sup> <https://github.com/tierney/iperf>, accessed 2017-01-03



Table 5.1: Overview of all collected measurements (as of 2017-02-24)

Type	4G	3G	2G	total
Signal Strength	21.8 M	7.8 M	0.2 M	32 M
Cells	2.5 k	9.7 k	2.0 k	14.9 k
RTT	961 k	507 k	14 k	1.7 M
Throughput	7.1 k	4.9 k	48	16.7 k

number of different features were integrated, both reducing the power consumption and limiting the amount of traffic consumed.

Measurements are run against emanicslab<sup>2</sup> servers and a dedicated measurement server. These are lightly used and well connected machines with sufficient capacity to act as remote servers. Further, these are distributed over Europe, thus allowing to run measurements from different countries and mobile operators within Europe. On each of these machines two *iperf* instances are running, each listening on its dedicated port for the mobile applications to connect. On connect, a TCP measurement from or to the remote device is initiated. Further, these are also configured with a maximum test duration of 5 s, thus limiting the amount of traffic received by the mobile device.

### 5.1.2 Description of the Measurements

The crowd-sourced data collection was started in December 2013 and is still running at the time of writing. The data set consists of 28 M coverage samples, 1.7 M RTT samples, and 17.6 k throughput samples. The bulk of the measurements are recorded in and around Darmstadt (Germany) which is also caused by a number of measurement studies focusing on the detailed analysis of the cellular network performance to later generate models for analysis and prediction. A summary of the crowd-sourced data set is given in Table 5.1.

Based on the collected data, network coverage and performance maps are created. These show the availability of different network technologies and their performance in different areas. From these it becomes clear that networks are primarily upgraded in urban centers and other more densely inhabited regions, while roads show gaps in network coverage, sometimes only providing 2G or no connectivity at all.

The RTT and throughput measurements are mapped according to the recorded locations. As the network performance depends on the available network technology, the created maps are split accordingly. This is particularly interesting for older phones, not supporting newer technologies like 4G/LTE.

A further, interesting aspect of the collected data is the number and size of observed cells. From all recorded samples the ones of the actively used cells are selected. Thus, not

<sup>2</sup> <https://www.emanicslab.org/>, accessed 2017-01-03

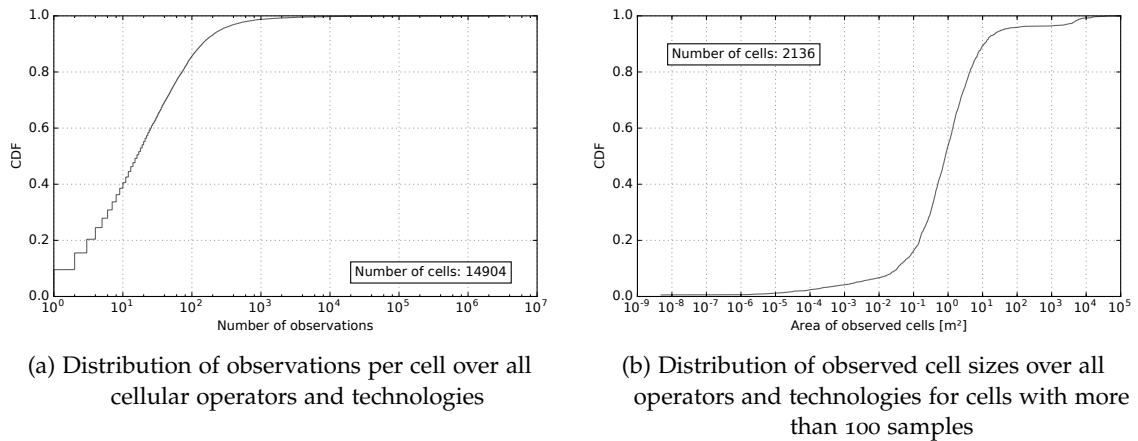


Figure 5.2: Comparison of the number of samples and derived cell size based on the crowd-sourced network measurements

only the theoretical, but a realistic cell coverage is determined. For this, the collected samples are grouped by mobile operator, location area code (LAC), and cell ID. For each cell, the minimum convex hull is determined using PostGIS' function `ST_ConvexHull`. Thus, the maximum covered area of each cell is determined.

Before this algorithm is run, the collected data is properly sanitized. Hence, data points are filtered by location accuracy, time after last location update, as well as time after the last cell update.. Thus stale values, which are frequently returned by the Android OS are removed.

Figure 5.2a gives an overview of the observed cells. The collected data shows a median count of 16 samples per cell. The mean count is 1036, indicating that for a number of cells a large number of samples was recorded. The plot shows an upper limit of over 1 million samples for the most frequently observed cells. Approximately 15% of the observed cells show a sample count of over 100, making it possible to derive cell-based performance metrics. This result is expected, as caused by different dedicated measurement campaigns, also a large number of measurements at fixed positions were recorded. Further, human mobility shows a quasi-stationary or nomadic behavior where people tend to stay at fixed locations for considerable time [CTS+11].

Based on the collected signal strength samples, the coverage area of the respective cells, and thus also their size can be derived. For this analysis, only cells with a minimum sample count of 100 have been selected. The derived area of the remaining 2136 cells is shown in Figure 5.2b. The CDF shows a median cell size of  $0.8 \text{ km}^2$  with a quite uniform distribution of cell sizes. 90% of the derived cell sizes are in the range  $50\,000 \text{ m}^2$  to  $10 \text{ km}^2$ . A few cells with an area of  $600 \text{ km}^2$  were also observed. These are the cells serving more rural areas, thus sharing their capacity with a larger, but less densely inhabited area. Still,

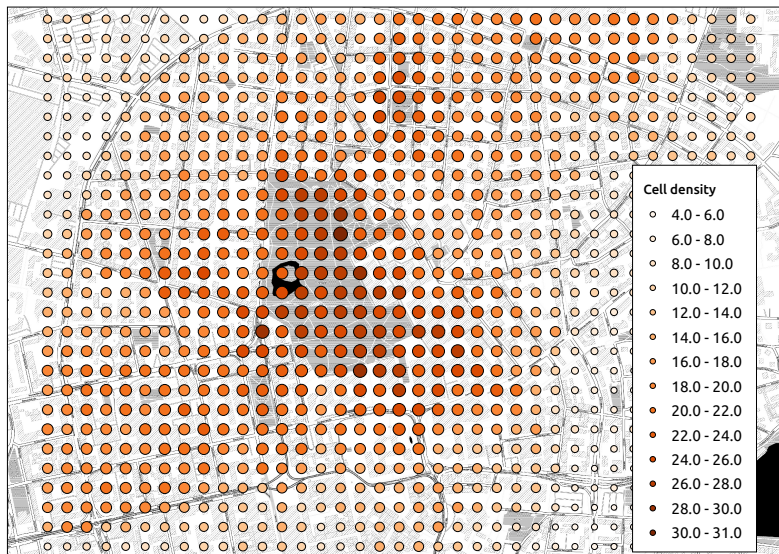


Figure 5.3: Map of cell densities in the city center of Darmstadt

due to the small sample size and focus of the measurements on a single urban region, no conclusive result can be given here. To achieve this, systematic measurements over a larger area would be required.

Based on the size of the respective cells, also cell densities at given locations can be determined. High data quality is assured by, besides the above filtering, including only cells based on a minimum of 100 samples in the analysis. Due to the high number of required samples, only cell densities for the city center of Darmstadt can be calculated. The map of derived cell densities is given in Figure 5.3. It shows a larger number of observed cells within the city center, while fewer cells are observed in the outer regions. Still, the effects of the comparatively open area of the park on the propagation of cellular signals are visible, increasing the relative signal quality of remote cells. Hence, while passing through, it is more likely to stay connected to the currently associated cell than being handed over to another cell, as would be expected in a more heavily built up area.

Analyzing the active measurements (e.g. RTT and throughput) and their relation to signal strength leads to some interesting observations. The dependency between RTT and downlink throughput for 3G and 4G networks is shown in Figure 5.4a. First, signal strength and RTT show no correlation. The same effect is visible when correlating RTT with throughput and throughput with signal strength. Generally, the RTT is lower on 4G networks, while the throughput is higher. The 3G measurements show a clear upper limit on throughput, which is caused by the maximum data rates supported by the network. Further, the RTT in the 4G network shows two distinct clusters at 20 ms and 32 ms independent of the measured throughput.

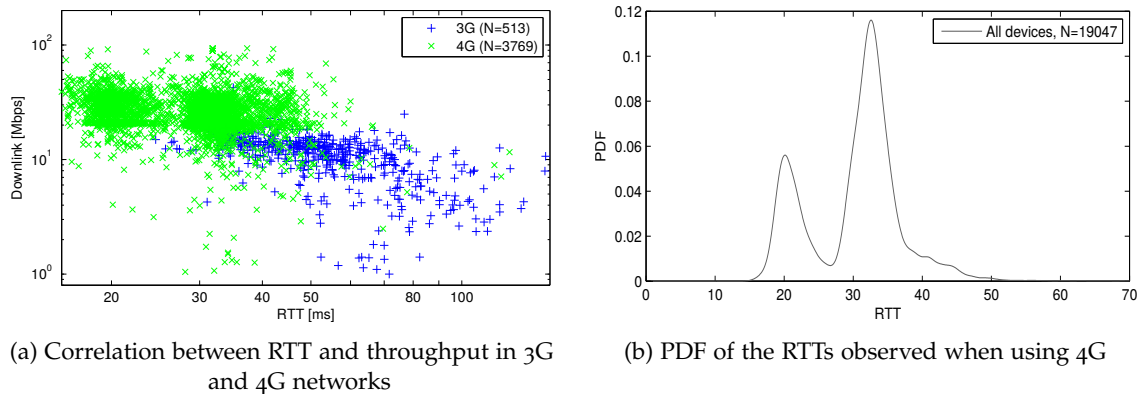


Figure 5.4: RTT throughput distributions as observed during the crowd-sensing study

Analyzing the distribution of the observed RTTs, in the probability density function (PDF) as given in Figure 5.4b is obtained. Here, two distinct peaks can be identified at 20 ms and 32 ms. As no cause for these abnormalities is known, a dedicated measurement study with the focus of identifying the root causes is set-up. These measurements are discussed in Section 5.3.

### 5.1.3 Summary and Conclusion

From these crowd-sensing measurements it becomes clear that the prediction of location-based network performance is not as simple as expected. Signal strength, RTT, and throughput generally do not correlate. From the data collected in the crowd-sourcing study, general results as also published by other researchers are confirmed. Further, the feasibility of determining cell sizes is assessed on the example of the inner city of Darmstadt. Additionally, the distribution of RTT and throughput in cellular networks is determined. These measurements confirm that the cellular service quality is mainly defined by the available network technology (e.g. 3G/4G). When this is known, the performance of the network can roughly be predicted without requiring any further active probing.

## 5.2 CELLULAR NETWORK PERFORMANCE ON TRAINS

An increasingly important topic is mobile data access on public transport. Commuters tend to use their mobile phones to read news or communicate with family and friends. Job related travelers use their time on trains typically writing and reading emails, using social media, or browsing [FS16]. Thus, for an increasing number of tasks continuous connectivity is already essential.

While network access on buses and trams is working well, the rail infrastructure provides additional challenges. These are on the one hand caused by environmental conditions, on the other by economic decisions. Environmental conditions include the traversal of different land-use zones from urban to rural, as well as geographical hurdles like hills, dams, and tunnels. Economic decisions are also caused by population density, where cell capacities are extended according to average cell utilization, and the decision of metalizing train windows to reduce the demand on air conditioning units.

From operator perspective, expanding capacity in rural areas covering train tracks is also often not an economically attractive option. Considering the burstiness of traffic demand, an expansion to support peak capacity is quite expensive. A train transporting a few hundred people may pass through a cell within just 30 s to 180 s, depending on velocity, cell size, and train type. Additionally, this route may be served only once or twice an hour, thus leaving the cell underutilized most of the time. Another challenge for the mobile operator to overcome is the sudden demand of a large number of devices requiring a handover between adjacent cells, creating considerable load on the network management infrastructure [ZPH+11].

An increasing number of trains are equipped with passenger WiFi. Thus, requirements on network management are decreasing, but due to the higher demand in networking, ever higher data rates are required. State-of-the-art approaches use multiple MNOs in parallel<sup>3</sup>. Therefore, a gateway is deployed on the train, maintaining a tunnel over multiple cellular modems to a remote server. Thus, always the best performance of the combined cellular networks is provided. The tunneling approach provides for TCP connections not to break. Still, connections may stall while the cellular service quality is insufficient.

This varying service quality is also often criticized by commuters or people traveling by train and trying to work [FS16]. Here, knowing the future network quality a-priori would at least allow the commuters to adjust their network consumption accordingly by scheduling network dependent tasks to times and locations with sufficient network coverage. In the long term mobile operators may be motivated to expand their network capacities in the currently under-provisioned areas.

Another challenge when analyzing the performance of cellular networks on trains is the localization. Due to the often metalized windows, incoming electromagnetic waves are highly attenuated. This affects both the cellular network and GPS. Thus, accurate localization of i) the measurement samples on the track, and ii) the localization of the end user when using pre-calculated network models provides another challenge.

The following measurements and network modeling are exemplary conducted on a single MNO's network. Besides location, the signal strength, RTT, and throughput are monitored over an exemplary route. Care is taken to systematically analyze influences on the cellular service quality, including seat location and train type. The following analysis is based on a Bachelor's Thesis conducted by Florian Fischer [Fis16], which was

<sup>3</sup> [https://www.bahn.de/p/view/service/zug/railnet\\_ice\\_bahnhof.shtml](https://www.bahn.de/p/view/service/zug/railnet_ice_bahnhof.shtml) accessed 2017-01-11

also published in [KFH17]. The following section describes the measurement setup (cf. Section 5.2.1), the collected measurements and their implications (cf. Section 5.2.2), and lessons learned (cf. Section 5.2.4).

### 5.2.1 *Measurement Methodology*

For the measurement of the cellular service quality the NetworkCoverage App as described in Section 5.1 is used. Still, due to the different environment, it is extended to suit the particular requirements of the train measurements. These are i) the implementation of an additional localization service using an API provided by Deutsche Bahn, ii) the collection of context information (e.g. seat position, wagon, train type), and iii) the further automation and configuration of the measurements and their intervals as described in Section 5.1.

As already stated, the localization itself is often a challenging problem. In particular on trains, where metalized windows hinder the propagation of electromagnetic radiation, GPS reception is often impeded. Hence, alternative localization approaches are evaluated. The localization using available networks as is done in the Google Play services<sup>4</sup> is of limited use when outside inhabited regions, because it is based on the service set identifiers (SSIDs) of the observed WiFi networks and cell IDs. If no SSIDs are available, the localization falls back to the use of cell IDs alone. As cells, in particular in rural areas, cover large areas, also the estimation based on the observed cells is comparatively inaccurate. Mean errors to expect are in the range of 1.5 km [MEMo8].

An alternative to these commonly available localization techniques is the Deutsche Bahn Zugradar<sup>5</sup>. There, the location of currently running trains is visualized on a website. The location published there is used in the following to derive the location of the train in question. Therefore, the API as used by the website is analyzed and the respective calls implemented in the mobile application.

The location of the train in question can be obtained by searching for trains leaving a given station. Therefore, first the departing station is requested from the experimenter. For this station, then the list of departures is fetched from the server. By default, this list contains trains in an interval of one hour around the time of the request. This list is shown to the experimenter to select the respective train.

Based on the derived train ID, information on the train is obtained from the web server. The detailed information is available from one of several API endpoints. For long-distance trains, the main region covering all of Germany is requested, while regional trains are tracked in specific sub-regions. After identifying the region, the detailed information is returned. The reply contains the full schedule of the train including all intermediate sta-

<sup>4</sup> <https://developers.google.com/android/reference/com/google/android/gms/location/package-summary> accessed 2017-01-11

<sup>5</sup> <https://www.bahn.de/p/view/service/auskunft/zugradar.shtml> accessed 2017-01-11

tions (also without halt), arrival and departure times. Additional fields indicate the actual arrival and departure dates. Besides these comparatively coarse locations, also a list of more detailed coordinates can be obtained using another ID from this response. This list contains the ordered list of coordinates in millionth of a degree. Thus, these locations can easily be converted to WGS 84 coordinates as used within GPS and any modern end user operated system.

The location as returned by the Zugradar API is derived from track occupancy information. Generally, the occupancy of a rail track is managed in segments of a minimum size of 500 m. This corresponds to the maximum length of trains supported by the system. This track occupancy information is used to control track switches and signals as well as safety systems automatically stopping a train if signals are missed. Based on this information, the locations of trains are derived. As the same information is used for both control and safety functions, the data available on the API is expected to be accurate. Still, as the absolute accuracy is not known, also the GPS position is recorded for all measurements.

First measurements show a maximum update frequency of the data available from the Zugradar API of 1 /min. Compared to GPS with a time resolution of 1 s, the resulting location accuracy is also predicted to be low. Still, based on the known track, the dimension of the error is reduced from two to one. Interpolating the location based on the time between location samples, a more precise location can be determined. A further advantage of using the Zugradar API is the reduced energy consumption compared to GPS. For each location update, one network interaction is required. Caused by the low update interval, considerable energy savings are possible.

The accuracy of the network measurements is maximized by keeping as many variables constant during the measurement. Where this is not possible, the variables are recorded to later discriminate between different situations. As described before, the same route is used for all measurements. Still, the type of train running on a particular day and time may be different. These may be single deck trains (DB Silberling<sup>6</sup>) or the newer double-deck coaches<sup>7</sup>. Further, on double-deck coaches, the position on the upper or lower deck is recorded. To simplify data collection, the user interface (UI) of the measurement application is extended to request the corresponding information from the experimenter. This information is stored besides the metrics recorded by the NetworkCoverage App as described in Section 5.1.

The measurement intervals were reduced compared to the published NetworkCoverage App. The ping interval is set to 10 s. This is the shortest feasible interval, considering that always five probe requests are sent, and may be lost due to intermittent connectivity. Hence, for each packet it must be possible to time-out. The throughput measurements were configured to run every 300 s. This interval is limited by the monthly traffic allowance on the available SIM cards. Care was taken to stay within the limits of the con-

---

<sup>6</sup> <https://en.wikipedia.org/wiki/Silberling> accessed 2017-01-11

<sup>7</sup> [https://en.wikipedia.org/wiki/Bombardier\\_Double-deck\\_Coach](https://en.wikipedia.org/wiki/Bombardier_Double-deck_Coach) accessed 2017-01-11

tract, such not distorting the collected measurements. As before, signal strength, network technology, associated information, as well as GPS location are recorded with maximum resolution. The only further limitation on these measurements is the energy consumption, which is not problematic on a 25 min train ride.

### 5.2.2 *Description of the Measurements*

The data presented in the following was collected on 59 individual train rides at different days and times. The collected data set consists of 29 607 signal strength samples, 6834 RTT measurements, and 514 throughput measurements. The majority of observed samples are collected in 3G and 4G networks. Each sample is annotated with time, location, and ride number, from which additional information on the specific ride (e.g. train type, seat location, etc.) can be derived. The distribution of samples over the network technologies is: EDGE: 1099, UMTS: 857, HSDPA: 11, LTE: 15053, HSPA+: 12407.

The accuracy of the collected GPS locations is analyzed based on the location accuracy as returned by the Android OS. The accuracy value defines a circle with a given radius around the returned position, which contains the actual position with a probability of 68%. The CDF of the recorded accuracy values is given in Figure 5.5a. Accuracies as low as 3 m have been observed. Still, the median accuracy is 13 m. 90% of the samples are more accurate than 22 m, while 99% are below 40 m. Thus, generally the accuracy of GPS on the train is sufficient for the measurement task.

The Zugradar locations are assessed based on the simultaneously recorded GPS values, due to lack of other, more accurate localization techniques. Hence, for each Zugradar location as recorded by the application, the closest GPS position in time is selected as reference. The differences between Zugradar and GPS locations are summarized in Figure 5.5b. Here, considerably larger differences are observed. The median difference between Zugradar and GPS location is 1000 m, with 19% of the samples showing deviations of more than 2 km.

Considering the large deviation between GPS location and Zugradar, the error inherent in the GPS location does not affect the validity of the presented results. Further, the large difference between both locations renders the use of the Zugradar API irrelevant for both measuring and predicting the cellular network quality. Knowing the location with an accuracy of 2 km on the selected route relates to a travel time of 1 min to 2 min. During this time, multiple changes in network performance may occur, thus rendering the location estimate superfluous. Consequently, the following evaluation is based on the GPS values recorded in combination with the performance measurements.

The collected data is first analyzed for anomalies. Therefore, the influence of different parameters on the measurements is analyzed. For this, variables were plotted for each of the parameters as identified before, and cross-correlated to other variables. The correla-



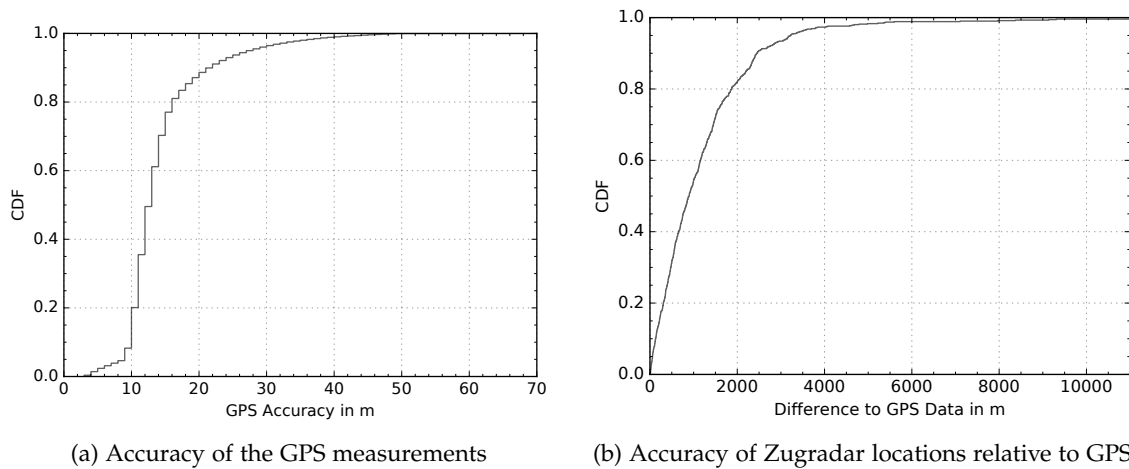


Figure 5.5: Comparison of the accuracy of GPS and Zugradar localization

tions and trends identified correlate well with results published in literature (e.g. [SSM13; LSR+12]). Hence, only the new observations are summarized here.

The distribution of observed network technologies on the route is given in Figure 5.6. For visualization and analysis purposes, the collected data was aggregated in segments of 500 m length. This results in a relatively accurate time and location estimate as well as sufficient data in each bin for a statistical analysis. Over the full route, clearly LTE and HSPA+ dominate. Still, in some areas the handset switches back to UMTS, EDGE, or loses connectivity entirely.

Figure 5.7 shows the aggregated signal strength measurements for LTE and HSPA+. Besides filtering by train type and deck, Figure 5.7a shows the signal strength of the HSPA+ network, while Figure 5.7b shows the LTE signal strength on the same track. Here, only these two technologies are discussed, as these cover the majority (93%) of the observed samples. The lines as indicated in the legend mark the mean observed signal strength of the respective network technology. The shaded areas in the respective color indicate the area between the 5<sup>th</sup> and 95<sup>th</sup> percentile, thus representing the range in which 90% of the samples have been observed.

It is interesting to observe that the received signal power is considerably higher in HSPA+ compared to LTE. This is caused by the different measurement method. While on 3G connections the RSSI is calculated based on the received signal power, in LTE the RSRP is used to judge the channel quality. Hence, no direct comparison between technologies based on the recorded signal strength is thus possible. Because measurements are collected on smartphones, no information on the used frequencies is available. Hence, no conclusion can be drawn from this. Peaks in both networks are observed within the same bins, indicating that the same cell sites are used for the deployment of both technolo-

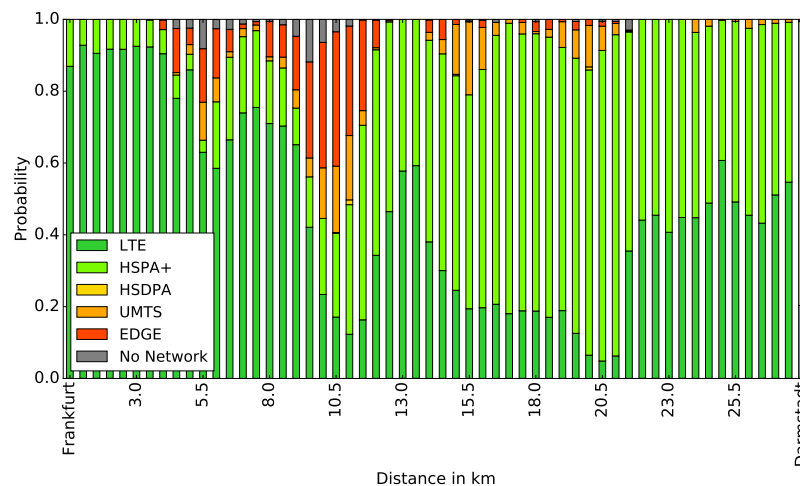


Figure 5.6: Distribution of observed on the route, aggregated for 500 m route segments (from [Fis16])

gies. Hence, also the dips in signal strength overlap in both networks, leading to service outages.

Considering the confidence intervals of the signal strength measurements, any significant influence of train type and deck can be dismissed. The mean as observed in each bin is slightly different, but the confidence intervals overlap in large regions. For the lower signal strength on the upper level in Figure 5.7b between kilometer 13 and 20 no explanation was found. It is counter-intuitive that the received signal power at a higher position is lower compared to other locations. The only possibility would be destructive interference caused by the surrounding environment for this particular height.

The analysis of the RTT performance is based on 7197 samples (LTE: 3827, HSPA+: 2976, HSDPA: 2, UMTS: 171, EDGE: 221). Generally, the RTT measurements confirm results as published in related work [LSR+12]. The median RTT of HSPA+ is at around 100 ms with the inner 50 % of values between 80 ms and 300 ms. The movement velocity and often insufficient signal strength cause frequent packet loss and thus lower layer retransmissions, resulting in a large number of outliers in the range of 1 s to 10 s. The LTE measurements show a similar picture. The median RTT is 50 ms with the inner 50 % of measurements covering the range 40 ms to 90 ms. Also on LTE a large number of outliers is detected. No linear correlation between RTT and signal strength is identified. However, RTT and packet loss depend on the location on the route.

As currently most smartphones are equipped with an LTE capable modem, the following analysis of the RTT measurements combines the performance of all available network technologies as selected by the mobile device. Thus, the presented data closely reflects the performance as experienced by end users on the respective route.

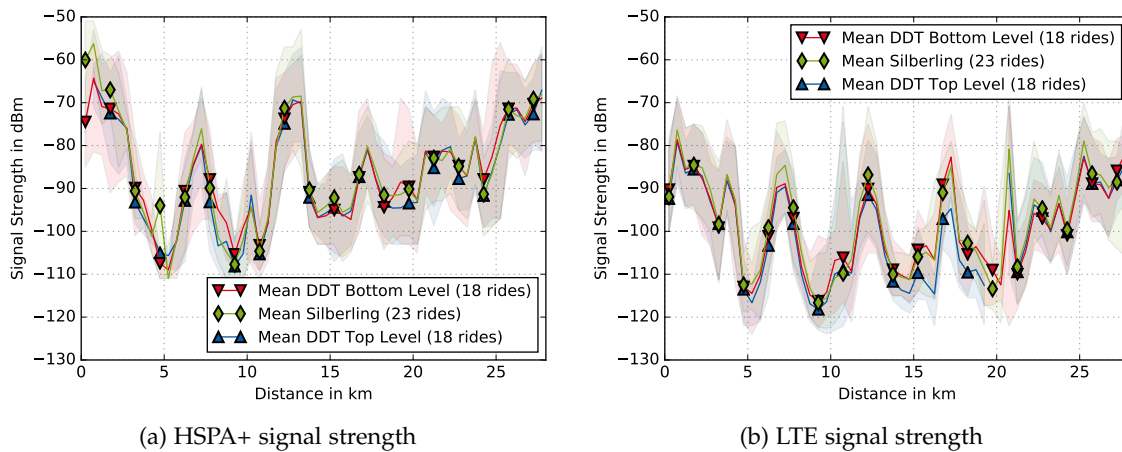


Figure 5.7: Comparison of the signal strength of HSPA+ and LTE on the route between Frankfurt (0 km) and Darmstadt (28 km) for different train types and seat positions

The ping measurements are summarized in Figure 5.8. The blue graph indicates the median RTT, while the dashed blue line describes the mean RTT in each bin. The blue shaded area relates the region where the inner 90% of measurements were observed. Similar to Figure 5.7, lower performance is observed at the same locations. The red line indicates the mean return rate of the sent probes. Here, the shaded area indicates the 95% confidence interval of the mean.

The packet loss as observed in some sections of the route appears negligible. However, considering that replies are sometimes received a few seconds later, a high impact on QoE is evident. Further, during service outages it was not possible to initiate an RTT measurement, thus the real-world impact is higher than estimated here.

The impact of the observed packet loss on any mobile service is much more grave. Considering that a web site currently requests tens to hundreds of different resources, and is usually only fully responsive after all resources are loaded, the impact of a single packet loss becomes evident. By either stalling a TCP session for considerable time, or even interrupting the session, a single packet loss can prevent a website from being fully loaded. Contrary, the large delay, and stalling of the page loading increases load on the network by causing the user to re-request the page. In both cases (e.g. automatic retransmissions and reloading) the demand on the network is increased without benefit to the end user.

Due to the adverse conditions on the route, only a small number of throughput measurements are available. This is both caused by frequent packet losses, but also the data requirements of the tests. Due to packet losses, a large number of throughput tests failed. Furthermore, the limited traffic budget restricts the maximum number of throughput tests to not exceed the traffic cap. If unlimited traffic was available, a continuous measurement

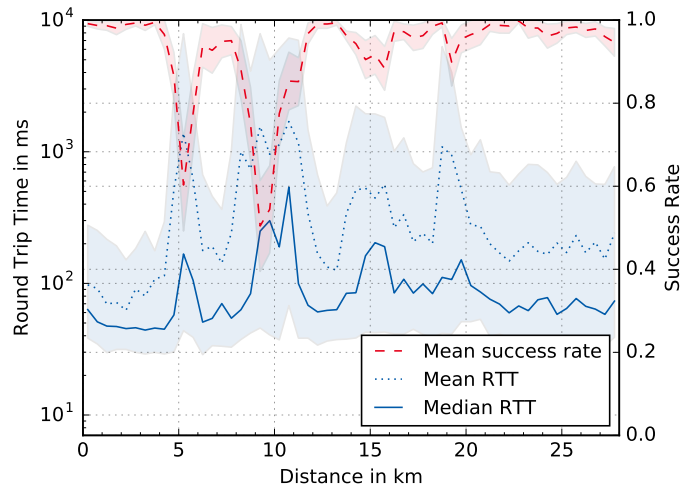


Figure 5.8: Measured RTT and success probability on the route between Frankfurt (0 km) and Darmstadt (28 km)

approach would have been used for a more accurate view on network performance. Still, 495 measurements are available, but not equally distributed over all sections, thus preventing a detailed statistical analysis. Generally, the throughput measurements, similar to the RTT measurements, show no linear correlation with signal strength or RTT. The performance of LTE was considerably higher than evolved HSPA (HSPA+).

### 5.2.3 Predicting the Cellular Service Quality on Trains

As already identified in the previous section, the success probability and RTT of the cellular connection have a significant influence on the user perceived service quality. Hence, a network quality metric is devised, based on transmission success probability and available network type. This is correlated with the subjective cellular service quality to derive thresholds for different services. Based on the quality metric, the subjective network performance and suggested possible service can be derived.

The quality metric is defined based on the success probability of packet transmissions, and the network performance parameters, which are defined by the available network. Based on the success probability, the reliability of the mobile connection is determined, while the network type identifies which services work in an acceptable manner. Thus, the quality metric is defined as a weighted sum of both.

$$Q = q_{\text{nettype}} \cdot w_{\text{nettype}} + p_{\text{success}} \cdot w_{\text{success}} \quad (5.1)$$

Here,  $q_{\text{nettype}}$  is a metric in the range 0 to 1 for the network quality, which depends on the currently available network type.  $p_{\text{success}}$  is the success probability of data transfers

Table 5.2: Network technology mapping used to derive the network quality metric

$q_{\text{nettype}}$	LTE	HSPA+	HSDPA	UMTS	EDGE	none
Value	1	0.8	0.6	0.4	0.2	0

Table 5.3: Classification of the subjective cellular service quality

Quality class	Description	Typical use case	Q interval
$C_1$	Excellent	Video streaming	$1.0 \geq Q > 0.96$
$C_2$	Good	Web browsing	$0.96 \geq Q > 0.87$
$C_3$	Poor	Chat & messaging	$0.87 \geq Q > 0.7$
$C_4$	N/A	–	$0.7 \geq Q$

in the respective section, derived from the success probability of the RTT measurements. Both metrics can be weighted according to user preferences, thus letting the user indicate his or her preference of reliability or performance of the cellular network. The sum of the weights  $w_{\text{nettype}}$  and  $w_{\text{success}}$  per definition is 1, thus resulting in a quality metric  $Q$  in the range 0 to 1.

The network quality  $q_{\text{nettype}}$  is calculated by mapping the available networks to the range 0 to 1. On the analyzed route five different network types are observed, although mostly HSPA+ and LTE were chosen by the device. These are mapped to the range 0 to 1 in ascending order according to their nominal bandwidth. This is also the inverse order of the observed RTTs on the respective technologies. The resulting mapping of network technology to network quality is summarized in Table 5.2.

The thresholds for mapping the network quality as determined by measurements are mapped to the possible use cases by defining service classes. These are identified by popular usage scenarios of smartphones and their bandwidth and latency requirements. Most demanding clearly is video streaming. The subsequent analysis does not distinguish between different quality levels and resolutions, as this is usually adapted by the mobile video player depending on the available bandwidth. The next, less demanding service class contains web browsing and music streaming. The bandwidth requirements are comparatively low, but reliable data transfers are required periodically. Least demanding are cloud-based messaging applications. These require intermittent connectivity, and handle data transfers and retransmissions in the background without requiring user interaction or interrupting the user experience. The largest effect visible is a delay in message delivery, which may be noticed if the conversation is highly interactive. The derived quality classes are summarized in Table 5.3

The thresholds mapping the service quality  $Q$  to the service classes  $C_i$  are derived by first defining the service classes for each route segment, calculating the quality metric, and

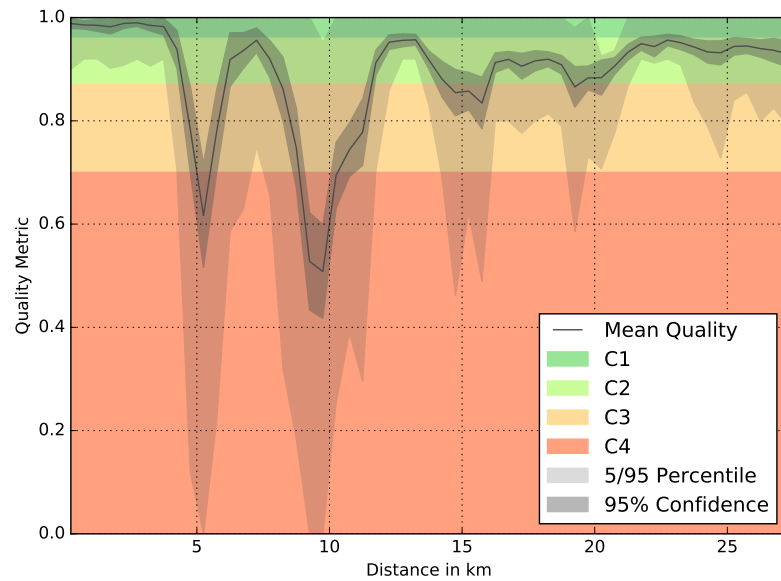


Figure 5.9: Calculated quality metric  $Q$  and subjective network quality

minimizing the resulting classification error over the available rides. The quality metric is calculated according to Equation 5.1. For each route segment, the success probability of the RTT measurements is averaged and weighted with the respective coefficient. Based on the distribution of network technologies, the network performance  $q_{\text{nettype}}$  is calculated and weighted with the weight for the network performance. For simplicity, the weights were chosen to slightly favor reliability over performance (0.6 - 0.4). Further studies (involving more than one participant) are required to analyze the effects of this decision and optimize the final weights.

The cellular service quality was manually classified in a first step, deriving the optimal thresholds to map the quality metric to the defined service classes. For this, the performance of the cellular network was recorded over a number of rides. Based on this preliminary classification, the thresholds for the quality metric  $Q$  are adjusted to result in the respective service classes. The resulting mapping of quality metric to associated service classes is given in Figure 5.9. The black line indicates the derived mean service quality as calculated based on Equation 5.1. Here, a similar behavior as in Figure 5.8 is visible. For reference, the 95% confidence of the mean is given in the darker shaded area around the mean. The lighter shaded area indicates the 5<sup>th</sup> and 95<sup>th</sup> percentile of the observed values. The colored areas indicate the service classes as also defined in Table 5.3. From this it can be seen that particularly sections with lower mean service quality still can show a performance between perfect and not working at all.

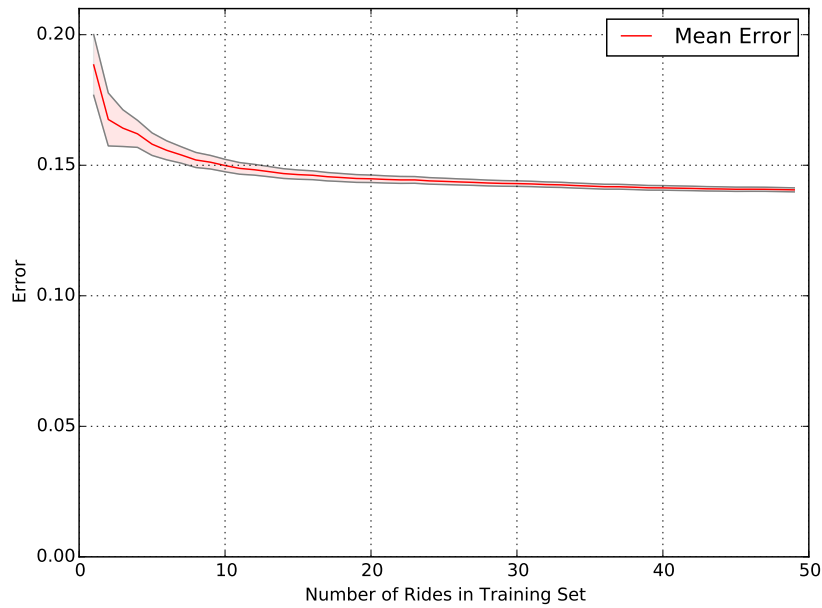


Figure 5.10: Residual error over the number of rides in the learning set (from [Fis16])

The overall accuracy of the prediction is assessed by running a  $k$ -fold cross validation on the collected data. Here,  $k$  is usually chosen as 10. As this leaves only 5 samples in each group,  $k$  is increased to 11, resulting in a larger number of reference groups of the same size. The resulting RMSE is 6.72%.

The required number of rides and resulting accuracy of predicting service quality classes are analyzed in the following. These calculations are conducted under the assumption that the measurements are executed as described in Section 5.2.2 (e.g. continuous location, signal strength and network technology updates, RTT every 10 s). The number of rides is determined by randomly sampling  $n$  out of the 59 rides, where  $n$  is increased from 1 to 49, thus leaving 10 rides as reference. Statistical significance is assured by repeating this process 30 times. Thus, the mean residual error as well as the confidence of the mean can be calculated. Figure 5.10 shows the derived dependency between number of rides and residual error. The RMSE for a single ride is 0.18. For 10 rides the error reduces to 0.15 and approximates 0.13 for a larger number of rides.

Considering the narrow range of the derived quality metric for the mapping to the service class, classification errors particularly in the higher service classes must be expected. This is caused by the small number of samples which can be captured on a single ride. As already visible in Figure 5.8 the RTT may vary considerably within a section. Thus, a larger number of rides would be required to increase the accuracy of the model. Still, based on the data from a single ride, an already reasonable estimate of the service quality and thus the service classes can be expected.

This initial study already shows good results, but clearly a more thorough approach involving a user study and a larger number of participants is required to derive a conclusive mapping. Still, the presented work shows a feasible approach to derive this mapping. Considering a larger deployment of the presented methodology, an interactive end-user application occasionally requesting feedback from end users would be a feasible approach to improve both service classification as well as the mapping of network performance metrics to service classes. Ideally, this may be included in another similarly themed App, or directly into the DB Navigator<sup>8</sup>. However, what is currently missing to make this approach viable is an energy efficient and accurate localization method.

Assuming a suitable localization technique was available, users may be motivated to participate in occasional network measurements if provided with a useful service. In the case of cellular network usage on trains, the direct benefit for the model may be user feedback according to the predicted network quality on future stretches of the journey. The benefit for the end user would be to know which services are expected to work during which phases of the journey. This can either be achieved by implementing a permanent notification in the status bar of the mobile phone, or a dedicated activity within the mobile application giving a more detailed insight into the current and predicted network performance.

During the course of this study, both user feedback mechanisms were implemented to check their usability and benefit. For the permanent notification it was decided to visualize the current status of the network similar to the signal strength graph using filled bars. When opening the notification, the verbal description of the current service class and an exemplary use case as given in Table 5.3 is displayed. Further, the remaining time to the next predicted quality class change and predicted quality class are indicated. An example of such a notification is given in Figure 5.11a. The derived service quality map for the selected route is shown in Figure 5.11b. Beginning from the departure station the predicted network quality is indicated in the left graph, with a legend to the right describing the quality classes and a usage scenario. The current location on the route is indicated by graying out the already passed route segments. The time scale on the left of the graph lets the user estimate the time within each service class, and thus plan how to spend the time on the train most efficiently. This feedback is expected to greatly improve user experience on trains by first communicating the current quality of the network and thus avoiding disappointments and failed connection attempts to use the network. Thus, the users may spend their time on trains more efficiently by planning their tasks accordingly. As a side effect, the energy consumption of the cellular modem can be reduced by not activating it during periods of poor coverage. On the other side, knowing the route segments showing poor coverage, mobile operators can expand their network accordingly, thus gradually eliminating the need to plan ahead.

---

<sup>8</sup> <https://play.google.com/store/apps/details?id=de.hafas.android.db> accessed 2017-01-14



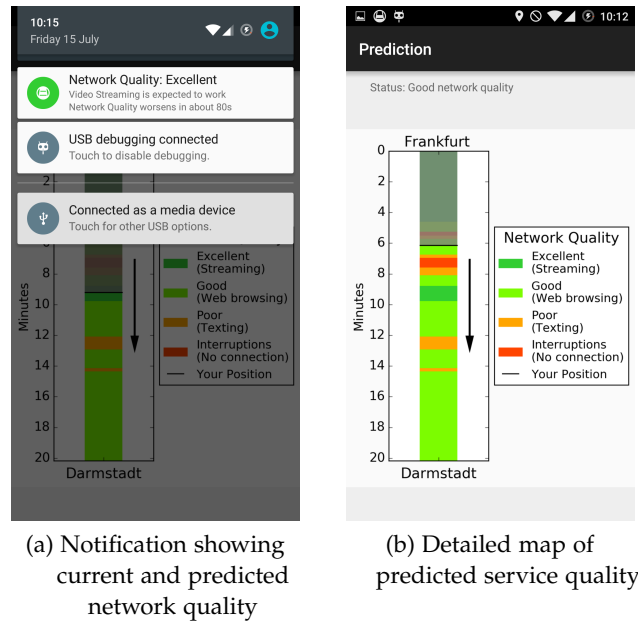


Figure 5.11: Examples of the user feedback implemented in the measurement application (from [Fis16])

#### 5.2.4 Summary and Conclusion

The presented work shows the feasibility of measuring and predicting the service quality on trains. First, the location accuracy of the DB Zugradar service is analyzed. The resulting error is determined to be too large to be useful for measuring or predicting the cellular service quality. Based on extensive measurements, the influence of seat position and train type on cellular network quality is rejected. Further, a quality metric for the cellular network was derived, based on the available network type and RTT. This quality metric is mapped to service classes relating to different usage scenarios like video streaming, web browsing, and messaging. These reflect the actual perceived service quality quite closely, although more extensive user studies are required. The analysis over all measured rides results in an error of 6.72%. The modeling of the service quality of a given route is determined to be possible with as few as 10 rides. Still, the major impediment for widely implementing and using this approach are the extensive energy consumption of GPS localization or the inaccuracy of the Zugradar system. Assuming a suitable localization technique was available, a highly beneficial service for commuters and travelers could be implemented.

### 5.3 INFLUENCE OF TRAFFIC MANAGEMENT DECISIONS ON NETWORK PERFORMANCE

The differences in RTT as identified in Section 5.1 are analyzed in detail by running additional, dedicated measurement studies. Therefore, the route of the packets taken through the network is analyzed by running traceroute measurements from the mobile device, as well as traceroute measurements in the opposite direction, from the measurement server towards the mobile device. Further, the location and response times of the DNS server are identified. As further reference, also measurements of popular websites are conducted, giving insight into the real-world implications of the observed effects.

The following section is based on work published in [KMB+15] and [KMB+16]. The section is structured as follows: First, the extended measurement setup is described in Section 5.3.1. The recorded network metrics are presented and discussed (cf. Section 5.3.2), from which conclusions on network structure and service placement are drawn (cf. Section 5.3.3).

#### 5.3.1 *Measurement Methodology*

As the crowd-sensing measurements show no significant correlation between location and network performance, the following measurements are executed from a stationary location. Therefore, the measurement application is extended to record, besides the metrics as described in Section 5.1, a traceroute to the measurement server, the DNS server used, and the RTT to the operator configured DNS server. All measurements are triggered periodically to measure the network performance during different times of the day and possibly identify weekly patterns. The study was run over the course of 4 weeks. The study duration was chosen to include two normal working weeks and as well as the Christmas break to identify any abnormalities in the network performance. The study began on December 14, 2015 and ended on January 15, 2016.

To maximize the number of measurements per cell, and eliminate as many variables as possible from the measurement setup, the devices are placed at fixed locations within the city. Thus, the probability of being connected to the same cell for a considerable amount of time is higher. Still, network initiated handover cannot fully be eliminated, as no option is available on Android locking the device to a single cell. Selected measurement locations are two office environments, a location close to the central park, and three residential areas. Thus, the influence of different traffic patterns within the cells on the measured network performance can be identified.

As only end-to-end measurements are feasible without the participation of the cellular network operator, the traversed network is considered to be a black box. Still, some information can be deduced. Running a traceroute measurement from the mobile device to the server, the path on the Internet backbone is visible. For reference, a reverse traceroute service is implemented on the measurement server. Mobile devices connect to a specific

port and leave their device identification. After that, the connection is terminated by the server. Based on the public IP address of the connecting device, a traceroute measurement back to the device is started. Thus, both participating interfaces of the intermediate nodes can be determined, resulting in a more detailed view of the network.

As further reference the response times of the 25 most popular websites are recorded. Thus, a more detailed insight into the performance of the cellular network compared to the wired backbone can be derived. The most popular websites are taken from the Alexa most popular domains list<sup>9</sup>. Each of these top 25 domains is first resolved using the operator provided DNS server, thus ensuring the association to the 'optimal' remote server. The resolved address is stored for the following measurements and later analysis. First, an RTT measurement to the resolved IP is run. Then, the main hypertext markup language (HTML) document is loaded from the respective server. For each, the connection time, time to first byte (TTFB), and time to finish (TTF) are recorded. According to Li et al. [LMW+15], both native Ping and HTTP connect show comparable performance. Hence, these serve as additional reference for the native RTT measurements. Becker et al. [BRF14] describe that HTTP requests in cellular networks are generally intercepted, and an ACK returned before the original server has replied. This is caused by PEPs in the mobile core network, intercepting all TCP connections. The presence of a PEP is confirmed running Netalyzer, which is described in [VSK15]. Hence, the connection time can serve as another indication for the existence of a PEP. As these are usually located close to the exit gateways of the cellular network provider, the connection time may also serve as additional demarcation point of the cellular operator's network domain. The first realistic value for the distance between the mobile device and the remote server is the TTFB, where the first actual data is received. Still, depending on the request type (e.g. HTTP/HTTPS), this delay corresponds to 3 to 4 RTT, minus one RTT between PEP and remote server.

### 5.3.2 Measurements

The data collected by the traceroute measurements to and from a dedicated server, as well as the response and load times of the 25 most popular websites is described in the following. First, the traceroute measurements are evaluated with the focus of identifying possible root causes of the anomalies observed in the crowd-sensed data. Then, implications on the real-world performance of end-user requests are assessed on the example of the page load time of popular websites.

From the traceroute and RTT measurements a number of interesting observations can be drawn. First, the performance of the individual devices changes over time. The observed time interval is 36 h and repeats over the course of the study on all devices. Contrary to first assumptions, the cellular parameters like signal strength and associated cell ID, LAC

---

<sup>9</sup> <http://www.alexa.com/topsites/countries/DE> [accessed 2015-10-06]

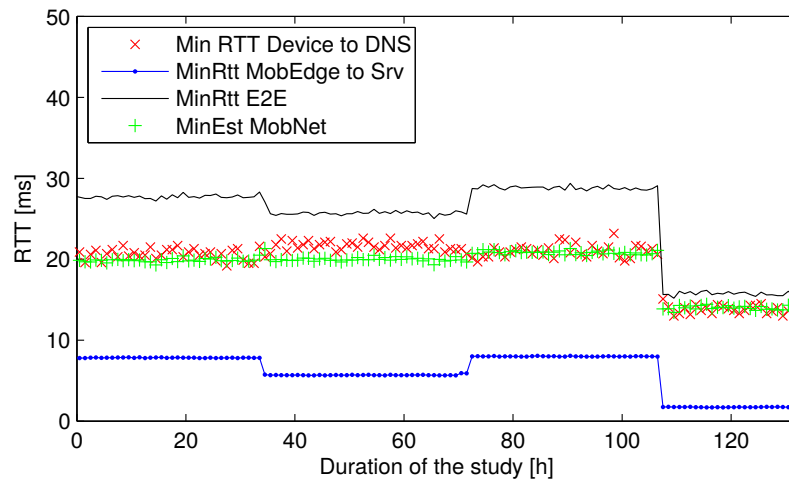


Figure 5.12: RTTs between mobile device and different demarcation points in the network

and network technology do not change during these changes. Hence, the root cause of this abnormal behavior must be caused by the upper layers of network connectivity.

Figure 5.12 shows the measured RTTs between a single device and the measurement server. The black line indicates the RTT between device and measurement server over the course of one week. As only minimum, mean, maximum, and median deviation of the RTT measurements are available, the minimum RTT is selected as the closest approximation of distance within the network. Larger delays are possible, but are expected to be caused by processing delays on the intermediate nodes, in particular as ICMP messages are handled with low priority. The blue line indicates the minimum RTT between the measurement server and the last hop replying to the ping messages. Again, the minimum of the measurements is chosen, approximating the best case performance. Calculating the difference between end-to-end RTT and server to last-hop RTT, the RTT as indicated by the green plus signs is derived. This serves as an estimate of the RTT within the cellular network, consisting of wireless and mobile backbone propagation delays. For reference, also the RTT to the operator configured DNS server is given in the form of red crosses. Its performance closely correlates with derived RTT within the cellular network, thus strengthening this argument.

The operator configured DNS server is always the same multicast address. The changing response times thus indicate that still different servers are used. Considering the structure of the cellular network, where tunnels are established between the mobile device and the SGW and PGW, this indicates that these are periodically changed.

Resolving the IP addresses of the intermediate nodes as returned by the traceroute measurements, conclusions on the location of the PGWs can be drawn. Based on the hostnames, three different locations in the core network are determined, receiving traffic

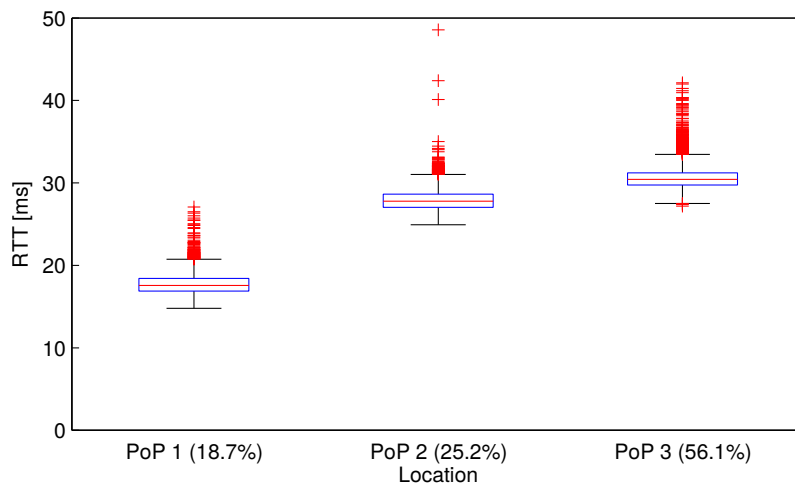


Figure 5.13: End-to-end RTTs as measured by the mobile devices, split by PoP as derived from the traceroute measurements

from the mobile operator. As the cellular network is completely opaque to traceroute measurements, the last public location of the reverse traceroutes is taken as demarcation between both networks, and in the following called PoP.

The derived PoPs are geographically distributed over Germany. This correlates well with the RTTs as observed between mobile device and DNS server and the ones derived from end-to-end measurements and reverse traceroutes. Splitting the measured end-to-end RTTs as measured on all devices by the derived PoP the performance implications of PoP assignment on end-to-end performance can be identified. Figure 5.13 shows the aggregated RTT of all devices using different PoPs. The narrow range of the boxes indicates a good fit. Thus, knowing the PoP explains the difference in end-to-end RTT well.

Figure 5.14 shows the derived network structure. All devices as well as the measurement server are placed in Darmstadt, thus are comparatively close to each other. The mobile devices connect to the cellular network and establish a connection via the SGW and PGW assigned by the network. These are geographically distributed all over Germany, as is derived from the host names. Comparing the geographical distance covered, a relative velocity  $v_0$  of 1% to 7% of the speed of light is observed. Considering common fiber deployments as discussed in [SCG+14] ( $2/3c_0$ , fiber length as  $2x$  geographical distance) a one-way distance of 300 km explains 6 ms of the 30 ms observed on this distance. The remaining time must be accounted to routing, MAC mechanisms, and device internal delays. A shorter distance of 30 km explains 0.5 ms of RTT, thus leaving 17 ms for routing, MAC and processing on the device. Assuming the device internal delays are similar to WiFi, approximately 7 ms are consumed by the smartphone [LMW+15]. Considering a medium access delay of 1 ms on both directions, still leaves 8 ms to 16 ms unexplained.

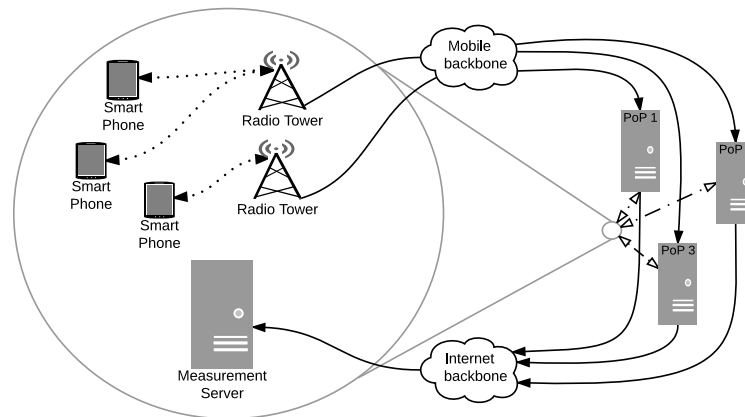


Figure 5.14: Derived structure of the cellular network

An interesting observation considering the PoP selection is the unequal distribution of time spent using the different PoPs. Only 18.7% of the time the most optimal PoP was used, while for 25.5% and 56.1% the second best respectively worst PoP was used. The behavior as observed is apparently caused by the network configuration, where always a random SGW and PGW are assigned to connecting devices. Hence it must be assumed that the association probabilities correlate to the capacities of the SGWs. Other possibilities are that just default parameters were used when setting up the network, or other costs (e.g. traffic) at another PoP are lower.

Considering the large increase in RTT, also the cellular backbone is used proportionally. Clearly, optical links have a high bandwidth compared to the cellular network, but as the cellular network technologies progress, also this factor becomes increasingly important, and as shown here, also measurable. Comparing these RTTs and calculating the relative performance penalty compared to the optimal PoP, the values in Table 5.4 are derived. These indicate the overhead in the different network domains. While the overall RTT increase for PoP 2 and PoP 3 is 58% and 73% respectively, the relative increase on the core network is considerably larger. While the absolute increase in RTT from 3 ms to 8 ms (cf. Figure 5.12) is comparatively low, the relative increase is considerable. Still, comparing this to the RTT increase in the mobile network (41% to 47%), also there considerable improvements are possible.

The real-world implication of the PoP assignment is evaluated by measuring the response times of the 25 most popular websites. Therefore, the connection time, TTFB, and TTF are measured. The results of the individual websites as received via the indicated PoPs are given in Figure 5.15. Contrary to the first measurements, here four different PoPs are discovered. It is interesting to observe that large differences in TTFB between different websites exist. This is expected to be caused by the server backend. Mostly static pages like `www.build.de` load comparatively fast, while more dynamic pages (e.g. `www.ebay.de`) take considerably longer to send the first data bytes.

Table 5.4: Possible reduction in RTT in the different domains compared to the optimal PoP. Here, the column *I-Net* denotes the RTT between server and mobile edge, *MobOp* the estimated RTT within the mobile operator's network, and *E2E* the end-to-end RTT.

	Median RTT overhead			Percentage of packets
	I-net	MobOp	E2E	
PoP 1	0	0	0	18.7%
PoP 2	247%	41%	58%	25.2%
PoP 3	365%	47%	73%	56.1%

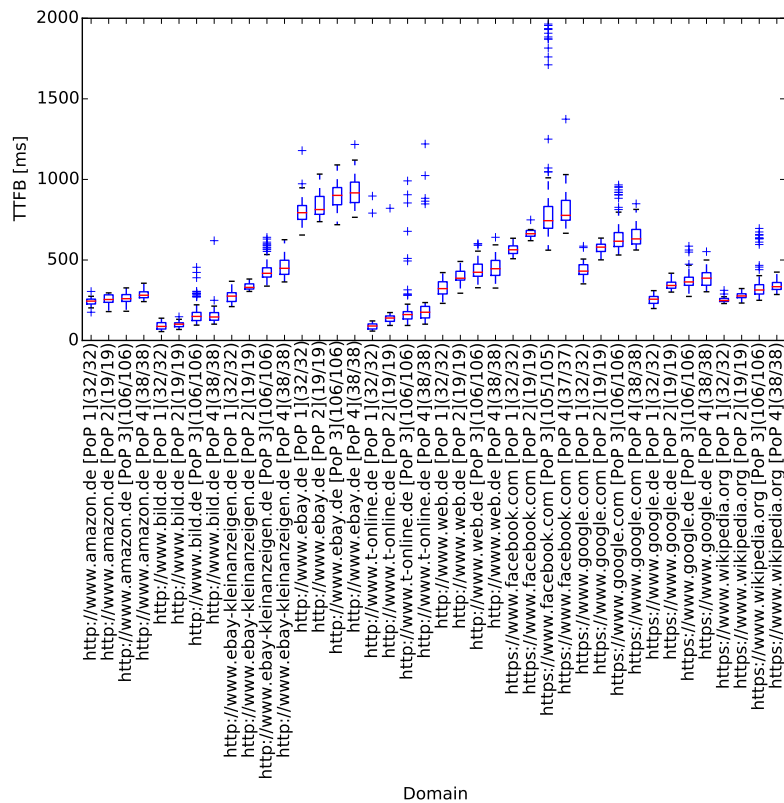


Figure 5.15: Time to first byte for the 10 most popular domains as received via different PoPs (Popularity determined on 2015-10-07)

The independence of the collected data for each website and PoP is verified using the Kruskal-Wallis test. This test is selected, as an analysis of variance (ANOVA) is not applicable for comparing data sets with non-Gaussian distribution. Further, the input data must be independent, and the distribution similar between groups. Independence of each sample is assured by the test setup. Tests are run every 30 min, which is much longer than most connection timeouts. No cache is used on the local devices, and the DNS resolution is freshly requested before each run. The remaining locations where state may be stored are the internals of the cellular network. As identified before, the association of the mobile device changes every 36 h. This is reflected in the selected PoP. Hence, the PoP is selected as dependent variable, distinguishing the different groups. Similarity of the observed distributions is assessed by comparing the histograms of the performance of each PoP for the different websites. These show a good fit. Thus, the prerequisites of the Kruskal-Wallis test are fulfilled.

The null hypothesis  $H_0$  is that the tested distributions are equal. For this, usually a confidence threshold of 0.05 is chosen, relating to a probability of 95 % of both groups not being drawn from the same distribution. Running the test on the collected data generally shows  $p$  values of lower than  $5 \cdot 10^{-5}$ , thus confirming a significant difference between those data sets, and hence performance implications caused by PoP selection.

The impact of PoP selection is compared by normalizing the performance of each website by the median TTFB via the best performing PoP, which in all cases was PoP 1. The normalization is required, as the performance of the different websites is too different to compare absolute numbers. The resulting performance penalties of the sub-optimal PoP selection are summarized in Figure 5.16. Figure 5.16a shows the increase in TTFB, which is an important indicator for the performance of short-lived flows and generally connections exchanging only little data, but requiring a frequent exchange of data between client and server. Generally, the observations made confirm the ones from the RTT analysis. Compared to these, the median increase is a little lower. The higher variance is caused by using the aggregated values of the selected 25 domains. For the best performing 25 % of sites, the TTFB increase was lower than 30 %, while for the worst 25 %, the penalty of using the worst PoP was 60 % to 120 %. The page load time (cf. Figure 5.16b) generally shows a similar behavior. The main difference is the lower variation in performance. This is expected to be caused by the lower influence of the RTT on the throughput of TCP connections after these are established. Some sites even show similar or better performance when using one of the alternative PoPs. This may be explained by their servers being located closer to one of the alternate PoPs.

Concluding, the measurements of the influence of PoP selection on the real-world performance of cellular networks confirm the results derived using RTT and traceroute measurements. From this, different conclusions can be drawn. First, in cellular networks it is sufficient to measure the RTT to a target server to approximate the load-times as expected



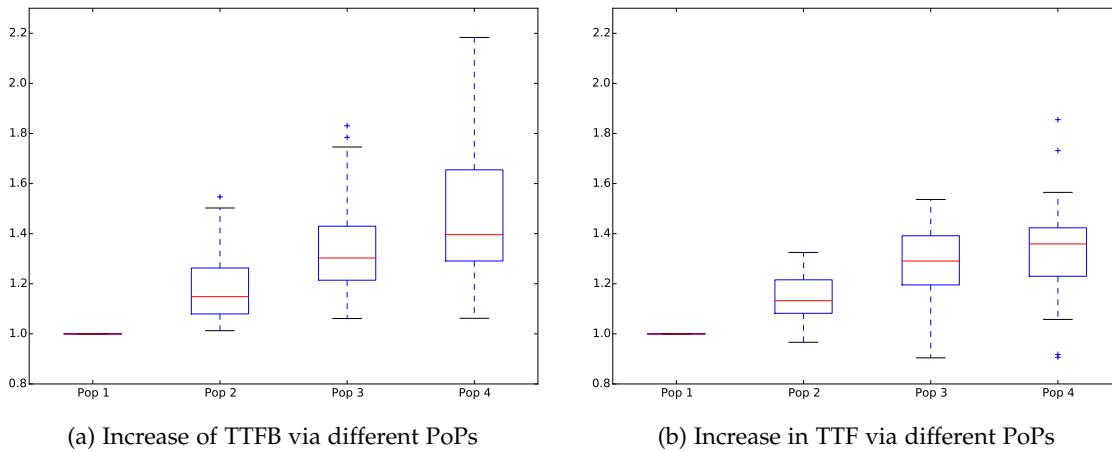


Figure 5.16: Comparison of the TTFB and TTF for different PoPs, normalized by the performance of the best performing PoP (PoP 1)

in an end-user scenario. This assumes that network capacity is not saturated, which was not observed in any of the studies.

Contrary, from these observations it can be concluded that by now RTT is the defining factor of cellular network performance. Considering the heavy increase in RTT, TTFB, and TTF depending on the allocated PoP, substantial performance increases can be achieved by optimizing the PoP allocation. This is expected to be possible in an economically feasible way by improving allocation algorithms, thus not requiring hardware upgrades to greatly increase end-to-end performance while simultaneously reducing traffic in the MNO's network.

### 5.3.3 Summary and Conclusion

Generally, from the above observations it can be concluded that PoPs closer to the end user are beneficial in most cases. Extrapolating this would require placing PoPs at more decentralized locations. This closely agrees with future visions of mobile communication networks, where latencies of lower than 1 ms are targeted [Qua13]. Clearly, this is not feasible with the current architecture. Considering the propagation speed of electromagnetic waves in wires (optical and electrical) of approximately  $2/3c_0$ , where  $c_0$  is the speed of light, a maximum distance of 200 km can be covered within 1 ms. Considering a round-trip, the distance is reduced to 100 km. Allowing some time for processing, a realistic distance between end user and server hence is more likely to be around 50 km. For this to be possible, the PoPs of the cellular network must be considerably closer to the end user than they are today. Finally, this would mean to placing one PoP in each city.

The possible improvements in RTT are of importance when considering the goal of reducing the latency to values below 1 ms, and thus making the Internet tactile [ITU14]. In particular when considering QoS requirements of interactive services like video conferencing or live cloud gaming, these can thus be satisfied. But also e-commerce websites benefit from reduced end-to-end latency, increasing sales by improving the click-through rates [KH02]. By already assigning the connecting mobile users to the optimum PoP, the perceived latency can be reduced by over 58 % in more than 80 % of the time.

PoP selection strategies in the simplest case consist of assigning the user to the closest PoP, such minimizing the distance between end-user device and PoP. Other strategies may include frequently used services and their locations, or also operator-internal metrics like power consumption or traffic cost into the decision.

#### 5.4 CONCLUSION ON PERFORMANCE OF CELLULAR NETWORKS

In the previous sections the performance of the cellular network for different usage scenarios is analyzed. These are guided by the research questions posed in the beginning, and are summarized in the following:

*RQ 2.1: What are the parameters affecting cellular service quality and user-perceived network performance when mobile?* Based on an extensive data set it is derived that the cellular service quality is mainly influenced by the available network technology (e.g. 2G/3G/4G). Within each technology class the observed performance is similar, and mostly uncorrelated to signal strength and other KPI. Thus, also the influence of location on network performance is limited as long as the connection technology does not change. Hence, when requiring an estimate of cellular network performance, knowing the available network technologies results in a useful estimate of the anticipated network metrics.

*RQ 2.2: How does cellular network access on trains differ from general mobile network access, and how can the network performance be predicted?* Knowing and predicting the cellular network performance on trains proved to be more challenging than in urban scenarios. The localization of the mobile device adds some difficulties, while simultaneously the network performance varies rapidly, requiring a sufficiently accurate location to assure repeatability and accuracy of the results. The performance of the network is identified to be mainly related to available network technology, and particularly on trains, also to the success probability of packet transmissions. This effect was not visible as such in the crowd-sensed measurement study. The collected measurements are analyzed with focus on predicting the suitability of the cellular network for different usage scenarios, concluding that a small number of rides is sufficient to capture and predict the service quality on a given route.

*RQ 2.3: What is the influence of network structure and management on end-user perceived network performance?* Variations in service quality as identified in the crowd-sensing and train measurements are augmented in a comprehensive manner by means of a stationary

reference study. Devices were located at representative locations for different usage scenarios (e.g. home, office, shopping, and recreation) and the network behavior analyzed using an extended set of measurements. Based on the collected data the main influence of RTT variations was determined to be caused by routing and gateway selection algorithms within the cellular backbone, increasing the end-to-end latency by up to 73% compared to the optimum observed performance.

Summarizing the different studies is derived that the cellular network performance is mainly determined by the available network technologies and their inherent performance. The second largest influence factor on cellular network performance are network management decisions within the cellular operator's network. These define the route taken by any packets through the backbone by attaching end-user devices randomly to SGWs and thus PGWs distributed over their network without considering device location or remote server locations. The observed signal strength has only marginal influence on the measured KPIs.

Combining the above observations for estimating and predicting the cellular service quality for e.g. optimizing services for various usage scenarios, it becomes obvious that mainly the available technology and thus the network upgrade process of the cellular network operator defines end-user performance. Optimizations by delaying traffic for short intervals within the same technology are thus not advisable. Still, if a handover to another technology is expected within a short time frame, larger optimizations are expected to be possible. Considering the current deployment and upgrade process of cellular networks, this optimization appears to be uncommon, in particular in urban scenarios. Still, on trains a fair optimization potential exists due to the frequent and drastic changes of cellular network performance. Caused by the network performance to be more dependent on location than time [YKH08], optimizations are also expected to be location based. This results in another challenge on trains, as localization using GPS is extremely power consuming, the Zugradar service is too inaccurate, and the localization by train run-time not reliable due to frequent delays.

Contrary to optimizations on the end-user device, mobile operators have a large potential of improving the end-user service quality. This ranges from relatively simple optimizations of end user to PoP mapping to upgrading and extending their networks with improved network technologies. The end user to PoP mapping may be conducted considering the location of the device when attaching to the network, and assigning it to the SGW and PGW at the geographically closest PoP, thus reducing RTT by 36% to 42%. Extending the network coverage is particularly promising for mobile data access on trains, as the performance for a large number of users with spare time to consume online services can be improved. Here, also any improvement in signaling efficiency pays off, as a large number of handover are required within a short time period, thus allowing the mobile network operator to provide optimum service to connected end users faster.



## INFLUENCE OF TRAFFIC MANAGEMENT ON ENERGY CONSUMPTION

---

THE power consumption of network infrastructure and the performance of cellular networks are analyzed in Chapters 4 and 5. Both are important aspects of communication networks, in particular in mobile environments. Still, their often contradictory optimization goals hinder simple optimizations. Hence, a trade-off between performance and energy consumption must be found.

One important aspect of mobile data access is the energy consumption of end-user devices. As smartphones are usually running on batteries, the available energy to perform their tasks is limited. Further, these are required to last at least a full day, while being used with changing intensity. The major influence on energy consumption with 31.2% is caused by wireless data interfaces [CDJ+15]. Their implications on battery life time depend on the available network technologies, their data rates, but also the frequency and volume of transferred data. A number of different mobile power saving approaches as presented in literature were discussed in Section 3.3. These generally promise large energy savings observed during test trials. Still, their results are difficult to verify, and their underlying data is usually not available. Hence, the first section of this chapter (cf. Section 6.1) analyses the performance of two selected approaches ([HQM+12; IWF13]) and compares their energy consumption to an unmodified system. This analysis is conducted based on collected user traces in an evaluation environment merging network availability and coverage data as described in Section 5.1, network performance as described in Section 5.3, and mobile power models as derived in Section 4.3. Thus, the energy consumption of the device under realistic conditions is compared to the cost as caused by the optimization approaches.

Also for infrastructure networks a number of optimizations are proposed. Often, these consider performance only, but recently the energy aspect of networking grew more important. Related work shows that already a number of mechanisms was proposed for both the fixed [VNS+11; CMN12; YWX+13] and wireless networks [HBB11; CFG+12; DBM+10]. Here, optimizations regarding the number of active devices, configured link speeds and user-to-cell association are proposed. Still, these are generally based on theoretical models and optimal assumptions concerning network performance. Hence, further research is recommended, including real-world measurements, models and observations into the optimization approaches. Clearly, flexible evaluation environments using more realistic assumptions including multiple domains modeling 5G networks are required to gain an accurate insight into different traffic management approaches.

From these observations, the following research questions are derived, targeting the analysis and optimization of energy consumption and network performance in both fixed and wireless networks, and guiding the subsequent analysis.

RQ 3.1 What is the energy cost of mobile communication for a regular smartphone user, and how is this affected by smartphone-based energy conservation approaches?

RQ 3.2 What is the potential of emerging network technologies on network infrastructure and mobile devices considering performance and energy consumption?

In both cases, the goal is finding the optimal trade-off between both, thus maximizing the QoE of the end user, while minimizing network and device energy consumption. Clearly, a detailed data basis is required to derive accurate results.

Future developments will further improve performance and energy efficiency of individual network technologies. The problem of finding the optimal trade-off becomes apparent when selecting network hardware and structuring the network. This problem will become increasingly complex as new network paradigms like SDN and NFV are introduced, largely increasing the solution space. Considering the requirement of further decentralizing the network to provide sufficient performance as is demanded by the *1000x mobile data challenge* [Qua13], clearly a versatile, but also automated approach a network optimization is needed. The following sections exemplary show how to derive conclusions for optimizations on the mobile handset and give recommendations on designing a comprehensive network optimization approach, also including functionality, performance, and energy consumption of the core network.

## 6.1 ANALYSIS OF THE POWER SAVINGS OF MOBILE SCHEDULING APPROACHES

The energy consumption of mobile handsets is analyzed based on empirical user-traces by combining the network performance data collected in the crowd-sensing study as described in Section 5.1 with the energy models as discussed in Section 4.3. For this analysis, an evaluation environment is implemented, calculating power estimates based on the observed user locations and traffic patterns. Using the location of the device, the available network, and its KPI, the power consumption is derived. Power models are available for the Nexus 5 and the Nexus S. Still, to gain insight into the full network performance including 4G, the energy model of the Nexus 5 is chosen for all traces.

The data sets used in the evaluation are discussed in Section 6.1.1. These include the user traces as well as network availability, RTT, performance and their aggregation for the later evaluation. Section 6.1.2 describes the evaluation framework and its various optimizations. The investigated power saving approaches are discussed in Section 6.1.3, based

on which the evaluation in Section 6.1.4 is derived. Finally, Section 6.1.5 summarizes the findings and gives suggestions on future research opportunities. The following analysis and results are based on a Master's thesis by Thomas Schnabel [Sch16].

#### 6.1.1 Data Sets

The user traces were collected during a user study over the course of two weeks at the beginning of 2016. Primary focus of the study was the analysis of Smartphone usage to develop functions assisting the user in its daily tasks. Secondary goal is the analysis of generated traffic patterns. Instructions to the study participants were to use the phone as they would do anyways. 99 students participated in the study. Still, as must be expected in every user study, some did not provide useful data (e.g. by installing the App on an otherwise unused phone), disabling location services, or not being in the area of interest. Further, due to the comparatively small size of the study, the App could not be tested on all devices on the market, so a few installations show a spotty data collection. Hence, the collected data is thoroughly filtered.

The data is filtered to select data from within Darmstadt containing continuous location and traffic updates over a 24 h interval. Thus, multiple data sets intervals may be recorded by a single user, while their number is different for other users. For an accurate analysis, the locations were filtered to have an accuracy of better than 100 m. Thus, the available network can be accurately mapped to the user locations.

Limiting the evaluation interval to 24 h should not affect the final results, as smartphones are commonly charged once per day, thus providing a full battery charge for every 24 h period. Further reducing the interval increases the size of the available data set, but leaves no clues on daily patterns, their influence on mobile data consumption, and thus on energy consumption.

Finally, data from 14 different users remained. Within these, 62 intervals of 24 h coverage were identified providing continuous location and network samples. These contain between 550 and 7353 data samples, with an average number of 2677 samples. The number of locations is between 3631 and 5015 location samples, relating to an average of 3.2 location samples per minute.

The network performance data as used in the following analysis is based on the data collected in the crowd-sensing study as described in Section 5.1. The data is collected by users of the public NetworkCoverage application. The dataset consists of signal strength, RTT and throughput measurements. The RTT is measured against the fastest of *emancip-slab* server, or the dedicated network measurement server. The throughput tests use the same server. Thus, RTT and throughput measurements can be directly related to each other. The throughput tests use *iperf* in both uplink and downlink direction, limited to a measurement interval of 5 s.

Besides cellular data samples, also data from WiFi networks is available. It consists of signal strength, RTT, and throughput samples. The metrics are recorded by the same functions as are also used for the active measurements on the cellular network. Thus, comparability between the performance measurements is guaranteed.

The majority of samples is collected in and around Darmstadt, thus mandating the limitation of the evaluation this area. Also alternative data sets were considered for the analysis, but no suitable data set or combination of different data sets is available. These network data sets are either limited to one technology only (e.g. WiFi/cellular), do not include performance data like RTT or throughput measurements, or are not publicly available. User traces are equally difficult to find, in particular the required combination of location and network utilization is not commonly found. Furthermore, the requirement of fine granularity of both, permitting the estimation of the resulting energy consumption, cannot be satisfied.

The coverage area of both WiFi and the cellular network are calculated by collecting all samples with the same basic service set identifier (BSSID) or cell ID and calculating the convex hull covering all samples. Still, thorough filtering is required to assure a high quality of the resulting estimate. Hence, samples with a location accuracy of higher than 15 m are removed from the set. Further, the time-series as collected by each device are analyzed, eliminating samples with updated location but unchanged signal strength. This is required, as the Android OS often returns stale coverage samples for cell measurements, or repeated WiFi samples for each new scan, still being based on an older beacon reception. Based on these cleaned samples, the coverage areas of the respective cells are determined. An example of the coverage estimation of a single cell is shown in Figure 6.1. Here, the circles indicate the samples recorded in the respective cell. This is limited to the data recorded while the handset was actually connected to the cell, thus assuring that the cell would also be selected by a mobile device at the respective location. The red/dark circles are eliminated from the data set, leaving the green/light samples, used for the estimation of the actual cell coverage.

As only a limited number of RTT and throughput measurements are can be run, the KPIs of each cell are determined based on aggregated data from other cells. This is required, because active measurements, in particular throughput measurements, cause a considerable traffic consumption, and the device must connect to the respective network to allow its measurement. Thus, in the case of the cellular network, the performance is aggregated over all samples observed when connected to the respective technology. This is a valid approach, as performance within a 4G network is largely independent of the observed signal strength, as long as the signal strength is above 30 % [SMS<sub>13</sub>].

On WiFi, the performance of the same BSSID is taken as reference. This gives the most accurate result, as individual APs may be connected via links of different bandwidth. In case no further samples are available, the measurements of the same SSID are aggregated to derive the performance of the respective WiFi network. The restriction on BSSID is



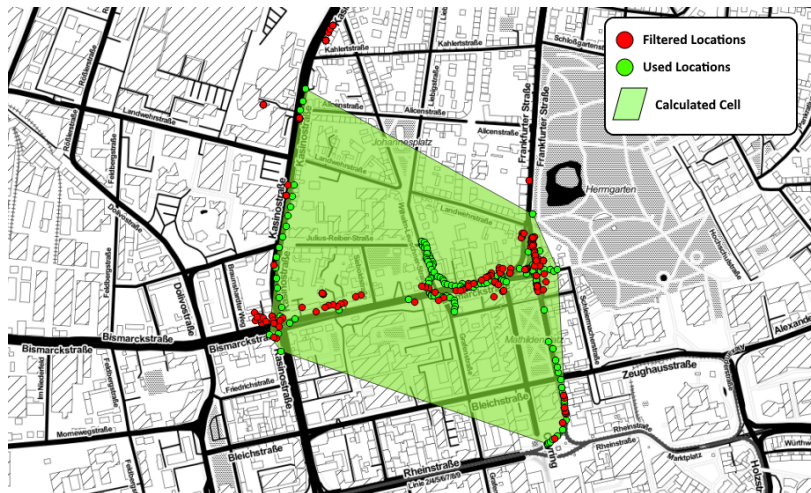


Figure 6.1: Example for the estimation of cell coverage (from [Sch16])

first attempted, because the performance of different APs may differ largely. If no other information on the respective BSSID is available, the performance of the associated SSID is the next best estimate.

An example of the observed WiFi coverage as determined by received beacons is shown in Figure 6.2. Exemplary, the visibility of *eduroam* and *Freifunk* WiFi SSIDs are shown as a heat map in orange/light and blue/dark respectively. Here it becomes apparent that it is futile to try to fully and systematically measure the full network coverage including all KPI. Doing so would require measuring all metrics in a grid of approximately 10x10 meters. For comparison the aggregated downlink measurements are shown in the hexagonal grid. Each cell has radius of approximately 15 m and indicates the mean network throughput, ranging from near zero (red/dark) to 20 Mbps (green/light).

For the analysis of the cellular network performance it is assumed that the smartphones are equipped with SIM cards not being limited to a specific network technology. Further, as most of the study participants were students at the Technische Universität Darmstadt, it was assumed that they may use the *eduroam* WiFi network, if available. Further, *Freifunk*<sup>1</sup> APs are available, and are assumed to be used by the study participants. *Freifunk* is an initiative of volunteers providing free WiFi connectivity to users. As these are commonly connected to the volunteers' Internet connection, their performance differs largely. Hence, the above attempt of first using the BSSID to determine the throughput, and only if no data is available the SSID, provides increased accuracy.

Measurements presented in Section 5.1 show that the performance of both the cellular network as well as the WiFi network is largely independent of the actual signal strength. Hence, for each cellular network technology and BSSID or SSID, the measured through-

<sup>1</sup> <https://freifunk.net/> accessed 2017-02-02

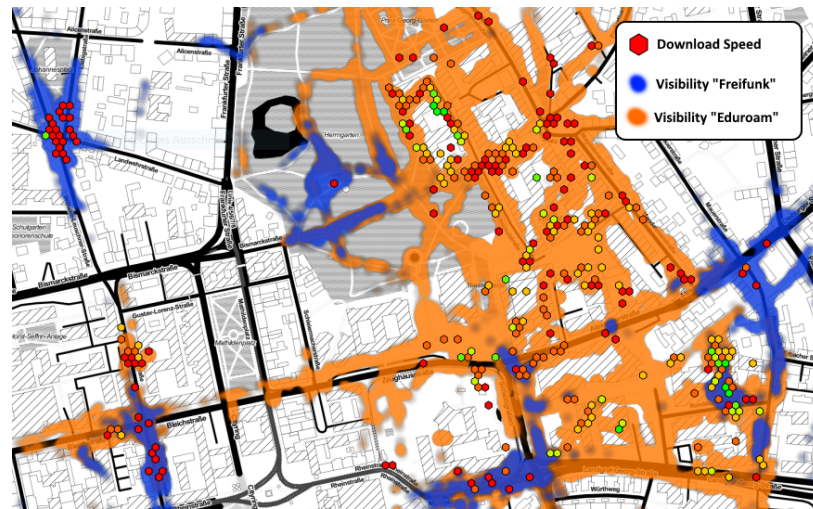


Figure 6.2: Example of WiFi coverage and aggregated throughput measurements (from [Sch16])

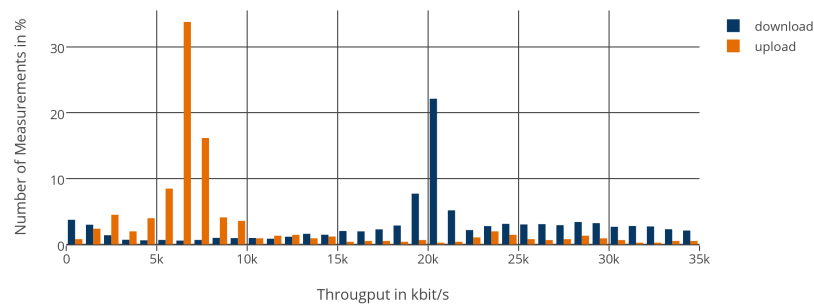


Figure 6.3: Histogram of the measured throughput in the LTE network (from [Sch16])

put is aggregated into a histogram. This is then used in the evaluation process to draw a random realization of the actual network throughput. As neither in the crowd-sensing study (cf. Section 5.1) nor the dedicated measurement studies (cf. Section 5.3) a dependency between any other of the recorded metrics could be identified, this is currently the best available estimate, considering the high cost of these measurements. The derived histograms of the LTE network are shown in Figure 6.3, the respective measurements of the Freifunk WiFi network in Figure 6.4.

The individual samples as available on the data collection server are pre-processed to facilitate determining the required metric during the evaluation phase. Therefore, the data is aggregated into hexagonal bins of 15 m diameter. Thus, for a given location the respective network performance can be identified by finding the bin covering this location. This aggregated data is used to first determine the availability of networks, and later to check for the availability of measured KPIs. Only if the requested point is covered by a

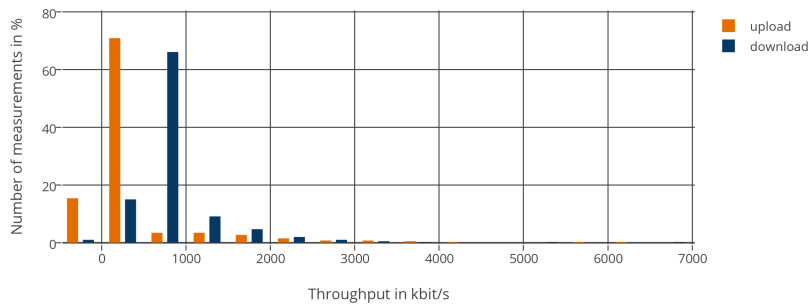


Figure 6.4: Histogram of the measured throughput in the Freifunk WiFi network (from [Sch16])

network, but no performance metrics are available, the above described random approach is used. Thus, the data is prepared to minimize the number of database requests and reduce the required computations to a minimum.

### 6.1.2 Evaluation Environment

The evaluation environment is custom built for the purpose of evaluating the energy consumption of mobile network access. It is written in Python, as being a high-level language with already a large number of packages providing the required functionality. Particularly, these are the comma separated values (CSV) and java script object notation (JSON) import and export functions, as well as the *psycopg2* package, interfacing the database on the central data collection server.

The extension of existing network simulators was also considered, but due to the requirements of the planned analysis, not much of the original functionality would be left. Even a high level network simulator like PeerfactSim.KOM [SGR+11], where already energy models are included [GKS+13], most of the functionality would be irrelevant. As the proposed analysis relies on measured network performance, the full network implementation would need to be made location aware, and thus network performance of links changing depending on location, time and used network technology. This would leave just the event management system untouched.

Hence, a modular evaluation framework is developed, conforming to the requirements of the network analysis. Based on the loaded user traces, the available networks and their performance are looked up in the database. The resulting data rates are used to determine the transmission duration of the transferred data within the connected network. Based on the transmission duration and data rates, the selected energy model calculates the consumed energy. Thus, for each user the overall consumed energy is determined.

All data is stored in a structured way in different tables of the PostgreSQL/PostGIS database. Hence, the conformity of the data is implicitly assured by the database. Fur-

thermore, the data can be requested in an orderly fashion, making use of the optimized filtering and access algorithms of the database server.

The evaluation framework is structured into independent components, providing functionality to assess the individual aspects based on incoming data. These are chained, because the output of one component is used as input for the next. The state representing the end-user device is kept in a global configuration object. This contains, besides the last evaluated timestamp and event, the last location, interface state, as well as further parameters and device states as required by the selected energy conservation approaches.

The user traces consist of different events ordered in time. These events describe changes in the device. These include data transfers (up-/downloads), user mobility, but also user interactions with the device (screen on/off, App interactions). For the subsequent analysis, these are either categorized as data events, directly affecting network activity, and device events, modifying the state of the end-user device. These modifications are location updates, requiring an update of the available networks and their throughput, screen on/off events, or scheduled changes in interface availability, thus affecting traffic scheduling.

Figure 6.5 shows the respective flow diagram for the default transmission mode. The normal flow of operation is the following: The first item of the user-trace is loaded. If this is a data event, meaning that data is to be transmitted, the position of the device is calculated based on last location and timestamp. If a location can be determined, the availability of networks at this location is looked up. When a valid network is found, its status is checked. In case the network is active, data is transmitted. After this, the next event is loaded and the cycle begins again.

In each state, the current device state, user interactions, or the environment may cause deviations from the desired behavior. If, for example, no valid location can be determined, the cycle is canceled and the user trace searched for the next location, from where the analysis begins again. Hence, a high granularity of the user traces and good coverage of network performance and availability is important. In case no network was found, the execution flow skips the following samples until a new location event is encountered, after which the cycle begins again.

The device state further defines the operating mode. In case the connection is not active as assumed in the normal execution flow, the interface must be activated. This results in an additional delay and energy consumption. This is processed immediately, but the incoming data event is only processed in the next iteration, where the interface state is changed to active. Only then, normal operation resumes.

Contrary, when no data is left for transmission, it is checked whether the next data event is after the tail time finishes. If the next data transmission is scheduled earlier, the time between the transmissions is considered to be spent with interface consuming idle power, otherwise a device event is created, deactivating the interface after the tail duration. If the next event is a device event it is executed, otherwise new data is directly loaded from the database.

Depending on the available networks, and the possibly different network conditions when recording the traces, it may also happen that for a given time interval the requested rate is higher than the available rate. For this a mechanism is introduced adding events to the data events queue. Thus, first the maximum traffic as determined by the current network availability is handled. The remaining load is then added as a new data event to the data events queue.

Based on device state and data transmissions, the power consumption is calculated. For this the power models as derived in Section 4.3 are used. These calculate the consumed power depending on selected interface and traffic rate. Here, only the energy spent on data transmissions is accounted for. The derived power consumption is identified for each time interval (e.g. between events) and written to a log file. This log file is later used for visualizations, evaluation and analysis purposes.

During all operations, log files are created keeping track of the changes in device state, location, network availability and quality. Thus, the evaluation run and the detailed decisions can later be analyzed in detail. Further, due to these logs being written to disk, also a later resuming of simulations is possible without loss of information.

Some additional processing on the collected data is conducted, providing reasonable throughput estimates for unsurveyed locations. Considering the effort required to fully sample the cellular network coverage, not for each location as contained within the user traces corresponding network coverage samples may be found. Hence, the evaluation environment supports the configuration of a base network technology assumed to be always available. In particular in urban scenarios where sufficient coverage can reasonably be expected, this serves as a fallback for the simulation in case of insufficient data. Considering the use of the evaluation environment for more rural scenarios, this feature can also be disabled.

A challenge caused by the different data sets is the performance estimation of the home WiFi performance of the study participants. Clearly, it can reasonably be expected that every student owning a smartphone and participating in such a study has access to a WiFi network at home. Still, as the main goal of the study was the identification of user behavior, any interference was minimized. Thus, also no active measurement of the network performance was conducted. Hence, the coverage and performance of the users' home WiFi is drawn in a process similar to the estimation of the performance of known networks. The coverage area is set to the maximum deviation of the majority of locations observed in the time between 2 am and 6 am. In case locations differed largely, these are removed from the estimation. Thus, the coverage area of a 'virtual' home WiFi for each user in the data set is created. The throughput is determined by drawing a sample from the distribution of WiFi throughput measurements from non-public SSIDs. This is done at the beginning of the simulation run, thus providing the same performance over the full evaluation run.

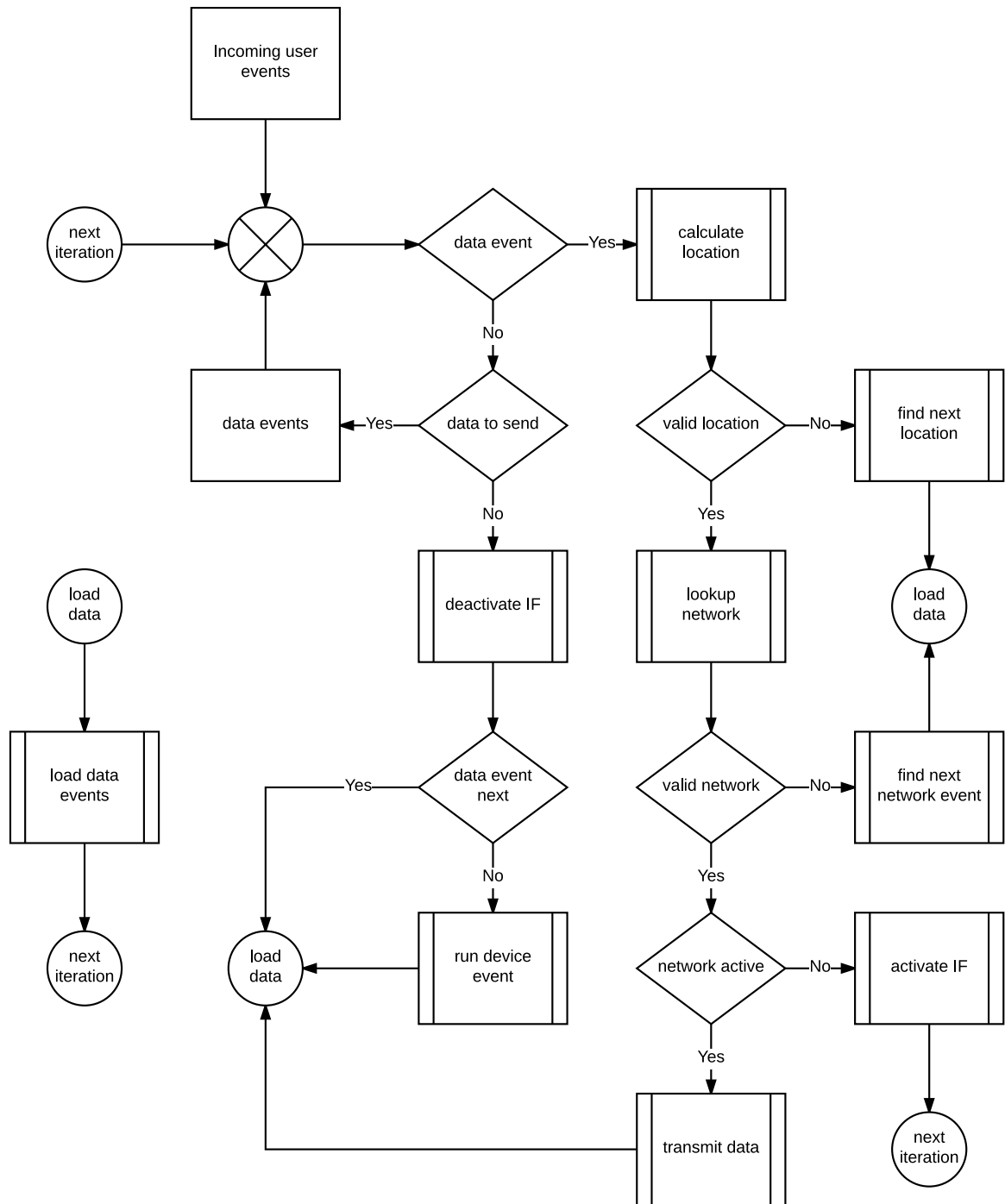


Figure 6.5: Flow diagram of the trace-based network energy estimation environment (adapted from [Sch16])

Generally, the evaluation of the deterministic traces should lead to the same results for each evaluation run. Still, in the process of generating estimates of the cellular network performance, some randomness is introduced. This is done in the throughput estimates for previously un-surveyed locations in both the cellular and WiFi networks. The reason for this is to reflect the variability and distribution of the available bandwidths on the power consumption of the devices. A more general view of the resulting energy consumption can be generated by running the evaluation on the same traces multiple times. Still, this is not desirable in any case. Considering the requirement of creating deterministic, and thus repeatable results, any randomness must be eliminated. This is achieved by seeding the random number generator with pre-determined state, thus always getting the same result. Still, as different scheduling mechanisms may affect the number of calls to the random number generator, a single seed is insufficient. Hence, for each location and time where an estimate of the cellular service quality is required, the random number generator is seeded with a string based on the current location and time. Thus, if a different user requests a network estimate at the exact same location and time, the same value would be returned. Still, as the user location as read from the recorded trace is used for this, the probability of this is almost zero. On the other hand, the same user at the same location, but having different traffic requirements due to the selected energy conservation mechanism observes the same network performance as in the default case. Thus the comparability of different traces can be guaranteed, while still allowing the same trace to be analyzed under different conditions, thus generating results more focused on the statistical distribution of network performance than caused by the end user.

### 6.1.3 *Energy Conservation Schemes*

The influence of different traffic scheduling approaches can be determined by first establishing the reference power consumption and comparing it with the power consumption when modifying the behavior of the system. The reference power consumption is established by calculating the energy consumption caused by the unmodified traces. As the available user traces are located in Darmstadt, and generally no coverage issues exist, the minimum available network technology is configured to universal mobile telecommunications system (UMTS). Hence, at locations without any network coverage samples, the availability of UMTS is assumed. The comparability of the energy conservation schemes is assured by setting deterministic seeds for the random components of the network performance as is discussed in the last section.

The evaluation environment is built to analyze different offloading and traffic scheduling scenarios. Offloading approaches as described in Section 3.3 propose large cost reductions, but also energy savings by transferring data via the WiFi interface. Still, the analysis of the collected data shows that at least in outdoor scenarios the performance of WiFi is generally insufficient, while the cellular network works well. This is also reflected in the

energy consumption, also when neglecting the additional energy required in the PA for lower signal strength connections. Due to the small throughput of WiFi in outdoor scenarios, data transfers take considerably longer, thus resulting in a larger battery utilization of the mobile device. Contrary, the performance of WiFi is better indoors, where cellular network coverage is worse due to attenuation by walls. Nonetheless, in indoor scenarios the limiting factor is the Internet up- and downlink bandwidth, which in nominal rates is often surpassed by cellular data rates. Hence, wireless offloading, both ad-hoc and scheduled, provides only minor potential.

Another class of mobile energy conserving traffic management schemes is traffic scheduling. Here, traffic as is requested by mobile applications is delayed until more favorable network conditions are observed, a number of requests have accumulated, or a specified time interval is reached. Generally, these approaches can only be applied to background traffic. Any interference with the network traffic while the user is interacting with the device immediately affects QoE. Also handling traffic of currently unused applications while the device is actively being used promises only marginal improvements, as this traffic volume is generally low and the interface likely active.

Selected for further analysis were the approaches by Huang et al. [HQM+12] and Ickin et al. [IWF13], both disabling the cellular interface while the device is idle. This is determined by monitoring the screen status. Based on this it can reasonably be derived that when the screen is off, the device is idle.

Huang et al. [HQM+12] analyze the energy consumption of a number of public applications as contained in a dataset collected on mobile users by the university of Michigan in 2011. Their optimization is based on i) disconnecting from the network earlier than configured by the operator by sending RRC messages to the network to reduce the tail energy, and ii) configuring intervals in which communication with the network is allowed while the device is idle. The parameters showing the best performance are to disconnect from the 3G network after 8 s while the device is used and after 4 s while the device is idle. Furthermore, the approach allows one 5 s communication interval in each 100 s slot. This data restriction is applied to both the WiFi and cellular interface. Combining both approaches, the authors claim to reduce the energy consumption by up to 60.92 %. This approach was developed and tested with 3G networks. Hence, savings are expected to be lower when applied to more recent networks. Tail durations within the measured 3G network were around 5 s, which are reduced to 4 s while testing the energy savings of this approach. Similarly, the possible savings when considering background traffic are expected to be lower compared to the ones reported in [HQM+12].

Ickin et al. [IWF13] focus their study mainly on the QoE of end users with a secondary goal of minimizing the energy consumption of the mobile devices. Their approach is similar to [HQM+12], but instead of 100 s intervals, 30 min intervals are chosen, with a connectivity period of 1 min within each interval. Thus, energy savings of up to 25 % without affecting QoE of the mobile users have been achieved ('34% energy gain' [IWF13]).



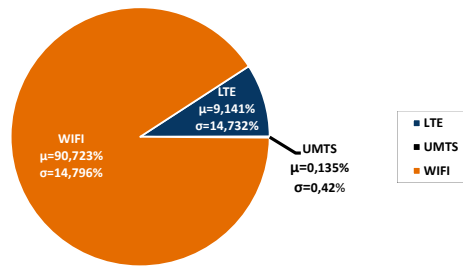


Figure 6.6: Availability of networks as determined based on the user traces (from [Sch16])

Still, also their data is solely based on 3G connectivity, although the observed tail durations seem to be smaller. As this is an operator configured parameter, it differs between networks. The same settings are configured for the analysis in the energy evaluation environment. Also here, energy savings are expected in comparison to the default case of no traffic control, but margins are expected to be considerably lower due to the use of LTE, which further reduces the tail duration to 2 s, while also providing higher mean data rates, thus further shortening transmission durations. Contrary to the approach by Huang et al., the data restrictions of this approach are only applied to the cellular network.

#### 6.1.4 Evaluation Results

The energy consumption derived based on the user traces in combination with the availability and performance of networks is compared in the following. Besides the default case of not altering the traffic patterns (except where the data rate was insufficient), the effectiveness of the energy conservation approaches as proposed by Huang et al. [HQM+12] and Ickin et al. [IWF13] are compared.

The general distribution of available networks over the full range of study participants and evaluation intervals is shown in Figure 6.6. Interestingly, during 90 % of the time WiFi networks are available. This includes the estimated home network coverage, the *eduroam* and *Freifunk* networks within the city and also covering the 'workplace'. Consequently, the optimizations of the cellular network performance are only possible within 10 % of the time, relating to 2.4 h daily. Another interesting observation is that only 0.135 % of time were spent connected to a 3G network. This means, that the activated base coverage function setting the minimum available network technology for the setup, was barely ever used. Further, this indicates that under normal conditions, full coverage of LTE can be expected, thus turning any optimizations focusing on the 3G network obsolete. Still, it must be noted that these results are only valid for an urban scenario, where people live and work within the same city. Repeating the study using a different population group likely changes the outcome of these models.

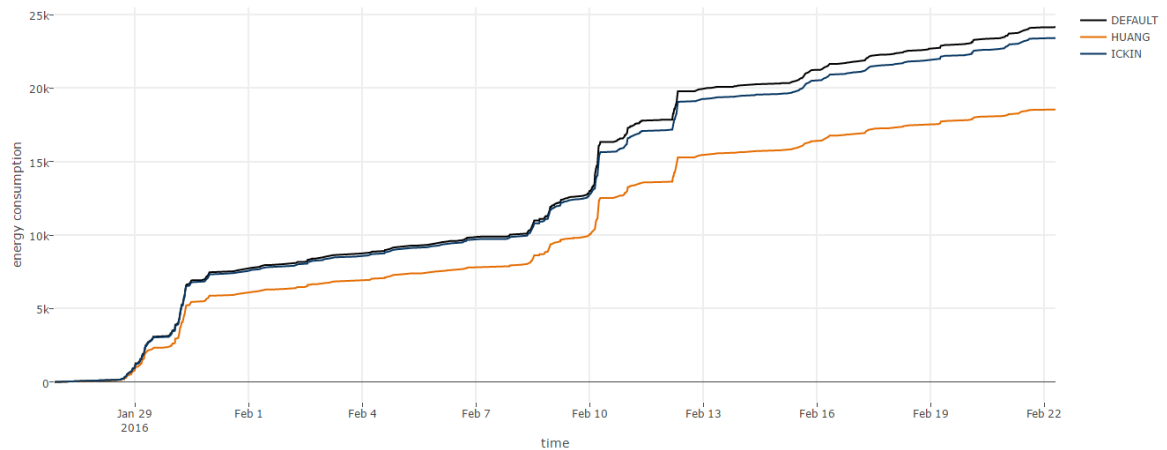


Figure 6.7: Comparison of the evaluated energy conservation schemes in Joule (from [Sch16])

Figure 6.7 shows the estimated power consumption over a full user trace for the three different approaches. The top graph shows the power consumption as derived using the unmodified traces. Directly below is the power consumption as estimated based on the approach proposed by Ickin et al. The lowest energy consumption is observed by using the approach by Huang et al. Generally, the estimated power consumption shows a similar behavior at the same times.

The power consumption over all trials is shown in Figure 6.8. The corresponding metrics are summarized in Table 6.1. As is also visible in the exemplary time series (cf. Figure 6.7), the default case consumes the most power. This is expected and shows that the implemented energy conservation policies do not affect the power consumption negatively. The highest power savings are observed using the approach by Huang et al., while the power consumption of Ickin et al. is largely identical to the unmodified energy consumption. These results are summarized in Table 6.1. Still, due to the relatively large standard deviation, only indicative statements can be made.

Compared to the power savings of up to 60% as claimed by Huang et al., under the given conditions power savings are as low as 27.4%. The approach proposed by Ickin et al. performs even worse. Instead of the claimed 54%, the estimated power saving is only 1.7%. This is both caused by the high fraction of WiFi connections as observed in the user study, but also the increasing energy efficiency of the cellular network. Caused by the generally shorter ramp and tail durations on LTE, but also on 3G networks, already considerable energy is saved. Furthermore, the increased data rates as available on LTE reduce transmission costs by requiring less active time of the cellular interface.

The observed behavior confirms expectations as derived from literature, where larger savings can be expected when restricting communication on all interfaces. By only dis-

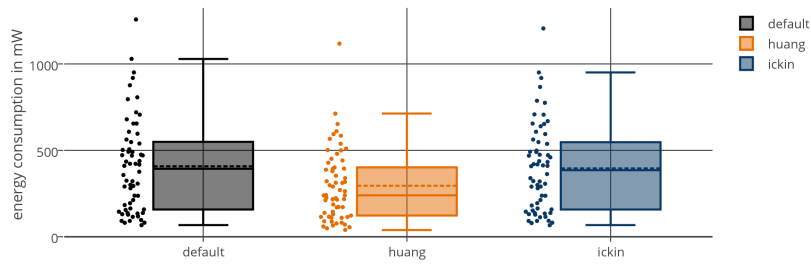


Figure 6.8: Mean power consumption of the selected energy conservation schemes over the full set of traces (from [Sch16])

Scheme	Minimum	Maximum	Median	Average	Standard Deviation
Default	66,8 mW	1.258,2 mW	393,5 mW	407,9 mW	268,7 mW
Huang	39,5 mW	1.118,4 mW	240,4 mW	294,1 mW	203,4 mW
Ickin	66,7 mW	1.206,5 mW	385,6 mW	396,5 mW	253,8 mW

Table 6.1: Energy consumption statistics of different energy conservation schemes (from [Sch16])

abling communication on the cellular interface during the radio-quiet intervals, Ickin et al. are wasting potential for saving energy, by being active in our case during only 10 % of the time. Still, as their focus is also QoE, and WiFi connectivity often coincides with the availability of grid power, the effect on end users is considered to be limited. Due to the shorter scheduling intervals in Huang et al., the effects on QoE also appear to be limited, still, additional studies analyzing this effect are required.

Another interesting aspect of any traffic scheduling approach is the additional transmission delay. Figure 6.9 summarizes the observed delay. As would be expected, the default approach transmits 100 % of the packets immediately, followed by Ickin et al. for 95 % of packets. The remaining packets are transmitted within the next 100 s. Huang et al. transmit only around 55 % of the packets within the first 10 s. Still, almost all packets are transmitted within the configured 100 s interval, leaving only a small number of packets to be transmitted later. This effect is expected to be caused by scheduling delays, where the surplus traffic is shifted to the next transmission interval. The longest delay with 4.25 min is observed when using Ickin et al., meaning that even in the case of cellular connectivity when idle, only marginal amounts of traffic are generated, or the availability of another WiFi APs is so likely that a transmission is possible already after a short duration.

Considerable effort was spend on analyzing and optimizing the performance of the evaluation environment. Due to the large number of required lookups and the underlying data aggregation, a large load on the system is generated. Hence, a number of optimizations are implemented, namely the pre-aggregation of frequently used data, and the

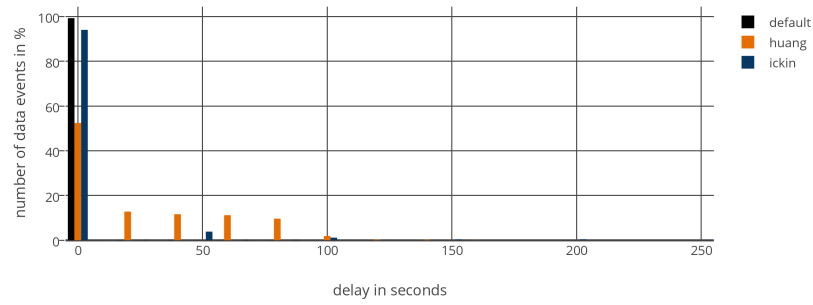


Figure 6.9: Observed delays in transmission of the compared energy conservation approaches (from [Sch16])

creation of database tables immediately providing the requested data without requiring database joins. The final result is a linear dependency between the number of events in the user traces and the required processing time. As all aspects of the energy evaluation depend on locations, these are taken as the baseline for comparison. For each user event (e.g. data transmission, screen on/off) two location updates are found in the user traces. Processing 10 000 locations results in duration of 100 s. Considering that the user traces contain approximately 3 location samples per minute, the processing speed is approximately 2000x real time, thus making it an attractive solution to swiftly compare different energy conservation approaches on large data sets. Hence, the location based analysis of traffic scheduling and energy conservation approaches for mobile data access is feasible with high performance.

The results generated and presented here are valid for the scenario described in the beginning. The area is currently limited to Darmstadt, as only there the required KPIs of the different networks are available. Furthermore, the presented results are indicative, as only a small number of user traces contained the required data in sufficient resolution and accuracy. For the analysis of the energy consumption, exemplary the power model of the Nexus 5 was used, but can easily be replaced in the evaluation environment. The feasibility of the selected evaluation approach is shown on the example of energy conservation disabling background communication in different configurations. Also these approaches may easily be extended and included in the evaluation environment, thus permitting the evaluation and comparison of a wider variety of approaches.

#### 6.1.5 Summary and Conclusions

Combining the observations made in Chapter 4 and Chapter 5, the energy evaluation environment estimates the energy consumption of mobile users based on recorded user traces for different energy conservation schemes. Therefore, locations, device interactions, and consumed traffic collected by the participants of a data collection study are used. For

each data transmission, the network coverage at the place of the occurrence is determined by finding the available networks and their performance in a central database. This data is collected by the NetworkCoverage App in various crowd-sensing studies as described in Section 5.1. Knowing the network availability and throughput, the resulting energy consumption is determined based on the power model of the Nexus 5 as was introduced in Section 4.3. Finally, two energy conservation approaches as proposed in literature are compared to the default energy consumption as directly derived from these traces.

The identified power savings are considerably smaller than claimed by the authors of the respective studies (i.e. [HQM+12; IWF13]). Instead of 60% and 54%, the energy savings in the analyzed scenario are as low as 27.4% and 1.7% respectively. This is caused by two main factors: first, the chosen scenario showing a high probability of WiFi connectivity, and secondly the increasing energy efficiency of mobile devices, in particular of cellular communication. Compared to 3G networks as used in these studies, the energy consumption was derived for LTE networks, significantly reducing ramp and tail durations. The considerably higher data rates result in lower transmission durations, further reducing the energy consumption on the mobile device.

The dependency on empirically measured network performance has both advantages and disadvantages. By using the measured network performance, the dependency on simulated network performance can be completely eliminated by abstracting end-to-end performance using the measured KPI. Still, this comes at the cost of measuring the cellular network performance in a fine-granular manner, or as was done in the presented evaluation, modeling the network behavior based on empirically determined performance metrics. Thus, the effort required to provide accurate and distributed network performance metrics can be balanced.

The quality of the generated energy consumption estimates can be improved by increasing the accuracy of the underlying network estimates by increasing the number of available throughput samples in the database. Further, the addition of power models of different smartphones, in particular different generations would further improve the value of the generated estimates, thus allowing the experimenter to determine possible weaknesses of the proposed energy conservation approaches on particular devices and configurations. Still, most important is the availability of complete, fine-granular user traces, containing locations, consumed traffic and device state. As the requested traffic volume, and the user behavior in general, is highly volatile, repeated studies would be required, periodically updating the user traces. These may also serve as reference data sets, allowing researchers at different locations to compare their approaches based on a common 'ground truth'.

A further, interesting extension of the currently available data set would be the annotation and drill-down of the collected data to application level accuracy. Thus, traffic management and scheduling approaches may base their scheduling decisions, besides user activity and device state, on derived priority classes of the requesting applications,

thus eliminating low-priority network requests if, for example, the requesting application was not used for a long time.

A number of the approaches proposed in literature are already included in newer versions of mobile operating systems. Where Apple's iOS in the past did not allow background processes at all, thus completely avoiding the problem of background activity, it now allows certain processes to run and communicate in the background. Waking devices up or triggering applications to update their data is implemented via a cloud service, bundling requests to the mobile device depending on priority, thus strictly limiting possibilities to inadvertently drain the device's battery. Android, in comparison relatively open, follows the paradigm of first allowing everything, and only later restricting functionality, which causes user dissatisfaction. Background services were always available, although on earlier devices often terminated due to low system resources. In recent Android releases, more focus is put on energy efficiency, emphasizing the use of google cloud messaging (GCM) to sync wakeups and avoid idle connections, waking processes up in batches in progressively increasing intervals ('Doze', Android 6.0 [And15]), or disabling network requests of rarely used Apps ('App Standby', Android 6.0 [And15]). Thus, App behavior harmful to the battery life-time is more and more restricted, thus maximizing the efficient use of the available energy.

## 6.2 ENERGY EFFICIENCY IN FUTURE NETWORKS

The increasing demand on communication networks increases traffic in all network domains. This begins at the mobile edge, where increasing data rates need to be supplied via advanced coding schemes, increasing spectrum re-use, and using additional spectrum, and continues over the network backbone to the content providers. Here, network virtualization provide the means to implement a number of promising network management approaches, ranging from the (wireless) access and aggregation layers to data centers.

In the wireless access networks, the promising future technologies are wireless SDN and centralized/cloud RAN (C-RAN). Further emerging technologies improving throughput in wireless access networks are (massive) multi user MIMO, non-orthogonal multiple access (NOMA), or full-duplex transmissions using self-interference cancellation.

Also in the fixed network backbone, functionality is increasingly migrated to SDN. Routers and switches may thus be configured on-the-fly from centralized controllers, dynamically adapting the network structure to the current demand. Thus, additional functionality can be implemented in the network. One example is SDM, implementing a multicast approach based on packet duplication features of SDN switches [RBH15]. Furthermore, virtual networks can be created and maintained. An example may be an ISP offering its business customers networks as a service (NaaS), thus allowing them to dynamically connect their sites without requiring a virtual private network (VPN). By increasing control over their network, higher link utilizations may be achieved [JKM+13].

The traffic consumed by end-user devices mostly originates in cloud data centers. These are supported in the delivery of static content by CDNs, providing storage at neural points of the network, e.g. data centers, IXPs, or central locations within ISP networks. This shows that content delivery is migrated from a central server or data center, located where convenient, closer to the end user.

Supporting the growing bandwidth requirements in cellular networks is also called the '1000x data challenge' [Qua13], extending the network to supply 1000 times the data rate of current systems. Main development directions target the use of femto-cells, immediately increasing throughput compared to macro cells, using additional spectrum in the licensed, licensed-shared access [MOP+14], and unlicensed bands [SBS+15], and increasing the efficiency of spectrum use and re-use by using improved coding schemes and more intelligent control. Still, these increasing data rates must also be handled by the network backbone, remote servers, and CDNs.

The implications of different technologies, their development goals, and their effect on energy efficiency of the network are discussed in the following.

### 6.2.1 *Potential of Emerging Network Technologies*

In wireless access networks the predominant technique for improving throughput is the deployment of increasingly smaller cells. Thus, the available spectrum can be used more efficiently by increasing its re-use. Furthermore, by improving coordination, the frequency use can be shared, thus allowing cells to use the same frequency. By increasing the frequency utilization, increases in throughput per area by a factor of  $>10\times$  [Qua13] are predicted. Possible reductions of energy consumption using femto-cells are analyzed in [TZZ13], showing that their deployment significantly reduces the area power consumption, in particular when serving indoor users. Savings of 75 % are expected to be possible when the area traffic demand surpasses the capacity of deployed macro cells.

The power consumption of the radio access network including fronthaul is assessed by Fiorani et al. in [FTF+16]. They compare the energy cost of different fronthaul technologies (e.g. using copper or optical links) for different links and also the placement of the radio signal processing (i.e. directly at the antenna or at a centralized location). For current networks the most energy efficient deployment is the radio network as currently used, but considering increasing bandwidth requirements, in particular within buildings, a femto-cell deployment using optical fiber backhaul is most energy efficient, followed by an approach also using fiber for the fronthaul [FTF+16].

Another proposal is the use of C-RAN (e.g. the processing of the radio signals in a centralized location), promising to reduce the energy consumption of radio networks by using co-location gains of cells with differing load within a central processing location [CMo12]. The overall available computing capacities can be reduced by leveraging the varying load distributions of cells over the course of the day. The proposed architec-

ture requires low latencies on the fronthaul (i.e.  $< 0.5$  ms) and low jitter ( $< 16$  ns), while at the same time requiring high bandwidths (2.5 Gbps per 20 MHz carrier) [CCY+15]. Due to the low latency requirement on the fronthaul, the physical distance between remote radio unit (RRU) and baseband unit (BBU) is limited to 20 km to 40 km. Satisfying the bandwidth requirement is currently mainly feasible using fiber optical links, or expensive, high bandwidth point-to-point radio links. Due to economic reasons this approach seems to be limited to a few special application scenarios like covering stadiums or other large-scale events. Fiorani et al. [FTF+16] further show that the C-RAN approach has an unfavorable energy consumption. Depending on the fronthaul architecture, C-RAN deployment can be energy efficient when the network load is high (e.g.  $>550$  Mbps/km<sup>2</sup>).

Another approach promising to considerably improve the throughput in wireless networks is cognitive radio (CR) [Hay05]. CR uses otherwise licensed spectrum as secondary user under the premise of not disturbing primary user's transmissions. The different aspects and inherent challenges of cognitive radio are summarized by Simon Haykin in [Hay05]. Challenges involved are amongst others the detection of primary user's transmissions, the coordination with remote end-points, and the optimal power allocation minimizing interference. Commonly, for cognitive radio applications software defined radios (SDRs) are used, providing the required frequency agility as well as flexibility in choosing modulations and coding schemes. Caused by this flexibility, the use of highly efficient ASICs is limited, leaving all signal processing to a generally less energy efficient general purpose CPUs (GP-CPU) or field programmable gate arrays (FPGAs). As cognitive radio is still in its infancy, current efforts are more focused on feasibility and performance than energy efficiency. After suitable approaches are found, the next important aspect will be the energy efficiency, in particular when considering mobile devices.

In wired networks, SDN is increasingly becoming attractive, as vendors provide a larger variety of devices. Generally, SDN may also be implemented in a single-vendor only way by not disclosing the communication protocols. Still, a growing number of devices also support the OpenFlow protocol. Thus, software-defined networks can be built without relying on a single hardware vendor. This reduces licensing costs in case of commercial software, and ensures that for different requirements within the network always the best implementation and hardware can be used. Still, performance differs largely, as is also shown in Section 4.2, where on the hardware switch some features are only available in software, thus drastically reducing performance when used. Energy efficiency in SDNs is also an ongoing topic in research, which was also discussed in Section 3.1.2.

Extending the paradigm of SDN to wireless networks is still a relatively new topic. In particular the energy consumption is not well considered. Vitale et al. [VM16] analyze the energy efficiency for the case of device to device (D2D) offloading in heterogeneous environments. A general solution for optimizing the trade-off between different aspects of energy consumption (i.e. deployment vs. operation vs. radiated power) in cellular networks is discussed in [CZX+11]. Their work gives guidelines on building and implement-



ing cellular networks in an energy efficient manner for the case of micro and macro cells. These approaches would need to be extended considering heterogeneous networks (i.e. using WiFi) and femto-cells, thus including the wireless network domain into their optimizations.

Another area, disjunct from wireless SDN, are self-organizing networks (SONs). Contrary to SDN, the control here is distributed between the eNodeB or base stations, coordinating transmission power and association of mobile stations, thus minimizing interference and optimizing service. An overview of the issues involved in SONs is given in [PB05], a survey of approaches is given in [MSB13]. Usually, only the issue of assigning radio frequency (RF) power is discussed (e.g. [XY13]), neglecting the power consumption of the overall deployment. Here, a large optimization potential exists, including the different approaches, which are likely already integrated into deployed networks, to reduce the overall system power consumption by e.g. temporally disabling under-utilized devices.

Considering the flexibility of SDN for managing traffic in all types of networks, also NFs must become more flexible. Currently deployed NFs implement firewalls, packet filters, PEPs, VPNs, and others. These are usually purchased and deployed as 'black boxes', wired and configured to serve their purpose. Obviously, this approach defies flexibility and scalability. Currently, upgrades are only possible by replacing the hardware unit, or parallelizing processing by adding load balancers. Thus, also these functions are increasingly virtualized, then called VNFs. In 2012, the ETSI began standardization and issued a first white paper detailing their vision of NFV [NFV12]. An overview of general network virtualization approaches is given in [JP13]. A recent classification of NFV approaches is given in [LMW+15] and [NHH16]. Current focus of development is achieving sufficient performance for deployment in high load scenarios and optimizing computational efficiency. As these VNFs are commonly run on COTS x86/64 hardware instead of ASICs, their efficiency is inherently lower. While sufficient performance is a strict requirement for deployment, the aspect of energy consumption should not be neglected in research and development. Here, important decisions may be taken before other design decisions may interfere. Particularly the heterogeneity of devices in future networks may prove promising to implement energy savings, where NFs may be run on x86/64 servers, offloaded to FPGAs [NRH17], or run on ARM platforms at the edge of networks, e.g. on home routers.

All these developments may be summarized under the term 5G. The next generation mobile network (ngmn) alliance also targets the adaptation of large areas of the fixed network infrastructure, to support the high bandwidths, low RTTs, and high number of devices. Sub-millisecond RTTs may only be achieved by decentralizing the network, and serving content and providing services from close to the end user. Clearly, if combined with WiFi offloading, which is required to support the envisioned data rates, also network functionality must be placed within the fixed access network. The challenge is finding the right abstractions, granting the participating parties still mutual benefits. SDN and NFV

may provide the basis for future solutions by virtualizing the network substrate, and thus letting network providers share their networks for a fair compensation.

### 6.2.2 *Network Structure and Management*

The increasing capabilities of home routers make these an increasingly attractive asset in provider networks. A first example of using these idle resources are the increasing number of WiFi services run on HGWs. Popular examples are Fon<sup>2</sup>, Unitymedia WifiSpot<sup>3</sup>, or Telekom Hotspots<sup>4</sup>. Thereby, the penetration of the operator's WiFi networks can be increased, reducing the load on the cellular infrastructure. Combining this with automated logins on smartphones managed by user-installable Apps or integrated into the operator-branded OS, automatic offloading to WiFi networks is already possible.

Extending the idea of providers adding functionality to HGWs, micro-services or caches may be located within end-user premises, thus reducing latency and increasing throughput to the end user. Exemplary this was implemented and evaluated in [LPB+15]. Still, conventional caching is not feasible anymore, as an increasing fraction of traffic is transferred over encrypted connections [San15]. However, placing network functions at end-user premises may be possible using lightweight containerization techniques like LXC<sup>5</sup> or Docker<sup>6</sup>. Thus, these resources may be sold to CDNs or other interested parties, allowing them to place nodes, and such content, closer to the end users than already possible. This is expected to increase the end-user perceived service quality while simultaneously reducing instantaneous bandwidth demand.

Considering the additional requirements as envisioned for 5G networks, also the migration of network functionality closer to the end user should be possible in the cellular network. The virtualization of the network in combination with appropriate QoS management may serve as a means of providing additional services to mobile end users while not affecting the primary user of a connection. An example would be running a PEP or the SGW and PGW on a user's personal pico-cell, directly interfacing with the Internet.

Still, these changes in network functionality require additional modifications of the network structure. As derived in Section 5.3, connections of mobile devices of the analyzed MNO are routed through one of four gateways within Germany, where also the additional network functions are placed. These include, besides the required network functionality like user management, network admission control and mobility management, also additional functionality like PEPs or other operator provided services. Considering a fast

---

<sup>2</sup> <https://fon.com/> accessed 2017-02-06

<sup>3</sup> <https://www.unitymedia.de/privatkunden/angebote/unitymedia-wifispot/wifi-fuer-unsere-kunden/> (German) accessed 2017-02-06

<sup>4</sup> <http://www.hotspot.de/> (German) accessed 2017-02-06

<sup>5</sup> <https://linuxcontainers.org/> accessed 2017-03-18

<sup>6</sup> <https://www.docker.com/> accessed 2017-03-18

handover to WiFi networks, or the availability of decentralized services, the currently deployed network structure clearly must also become more decentralized.

The deployment of these virtualized micro-services also affects currently implemented network management approaches. Load balancing of CDNs is achieved via DNS [NSS10], where the response has a validity of a few minutes. For each new content element requested after some time, another DNS request is issued possibly resolving to another server. Considering the deployment of services on a HGW, either also a DNS service must be running as a NF on the gateway, or the remote DNS be informed of the currently used connection. This must then be mapped to the available content or services on the local gateway, thus allowing the CDN to direct the mobile device to the closest CDN node.

Considering the vertical slicing of the underlying network substrate, also resource utilization, in particular on resource-constrained devices leads to some interesting research topics. First, the mapping of the requirements of VNFs to resource consumption depending on the underlying hardware architecture should be analyzed. Here, representative metrics describing the workload, possibly on a per-user basis, must be found and mapped to the underlying hardware, thus permitting optimization algorithms to determine a possible placement of the respective network function. This becomes particularly important, when considering resource scarcity of the hardware running these functions and serving multiple users in parallel. Secondly, the gain of running these functions in increasing QoS and bandwidth utilization and reducing RTT must be determined.

These modifications of the network require a tight alignment between currently mostly disjunct network domains. When a handover between the cellular network and WiFi is conducted, all connections need to be reestablished. Here, MPTCP may help alleviating the most severe interruptions [PDD+12]. Still, if low latencies are required, the current cellular network architecture must be adapted. Instead of four PoPs distributed over Germany, one PoP per city is required to support latencies of below 1 ms. Similar modifications would be required in fixed access networks to support fast handover between the fixed and cellular networks. Thus, services placed in data centers within each city benefit from both low latency when connected to one technology, and fast inter-technology handover.

### 6.2.3 *Energy Efficiency*

The overall network energy consumption of conventional networks can be modeled as described by Vereecken et al. in [VHD+11]. The authors model the static power consumption of both the fixed and wireless network domain. According to measurements conducted in Chapter 4 this is a reasonably accurate approach. Only, if device vendors considerably reduce the idle power consumption of networking devices compared to the dynamic part, more advanced power models are required. What this approach does not consider is the processing load of network functions or the increased flexibility as offered by SDN.

Section 4.3 shows that wireless data access on smartphones is most energy efficient when connected to a WiFi network with sufficiently high data rates. Still, a comparison of the energy consumption within the network backbone of the same data transfer on cellular and WiFi networks was not conducted yet. Due to the high idle consumption of the network at periodically low utilization, the energy cost of data transmissions is expected to vary considerably over the course of the day. Particularly the high cost of cellular base stations influences the overall transmission cost.

Considering the large increase in capabilities at near constant power consumption of smartphones, considerable reductions are also expected to be possible on HGWs and similar devices. As these are often idle for long time intervals, similar sleep modes appear to be promising. Assuming a larger penetration of ARM chipsets in the home gateway market, energy savings may be possible using similar functionality in HGWs.

The network reconfiguration approaches as discussed in Section 4.2 (e.g. reducing link speed) or switching off ports or full devices [GGS04; CSB+08; CMN12; YWX+13] also apply to fixed network infrastructure. Reducing the idle consumption promises considerable improvements, converging to energy-proportional forwarding [VK10]. In both areas, the commitment of hardware vendors is required, strengthened by the market requirements of reducing OPEX, particularly in high density environments, where the cost of air conditioning proportionally increases the cost of equipment energy consumption. Considering services to move closer to the user, relating to a decentralization of the network, the currently deployed network capacities may be used for a longer duration, as the burstiness of traffic can be reduced using caching, and better controlled using SDN.

The cost of cellular networks can be estimated using approximations of the base-station power consumption as published in related work [DDG+12], while the influence of the remaining network can be deduced from aggregate numbers as published in [CM012]. Assuming that the existing infrastructure will only be extended, but no significant power saving approaches implemented, the additional cost or savings of network decentralization can be estimated. Absolute savings are difficult to achieve under these assumptions, but the potential for large relative cost reductions considering the processed data volume is given. Still, the correct trade-off between decentralized and central devices, including the transmission cost, must be found.

Currently, no mechanism exists to estimate the cost and benefit of these decentralization approaches. Depending on the network functions to be deployed and their specific requirements, their demand on network and computing hardware is highly variable. Furthermore, the underlying computing hardware influences its suitability for different computing and traffic processing tasks. Considering virtual CDN nodes serving content via a secure connection and a PEP, the demands are highly different. Where the CDN node possibly requires large storage, the PEP is mainly limited by raw network throughput. The cost and energy consumption of each of these approaches also depends on the un-

derlying hardware. Furthermore, the hardware at different locations may have different characteristics and limitations, possibly causing further restrictions to the NF selection.

These possibly conflicting optimization goals may be resolved by i) mapping the requirements of NFs depending on network traffic to hardware requirements and ii) mapping the hardware requirements to power consumption of the network, based on both system and network utilization of the NFs. For this an evaluation framework extending the one presented in Section 6.1 by network infrastructure demands and energy consumption is proposed. This may include, in different abstractions, the individual network domains [BBD+14]. Thus the cost of different traffic management, routing, and content placement algorithms can be determined, based on either recorded user traces as described in Section 6.1, or synthetic usage scenarios, e.g. created using mobility and traffic generators.

Knowing the resource utilization and energy consumption models of specific network demands, an optimization problem can be formulated. Thus, optimal solutions for a given scenario can be found, and heuristic solutions reducing the computational complexity to manageable times. This allows the proposed approaches to be deployed and continually adapted to real-world networks.

Summarizing the above considerations, a few conclusions can be drawn. Energy optimization in conventional networks has limited potential, due to source and destination of the traffic being fixed. This is caused by the energy requirements of the analyzed devices mostly showing high idle power and a linear increase under load. Hence, scheduling traffic has no influence on the overall energy consumption. Still, routing and disabling of interfaces and devices may reduce the power consumption. However, for currently deployed network infrastructure frequent reboots are discouraged.

More interesting are wireless networks, where second order terms are visible. There, a higher utilization of the interface reduces the cost per byte. Thus energy savings by scheduling and aggregating traffic are possible. Considering ramp and tail energies as inherent in 3G and to limited extend also in 4G networks, these savings can further be increased.

The advanced capabilities of SDN and NFV in contrast show performance improvement and energy saving potential. By dynamically placing VNFs in the network, the optimal performance for the end user can be achieved. By dynamically adding and removing computational capabilities, the system utilization can be optimized. Thus also the energy consumption can be reduced. Configuring the network to re-route traffic to the required destinations, the placement of these VNFs is completely flexible. Thus, network functions can be run where the optimal performance and energy efficiency is identified.

By modeling the requirements of these network functions, and knowing the energy models of the underlying hardware, the effectiveness of scheduling approaches may be analyzed using an energy evaluation framework similar to the one described in Section 6.1. Instead of focusing on a single device, the source and destination of each traffic flow with

its associated requirements is defined. These evaluations may be based on realistic traffic patterns recorded by end-users, thus resulting in a realistic estimate of the energy saving and performance improvement potential of possible solutions. From these, optimizations as well as suggestions on further directions for hardware and algorithm development can be derived.

### 6.3 CONCLUSION

In the previous sections the influence of mobile data communication on the power consumption of network equipment is analyzed. This is conducted in a quantitative manner for the example of the Nexus 5 in an urban scenario in the first part of this chapter. The influence of data transfers and possible improvements on infrastructure networks considering the proposed decentralization of the network in combination with emergent technologies like SDN and NFV is discussed in the second part of the chapter.

The analysis of the load dependent power consumption of communication networks was guided by two research questions posed in the beginning, based on which the respective findings are summarized in the following.

*RQ 3.1: What is the energy cost of mobile communication for a regular smartphone user, and how is this affected by smartphone-based energy conservation approaches?* The analysis in Section 6.1 has shown that the cost of mobile communication nowadays is mostly proportional to the load on the network interface. Ramp and tail energies as the pre-dominant influence on the power consumption as was common in 3G networks have almost disappeared. The tail times in the measured network are already comparable to the ones proposed by Huang et al. [HQM+12]. Limiting background traffic reduces the power consumption, but savings are considerably smaller. The improvements as proposed by Ickin et al. [IWF13] only modifying the behavior of the cellular interface while the device is inactive have marginal effects on the energy consumption, when the majority of time is spent on WiFi. Based on the available traffic traces, the energy consumption of two traffic saving approaches (i.e. [HQM+12; IWF13]) is analyzed. The evaluation of the consumed energy shows that the benefit on WiFi and 4G networks is significantly smaller than originally claimed by the authors. This is caused by the extensive 4G coverage in the area where traces are available, and the high fraction of WiFi coverage.

*RQ 3.2: What is the potential of emerging network technologies on network infrastructure and mobile devices considering performance and energy consumption?* Assuming an increase in traffic demand as predicted in [Cis16], the load on both cellular and infrastructure networks, but also on content providers is increasing considerably. This is also expected to increase the network power consumption [BBD+11]. Based on measurements of network equipment power consumption as conducted in Chapter 4 and is available from literature, possibilities of reducing the power consumption of communication networks under consideration of the goal of further reducing the latency of the network are discussed. Energy

savings are anticipated when leveraging the flexibility of SDN and NFV to dynamically reroute traffic and move VNFs to the optimal locations. The end-user perceived service quality is expected to improve, while simultaneously increasing the utilization of network hardware. Thus, currently deployed hardware may be used for a longer time. This is expected to reduce the CAPEX of the network provider, and simultaneously saving energy by delaying upgrades to more potent, but usually more energy consuming hardware. The service placement is to be optimized by considering the load generated by a single user on the different hardware platforms, based on which the minimum energy cost is to be derived. The performance of the derived optimization algorithms can then be analyzed based on empirical traces and power models as presented in Chapter 4 and literature, to assess the real-world performance and effect on overall energy consumption.

Summarizing, this chapter has shown the feasibility of determining the power consumption of end-user devices based on empirical traffic traces. The effectiveness of two approaches presented in literature is compared, concluding that both mechanisms, developed with 3G networks in mind, show limited savings in current networks. Combining the requirements of lower latency of the network and reducing the energy consumption, the decentralization of networks is proposed, leveraging the flexibility of SDN and NFV to dynamically reconfigure routing and service placement. The performance for the end-user is expected to be maximized, while making use of idle network components, thus increasing network utilization, which is also predicted to reduce the overall power consumption of the network.





## CONCLUSIONS

---

**E**NERGY EFFICIENCY and performance are two of the main aspects when assessing modern communication networks. This is true for both wireless and fixed access networks. While in cellular networks the base stations are the largest energy consumer, HGWs consume the major fraction of energy in fixed access networks [VHD+11]. Due to the increasing load on the network [Cis16], the energy demand is predicted to continue to grow [BBD+11]. In particular the increasing popularity of wireless data access results in a high energy consumption within access networks [VHD+11].

This thesis shows the feasibility of modeling both device power consumption and network performance, thus quantifying both of these two aspects for mobile data access. Based on these, mobile energy conservation approaches are compared, from which suggestions for improving both performance and energy efficiency in future 5G networks were derived. The following sections summarize the findings as presented in this thesis, and give an outlook on promising areas for future research, considering both energy efficiency and network performance.

### 7.1 SUMMARY OF THE THESIS

The power consumption of network entities, the performance of mobile data access, and the analysis of energy conservation approaches as discussed in the previous chapters are summarized in the following. The conclusions are structured along the research questions posed in the beginning, from which the contributions of this thesis and the overall conclusions are drawn. First, the energy consumption of communication networks is summarized in Section 7.1.1, followed by the analysis of cellular network performance in Section 7.1.2. Combining these, conclusions on their simultaneous optimization are summarized in Section 7.1.3.

#### 7.1.1 *Power Consumption of Network Entities*

Based on a thorough analysis of power models for networking hardware, the lack of power models for the device classes of SBCs and SDN switches was derived. Hence, the measurement methodology to calibrate dynamic power models for these was developed. The adapted measurement procedures were applied to a number of different devices, showing the feasibility of inferring power models depending on operating state and load. Thus, exemplary power models for the measured devices were generated. These power

models were assessed using arbitrary loads by comparing the power consumption as estimated using the observed load with the measured power consumption. The obtained results for the guiding research questions are summarized in the following.

*RQ 1.1: What is the energy cost of decentralized caching and computational offloading using SBCs and how can it be determined?* The power models for SBCs as presented in Section 4.1 describe the load-dependent power consumption of a new class of devices. Therefore, extensive performance and power measurements were conducted based on which power models of the respective devices are generated. The resulting accuracy, tested under arbitrary load is between 1 % and 10 %. As the generated model is based on system utilization values read from `/proc`, the model may be used with minimal additional effort. These models are exemplary used to derive the power consumption of a large deployment of SBCs providing services to end users. By comparing changes in the peak performance and energy consumption of SBCs over time, a similar increase in computational efficiency as observed by Koomey et al. [KBS+11] are confirmed.

*RQ 1.2: How does the energy efficiency of hardware and software OpenFlow switches compare and what are their respective benefits?* The power consumption of SDN switches was presented in Section 4.2 based on the example of a professional hardware switch and an Open vSwitch running on a bare metal server. Similar to conventional switches, the power consumption of a hardware switch is mainly defined by its idle consumption and the configuration of connected ports. The influence of OpenFlow matches and actions on performance and power consumption is negligible, as long as the functionality is implemented in hardware. More complex matches and actions reduce the performance and increase the relative power consumption per packet by a factor of 1000. The Open vSwitch does not show dependencies between the number and type of matches and actions applied to the processed traffic on the resulting power consumption, thus making it the preferred choice for more complex tasks. Still, the power per port and cost per packet are higher on the Open vSwitch. Concluding, hardware switches showing high performance and low influence on power consumption for simple packet processing may preferably be used in aggregation networks, while advanced packet manipulation should be conducted by virtualized switches, possibly close to the core of the network.

*RQ 1.3: What is the energy cost of increasing reliability and throughput of mobile communication using MPTCP for CBR streaming?* Similar to the SBCs, the power consumption of smartphones for the use case of CBR streaming was analyzed and modeled. The focus of the analysis was on real-time, bandwidth-intensive communication like live video streaming or cloud gaming. The results show that the energy consumption is minimized when using only a single interface, with a preference on the one with the lower RTT. Only if the available throughput on a single interface is insufficient for the requested data rate, the second interface should be added. Still, the majority of data should be transmitted on the interface with lower RTT. Considering that generally WiFi interfaces show a lower RTT than the cellular network, also the cost of data transmissions is minimized.

Generally, the presented power models result in a good fit between the measurement and power estimation based on utilization values. The observed error is between 1% and 10% depending on device class and device. The derived power models build the foundation for further analysis and optimization of different network domains regarding performance and energy consumption.

### 7.1.2 Analysis of Network Performance

The power consumption of mobile devices in cellular networks causes a considerable fraction (i.e. 31.2%) of the device energy consumption [CD]<sup>+15</sup>. This again depends on the KPIs of the connected network. The higher the data rates, the shorter time the interface must be active, resulting in considerable energy savings. To analyze the cost under realistic conditions, the performance of the cellular network is analyzed in a crowd-sensing based manner for an urban and train scenario in Sections 5.1 and 5.2 respectively. Irregularities observed in the collected data were analyzed using dedicated measurement studies in Section 5.3. In the following the main conclusions are summarized along the research questions posed in the beginning.

*RQ 2.1: What are the parameters affecting cellular service quality and user-perceived network performance when being mobile?* The crowd-sensing study shows that the performance of the cellular network mainly depends on the available network technology. The received signal strength has only marginal influence on RTT and throughput as long as a reasonable signal strength is observed. Measurements of cell sizes show the coverage and number of cells within the surveyed region. The collected data set is the first to the best knowledge of the author covering both cellular and WiFi networks, containing signal strength, RTT, and throughput values for the surveyed locations. Thus, the subsequent study merging both networks types with the presented energy models was only made possible using these measurements.

*RQ 2.2: How does cellular network access on trains differ from general mobile network access, and how can the network performance be predicted?* The measurements conducted on trains show similar results to the urban crowd-sensing study, although showing higher variability due to movement velocity. Both the localization and accurate measurement proved more challenging due to frequent handovers and changes in network availability. Major impediment of mobile network access on trains is packet loss, sometimes causing every second messages to be lost, although the cellular network employs extensive retransmission functionality. Caused by these retransmissions, a temporary higher latency is observed on the connection. Knowing the availability of network technologies and loss rate, the usability of the network for different service classes was predicted.

*RQ 2.3: What is the influence of network structure and management on end-user perceived network performance?* Abnormalities detected in the collected data were analyzed using dedicated studies running an extended set of stationary measurements. Thus, the influ-

ence of network management on the allocation of mobile devices to PoPs was found to significantly influence RTT performance. Instead of being allocated to the closest PoP, a random user allocation was observed. In case of more distant PoPs, an RTT increase of more than 58 % was identified in more than 80 % of the time. This is also reflected in the response times of popular web sites, showing a similar increase. This effect is also visible in related work [BRF14], but was not explained before. Improving the user-to-PoP allocation by always routing traffic via the closest PoP is expected to significantly improve page load times.

Summarizing, the performance of the cellular networks as used by the participants of the crowd-sensing study was profiled. Based on the collected data, the relative independence between signal strength and network performance was derived. Hence, based on the coverage and packet losses the cellular service quality and usability for different service classes can be predicted. The dedicated measurements focusing on the identification of observed anomalies show the effect of network management in form of PoP allocation considerably influencing the end-to-end RTT. The identified latency overhead is 58 % in over 80 % of the time, which is confirmed by measurements of the page load time of the 25 most popular websites.

### 7.1.3 *Influence of Traffic Management on Energy Consumption*

Combining the above observations on cellular network performance with the smartphone power models, the energy consumption of network transmissions was derived and energy optimizations as proposed in literature are analyzed. Based on the general conclusions drawn from the described measurements and literature, suggestions on improving both performance and energy consumption of 5G networks were given. The conclusions are again structured along the research questions posed initially.

*RQ 3.1: What is the energy cost of mobile communication for a regular smartphone user, and how is this affected by smartphone-based energy conservation approaches?* Based on the smartphone power models, the location based cellular service quality, and additionally measured WiFi availability and performance, the effectiveness of two selected mobile energy conservation approaches was analyzed and compared to the energy consumption of an unmodified system. The results show that the selected approaches developed for 3G networks show considerably lower energy savings than claimed in the original publications. These lower savings are allocated to improvements in cellular network access. Hence, it is concluded that results obtained by earlier network optimization studies should be considered with care. Large-scale measurements of the actual network performance are required to derive correct network performance values for contemporary networks. Based on these, optimization potentials can be assessed. Generally, increasing network throughput reduces transmission times, thus enabling energy savings on the mobile device.

*RQ 3.2: What is the potential of emerging network technologies on network infrastructure and mobile devices considering performance and energy consumption?* The goal of reducing latency and increasing network throughput as targeted for 5G networks results in the conclusion that services must move closer to the user. This applies to both content and network functionality. The minimum requirement to achieve sub-ms RTTs is to place services within each city. On the cellular network side, this requires establishing PoPs in each city connected with the public Internet. Taking this approach further, virtualized services may be run on extended HGWs, thus even closer to the end user. This requires a large scale deployment of SDN and NFV solutions to flexibly redirect traffic and migrate services depending on user mobility. Considering that thus network utilization can be increased, further energy savings are expected.

Concluding, the energy efficiency of mobile communication mainly depends on the active time of the interface, the implementation of energy conservation techniques like sleep modes on the mobile device, and returning to low power modes with minimal delays after activity periods. By increasing throughput, and thus reducing transmission times, energy on smartphones can be saved. In fixed communications networks, the high idle consumption and missing energy saving states prevent effective energy conservation approaches. Nonetheless, increasing virtualization promises to increase network utilization, thus reducing the relative cost of communication. Still, the participation of device vendors including low-power modes in their wired network devices is highly desired to leverage the full energy saving potential, in particular during periods of low demand.

#### 7.1.4 Contributions

The contributions presented in this work are the following: load-dependent power models for various devices were calibrated and the performance of wireless network performance was measured and evaluated. Based on these the influence of empirical traffic patterns on the power consumption of mobile communication was derived.

Power models for two new device classes (SBCs, OpenFlow switches) and traffic management approaches (MPTCP, SDN) were calibrated. The power consumption of at least two devices in each class has been exemplarily derived, thus establishing power models for further analysis and optimization of communication network infrastructure. Similarly, the power consumption of MPTCP on smartphones for the case of CBR streaming was derived. The defining characteristics of the analyzed device classes are worked out, thus deriving their strength and weaknesses for different use cases.

The cellular network performance for an urban scenario was analyzed using a crowd-sensing study. The collected data is available to the research community on a public web server in the form of compiled map tiles. The performance of cellular networks on trains was analyzed in an extensive measurement study. Based on these measurements, the feasibility of predicting the cellular service quality on trains was established. Anomalies

detected within the studies were analyzed using dedicated measurements. From these, the previously unknown influence of traffic management and user allocation to PoPs on cellular service quality was derived and published.

Combining the energy models and network performance measurements, the first reproducible comparison of two mobile energy conservation approaches based on empirical data and realistic power models was presented. For this, an energy evaluation environment was implemented and described in this thesis. Based on the observations obtained from literature and presented in this thesis, promising approaches increasing both performance and energy efficiency in future networks were derived.

#### 7.1.5 *Conclusions*

In the presented work, a comprehensive overview of power consumption and performance of communication networks was given. Previously available MPTCP power models were extended with the analysis of CBR streaming on two different smartphones and network interfaces. In addition to the power models found in literature, models describing the power consumption of SDN switches using the OpenFlow protocol were developed. Finally, with power models for SBCs the energy consumption of a new class of devices was derived.

The location-based cellular network performance was analyzed in an urban scenario. Based on the results it was concluded that RTT and throughput within a network mainly depend on the available network technology instead of the signal strength, as often assumed. A further study analyzing the cellular network performance on trains shows that knowing the location, the usability of the cellular network for different service classes can be predicted. Finally, using dedicated, stationary measurements, the influence of network management on the end-to-end RTT of cellular networks was determined. It was found that relatively simple modifications in the cellular network management can significantly improve latency for a large number of connections.

Finally, based on the combined cellular and WiFi network performance data set in combination with the device power models, the effectiveness of two energy optimization schemes as described in literature is analyzed. The presented energy evaluation framework presents an approach for comparing different energy optimization approaches based on empirically derived network performance. Thus, for the first time reproducible comparisons of existing and future mobile energy optimization approaches are possible. Extending these observations with additional energy models of fixed network devices, the optimization potential of network virtualization and decentralization approaches concerning both performance and energy consumption are inferred. Thus, promising approaches further improving performance and energy consumption of fixed and mobile network access are derived.

## 7.2 OUTLOOK

The constantly increasing demand in network services leads to a growing energy consumption of fixed and wireless networks [BBD+11]. Over the last years, a large number of optimizations have been proposed, and partially integrated into fixed and wireless networks, both on the client and network side. Still, a considerable optimization potential has been identified, in particular considering the idle consumption of infrastructure networks.

Hardware vendors of fixed network infrastructure are urged to improve their energy footprint. This may be achieved by using more efficient hardware, employing power saving sleep modes, and providing remote configuration options to shut down interfaces and full devices when temporarily not required. The evolution of smartphones has shown that considerable improvements are possible without sacrificing performance. Still, also smartphones are expected to benefit from further improvements like increased wireless bandwidth, shortening data transmission durations, and thus reducing the overall power consumption. Here, additional spectrum, advanced modulation schemes, multi user MIMO (MU-MIMO), NOMA, and CR promise a large optimization potential.

To compare the power consumption of different network optimization approaches, models defining their load, the active interfaces and nodes are required. These should include, besides the overall network traffic, its temporal patterns and the system utilization on the respective node. This is particularly important when utilizing idle resources at the network edge (e.g. HGWs), being strictly resource limited. Thus, the cost and benefit of these approaches can be analytically analyzed before being deployed on a large scale.

These power models and optimization approaches can only be tested successfully when the KPIs of the network are known. Therefore, additional measurements in different networks are required. As this and other works have shown, these cannot be derived from signal strength or other inexpensive measurements. Besides location, time, and connected network, these must also include the location of the remote measurement endpoint, as performance may vary significantly between locally delivered and remote content. A major limitation of capacity loading bandwidth measurements, as most suitable for highly variable networks, is the amount of transferred data. Here, more efficient measurement approaches should be developed, thus permitting the accurate quantification of network performance at low cost.

As one of the approaches improving capacity in 5G is introducing smaller cells, also the relocation of content and network services closer to the user promises to reduce the power consumption, both on the mobile device and within the network. By intelligently controlling which content is available on which HGWs or femto-cells, idle resources may be used to reduce the peak load on the network backbone. Here, the functionality of SDN and NFV provides the required flexibility to dynamically reconfigure the network and place network functionality where optimally suited. By instantiating PEPs or other network functions on the node, the latency of retransmissions may be reduced, or the

quality of content adapted to the currently connected device, thus optimizing the trade-off between power consumption, network capacity, and quality. Still, these approaches require accurate knowledge of network performance and load to reduce the overall energy consumption.



## REFERENCES

---

- [3GPP04] 3rd Generation Partnership Project. *TS 25.331; Technical Specification Group Radio Access Network; Radio Resource Control (RRC) protocol specification (version 3.21.0, Release 1999)*. 2004 (cit. on p. 16).
- [3GPP08] 3rd Generation Partnership Project. *TS 27.007; Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); AT command set for User Equipment (UE) (version 8.5.0, Release 8)*. 2008 (cit. on p. 18).
- [3GPP14] 3rd Generation Partnership Project. *TS 36.304; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode (version 12.2.0, Release 12)*. 2014 (cit. on p. 17).
- [3GPP16] 3rd Generation Partnership Project. *TS 36.331; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification (version 13.0.0, Release 13)*. 2016 (cit. on p. 17).
- [ACO+06] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. “Avoiding Traceroute Anomalies with Paris Traceroute Brice.” In: *ACM Internet Measurement Conference (IMC)*. 2006, pp. 153–158 (cit. on pp. 40, 47).
- [AMW+10] Dennis Abts, Michael R Marty, Philip M Wells, Peter Klausler, and Hong Liu. “Energy Proportional Datacenter Networks.” In: *International Symposium on Computer Architecture (ICSA)*. 2010, pp. 338–347 (cit. on p. 31).
- [And15] *Android 6.0 Changes*. 2015. URL: <https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html> (cit. on p. 164).
- [BBD+11] Raffaele Bolla, Roberto Bruschi, Franco Davoli, and Flavio Cucchiatti. “Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-aware Fixed Network Infrastructures.” In: *IEEE Communications Surveys and Tutorials* 13.2 (2011), pp. 223–244 (cit. on pp. 2, 26, 31, 172, 175, 181).
- [BBD+14] Raffaele Bolla, Roberto Bruschi, Franco Davoli, Pasquale Donadio, Leonardo Fialho, Martin Collier, Alfio Lombardo, Diego Reforgiato, Vincenzo Riccobene, and Tivadar Szemethy. “A Northbound Interface for Power Management in Next Generation Network Devices.” In: *IEEE Communications Magazine* 52.1 (2014), pp. 149–157 (cit. on pp. 30, 38, 171).

- [BCL+10] Aruna P. Bianzino, Claude Chaudet, Federico Larroca, Dario Rossi, and Jean-Louis Rougier. "Energy-Aware Routing: a Reality Check." In: *IEEE Globecom Workshop on Green Communications*. 2010, pp. 1422–1427 (cit. on pp. 30, 38).
- [BKW+13] Andreas Blenk, Wolfgang Kellerer, Florian Wamser, Thomas Zinner, and Phuoc Tran-Gia. "Dynamic HTTP Download Scheduling with Respect to Energy Consumption." In: *Tyrrhenian International Workshop on Digital Communications - Green ICT (TIWDC)*. 2013, pp. 1–6 (cit. on p. 114).
- [BMS13] Emily Blem, Jaikrishnan Menon, and Karthikeyan Sankaralingam. "Power Struggles: Revisiting the RISC vs . CISC Debate on Contemporary ARM and x86 Architectures." In: *IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 2013, pp. 1–12 (cit. on p. 27).
- [BMV10] Aruna Balasubramanian, Ratul Mahajan, and Arun Venkataramani. "Augmenting Mobile 3G using WiFi." In: *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 2010, pp. 209–222 (cit. on pp. 48, 51).
- [BPB11] Sébastien Barré, Christoph Paasch, and Olivier Bonaventure. "MultiPath TCP: From Theory to Practice." In: *IFIP Networking Conference (NETWORKING)*. 2011 (cit. on pp. 19, 33, 94).
- [BRF14] Nico Becker, Amr Rizk, and Markus Fidler. "A Measurement Study on the Application-level Performance of LTE." In: *IFIP Networking Conference (NETWORKING)*. 2014 (cit. on pp. 43, 47, 137, 178).
- [CCD+14] Xiang Chen, Yiran Chen, Mian Dong, and Charlie Zhang. "Demystifying Energy Usage in Smartphones." In: *Design Automation Conference (DAC)*. 2014 (cit. on pp. 2, 32, 38).
- [CCY+15] Aleksandra Checko, Henrik L. Christiansen, Ying Yan, Lara Scolari, Georgios Kardaras, Michael S. Berger, and Lars Dittmann. "Cloud RAN for Mobile Networks - A Technology Overview." In: *IEEE Communications Surveys and Tutorials* 17.1 (2015), pp. 405–426 (cit. on p. 166).
- [CDJ+15] Xiaomeng Chen, Ning Ding, Abhilash Jindal, Y. Charlie Hu, Maruti Gupta, and Rath Vannithamby. "Smartphone Energy Drain in the Wild." In: *ACM SIGMETRICS Performance Evaluation Review* 43.1 (2015), pp. 151–164 (cit. on pp. 2, 25, 31, 94, 113, 147, 177).
- [CFG+12] Antonio Capone, Ilario Filippini, Bernd Gloss, and Ulrich Barth. "Rethinking Cellular System Architecture for Breaking Current Energy Efficiency Limits." In: *Sustainable Internet and ICT for Sustainability (SustainIT)*. 2012, pp. 1–5 (cit. on p. 147).

- [CGB05] Shuguang Cui, Andrea J. Goldsmith, and Ahmad Bahai. “Energy-constrained modulation optimization.” In: *IEEE Transactions on Wireless Communications* 4.5 (2005), pp. 2349–2360 (cit. on p. 2).
- [CH10] Aaron Carroll and Gernot Heiser. “An Analysis of Power Consumption in a Smartphone.” In: *USENIX annual technical conference*. 2010, pp. 1–14 (cit. on pp. 31, 38).
- [CH13] Aaron Carroll and Gernot Heiser. “The Systems Hacker’s Guide to the Galaxy Energy Usage in a Modern Smartphone.” In: *Asia-Pacific Workshop on Systems*. 2013, pp. 1–7 (cit. on pp. 31, 37, 38).
- [Cis16] *Cisco Visual Networking Index: Forecast and Methodology, 2015-2020*. Tech. rep. Cisco, 2016. URL: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html> (cit. on pp. 1, 13, 25, 31, 113, 172, 175).
- [CLG+13] Yung-Chih Chen, Yeon-Sup Lim, Richard J. Gibbens, Erich M. Nahum, Ramin Khalili, and Don Towsley. “A Measurement-based Study of MultiPath TCP Performance over Wireless Networks.” In: *ACM Internet Measurement Conference (IMC)*. 2013, pp. 1–14 (cit. on pp. 34, 36, 94).
- [CMN12] Luca Chiaraviglio, Marco Mellia, and Fabio Neri. “Minimizing ISP Network Energy Cost: Formulation and Solutions.” In: *IEEE/ACM Transactions on Networking* 20.2 (2012), pp. 463–476 (cit. on pp. 93, 147, 170).
- [CM012] *C-RAN: The Road Towards Green Radio Access Network*. Tech. rep. China Mobile, 2012. URL: [http://labs.chinamobile.com/cran/wp-content/uploads/CRAN\\_white\\_paper\\_v2\\_5\\_EN.pdf](http://labs.chinamobile.com/cran/wp-content/uploads/CRAN_white_paper_v2_5_EN.pdf) (cit. on pp. 1, 25, 53, 165, 170).
- [Coo77] R. Dennis Cook. “Detection of Influential Observation in Linear Regression.” In: *Technometrics* 19.1 (1977), pp. 15–18 (cit. on p. 104).
- [CSB+08] Joseph Chabarek, Joel Sommers, Paul Barford, Cristian Estan, David Tsiang, and Stephen Wright. “Power Awareness in Network Design and Routing.” In: *IEEE International Conference on Computer Communications (INFOCOM)*. 2008 (cit. on pp. 29, 38, 170).
- [CTM+13] Maria Charalambides, Daphné Tuncer, Lefteris Mamatras, and George Pavlou. “Energy-Aware Adaptive Network Resource Management.” In: *International Symposium on Integrated Network Management (IM)*. 2013, pp. 1–9 (cit. on pp. 3, 53).
- [CTS+11] Yohan Chon, Elmurod Talipov, Hyojeong Shin, and Hojung Cha. “Mobility Prediction-based Smartphone Energy Optimization for Everyday Location Monitoring.” In: *ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 2011, pp. 82–95 (cit. on p. 120).

- [CYM13a] Shengyang Chen, Zhenhui Yuan, and Gabriel-Miro Muntean. "A Traffic Burstiness-based Offload Scheme for Energy Efficiency Deliveries in Heterogeneous Wireless Networks." In: *IEEE Globecom Workshops*. 2013, pp. 1–6 (cit. on pp. 34, 36, 38, 46, 51).
- [CYM13b] Shengyang Chen, Zhenhui Yuan, and Gabriel-Miro Muntean. "An Energy-aware Multipath-TCP-based Content Delivery Scheme in Heterogeneous Wireless Networks." In: *IEEE Wireless Communications and Networking Conference (WCNC)*. 2013, pp. 1291–1296 (cit. on pp. 34, 38, 46, 51, 54, 94).
- [CZX+11] Yan Chen, Shunqing Zhang, Shugong Xu, and Geoffrey Ye Li. "Fundamental Trade-offs on Green Wireless Networks." In: *IEEE Communications Magazine* 49.6 (2011), pp. 30–37 (cit. on p. 166).
- [DAB+06] Luca Dall'Asta, Ignacio Alvarez-Hamelin, Alain Barrat, Alexei Vázquez, and Alessandro Vespignani. "Exploring Networks with Traceroute-like Probes: Theory and Simulations." In: *Theoretical Computer Science* 355.1 (2006), pp. 6–24 (cit. on pp. 40, 47).
- [DBM+10] Katerina Dufkova, Milan Bjelica, Byongkwon Moon, Lukas Kencl, and Jean-Yves Le Boudec. "Energy Savings for Cellular Network with Evaluation of Impact on Data Traffic Performance." In: *European Wireless Conference (EW)*. 2010, pp. 916–923 (cit. on p. 147).
- [DDG+12] Claude Desset, Björn Debaillie, Vito Giannini, Albrecht Fehske, Gunther Auer, Hauke Holtkamp, Wieslawa Wajda, Dario Sabella, Fred Richter, Manuel J. Gonzalez, Henrik Klessig, István Gódor, Magnus Olsson, Muhammad Ali Imran, Anton Ambrosy, and Oliver Blume. "Flexible Power Modeling of LTE Base Stations." In: *IEEE Wireless Communications and Networking Conference (WCNC)*. 2012, pp. 2858–2862 (cit. on p. 170).
- [DG12] Trinh Minh Tri Do and Daniel Gatica-Perez. "Contextual Conditional Models for Smartphone-based Human Mobility Prediction." In: *ACM Conference on Ubiquitous Computing (UbiComp)*. 2012, pp. 163–172 (cit. on p. 114).
- [DNS+14] Shuo Deng, Ravi Netravali, Anirudh Sivaraman, and Hari Balakrishnan. "WiFi, LTE, or Both? Measuring Multi-Homed Wireless Internet Performance." In: *ACM Internet Measurement Conference (IMC)*. 2014, pp. 1–14 (cit. on p. 94).
- [DWC+13] Ning Ding, Daniel Wagner, Xiaomeng Chen, Y. Charlie Hu, and Andrew Rice. "Characterizing and Modeling the Impact of Wireless Signal Strength on Smartphone Battery Drain." In: *International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*. 2013, pp. 29–40 (cit. on pp. 32, 38).
- [Eri16] *Mobility Report*. Tech. rep. Ericsson, 2016. URL: <https://www.ericsson.com/res/docs/2016/ericsson-mobility-report-2016.pdf> (cit. on p. 94).

- [FHL+14] Moufida Feknous, Thierry Houdoin, Bertrand Le Guyader, Joseph De Biasio, Annie Gravey, and Jose Alfonso Torrijos Gijón. “Internet Traffic Analysis: A Case Study from two Major European Operators.” In: *IEEE Symposium on Computers and Communications (ISCC)*. 2014, pp. 1–7 (cit. on p. 118).
- [Fis16] Florian Fischer. “Analysis of the Cellular Service Quality in Trains.” Bachelor’s Thesis. Technische Universität Darmstadt, 2016, pp. 1–80 (cit. on pp. 123, 128, 133, 135).
- [FLG+16] Jose Oscar Fajardo, Fidel Liberal, Ioannis Giannoulakis, Emmanouil Kafetzakis, Vincenzo Pii, Irena Trajkovska, Thomas Michael Bohnert, Leonardo Goratti, Roberto Riggio, Javier Garcia Lloreda, Pouria S. Khodashenas, Michele Paolino, Pavel Bliznakov, Jordi Perez-Romero, Claudio Meani, Ioannis Chochliouros, and Maria Belesioti. “Introducing Mobile Edge Computing Capabilities through Distributed 5G Cloud Enabled Small Cells.” In: *Mobile Networks and Applications* 21 (2016), pp. 564–574 (cit. on p. 55).
- [FS16] Toni Feather and John Swanson. *Mobile connectivity research study*. Tech. rep. Steer Davies Gleave, 2016 (cit. on pp. 44, 114, 122, 123).
- [FTF+16] Matteo Fiorani, Sibel Tombaz, Fabricio Farias, Lena Wosinska, and Paolo Monti. “Joint Design of Radio and Transport for Green Residential Access Networks.” In: *IEEE Journal on Selected Areas in Communications* 34.4 (2016), pp. 812–822 (cit. on pp. 165, 166).
- [GCN05] Chamara Gunaratne, Ken Christensen, and Bruce Nordman. “Managing Energy Consumption Costs in Desktop PCs and LAN Switches with Proxying, Split TCP Connections, and Scaling of Link Speed.” In: *International Journal of Network Management* 15.5 (2005), pp. 297–310 (cit. on p. 80).
- [GDK+13] Vijay Gabale, UmaMaheswari Devi, Ravi Kokku, Vinay Kolar, Mukundan Madhavan, and Shivkumar Kalyanaraman. “Async: De-congestion and Yield Management in Cellular Data Networks.” In: *IEEE International Conference on Network Protocols (ICNP)*. 2013, pp. 1–10 (cit. on pp. 46, 48, 49, 51).
- [GGS04] Maruti Gupta, Satyajit Grover, and Suresh Singh. “A Feasibility Study for Power Management in LAN Switches.” In: *IEEE International Conference on Network Protocols (ICNP)*. 2004, pp. 361–371 (cit. on pp. 53, 170).
- [GK15] Vijay Gabale and Dilip Krishnaswamy. “MobInsight: On Improving the Performance of Mobile Apps in Cellular Networks.” In: *ACM International Conference on World Wide Web (WWW)*. 2015, pp. 355–365 (cit. on pp. 46, 51).
- [GKS+13] Christian Gross, Fabian Kaup, Dominik Stingl, Björn Richerzhagen, David Hausheer, and Ralf Steinmetz. “EnerSim: An Energy Consumption Model for Large-Scale Overlay Simulators.” In: *IEEE Conference on Local Computer Networks (LCN)*. 2013, pp. 252–255 (cit. on p. 153).

- [GPN13] Navin Gautam, Henrik Petander, and Joseph Noel. "A Comparison of the Cost and Energy Efficiency of Prefetching and Streaming of Mobile Video." In: *ACM Workshop on Mobile Video (MoVid)*. 2013, pp. 1–7 (cit. on pp. 46, 51).
- [GPS+10] Alexandre Gerber, Jeffrey Pang, Oliver Spatscheck, and Shobha Venkataraman. "Speed Testing without Speed Tests: Estimating Achievable Download Speed from Passive Measurements." In: *ACM Internet Measurement Conference (IMC)*. 2010, pp. 424–430 (cit. on pp. 42, 47).
- [GRR+12] Karina Gomez, Roberto Riggio, Tinku Rasheed, Daniele Miorandi, and Fabrizio Granelli. "Energino: a Hardware and Software Solution for Energy Consumption Monitoring." In: *International Workshop on Wireless Network Measurements (Winmee)*. 2012, pp. 311–317 (cit. on pp. 28, 37, 38).
- [GWC+15] Utkarsh Goel, Mike P. Wittie, Kimberly C. Claffy, and Andrew Le. "Survey of End-to-End Mobile Network Measurement Testbeds." In: *IEEE Communications Surveys and Tutorials* 18.1 (2015), pp. 105–123 (cit. on p. 114).
- [Hay05] Simon Haykin. "Cognitive Radio: Brain-empowered Wireless Communications." In: *IEEE Journal on Selected Areas in Communications* 23.2 (2005), pp. 201–220 (cit. on p. 166).
- [HBB11] Ziaul Hasan, Hamidreza Boostanimehr, and Vijay K Bharghava. "Green Cellular Networks: A Survey, Some Research Issues and Challenges." In: *IEEE Communications Surveys and Tutorials* 13.4 (2011), pp. 524–540 (cit. on p. 147).
- [HDP09] Helmut Hlavacs, Georges Da Costa, and Jean-Marc Pierson. "Energy Consumption of Residential and Professional Switches." In: *IEEE International Conference on Computational Science and Engineering (CSE)*. 2009, pp. 240–246 (cit. on pp. 29, 38, 53).
- [HPT+09] Brandon Heller, Ben Pfaff, Dan Talayco, David Erickson, Glen Gibb, Guido Appenzeller, Jean Tourrilhes, Justin Pettit, KK Yap, Martin Casado, Masayoshi Kobayashi, Nick McKeown, Peter Balland, Reid Price, Rob Sherwood, and Yannis Manolopoulos. *OpenFlow Switch Specification Version 1.0.0*. Tech. rep. OpenFlow Switch Consortium, 2009. URL: <http://archive.openflow.org/documents/openflow-spec-v1.0.0.pdf> (cit. on p. 20).
- [HQG+12] Junxian Huang, Feng Quian, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. "A Close Examination of Performance and Power Characteristics of 4G LTE Networks." In: *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 2012, pp. 225–238 (cit. on pp. 32, 33, 35, 38, 44, 48, 101).

- [HQG+13] Junxian Huang, Feng Quian, Yihua Guo, Yuanyuan Zhou, Qiang Xu, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. “An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance.” In: *ACM SIGCOMM Conference*. 2013, pp. 363–374 (cit. on pp. 42, 47).
- [HQM+12] Junxian Huang, Feng Qian, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. “Screen-off Traffic Characterization and Optimization in 3G/4G Networks.” In: *ACM Internet Measurement Conference (IMC)*. 2012, pp. 357–363 (cit. on pp. 5, 32, 33, 38, 46, 50, 51, 147, 158, 159, 163, 172).
- [HSJ+12] Sangtae Ha, Soumya Sen, Carlee Joe-Wong, Youngbin Im, and Mung Chiang. “TUBE: Time-Dependent Pricing for Mobile Data.” In: *ACM SIGCOMM Conference*. 2012, pp. 247–258 (cit. on pp. 48, 51).
- [HSM+10] Brandon Heller, Srinu Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Suajta Banerjee, and Nick McKeown. “ElasticTree: Saving Energy in Data Center Networks.” In: *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 2010, pp. 1–16 (cit. on pp. 30, 38).
- [Ipf10] *Iperf*. 2010. URL: <http://software.es.net/iperf/> (cit. on pp. 41, 47).
- [ITU14] ITU. *The Tactile Internet: ITU-T Technology Watch Report*. Tech. rep. 2014 (cit. on p. 144).
- [IWF13] Selim Ickin, Katarzyna Wac, and Markus Fiedler. “QoE-Based Energy Reduction by Controlling the 3G Cellular Data Traffic on the Smartphone.” In: *ITC Specialist Seminar on Energy Efficient and Green Networking (SSEGN)*. 2013, pp. 13–18 (cit. on pp. 33, 38, 46, 50, 51, 147, 158, 159, 163, 172).
- [JD02] Manish Jain and Constantinos Dovrolis. “Pathload: a measurement tool for end-to-end available bandwidth.” In: *Passive and Active Measurements (PAM) Workshop*. 2002, pp. 1–12 (cit. on pp. 41, 47).
- [JKM+13] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jonathan Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. “B4: Experience with a Globally-Deployed Software Defined WAN.” In: *ACM SIGCOMM Conference*. 2013, pp. 3–14 (cit. on p. 164).
- [JP13] Raj Jain and Subharthi Paul. “Network Virtualization and Software Defined Networking for Cloud Computing: A Survey.” In: *IEEE Communications Magazine* 51.11 (2013), pp. 24–31 (cit. on p. 167).
- [JVO+13] Mateusz Jarus, Sébastien Varette, Ariel Oleksiak, and Pascal Bouvry. “Performance Evaluation and Energy Efficiency of High-Density HPC Platforms Based on Intel, AMD and ARM Processors.” In: *Energy Efficiency in Large Scale Distributed Systems*. Ed. by Jean-Marc Pierson, Georges Da Costa, and Lars Dittmann. Vol. 8046. Lecture Notes in Computer Science November

2012. Springer Berlin Heidelberg, 2013, pp. 182–200 (cit. on pp. 27, 28, 38, 73).
- [KBS+11] Jonathan G. Koomey, Stephen Berard, Marla Sanchez, and Henry Wong. “Implications of Historical Trends in the Electrical Efficiency of Computing.” In: *IEEE Annals of the History of Computing* 33.3 (2011), pp. 46–54 (cit. on pp. 72, 75, 176).
- [KFH17] Fabian Kaup, Florian Fischer, and David Hausheer. “Measuring and Predicting Cellular Network Quality on Trains.” In: *International Conference on Networked Systems (NetSys)*. 2017, pp. 1–8 (cit. on p. 124).
- [KGH14] Fabian Kaup, Philip Gottschling, and David Hausheer. “PowerPi: Measuring and Modeling the Power Consumption of the Raspberry Pi.” In: *IEEE Conference on Local Computer Networks (LCN)*. 2014, pp. 236–243 (cit. on p. 56).
- [KH02] Stas Khirman and Peter Henriksen. “Relationship between Quality-of-Service and Quality-of-Experience for Public Internet Service.” In: *Workshop on Passive and Active Measurement Conference (PAM Workshop)*. 2002, pp. 1–6 (cit. on p. 144).
- [KJH15] Fabian Kaup, Florian Jomrich, and David Hausheer. “Demonstration of NetworkCoverage – A Mobile Network Performance Measurement App.” In: *International Conference on Networked Systems (NetSys)*. 2015 (cit. on p. 115).
- [KLW12] Ouldooz B. Karimi, Jiangchuan Liu, and Chonggang Wang. “Seamless Wireless Connectivity for Multimedia Services in High Speed Trains.” In: *IEEE Journal on Selected Areas in Communications* 30.4 (2012), pp. 729–739 (cit. on p. 114).
- [KMB+15] Fabian Kaup, Foivos Michelinakis, Nicola Bui, Joerg Widmer, Katarzyna Wac, and David Hausheer. “Behind the NAT – A Measurement Study of Cellular Service Quality.” In: *Conference on Network and Service Management (CNSM)*. 2015, pp. 228–236 (cit. on pp. 115, 136).
- [KMB+16] Fabian Kaup, Foivos Michelinakis, Nicola Bui, Joerg Widmer, Katarzyna Wac, and David Hausheer. “Assessing the Implications of Cellular Network Performance on Mobile Content Access.” In: *IEEE Transactions on Network and Service Management* 13.2 (2016), pp. 168–179 (cit. on pp. 115, 136).
- [KMH14a] Fabian Kaup, Sergej Melnikowitsch, and David Hausheer. *A Model-based OpenFlow Power Monitoring Framework based on Real-World Measurements*. Tech. rep. PS Technical Report No. PS-TR-2014-02, 2014. URL: <http://www.ps.tu-darmstadt.de/fileadmin/publications/PS-TR-2014-02.pdf> (cit. on p. 76).



- [KMH14b] Fabian Kaup, Sergej Melnikowitsch, and David Hausheer. “Measuring and Modeling the Power Consumption of OpenFlow Switches.” In: *International Conference on Network and Service Management (CNSM)*. 2014, pp. 181–186 (cit. on p. 76).
- [KSB+13] Aman Kansal, Scott Saponas, A. J. Bernheim Brush, Kathryn S. McKinley, Todd Mytkowicz, and Ziola Ryder. “The Latency, Accuracy, and Battery (LAB) Abstraction: Programmer Productivity and Energy Efficiency for Continuous Mobile Context Sensing.” In: *ACM Object-Oriented Programming, Systems, Languages & Applications (OOPSLA)*. 2013, pp. 1–16 (cit. on p. 118).
- [KWR+15a] Fabian Kaup, Matthias Wichtlhuber, Stefan Rado, and David Hausheer. *Analysis and Modeling of the Multipath-TCP Power Consumption for Constant Bitrate Streaming*. Tech. rep. no. PS-TR-2015-02. Darmstadt: Technische Universität Darmstadt, 2015. URL: [http://www.ps.tu-darmstadt.de/nc/publications/publications-details/?pub\\_id=KWR+15-2](http://www.ps.tu-darmstadt.de/nc/publications/publications-details/?pub_id=KWR+15-2) (cit. on p. 94).
- [KWR+15b] Fabian Kaup, Matthias Wichtlhuber, Stefan Rado, and David Hausheer. “Can Multipath TCP Save Energy? A Measuring and Modeling Study of MP-TCP Energy Consumption.” In: *IEEE Conference on Local Computer Networks (LCN)*. 2015, pp. 442–445 (cit. on p. 94).
- [LAH+15] Tao Lin, Tansu Alpcan, Kerry Hinton, and Arun Vishwanath. “A Distributed Multi-Objective Optimisation Framework for Energy Efficiency in Mobile Backhaul Networks.” In: *IEEE Conference on Control Applications (CCA)*. 2015, pp. 1630–1636 (cit. on pp. 30, 38).
- [LCB15] Arthur F. Lorenzon, Marcia C. Cera, and Antonio Carlos S. Beck. “On the influence of static power consumption in multicore embedded systems.” In: *IEEE International Symposium on Circuits and Systems (ISCAS)*. 2015, pp. 1374–1377 (cit. on pp. 3, 27).
- [LCN+14] Yeon-sup Lim, Yung-Chih Chen, Erich M. Nahum, Don Towsley, and Richard J. Gibbens. “Improving Energy Efficiency of MPTCP for Mobile Devices.” In: *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. 2014, pp. 1–13 (cit. on pp. 3, 34, 38, 46, 51, 54, 94).
- [LHHo8] Matthew Luckie, Young Hyun, and Bradley Huffaker. “Traceroute Probe Method and Forward IP Path Inference.” In: *ACM Internet Measurement Conference (IMC)*. 2008, pp. 311–324 (cit. on pp. 40, 47).
- [LHR+12] Tuan Anh Le, Choong Seon Hong, Md. Abdur Razzaque, Sungwon Lee, and Heeyoung Jung. “ecMTCP: An Energy-Aware Congestion Control Algorithm for Multipath TCP.” In: *IEEE Communications Letters* 16.2 (2012), pp. 275–277 (cit. on pp. 34, 46, 51).

- [LMW+15] Weichao Li, Ricky K. P. Mok, Daoyuan Wu, and Rocky K. C. Chang. "On the Accuracy of Smartphone-based Mobile Network Measurement." In: *IEEE International Conference on Computer Communications (INFOCOM)*. 2015, pp. 370–378 (cit. on pp. 40, 47, 117, 137, 139, 167).
- [LPB+15] Andri Lareida, George Petropoulos, Valentin Burger, Michael Seufert, Sergios Soursos, and Burkhard Stiller. "Augmenting Home Routers for Socially-Aware Traffic Management." In: *IEEE Conference on Local Computer Networks (LCN)*. 2015, pp. 347–355 (cit. on pp. 53, 55, 70, 168).
- [LSR+12] Markus Laner, Philipp Svoboda, Peter Romirer-Maierhofer, Navid Nikaein, Fabio Ricciato, and Markus Rupp. "A Comparison Between One-way Delays in Operating HSPA and LTE Networks." In: *Workshop on Wireless Network Measurements (WinMee)*. 2012, pp. 286–292 (cit. on pp. 40, 47, 127, 128).
- [MAB+08] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. "OpenFlow: Enabling Innovation in Campus Networks." In: *ACM SIGCOMM Computer Communication Review* 38.2 (2008), pp. 69–74 (cit. on p. 20).
- [Mas16] Teresa Mastrangelo. *Virtual Reality Check: Are Our Networks Ready for VR?* 2016. URL: <http://blog.advaoptical.com/virtual-reality-check-are-our-networks-ready-for-vr> (cit. on p. 113).
- [MBF+15] Foivos Michelinakis, Nicola Bui, Guido Fioravanti, Joerg Widmer, Fabian Kaup, and David Hausheer. "Lightweight Mobile Bandwidth Availability Measurement." In: *IFIP Networking Conference (NETWORKING)*. 2015, pp. 1–7 (cit. on p. 42).
- [MBF+16] Foivos Michelinakis, Nicola Bui, Guido Fioravanti, Joerg Widmer, Fabian Kaup, and David Hausheer. "Lightweight capacity measurements for mobile networks." In: *Computer Communications* 84 (2016), pp. 73–83 (cit. on pp. 42, 47).
- [MBGoo] Bob Melander, Mats Björkman, and Per Gunningberg. "A New End-to-End Probing and Analysis Method for Estimating Bandwidth Bottlenecks." In: *IEEE Global Telecommunications Conference (GLOBECOM)*. 2000, pp. 415–420 (cit. on pp. 41, 47).
- [Mel14] Sergej Melnikowitsch. "Measuring and Modeling the Power Consumption of Software Defined Networks." Master's Thesis. Technische Universität Darmstadt, 2014 (cit. on pp. 76, 80, 82–85, 89–91).
- [MEMo8] Marian Mohr, Christopher Edwards, and Ben McCarthy. "A Study of LBS Accuracy in the UK and a Novel Approach to Inferring the Positioning Technology Employed." In: *Computer Communications* 31.6 (2008), pp. 1148–1159 (cit. on p. 124).

- [MH15] Maria Malik and Houman Homayoun. “Big Data on Low Power Cores - Are Low Power Embedded Processors a good fit for the Big Data Workloads?” In: *IEEE International Conference on Computer Design (ICCD)*. 2015, pp. 379–382 (cit. on pp. 26–28).
- [MHR03] Matt Mathis, John Heffner, and Raghu Reddy. “Web100: Extended TCP Instrumentation for Research, Education and Diagnosis.” In: *ACM SIGCOMM Computer Communication Review* 33.3 (2003), pp. 69–79 (cit. on p. 42).
- [Mil13] Mark P. Mills. *The Cloud Begins with Coal*. Tech. rep. Digital Power Group, 2013 (cit. on pp. 23, 25, 53).
- [MOF15] Jahanzeb Maqbool, Sangyoon Oh, and Geoffrey C. Fox. “Evaluating ARM HPC Clusters for Scientific Workloads.” In: *Concurrency and Computation: Practice and Experience* 27 (Dec. 2015), pp. 5390–5410 (cit. on pp. 26, 27, 72).
- [MOP+14] Marja Matinmikko, Hanna Okkonen, Marko Palola, Seppo Yrjölä, Petri Aho-kangas, and Miia Mustonen. “Spectrum Sharing Using Licensed Shared Access: The Concept and its Workflow for LTE-Advanced Networks.” In: *IEEE Wireless Communications* 21.2 (2014), pp. 72–79 (cit. on p. 165).
- [MSB+09] Priya Mahadevan, Puneet Sharma, Sujata Banerjee, and Parthasarathy Ranganathan. “A Power Benchmarking Framework for Network Devices.” In: *IFIP Networking Conference (NETWORKING)*. 2009, pp. 795–808 (cit. on pp. 29, 38, 76).
- [MSB13] Fadoua Mhiri, Kaouthar Sethom, and Ridha Bouallegue. “A Survey on Interference Management Techniques in Femtocell Self-organizing Networks.” In: *Journal of Network and Computer Applications* 36.1 (2013), pp. 58–65 (cit. on p. 167).
- [MSS+15] Christian Meurisch, Alexander Seeliger, Benedikt Schmidt, Immanuel Schweizer, Fabian Kaup, and Max Mühlhäuser. “Upgrading Wireless Home Routers for Enabling Large-Scale Deployment of Cloudlets.” In: *International Conference on Mobile Computing, Applications and Services (MobiCASE)*. 2015, pp. 1–18 (cit. on p. 55).
- [NFV12] Margaret Chiosi, Don Clarke, Peter Willis, Andy Reid, James Feger, Michael Bugenhagen, Waqar Khan, Michael Fargano, Chunfeng Cui, Hui Deng, and Javier Benitez. *Network Functions Virtualisation*. Tech. rep. 1. 2012 (cit. on pp. 22, 167).
- [NGNM15] *5G White Paper*. Tech. rep. NGMN Alliance, 2015. URL: [https://www.ngmn.org/uploads/media/NGMN\\_5G\\_White\\_Paper\\_V1\\_0\\_01.pdf](https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0_01.pdf) (cit. on p. 2).

- [NHH16] Leonhard Nobach, Oliver Hohlfeld, and David Hausheer. "New Kid on the Block: Network Functions Virtualization: From Big Boxes to Carrier Clouds." In: *ACM SIGCOMM Computer Communication Review* July (2016), pp. 1–8 (cit. on p. 167).
- [NL13] Jose Nunez-Yanez and Geza Lore. "Enabling Accurate Modeling of Power and Energy Consumption in an ARM-based System-on-Chip." In: *Microprocessors and Microsystems* 37.3 (2013), pp. 319–332 (cit. on pp. 27, 38).
- [NN08] Anthony J. Nicholson and Brian D. Noble. "BreadCrumbs: Forecasting Mobile Connectivity." In: *ACM International Conference on Mobile Computing and Networking (MobiCom)*. 2008, pp. 46–57 (cit. on pp. 48, 51).
- [NRH17] Leonhard Nobach, Benedikt Rudolph, and David Hausheer. "Benefits of Conditional FPGA Provisioning for Virtualized Network Functions." In: *Workshop on Software-Defined Networking and Network Functions Virtualization for Flexible Network Management (SDNFlex)*. 2017 (cit. on p. 167).
- [NSA+13] Ewa Niewiadomska-Szynkiewicz, Andrzej Sikora, Piotr Arabas, Mariusz Kamola, Krzysztof Malinowski, Przemyslaw Jaskola, and Michal Marks. "Network-Wide Power Management in Computer Networks." In: *ITC Specialist Seminar on Energy Efficient and Green Networking (SSEEGN)*. 2013, pp. 25–30 (cit. on pp. 30, 38).
- [NSS10] Erik Nygren, Ramesh K. Sitaraman, and Jennifer Sun. "The Akamai Network: A Platform for High-performance Internet Applications." In: *ACM SIGOPS Operating Systems Review* 44.3 (2010), pp. 2–19 (cit. on p. 169).
- [ODL14] Anne-Cecile Orgerie, Marcos Dias de Assuncao, and Laurent Lefevre. "A Survey on Techniques for Improving the Energy Efficiency of Large Scale Distributed Systems." In: *ACM Computing Surveys* 46.4 (2014), pp. 1–35 (cit. on p. 25).
- [Ofc15] *The Communications Market Report*. Tech. rep. 2015 (cit. on p. 31).
- [OLG+11] Anne Cécile Orgerie, Laurent Lefèvre, Isabelle Guérin-Lassous, and Dino M. Lopez Pacheco. "ECOFEN: An End-to-end Energy Cost Model and Simulator for Evaluating Power Consumption in Large-scale Networks." In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2011, pp. 1–6 (cit. on pp. 29, 38).
- [OpS] *OpenSignal*. URL: <http://opensignal.com/> (cit. on pp. 44, 47, 113, 115).
- [PB05] Christian Prehofer and Christian Bettstetter. "Self-organization in Communication Networks: Principles and Design Paradigms." In: *IEEE Communications Magazine* 43.7 (2005), pp. 78–85 (cit. on p. 167).

- [PCK15] Apostolos Papageorgiou, Bin Cheng, and Erno Kovacs. “Real-time Data Reduction at the Network Edge of Internet-of-Things Systems.” In: *International Conference on Network and Service Management (CNSM)*. 2015, pp. 284–291 (cit. on pp. 3, 55).
- [PDD+12] Christoph Paasch, Gregory Detal, Fabien Duchene, Costin Raiciu, and Olivier Bonaventure. “Exploring Mobile/WiFi Handover with Multipath TCP.” In: *ACM SIGCOMM workshop on Cellular Networks (CellNet)*. 2012, pp. 1–6 (cit. on pp. 34, 38, 53, 54, 94, 169).
- [PEK11] Christopher Pluntke, Lars Eggert, and Niko Kiukkonen. “Saving Mobile Device Energy with Multipath TCP.” In: *ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*. 2011, pp. 1–6 (cit. on pp. 34, 38, 46, 51).
- [PFC+12] Luca Prete, Fabio Farina, Mauro Campanella, and Andrea Biancini. “Energy Efficient Minimum Spanning Tree in OpenFlow Networks.” In: *European Workshop on Software Defined Networks (EWSDN)*. 2012, pp. 36–41 (cit. on pp. 30, 75).
- [PLH+11] Ben Pfaff, Bob Lantz, Brandon Heller, Casey Barker, Dan Cohn, Dan Talayco, David Erickson, Edward Crabbe, Glen Gibb, Guido Appenzeller, Jean Tourrilhes, Justin Pettit, K. K. Yap, Leon Poutievski, Martin Casado, Masahiko Takahashi, Masayoshi Kobayashi, Nick McKeown, Peter Balland, Rajiv Ramanathan, Reid Price, Rob Sherwood, Saurav Das, Tatsuya Yabe, Yiannis Yakoumis, and Zoltán Lajos Kis. *OpenFlow Switch Specification Version 1.1.0*. Tech. rep. 2011. URL: <http://www.openflow.org/> (cit. on pp. 30, 77, 79).
- [PMD+03] Ravi Prasad, Margaret Murray, Constantinos Dovrolis, and KC Claffy. “Bandwidth Estimation: Metrics, Measurement Techniques, and Tools.” In: *IEEE Network* 17.6 (2003), pp. 27–35 (cit. on pp. 18, 41, 47).
- [Pos81] J. Postel. *ICMP*. Tech. rep. 1981. URL: <https://tools.ietf.org/html/rfc792> (cit. on pp. 39, 47).
- [PW12] Tobias Pögel and Lars Wolf. “Analysis of Operational 3G Network Characteristics for Adaptive Vehicular Connectivity Maps.” In: *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)* (2012), pp. 121–128 (cit. on pp. 44, 47).
- [Qua13] *Qualcomm 1000x data challenge*. 2013. URL: <https://www.qualcomm.com/invention/1000x> (cit. on pp. 143, 148, 165).
- [QWG+10] Feng Qian, Zhaoguang Wang, Alexandre Gerber, Z. Morley Mao, Subhabrata Sen, and Oliver Spatscheck. “TOP: Tail Optimization Protocol For Cellular Radio Resource Allocation.” In: *International Conference on Network Protocols (ICNP)*. 2010, pp. 285–294 (cit. on pp. 33, 38, 46, 51).

- [Rad14] Stefan Rado. "Optimizing the Energy Efficiency of Multipath TCP for Mobile Devices." Bachelor's Thesis. Technische Universität Darmstadt, 2014 (cit. on pp. 94, 98).
- [RBH15] Julius Rückert, Jeremias Blendin, and David Hausheer. "Software-Defined Multicast for Over-the-Top and Overlay-based Live Streaming in ISP Networks." In: *Journal of Network and Systems Management* 23.2 (2015), pp. 280–308 (cit. on pp. 92, 164).
- [RDC+15] Filippo Rebecchi, Marcelo Dias de Amorim, Vania Conan, Andrea Passarella, Raffaele Bruno, and Marco Conti. "Data Offloading Techniques in Cellular Networks: A Survey." In: *IEEE Communications Surveys & Tutorials* 17.2 (2015), pp. 580–603 (cit. on p. 25).
- [RHRo8] Fabio Ricciato, Eduard Hasenleithner, and Peter Romirer-Maierhofer. "Traffic Analysis at Short Time-Scales: An Empirical Case Study from a 3G Cellular Network." In: *IEEE Transactions on Network and Service Management* 5.1 (2008), pp. 11–21 (cit. on pp. 40, 47, 114).
- [RM11] Barath Raghavan and Justin Ma. "The Energy and Emery of the Internet." In: *ACM Workshop on Hot Topics in Networks (Hotnets)*. 2011, pp. 1–6 (cit. on pp. 22, 23).
- [RPB+12] Costin Raiciu, Christoph Paasch, Sebastien Barre, Alan Ford, Michio Honda, Fabien Duchene, Olivier Bonaventure, and Mark Handley. "How Hard Can It Be? Designing and Implementing a Deployable Multipath TCP." In: *USENIX Symposium of Networked Systems Design and Implementation (NSDI)*. 2012, pp. 1–14 (cit. on p. 34).
- [RPO+12] Upendra Rathnayake, Henrik Petander, Maximilian Ott, and Aruna Seneviratne. "EMUNE: Architecture for Mobile Data Transfer Scheduling with Network Availability Predictions." In: *Mobile Networks and Applications* 17.2 (2012), pp. 216–233 (cit. on pp. 48, 51).
- [RRB+03] Vinaj J. Ribeiro, Rudolf H. Riedi, Richard G. Baranjuk, Jiri Navratil, and Les Cottrell. "PathChirp: Efficient Available Bandwidth Estimation for Network Paths." In: *Passive and Active Measurement Conference (PAM)*. 2003, pp. 1–11 (cit. on pp. 41, 47).
- [RSU+12] Charalampos Rotsos, Nadi Sarrar, Steve Uhlig, Rob Sherwood, and Andrew W. Moore. "OFLOPS: An Open Framework for OpenFlow Switch Evaluation." In: *International Conference on Passive and Active Measurement (PAM)*. 2012, pp. 85–95 (cit. on p. 30).
- [San15] Sandvine Incorporated ULC. *Global Internet Phenomena Spotlight: Encrypted Internet Traffic*. Tech. rep. 2015 (cit. on p. 168).

- [SBC+09] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Cáceres, and Nigel Davies. “The case for VM-based cloudlets in mobile computing.” In: *IEEE Pervasive Computing* 8.4 (2009), pp. 14–23 (cit. on p. 55).
- [SBS+15] Shweta Sagari, Samuel Baysting, Dola Saha, Ivan Seskar, Wade Trappe, and Dipankar Raychaudhuri. “Coordinated Dynamic Spectrum Management of LTE-U and Wi-Fi Networks.” In: *International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. 2015, pp. 209–220 (cit. on p. 165).
- [SCG+14] Ankit Singla, Balakrishnan Chandrasekaran, P. Brighten Godfrey, and Bruce Maggs. “The Internet at the Speed of Light.” In: *ACM Workshop on Hot Topics in Networks (HotNets)*. 2014, pp. 1–7 (cit. on p. 139).
- [Sch16] Thomas Schnabel. “Simulative Comparison of Mobile Energy Conservation Schemes.” Master’s Thesis. Technische Universität Darmstadt, 2016 (cit. on pp. 149, 151–153, 156, 159–162).
- [Sen] *Sensorly*. URL: <http://sensorly.com/> (cit. on pp. 44, 47, 113, 115).
- [SGR+11] Dominik Stingl, Christian Groß, Julius Rückert, Leonhard Nobach, Alexandra Kovacevic, and Ralf Steinmetz. “PeerfactSim.KOM: A Simulation Framework for Peer-to-Peer Systems.” In: *IEEE International Conference on High Performance Computing & Simulation (HPCS)*. 2011, pp. 577–584 (cit. on p. 153).
- [SMS13] Sebastian Sonntag, Jukka Manner, and Lennart Schulte. “Netradar – Measuring the Wireless World.” In: *Wireless Network Measurements*. 2013, pp. 29–34 (cit. on pp. 115, 150).
- [SOC12] Varun Singh, Jörg Ott, and Igor D. D. Curcio. “Predictive Buffering for Streaming Video in 3G Networks.” In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2012, pp. 1–10 (cit. on pp. 48, 51).
- [SRS16] Andreas Selinger, Karl Rupp, and Siegfried Selberherr. “Evaluation of Mobile ARM-Based SoCs for High Performance Computing.” In: *High Performance Computing Symposium (HPC)*. 2016, pp. 1–7 (cit. on pp. 26–28, 38, 55, 72).
- [SSM13] Sebastian Sonntag, Lennart Schulte, and Jukka Manner. “Mobile Network Measurements — It’s not all about Signal Strength.” In: *IEEE Wireless Communications and Networking Conference (WCNC)*. 2013, pp. 4624–4629 (cit. on pp. 44, 47, 113, 127).
- [TCD03] Ajay Tirumala, Les Cottrell, and Tom Dunigan. “Measuring end-to-end bandwidth with Iperf using Web100.” In: *Passive and Active Measurement Conference (PAM)*. 2003, pp. 1–8 (cit. on pp. 42, 47).
- [Tra] *Traceroute*. URL: <http://traceroute.sourceforge.net/> (cit. on pp. 39, 47).

- [TT13] Bogdan Marius Tudor and Yong Meng Teo. "On Understanding the Energy Consumption of ARM-based Multicore Servers." In: *International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*. 2013, pp. 267–278 (cit. on pp. 27, 38).
- [TW10] Jeffrey Y Tsao and Paul Waide. "The World's Appetite for Light: Empirical Data and Trends Spanning Three Centuries and Six Continents." In: *Leukos* 6.4 (2010), pp. 259–281 (cit. on p. 1).
- [TW]+13] Fung Po Tso, David R. White, Simon Jouet, Jeremy Singer, and Dimitrios P. Pezaros. "The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures." In: *International Workshop on Resource Management of Cloud Computing*. 2013, pp. 1–5 (cit. on p. 55).
- [TZZ13] Sibel Tombaz, Zhihao Zheng, and Jens Zander. "Energy Efficiency Assessment of Wireless Access Networks Utilizing Indoor Base Stations." In: *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. 2013, pp. 1–6 (cit. on p. 165).
- [Usboo] *Universal Serial Bus Specification - Revision 2.0*. 2000. URL: [http://www.usb.org/developers/docs/usb20\\_docs/#usb20spec](http://www.usb.org/developers/docs/usb20_docs/#usb20spec) (cit. on p. 57).
- [VHD+11] Willem Vereecken, Ward Van Heddeghem, Margot Deruyck, Bart Puype, Bart Lannoo, Wout Joseph, Didier Colle, Luc Martens, and Piet Demeester. "Power Consumption in Telecommunication Networks: Overview and Reduction Strategies." In: *IEEE Communications Magazine* 49.6 (2011), pp. 62–69 (cit. on pp. 2, 25, 26, 36, 38, 169, 175).
- [VK10] Nedeljko Vasić and Dejan Kostić. "Energy-aware Traffic Engineering." In: *International Conference on Energy-Efficient Computing and Networking (e-Energy)*. 2010, pp. 169–178 (cit. on p. 170).
- [VM14] Vladimir Vujovic and Mirjana Maksimovic. "Raspberry Pi as a Wireless Sensor node: Performances and constraints." In: *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2014, pp. 1013–1018 (cit. on p. 55).
- [VM16] Christian Vitale and Vincenzo Manusco. "Energy Efficiency in Mixed Access Networks." In: *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiN)*. 2016, pp. 1–8 (cit. on pp. 75, 93, 166).
- [VNS+11] Nedeljko Vasic, Dejan Novakovic, Satyam Shekhar, Prateek Bhurat, Marco Canini, and Dejan Kostic. "Identifying and Using Energy-Critical Paths." In: *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. 2011, pp. 1–12 (cit. on pp. 30, 38, 75, 93, 147).



- [VSK15] Narseo Vallina-Rodriguez, Srikanth Sundaresan, and Christian Kreibich. "Beyond the Radio: Illuminating the Higher Layers of Mobile Networks." In: *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 2015, pp. 375–387 (cit. on pp. 10, 43, 47, 137).
- [WE15] Richard Withnell and Christopher Edwards. "Towards a Context Aware Multipath-TCP." In: *IEEE Conference on Local Computer Networks (LCN)*. 2015, pp. 434–437 (cit. on p. 55).
- [WS10] Marilyn P. Wylie-Green and Tommy Svensson. "Throughput, Capacity, Handover and Latency Performance in a 3GPP LTE FDD Field Trial." In: *IEEE Globecom*. 2010, pp. 1–6 (cit. on pp. 43, 47).
- [XY13] Renchao Xie, F. Richard Yu, and Hong Ji. "Interference Management and Power Allocation for Energy-Efficient Cognitive Femtocell Networks." In: *Mobile Networks and Applications* 18.4 (2013), pp. 578–590 (cit. on p. 167).
- [YKH08] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. "An Empirical Study of Bandwidth Predictability in Mobile Computing." In: *ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*. 2008, pp. 11–18 (cit. on pp. 43, 47, 145).
- [YKH11] Jun Yao, Salil S. Kanhere, and Mahbub Hassan. "Mobile Broadband Performance Measured from High-Speed Regional Trains." In: *IEEE Vehicular Technology Conference*. 2011, pp. 1–5 (cit. on pp. 44, 47).
- [YWX+13] Yuan Yang, Dan Wang, Mingwei Xu, and Suogang Li. "Hop-by-hop Computing for Green Internet Routing." In: *IEEE International Conference on Network Protocols (ICNP)*. 2013, pp. 1–10 (cit. on pp. 30, 38, 53, 93, 147, 170).
- [ZGC+13] Xuejun Zhuo, Wei Gao, Guohong Cao, and Sha Hua. "An Incentive Framework for Cellular Traffic Offloading." In: *IEEE Transactions on Mobile Computing* (2013), pp. 1–15 (cit. on pp. 49, 51).
- [ZLH+16] Yiran Zhao, Shen Li, Shaohan Hu, Hongwei Wang, Shuochao Yao, Huajie Shao, and Tarek Abdelzaher. "An Experimental Evaluation of Datacenter Workloads on Low-power Embedded Micro Servers." In: *Very Large Data Bases (VLDB) Endowment* 9.9 (2016), pp. 696–707 (cit. on pp. 3, 26–28, 38).
- [ZPH+11] Yiqing Zhou, Zhengang Pan, Jinlong Hu, Jinglin Shi, and Xinwei Mo. "Broadband Wireless Communications on High Speed Trains." In: *Wireless and Optical Communications Conference (WOCC)*. 2011, pp. 1–6 (cit. on p. 123).
- [ZTD+10] Lide Zhang, Birjodh Tiwana, Robert P. Dick, Zhiyun Qian, Z. Morley Mao, Zhaoguang Wang, and Lei Yang. "Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones." In: *IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*. 2010, pp. 105–114 (cit. on pp. 32, 38).

*All web pages cited in this work have been checked in March 2017. However, due to the dynamic nature of the World Wide Web, the long-term availability of web pages cannot be guaranteed.*

APPENDIX

---

## A.1 LIST OF ACRONYMS

<b>3GPP</b>	3rd Generation Partnership Project
<b>ADB</b>	Android debug bridge
<b>AOSP</b>	Android open source project
<b>AP</b>	access point
<b>API</b>	application programming interface
<b>AS</b>	autonomous system
<b>ASIC</b>	application specific integrated circuit
<b>ASU</b>	arbitrary strength unit
<b>BBU</b>	baseband unit
<b>BRAS</b>	broadband remote access server
<b>BSSID</b>	basic service set identifier
<b>BTC</b>	bulk transfer capacity
<b>BTS</b>	base transceiver station
<b>C-RAN</b>	centralized/cloud RAN
<b>CAGR</b>	compound annual growth rate
<b>CAPEX</b>	capital expenditure
<b>CBR</b>	constant bit rate
<b>CDF</b>	cumulative distribution function
<b>CDN</b>	content distribution network
<b>CISC</b>	complex instruction set computing
<b>COTS</b>	commercial off-the-shelf
<b>CPU</b>	central processing unit
<b>CR</b>	cognitive radio

<b>CSV</b>	comma separated values
<b>D2D</b>	device to device
<b>DCH</b>	dedicated channel
<b>DHCP</b>	dynamic host configuration protocol
<b>DHT</b>	distributed hash table
<b>DNS</b>	domain name system
<b>DSL</b>	digital subscriber line
<b>DSLAM</b>	DSL access multiplexer
<b>DUT</b>	device under test
<b>EDP</b>	energy delay product
<b>eNB</b>	eNodeB
<b>EPC</b>	evolved packet core
<b>FACH</b>	fast access channel
<b>FIFO</b>	first in, first out
<b>FPGA</b>	field programmable gate array
<b>GCM</b>	google cloud messaging
<b>GP-CPU</b>	general purpose CPU
<b>GPIO</b>	general purpose input output
<b>GPS</b>	global positioning system
<b>HARQ</b>	hybrid automatic repeat request
<b>HGW</b>	home gateway
<b>HPC</b>	high-performance computing
<b>HSDPA</b>	high speed downlink packet access
<b>HSPA</b>	high speed packet access
<b>HSPA+</b>	evolved HSPA
<b>HTB</b>	hierarchical token bucket
<b>HTML</b>	hypertext markup language
<b>HTTP</b>	hypertext transfer protocol

<b>HTTPS</b>	HTTP secure
<b>ICMP</b>	Internet control message protocol
<b>IC</b>	integrated circuit
<b>ICT</b>	information and communications technology
<b>IoT</b>	Internet of things
<b>IP</b>	Internet protocol
<b>IPv6</b>	Internet protocol version 6
<b>ISP</b>	Internet service provider
<b>IXP</b>	Internet exchange point
<b>JSON</b>	java script object notation
<b>KPI</b>	key performance indicator
<b>LAC</b>	location area code
<b>LTE</b>	long term evolution
<b>MAC</b>	medium access control
<b>MIMO</b>	multiple-input multiple-output
<b>MNO</b>	mobile network operator
<b>MPTCP</b>	MultiPath TCP
<b>MU-MIMO</b>	multi user MIMO
<b>MVNO</b>	mobile virtual network operator
<b>NaaS</b>	network as a service
<b>NAT</b>	network address translation
<b>NF</b>	network function
<b>NFC</b>	near field communication
<b>NFV</b>	network functions virtualization
<b>ngmn</b>	next generation mobile network
<b>NOMA</b>	non-orthogonal multiple access
<b>NodeB</b>	3G base station
<b>NTP</b>	network time protocol

<b>OPEX</b>	operational expenditure
<b>OS</b>	operating system
<b>PA</b>	power amplifier
<b>PC</b>	personal computer
<b>PCH</b>	paging channel
<b>PDF</b>	probability density function
<b>PEP</b>	performance enhancing proxy
<b>PGW</b>	packet gateway
<b>PID</b>	process ID
<b>PON</b>	passive optical network
<b>PoP</b>	point of presence
<b>QoE</b>	quality of experience
<b>QoS</b>	quality of service
<b>RAM</b>	random access memory
<b>RAN</b>	radio access network
<b>RB</b>	resource block
<b>RED</b>	random early discard
<b>REST</b>	representational state transfer
<b>RF</b>	radio frequency
<b>RISC</b>	reduced instruction set computing
<b>RMS</b>	root mean square
<b>RMSE</b>	root mean square error
<b>RNC</b>	radio network controller
<b>RR</b>	round robin
<b>RRU</b>	remote radio unit
<b>RSRP</b>	reference signal received power
<b>RSSI</b>	received signal strength indicator
<b>RTT</b>	round-trip time

<b>SATA</b>	serial ATA
<b>SBC</b>	single-board computer
<b>SDN</b>	software defined networking
<b>SDM</b>	software defined multicast
<b>SDR</b>	software defined radio
<b>SGW</b>	serving gateway
<b>SIC</b>	self-interference cancellation
<b>SOC</b>	system on chip
<b>SON</b>	self-organizing network
<b>SSH</b>	secure shell
<b>SSID</b>	service set identifier
<b>TCAM</b>	ternary content addressable memory
<b>TCP</b>	transport control protocol
<b>TLS</b>	transport layer security
<b>TOPP</b>	train of packet pairs
<b>TTF</b>	time to finish
<b>TTFB</b>	time to first byte
<b>TTL</b>	time to live
<b>UDP</b>	user datagram protocol
<b>UE</b>	user equipment
<b>UI</b>	user interface
<b>UMTS</b>	universal mobile telecommunications system
<b>USB</b>	universal serial bus
<b>VDSL</b>	Very-high-bit-rate digital subscriber line
<b>VNF</b>	virtual network function
<b>VPN</b>	virtual private network
<b>VR</b>	virtual reality
<b>WiFi</b>	wireless fidelity





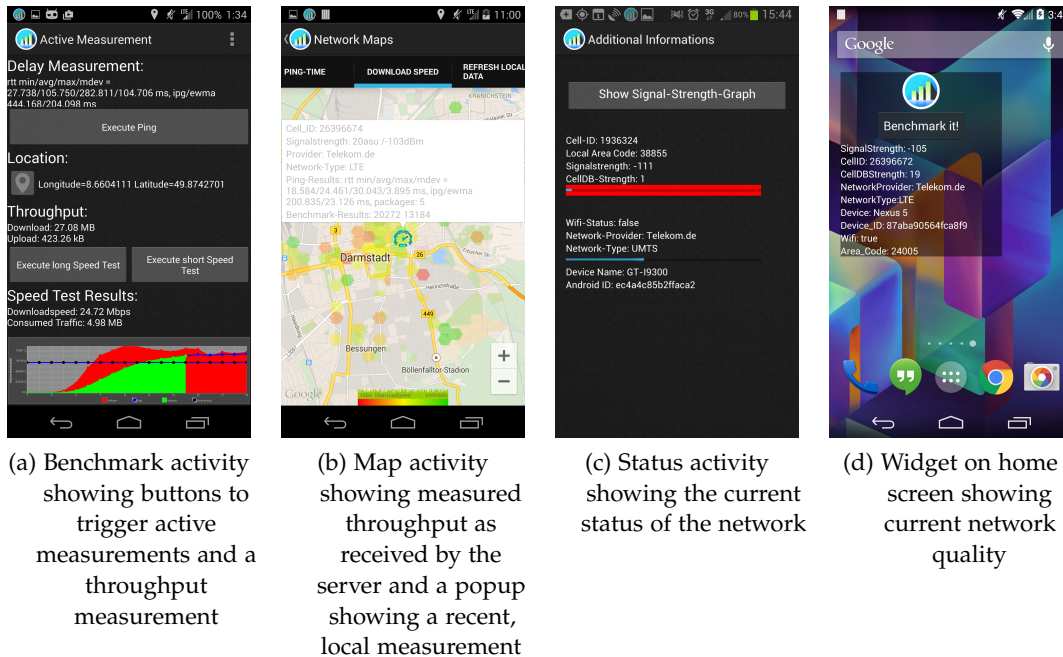


Figure A.1: Screenshots of the NetworkCoverage App

## A.2 ARCHITECTURE OF THE NETWORK COVERAGE APP

The application running on Android smartphones is called *NetworkCoverage App* and available on GooglePlay<sup>1</sup>. It is structured into UI and a service, which is responsible for monitoring, measuring, and data handling. The UI consists of a number of different screens, which are called *Activities* on Android, letting the user see the data collected on the device as well as aggregated data from the measurement server, run measurements, or check the network status.

The application was developed to provide benefits for both the user and the experimenter. Hence, a considerable effort was put into the presentation of the collected data, ease of use, and general user friendliness and energy efficiency. The application consists of four *Activities*, showing the current network status, a map of the aggregated data as available on the data collection server, an overview of the locally available measurements, and another activity, where the user can manually run throughput or RTT measurements at locations of interest. Screenshots of these activities are given in Figure A.1

Data collection and measurement are handled by a background service. This is structured into a passive monitoring component, registering on system callbacks for signal strength, location, or battery updates, plus an active measurement component, running

<sup>1</sup> <https://play.google.com/store/apps/details?id=de.tudarmstadt.networkcoverage>, accessed 2017-01-02

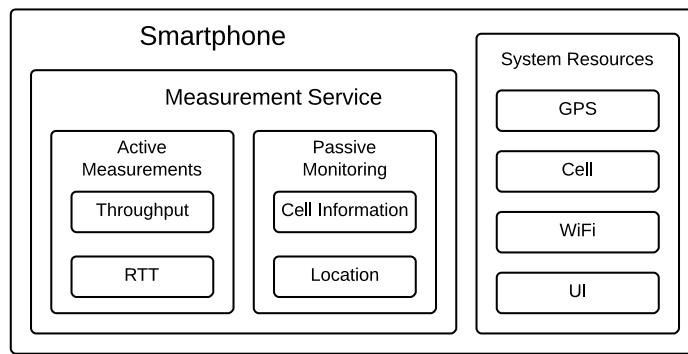


Figure A.2: Block diagram of the NetworkCoverage App

periodic throughput and RTT measurements, and the data handling routines, responsible for storing and uploading the collected data, as well as providing a local cache for the visualization of recorded network metrics. The application provides two main operating modes: *active* and *passive*. In active mode, the user interacts with the application, while in passive mode the application is configured to periodically run measurements in the background. Figure A.2 gives an overview of the structure of the App.

When the application is started, the background service starts the location monitoring with a preference on GPS and registers for the callbacks. It also registers on the signal strength changes periodically broadcasted by the telephony manager. Thus, signal strength, and network technology changes are recorded together with the latest location and timestamp. This information is stored in the local database to periodically be uploaded to the central data collection server.

A further feature implemented in the NetworkCoverage Application is an indoor/outdoor detection algorithm based on cellular signal strength samples. Therefore, a classification approach is implemented, observing the available cells and their signal strengths. During the learning phase it was observed that in indoor scenarios often only one cell shows a strong signal, while the remaining cells are comparatively weak. Contrary in outdoor settings, the observed signal strengths are more similar, and the strongest cell tends to change frequently. Based on these observations it becomes possible to derive a classification tree, returning the setting of the device, and the certainty of this decision. This is appended to the location data to assess the measurements depending on the environment and setting of the device. Further, this is an important metric for later modeling and evaluation of network optimization approaches, as usually the performance of cellular network is better outside, while WiFi performance there is poor, and vice versa indoors.

Besides cellular data, also the available WiFi networks and their signal strengths are recorded. For this, a periodic scan is triggered. Similar to the cellular signal strength, this data is annotated with the latest location and timestamp.

Active measurements in the published version comprise of RTT, throughput, with an extended feature set for dedicated measurements as will be described in Section 5.3. RTT and throughput measurements are either started automatically while in active scanning mode, or triggered by the user at locations or times of interest.

The amount of traffic consumed by the measurements is limited by requesting a monthly traffic allowance from the user when first starting the application. By monitoring the traffic consumed by the measurements, and comparing it with the allowance interpolated to the current day, a decision on the execution of RTT or throughput measurements is made. For each connected network technology an average consumed traffic volume per test is configured in the application, thus allowing the scheduler to decide whether further tests may be run without exceeding the traffic budget. Clearly, thus the measurements are more probable to the end of the month. Still, due to differing accounting intervals (as defined by the network operators), measurements are again distributed over the full month.

To reduce energy consumption, a special background mode was introduced. This registers a callback on a system broadcast, and only then activates the App for a single measurement when the devices become active. Further, only network localization (e.g. triangulation based on the observed WiFi SSIDs and cell IDs) is used. The observed location accuracies are generally better than 100 m in an urban or suburban context. As urban users spend most of the time in vicinity of some form of WiFi coverage, the accuracy of the collected measurement samples in urban environments is generally sufficient [CTS+11]. Further, GPS is not suitable for indoor localization; hence no accuracy is lost when resorting to WiFi localization for the background measurements.

For outdoor localization in remote areas, accurate localization is achieved by passively listening on location updates, which may be requested by some other application on the end-user device. Thus, whenever an updated location is received, another coverage sample is recorded. Depending on the configured probing intervals, also an RTT or throughput measurement may be triggered.

All measurements are first collected on the mobile device, to be periodically uploaded to the data collection server. For this, a custom representational state transfer (REST) endpoint is installed, authenticating the connecting devices, accepting and checking the uploaded data, as well as storing it for later processing in a local database.

This server runs a modified version of the *da\_sense*<sup>2</sup> server, adapted to the requirements of the network measurements described above. This server is split into several components, handling different tasks. The API endpoints for data retrieval are written in plain PHP, while the website consists of PHP templates combined with JavaScript files providing interactive elements. For data upload and user management, a custom solution

---

<sup>2</sup> <http://www.da-sense.de/> accessed 2017-01-03

based on *laravel*<sup>3</sup> is used. The data is stored in a PostgreSQL database<sup>4</sup> with the PostGIS extension<sup>5</sup> installed. Details on the individual components are given in the following.

The user management is handled by an API endpoint written in *laravel*. There, users may register with a username, password and email address. From the password string a secure hash string is derived using a salt only known to the server. On each user login, a token is generated, which is used in the following interactions to authenticate the incoming connections and map the data to the respective user accounts.

The data sent by the mobile application is recorded using another API endpoint. After authentication and integrity of the received data are assured, the data is stored in respective tables in the PostgreSQL database.

The JSON upload is structured as follows: The main object contains a device identifier, measurement type (for compatibility reasons with *da\_sense*), and a series object, containing the actual measurement samples. The *series* object has a name, visibility, and timestamp field as well as an array of measurements. Each of these measurements contains a timestamp, version field, and – depending on the measurement – cells, pings, throughput, WiFi scans and tags. For RTT and throughput measurements, individual measurement samples are stored. Thus, the versatility of the data protocol is maximized, including the possibility to add additional measurement modes with increased accuracy in a future App release. This versatility is also visible in the various timestamps. Using these, the location accuracy of the measurements is maximized by allowing the post-processing steps to interpolate different metrics based on the path of the user. Further, differences in measurements over time can be identified, thus giving a more fine-granular insight into the measured network performance. An example object as received by the server is given in Listing A.1.

In the database, tables for series, measurement samples (termed *values*), cells, WiFi networks, ping measurements, throughput measurements, and the respective measurement samples are created and referenced accordingly. As database server PostgreSQL in combination with PostGIS is used. Here, PostGIS extends the PostgreSQL database with additional data formats, supporting the efficient storage and retrieval of 2-dimensional data such as location data. Thus, data from a given region can efficiently be requested from the database, simplifying later analysis and processing. The combination of both is used, as it is well tested, free of charge, and widely used.

Storing raw data assures that all data is available as recorded by the devices, thus allowing the experimenter to later change aggregation and visualization algorithms as required as well as extract metrics not identified during planning, setup and execution of the measurement. The collected data is processed in a pre-processing step to assure high quality of the data for visualization. This filtering eliminates samples with insufficient accuracy, missing values, impossible combinations of values, and otherwise invalid data.

<sup>3</sup> <https://www.laravel.com/> accessed 2017-01-04

<sup>4</sup> <https://www.postgresql.org/> accessed 2017-01-03

<sup>5</sup> <http://postgis.net/> accessed 2017-01-03

The processed data is stored in a separate table used for calculating map tiles for the website<sup>6</sup> and the mobile application. For each network provider, map tiles visualizing the signal strength, RTT, and throughput for the different network technologies can be requested from the server. These are generated on-demand. If these have previously been requested, these are served from the built-in cache.

The *da\_sense* framework already provides two different visualization algorithms: conventional squares, and a visually more pleasing hexagonal visualization. To plot the collected network performance data, these are adapted to useful variable ranges and color schemes (e.g. higher throughput in green, while higher RTT in red).

These map tiles provide a means of increasing user participation by showing availability and performance of the network at previously surveyed locations. This increases the motivation to contribute by giving the users a sense of participation in the measurement study. Further, when showing and aggregating measurements from multiple participants, a more detailed view of the network is available at the locations of interest of the particular user.

Listing A.1: Example of an uploaded, JSON formatted, measurement sample

```
{
  "deviceId": "a0881a8b8802f99d2c0abdc02aa551650fe2db38",
  "measurementType": 7,
  "series": [{
    "name": "testseries",
    "visibility": 1,
    "timestamp": "2017-01-23 01:01:16+01",
    "values": [{
      "timestamp": "2017-01-23 01:01:16+01",
      "app_version": "33",
      "locations": [{
        "longitude": 49.8145200,
        "latitude": 8.6459600,
        "altitude": 122,
        "accuracy": 5,
        "speed": 7.5,
        "indoor": false,
        "indoorConfidence": 0.9,
        "timestamp": "2017-01-23 01:01:16+01"
      }],
      "cells": [{
        "cellId": 26414338,
        "lac": 24005,
        "networkType": "LTE",
        "networkProvider": "Telekom.de",
```

<sup>6</sup> <https://mona.ps.e-technik.tu-darmstadt.de/> accessed 2017-01-03

```

    "asu": 30,
    "signalStrengthDb": -80,
    "isActive": true,
    "updateTimestamp": "2017-01-23 01:01:16+01"
  }],
  "pings": [{
    "timeStart": "2017-01-23 01:01:16+01",
    "timeEnd": "2017-01-23 01:02:16+01",
    "remoteServer": "someServer",
    "samples": [{
      "value": 123,
      "timestamp": "2017-01-22 01:01:16+01"
    }],
    "receivedPingCount": 2,
    "pingCount": 5
  }],
  "throughput": [{
    "direction": "up",
    "benchmarkType": "sometype",
    "remoteServer": "someServer",
    "timeStart": "2017-01-23 01:01:16+01",
    "timeEnd": "2017-01-23 01:02:16+01",
    "errorCode": 10,
    "samples": [{
      "value": 113,
      "timestamp": "2017-01-23 01:01:16+01"
    }]
  }],
  "wifi": [{
    "signalStrength": -90,
    "ssid": "Meins",
    "bssid": "74:ea:3a:be:1f:aa",
    "capabilities": "wpa2",
    "frequency": 2412,
    "isActive": false,
    "updateTimestamp": "2017-01-23 01:01:16+01"
  }],
  "tags": [{
    "name": "someKey",
    "value": "someValue"
  }]
}
}]
}]

```

## A.3 ARCHITECTURE OF THE OPENFLOW POWER MEASUREMENT ENVIRONMENT

The architecture of the measurement environment is detailed in Figure A.3. The Floodlight OpenFlow controller is extended by a number of components implementing the above described functionality. Namely, the added components are the *TestCoordinator*, *TestInterpreter*, the test cases as defined in Figure 4.9, the *EnergyMonitor*, and the *MessageHandler*. The *MessageHandler* connects to the corresponding component at the *SyncClient*. Via this channel the remote *TrafficGenerator* and *DataCollector* are configured.

The *TestCoordinator* is the high level coordinator of the tests. It executes the *TestCases* sequentially or in random order, while coordinating via the *MessageHandler* with the *SyncClient* to start traffic generation and power measurement, or collecting recorded data.

The *TestCases* are read by the *TestCoordinator* to configure the connected DUT. The *TestCases* contain the full configuration required for configuring the DUT via OpenFlow commands, traffic generation on the *SyncClient* and power measurements.

On the *SyncClient* side, the *TrafficGenerator* is started as soon as the respective message is received. Simultaneously, the *DataCollector* is started, polling the connected power meter for measurements. These are stored on the *SyncClient* but also made available to the *SyncMaster* via a REST interface.

The *TestInterpreter* reads the recorded data, combines it with the configuration of the DUT as defined in the *TestCases* and aggregates both into power models. Such, after finishing the tests, the power model of the DUT can be automatically generated.

The generated energy models are then used by the *EnergyMonitor* to estimate the power consumption of connected devices solely based on device configuration and observed traffic patterns. Thus, the power consumption of the device can be estimated in real-time. For the analysis of the measurement accuracy, a live energy monitor is implemented, showing the power consumption as measured on the *SyncClient*, estimated by the *EnergyMonitor*, and the error between both.

The remaining modules shown in the Floodlight container in Figure A.3 are the default components included in the Floodlight framework. These are required for the correct functioning of the power measurement and modeling framework, but were not modified.

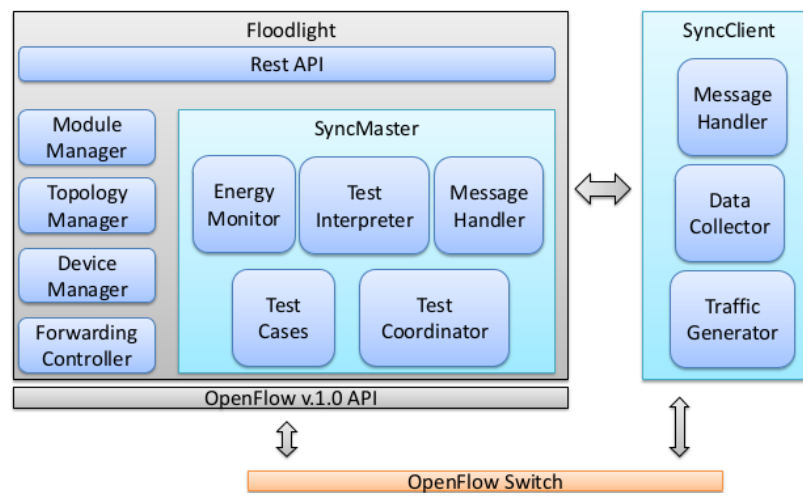


Figure A.3: Modular representation of the system design (from [Mel14])



AUTHOR'S PUBLICATIONS

---

## B.1 MAIN PUBLICATIONS

*Journal paper*

- [KMB+16] Fabian Kaup, Foivos Michelinakis, Nicola Bui, Joerg Widmer, Katarzyna Wac, and David Hausheer. "Assessing the Implications of Cellular Network Performance on Mobile Content Access." In: *IEEE Transactions on Network and Service Management* 13.2 (2016), pp. 168–179.

*Conference paper*

- [KFH17] Fabian Kaup, Florian Fischer, and David Hausheer. "Measuring and Predicting Cellular Network Quality on Trains." In: *International Conference on Networked Systems (NetSys)*. 2017, pp. 1–8.
- [KGH14] Fabian Kaup, Philip Gottschling, and David Hausheer. "PowerPi: Measuring and Modeling the Power Consumption of the Raspberry Pi." In: *IEEE Conference on Local Computer Networks (LCN)*. 2014, pp. 236–243.
- [KMB+15] Fabian Kaup, Foivos Michelinakis, Nicola Bui, Joerg Widmer, Katarzyna Wac, and David Hausheer. "Behind the NAT – A Measurement Study of Cellular Service Quality." In: *Conference on Network and Service Management (CNSM)*. 2015, pp. 228–236.
- [KMH14b] Fabian Kaup, Sergej Melnikowitsch, and David Hausheer. "Measuring and Modeling the Power Consumption of OpenFlow Switches." In: *International Conference on Network and Service Management (CNSM)*. 2014, pp. 181–186.
- [KWR+15b] Fabian Kaup, Matthias Wichtlhuber, Stefan Rado, and David Hausheer. "Can Multipath TCP Save Energy? A Measuring and Modeling Study of MP-TCP Energy Consumption." In: *IEEE Conference on Local Computer Networks (LCN)*. 2015, pp. 442–445.

*Other*

- [Kau12] Fabian Kaup. "Measurement and Modeling of WiFi ad hoc-based Communication in Mobile Peer-to-Peer Networks using Smartphones." Diplomarbeit. Technische Universität Darmstadt, 2012.
- [KH13] Fabian Kaup and David Hausheer. "Optimizing Energy Consumption and QoE on Mobile Devices." In: *IEEE International Conference on Network Protocols (ICNP), PhD Workshop*. 2013.
- [KJH15] Fabian Kaup, Florian Jomrich, and David Hausheer. "Demonstration of NetworkCoverage – A Mobile Network Performance Measurement App." In: *International Conference on Networked Systems (NetSys)*. 2015.
- [KMH14a] Fabian Kaup, Sergej Melnikowitsch, and David Hausheer. *A Model-based OpenFlow Power Monitoring Framework based on Real-World Measurements*. Tech. rep. PS Technical Report No. PS-TR-2014-02, 2014. URL: <http://www.ps.tu-darmstadt.de/fileadmin/publications/PS-TR-2014-02.pdf>.
- [KWR+15a] Fabian Kaup, Matthias Wichtlhuber, Stefan Rado, and David Hausheer. *Analysis and Modeling of the Multipath-TCP Power Consumption for Constant Bitrate Streaming*. Tech. rep. no. PS-TR-2015-02. Darmstadt: Technische Universität Darmstadt, 2015. URL: [http://www.ps.tu-darmstadt.de/nc/publications/publications-details/?pub\\_id=KWR+15-2](http://www.ps.tu-darmstadt.de/nc/publications/publications-details/?pub_id=KWR+15-2).

## B.2 CO-AUTHORED PUBLICATIONS

*Journal paper*

- [MBF+16] Foivos Michelinakis, Nicola Bui, Guido Fioravanti, Joerg Widmer, Fabian Kaup, and David Hausheer. “Lightweight capacity measurements for mobile networks.” In: *Computer Communications* 84 (2016), pp. 73–83.

*Conference paper*

- [BKS+15] Valentin Burger, Fabian Kaup, Michael Seufert, Matthias Wichtlhuber, David Hausheer, and Phuoc Tran-Gia. “Energy Considerations for WiFi Offloading of Video Streaming.” In: *EAI International Conference on Mobile Networks and Management (MONAMI)*. 2015, pp. 181–195.
- [BSK+15] Valentin Burger, Michael Seufert, Fabian Kaup, Matthias Wichtlhuber, David Hausheer, and Phuoc Tran-gia. “Impact of WiFi Offloading on Video Streaming QoE in Urban Environments.” In: *IEEE International Conference on Communication Workshop (ICCW)*. 2015, pp. 1717–1722.
- [GKS+13] Christian Gross, Fabian Kaup, Dominik Stingl, Björn Richerzhagen, David Hausheer, and Ralf Steinmetz. “EnerSim: An Energy Consumption Model for Large-Scale Overlay Simulators.” In: *IEEE Conference on Local Computer Networks (LCN)*. 2013, pp. 252–255.
- [MBF+15] Foivos Michelinakis, Nicola Bui, Guido Fioravanti, Joerg Widmer, Fabian Kaup, and David Hausheer. “Lightweight Mobile Bandwidth Availability Measurement.” In: *IFIP Networking Conference (NETWORKING)*. 2015, pp. 1–7.
- [MSS+15] Christian Meurisch, Alexander Seeliger, Benedikt Schmidt, Immanuel Schweizer, Fabian Kaup, and Max Mühlhäuser. “Upgrading Wireless Home Routers for Enabling Large-Scale Deployment of Cloudlets.” In: *International Conference on Mobile Computing, Applications and Services (MobiCASE)*. 2015, pp. 1–18.
- [SBS+15] Immanuel Schweizer, Roman Bärthel, Benedikt Schmidt, Fabian Kaup, and Max Mühlhäuser. “Kraken.me mobile: The energy footprint of mobile tracking.” In: *International Conference on Mobile Computing, Applications and Services (MobiCASE)*. 2015, pp. 82–89.

*Other*

- [WKR+15] Matthias Wichtlhuber, Fabian Kaup, Robert Reinecke, Jeremias Blendin, and David Hausheer. "Demo: A holistic energy-monitoring framework for the IT service delivery chain." In: *IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015, pp. 1145–1146.

CURRICULUM VITÆ

---

## PERSONAL INFORMATION

Name	Fabian Kaup
Date of Birth	August 2, 1984
Place of Birth	Lahnstein
Nationality	German

## EDUCATION

Since 03/2013	Technische Universität Darmstadt Department of Electrical Engineering and Information Technology Associate researcher at Peer-to-Peer Systems Engineering Lab
10/2005–01/2012	Technische Universität Darmstadt Department of Electrical Engineering and Information Technology Diplom, <i>Communication Technology</i>
02/2009–11/2009	Curtin University (Perth, Australia) Department of Electrical and Computer Engineering equiv. Bachelor's thesis
03/2004	Marion Dönhoff Gymnasium, Oberlahnstein Abitur, majoring in Mathematics and German

## PROFESSIONAL EXPERIENCE

02/2012–03/2013	Bosch SoftTec iOS Developer
-----------------	--------------------------------

## TEACHING ACTIVITIES

- |           |                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------|
| 2012–2017 | Lab exercise Communication Networks Lab<br>Organization of the course<br>Supervision of student groups |
| 2012–2017 | Tutor for 4 Bachelor's and 5 Master's theses                                                           |

## SCIENTIFIC ACTIVITIES

- |          |                                                                                                    |
|----------|----------------------------------------------------------------------------------------------------|
| Reviewer | IEEE Communications Magazine, 2016 (2x).                                                           |
| Reviewer | Conference on Network and Service Management (CNSM), 2013, 2015, 2016.                             |
| Reviewer | IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015, 2017.               |
| Reviewer | IEEE International Conference on Local Computer Networks (LCN), 2014, 2015, 2016.                  |
| Reviewer | IFIP Networking, 2014, 2016 (2x).                                                                  |
| Reviewer | IEEE Conference on Network Softwarization (NetSoft), 2015, 2016.                                   |
| Reviewer | IEEE/IFIP Network Operations and Management Symposium (NOMS), 2014, 2016.                          |
| Reviewer | IEEE Wireless Communications and Networking Conference (WCNC), 2017.                               |
| Reviewer | IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2016 |

## SUPERVISED THESES

- |           |                                                                                                                                                                                                                          |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PS-D-0008 | Sergej Melnikowitsch, <i>Measuring and Modeling the Power Consumption of Software Defined Networks</i> . Master's thesis, Technische Universität Darmstadt, March 2014. (Supervised in cooperation with Julius Rückert.) |
| PS-S-0009 | Stefan Rado, <i>Optimizing the Energy Efficiency of Multipath TCP for Mobile Devices</i> . Bachelor's thesis, Technische Universität Darmstadt, August 2014. (Supervised in cooperation with Matthias Wichtlhuber.)      |

- PS-S-0011 Manuel Fuentes Sifuentes, *Evaluating the Feasibility of Land-use regression for the Prediction of Cellular Network Performance*. Studienarbeit, Technische Universität Darmstadt, September 2014. (Supervised in cooperation with Matthias Wichtlhuber.)
- PS-S-0014 Carsten Englert, *Passive Throughput Measurements on Android*. Bachelor's Thesis, Technische Universität Darmstadt, May 2015. (Supervised in cooperation with Matthias Wichtlhuber.)
- PS-D-0021 Hendryk Herzog, *Analysis of Inter- and Intra-technology Handovers on Network Performance and Energy Consumption of Smartphones*. Master's thesis, Technische Universität Darmstadt, November 2015. (Supervised in cooperation with Leonhard Nobach.)
- PS-S-0019 Florian Fischer, *Analysis of the Cellular Service Quality in Trains*. Bachelor's thesis, Technische Universität Darmstadt, April 2016. (Supervised in cooperation with Jeremias Blendin.)
- PS-D-0029 Thomas Schnabel, *Simulative Comparison of Mobile Energy Conservation Schemes*. Master's thesis, Technische Universität Darmstadt, May 2016. (Supervised in cooperation with Christian Koch.)
- PS-D-0035 Rohit Shah, *Analysis of Software-defined Power Saving on Android on the Nexus 5*. Master's thesis, Technische Universität Darmstadt, November 2016.
- PS-D-0043 Rainer Wahler, *Analysis of Handover Processes in Cellular Networks using Software-Defined Radio*. Master's thesis, Technische Universität Darmstadt, April 2017. (Supervised in cooperation with Florian Jomrich.)

Darmstadt, 22. März 2017





ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG

---

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe. Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

*Darmstadt, 22. März 2017*

---

Dipl.-Ing. Fabian Kaup