Copyright © May 2015, Junaid Shaikh. All rights reserved.

Blekinge Institute of Technology
Doctoral Dissertation Series No. 2015:xx
ISSN xxxx-xxxx
ISBN x

Published 201x
Printed by xx
Karlskrona xx
Sweden

This publication was typeset using \LaTeX.
Dedicated to my grandparents
Abstract

Recent years have observed a tremendous shift from the technology-centric assessment to the user-centric assessment of network services. Obviously, a sustainable network management approach cares about the user demands and expectations. Consequently, the measurement and modeling of Quality of Experience (QoE) attracted many contributions from researchers and practitioners. Generally, QoE is assessed at two levels, i.e., Application and Network level. While the former usually allows QoE assessment on the test traffic with control on client-side instrumentation, the latter opens the avenues for continuous QoE assessment on the traffic generated by the real users. This thesis contributes towards passive network-level assessment of QoE.

This thesis document begins with a background on the fundamentals of Network Management and objective QoE assessment. It extends the discussion further to the QoE-centric monitoring and management of network, complimented by the details about QoE estimator agent developed within the Celtic project QuEEN (Quality of Experience Estimators in Network).

The discussion on findings start with results from subjective tests to understand the relationship between waiting times and user subjective feedback over time. These results help strengthen the understanding of timescales on which users react, as well as, the role of memory effect. The findings show that QoE drops significantly with delays on the timescales of 1–4 s. With recurring delays, the user tolerance to waiting times decreases constantly showing the signs of memory effect.

Subsequently, this document introduces and evaluates a passive wavelet-
based QoE monitoring method. The method detects the timescales on which transient outages occur frequently. A study presents results from Qualitative measurements, showing the ability of wavelet to differentiate on-fly between the “Good” and the “Bad” streams. In sequel, a quantitative study illustrates the ability of method to monitor the duration and frequency of traffic gaps. The discussion also guides practical implementation of this method using QoE agent developed within QuEEN project.

Finally, this thesis investigates a method for passive monitoring of user reactions to the bad network performance. The method is based on the TCP termination flags. With a systematic evaluation in test environment, the results characterize termination of data transfers in the case of different user actions in the Web browser.
First of all, I would like to thank Professor Markus Fiedler for accepting me in PhD studies. He has been a great mentor who guided me over the years with a lot of patience and hardwork. Working in his team has been a rewarding experience, which I will always treasure.

I am grateful to Dr. Patrik Arlos for his guidance during those many hours, which I spent in network performance lab at Blekinge Institute of Technology (BTH). He practically taught me the ABC of network measurements. I am also thankful to him for providing comments on my thesis.

There are several other persons who supported and guided me throughout my PhD studies. I am really thankful to Denis Collange at Orange labs for his continuous willingness to collaborate and discuss on various ideas during my research studies. I am thankful to Professor Adrian Popescu for the discussions during PhD course work and research. I am deeply thankful to Monica Nilsson for her continuous availability and the administrative support that she provided during last couple of years. I am also grateful to Eva-Lotta and Camilla for her support throughout the stay at BTH. I acknowledge my fellow PhD students including Tahir Minhas, Charlott Lorentzen, Yong Yao, Selim Ickin, Said Ngoga and others for being supportive and friendly during my stay at BTH.

I acknowledge the projects QoEWeb funded by the European Network of Excellence (EuroNF) and QuEEN (Celtic project) by Swedish funding agency VINNOVA for funding and supporting my PhD research work.

Finally, I am extremely thankful to my family members. Without them, this journey would have not been possible at all. First of all, I am thankful to my
parents and grandparents who always showed trust and confidence in me. I am deeply grateful to my uncle Abdul Khaliq who was always available for all kinds of guidance, help and advices. I thank my wife Rabail for her company and support, which made me further stronger to face bigger challenges during the last few years. I am grateful to my siblings for always being so friendly and supportive. Moreover, my son Naufel has joined me for last 2.5 years to make this journey further enjoyable.

Junaid Shaikh
Karlskrona, May 2015
# CONTENTS

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
<tr>
<td>List of Included Papers</td>
<td>xi</td>
</tr>
<tr>
<td>List of Additional Papers</td>
<td>1</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Research Objectives</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Outline</td>
<td>5</td>
</tr>
<tr>
<td><strong>2 Network Management</strong></td>
<td>7</td>
</tr>
<tr>
<td>2.1 Models of Network Management</td>
<td>8</td>
</tr>
<tr>
<td>2.2 QoE-Centric Network Management</td>
<td>11</td>
</tr>
<tr>
<td><strong>3 Quality of Experience Assessment</strong></td>
<td>13</td>
</tr>
<tr>
<td>3.1 Objective QoE assessment models</td>
<td>13</td>
</tr>
<tr>
<td><strong>4 QoE-Centric Network Management</strong></td>
<td>19</td>
</tr>
<tr>
<td>4.1 QoE Monitoring from Sessions to Packets</td>
<td>21</td>
</tr>
<tr>
<td>4.2 QoE estimator agent in QuEEN Project</td>
<td>23</td>
</tr>
</tbody>
</table>
5 Problem Statements  25

6 Conclusions & Outlook  31
   6.1 Conclusions  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
   6.2 Future Work  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

Bibliography  35

I Quality of Experience From User and Network Perspectives  39

II Back to Normal? Impact of Temporally Increasing Network Disturbances on QoE  71

III In Small Chunks or All at Once? User Preferences of Network Delays in Web Browsing Sessions  91

IV Modeling and Analysis of Web Usage and Experience Based on Link-Level Measurements  111

V Quantitative Evaluation of Wavelet-based Traffic Gap Detection  135

VI Inferring User-Perceived Performance of Network by Monitoring TCP Interruptions  163
List of Figures

1.1 Thesis structure .................................................. 6
2.1 Communication between Manager and Agent on Network....... 11
4.1 QoE-Centric Network Management. ............................. 20
4.2 Monitoring scale: From Session to Packets ...................... 21
4.3 The qoe-monitoring subtree. Figure 12 in [28] ................. 24
List of Included Papers

The listed order of papers corresponds to the order in which they appear in thesis.

Included papers


xi
LIST OF ADDITIONAL PAPERS

The following is a list of additional papers, which are not included in this thesis document.

NOT INCLUDED PAPERS


Today, the Internet drives a large portion of daily life activities. It has in fact become integral part of everyday tasks, related to health, education, business, entertainment, social life and news etc. Thus, networks now, more than ever, need to operate dynamically in a diverse range of scenarios and still assure a good user experience. Specifically, networks require intelligent operation and management techniques, which are able to meet the growing expectations of users under a variety of above-mentioned usage contexts.

Formally, the objective of network management is to meet user demands [1]. To meet this objective, network management activities and methods need to be user-centric, which understand expectations of the users and provide services accordingly. In contrast, traditionally, the practitioners followed a rather technology-centric approach to the management of networks, which often overlooked the above-mentioned fundamental goal of network management. However, with the changing landscape of network usage and a stiff competition between network operators, a rapid shift is being observed from technology-centric to the user-centric management of networks.

Consequently, Quality of Experience (QoE) emerged as a popular topic among researchers and practitioners during recent years. It is also referred to as the user perception of a service. QoE factors include network-, application-
CHAPTER 1. INTRODUCTION

and device-performance, as well as, content characteristics and user background to name a few.

The white paper by the Qualinet (European Network on Quality of Experience in Multimedia Systems and Services) defines QoE as [2]:

“Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfilment of his or her expectations with respect to the utility and or enjoyment of the application or service in the light of the user’s personality and current state.”

The ITU-T P.10/G.100 defines QoE as [3]:

“The overall acceptability of an application or service, as perceived subjectively by the end-user.”

The above-mentioned definitions of QoE express the multi-disciplinary nature of QoE. Based on its dependency on many aspects, measurement and modeling of QoE have been a challenge. Several studies proposed models for the estimation of QoE[4][5][6]. The studies presented in papers [7][8][9] propose models for web browsing QoE estimation. Similarly, authors also studied factors, which impact video streaming QoE [10][11][12]. These models estimate QoE based on the measurable network QoS parameters. It implies that these models may be implemented on the network for the QoE estimation of relevant applications. The ITU-T recommendations G. 1030 and P. 1201 presented standardised QoE models for web browsing and audiovisual services, respectively [13][14].

A large number of the proposed QoE models are developed based on the user subjective tests, which take into account a nominal (usually small, i.e., from a few seconds to a few minutes duration) timescale. This approach often overlook the dynamics of user satisfaction against fluctuating network performance over relatively longer period of times. For example, a user watching a long video clip (movie) or surfing many web pages in a session, which typically last from several minutes to hours, represent rather realistic scenarios today. In these usage contexts, the user memory or the recency effect may play a vital role in shaping the overall QoE [15][16], which needs to be taken into account for
the assessment of QoE. Thus, an evaluation that provides a view of network performance and QoE, flexibly over multiple timescales can help a great deal in painting a real picture of the perceived quality.

Moreover, the traditional QoS parameters, such as loss percentage, mean inter-packet time, mean throughput or data rate and mean Round Trip Times (RTTs) of data streams are coarse-grained parameters and thus, they may not sketch the continuously evolving picture of QoE over time [17]. Particularly, it becomes difficult to relate QoE issues to their root causes, typically due to the inappropriate choice of measurable metrics and the time granularity involved in their measurement. For example, the average data rate of a connection may not highlight the short patches of network outages, and the subsequent bursts of arriving data at the user end. The average value of data rate may hide those short intervals of waiting times at all depending on the duration of a transfer.

Henceforth, appropriate methods are required to match what we monitor on the network to what the user feels about service. The methods must also take into account practicalities w.r.t close-to-real-time implementation of models. On the other hand, the implementable models may not consider measurement of all the factors on which QoE depends, as it may not be trivial to acquire all the required parameters, due to the high complexity involved in retrieving their values. Therefore, a trade-off is involved between the accuracy and practicality of approaches for the measurement of QoE.

1.1 Research Objectives

The main focus of this thesis is to propose and evaluate a method that allows passive network-based monitoring of QoE at various timescales. The method is particularly relevant for the scenarios when client-side instrumentation is not available, and when there is no access to the original stream at the content-provider end (i.e. no-reference scenarios). When the only information available is the packet stream on network level captured within operator’s domain.

This thesis explores indicators of performance issues that potentially degrade
CHAPTER 1. INTRODUCTION

QoE, and conversely, the indicators of user reactions to the performance problems. Thus, this work contributes towards creating feedback loop between the network and the user with its implementation in network. Along the way, this work assesses the impact of waiting times on the user subjective opinions using subjective tests. The subjective tests help understand the dynamics of users in response to the delays occurring on the network.

Concisely, in context of the aforementioned description, this thesis deals with the following three research objectives:

Research Objective I: To understand the relationship between waiting times and user subjective feedback over time.

The first objective of this thesis work is to understand the fundamental relationship between waiting times and user subjective opinions. To achieve this objective, three subjective tests were designed to assess the impact of waiting times on QoE for a web browsing service. These subjective tests studied QoE at page and task-based session levels. The results of tests, amongst others, strengthened the understanding of user reactions to delays over time and the role of user memory at the page as well as the session levels. Paper I – Paper III discusses results from subjective tests.

Research Objective II: Monitoring and visualization of network performance issues at multiple timescales, that potentially degrade QoE over time.

The second objective of this thesis is to propose method for passive network-based detection of performance issues, which may potentially degrade QoE. The first step towards this objective is to propose metric reflecting performance issue from user perspective, i.e., the issues that may result in recurring waiting times. The second step is to devise an approach to detect recurring performance problems for quantification of user waiting times. Thus, the method must take into account multi-timescale view of network performance problems to be able to relate them to QoE. To meet this objective, this thesis proposes transient outage within data transfers as a metric to express QoE degradation issues. Subsequently, this thesis discusses and evaluate a wavelet-based method to monitor and visualize transient outages at various timescales. Paper IV – Paper V presents wavelet-based method for outage detection.
1.2 OUTLINE

Research Objective III: Network-based monitoring of user reactions to performance problems.

The third and the final objective of this thesis work is to devise method for monitoring of passive network-based indications of user reactions to the performance issues. The users may lose patience and break the ongoing data transfers, if the waiting times are high above their expectations. To be able to monitor these user reactions, this thesis evaluates the indications that appear in the network traffic in case of different user actions in the web browser. This research objective compliments the Research Objective II, as the detection of recurring transient outages followed by the detection of transfer terminations alarm network operators about the existence of serious QoE degradations. Paper VI discusses findings related to the systematic detection of termination of transfers using TCP flags.

1.2 OUTLINE

This thesis is divided into two parts. The first part introduces thesis with research objectives, followed by detailed background, problem statements, research contributions, main conclusions and the future work. The second part constitutes of research papers published in peer-reviewed conferences and journals. Each paper addresses a certain research objective discussed in the previous section. Figure 1.1 sketches the structure of this thesis.

Chapter 1 presents introduction and research objectives of thesis. Chapter 2 briefly describes the background on the network management. Considering the big picture, this thesis belongs to the area of Network Management. Chapter 3 introduces Quality of Experience (QoE) assessment and places this thesis into the relevant investigation area of QoE assessment. Chapter 4 with title QoE-Centric Network Management extends the discussion further to an overall aim of QoE-based Network Management, where the contributions made in this thesis can be utilised. Chapter 5 lists the problem statements followed by research contributions of attached papers. Finally, chapter 6 concludes this thesis with the main conclusions according to the research objectives described in this


**CHAPTER 1. INTRODUCTION**

This part introduces the scope and research objectives of this thesis. It also discusses the background of research area, followed by problem statements and conclusions.

<table>
<thead>
<tr>
<th>First Part: Chapters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 1: Introduction</td>
</tr>
<tr>
<td>Chapter 2: Network Management</td>
</tr>
<tr>
<td>Chapter 3: Quality of Experience</td>
</tr>
<tr>
<td>Chapter 4: QoE-Centric Network Management</td>
</tr>
<tr>
<td>Chapter 5: Problem Statements</td>
</tr>
<tr>
<td>Chapter 6: Conclusions &amp; Outlook</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Second Part: Papers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Paper I–III</strong> address the <strong>Research Objective I</strong>: To understand the relationship between waiting times and user subjective feedback</td>
</tr>
<tr>
<td>Paper I: Quality of Experience from User and Network Perspectives</td>
</tr>
<tr>
<td>Paper II: Back to Normal? Impact of Temporally Increasing Network Disturbances on QoE</td>
</tr>
<tr>
<td>Paper III: In Small Chunks or All at Once? User Preferences of Network Delays in Web Browsing Sessions</td>
</tr>
</tbody>
</table>

| **Paper IV–V** contribute towards the **Research Objective II**: QoE monitoring on multiple timescales |
| Paper IV: Modeling and Analysis of Web Usage and Experience Based on Link-Level Measurements |
| Paper V: Quantitative Evaluation of Wavelet-Based Traffic Gap Detection |

| **Paper VI** addresses the **Research Objective III**: Monitoring of user reactions to network performance |
| Paper VI: Inferring User-Perceived Performance of Network by Monitoring TCP Interruptions |

---

**Figure 1.1: Thesis structure**

...

chapter.
Chapter 2

Network Management

The communications network is a collection of nodes, and links that interconnect these nodes in order to enable communication between two terminals. These nodes and links require management to function and deliver services according to the set objectives. The main task of network management is directed towards the planning and execution of activities and tools, which keep the network running according to the specified goals. At the same time, network management must also care about resources, such that, it keeps the expenditures under control, and achieve planned revenues for the organization.

The tasks for which networks are designed may vary. Some networks are very small in size based on a very few nodes, while others are large, spread over several geographical regions, consisting of thousands of nodes and links. Similarly, the nature of tasks for which these networks are designed also differ, depending on the context. Some tasks are more time-critical than others, such as, interactive services like gaming and tele-meetings are more time-sensitive than the traditional file downloads. These differences also bring heterogeneity in the technology involved in the management of networks, which brings a lot of challenges for network managers to choose between the right set of tools to manage resources for meeting the given demands.

In [1], Network Management is formally defined as:
"Network Management refers to the activities, methods, procedures and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems." – (Network Management: Principles and Practices by Mani Subramanian)

The book “Network Management Fundamentals” written by Alexander Clemm also provides a similar definition of network management. It defines network management as operation, administration, maintenance and provisioning of networked systems [18]. The Cisco handbook on Internetworking technology defines network management as tools and devices that assist in monitoring and maintenance of network [19].

Practically, Network Management is rather seen as FCAPS management, i.e., Fault Configuration, Accounting, Performance and Security management [20]. The notion FCAPS management is created by International Standard Organization (ISO) under the proposed network management framework. All the functions in FCAPS are based on monitoring and analysis, which are deemed as the backbone of functional dimension of network management.

### 2.1 MODELS OF NETWORK MANAGEMENT

The (ISO) has defined the following four models of Network Management [1]:

- Organization model
- Information model
- Communication model
- Functional model

The following subsection briefly introduces each model before sketching a bigger picture about the monitoring and analysis on network, which is deemed as the backbone of Network Management.
2.1. MODELS OF NETWORK MANAGEMENT

2.1.1 Organization model

Organization model defines the role for entities, which communicate on the network to exchange information for performing network management functions. These entities are mainly divided into two categories: Manager and Agent. In order to monitor and analyze the network, the manager acquires information from the agents on network. The agents are simply nodes on a network performing different functions, such as counting of bytes and packets, recording up and down times of system, keeping configuration details, so on and so forth. These agents can be routers, switches or any additional system on the network deployed to probe the details about network.

2.1.2 Information Model

Information model specifies the structure and storage of information on the nodes. In this regard, the ISO has defined Management Information Base (MIB) for nodes on a network. The MIB is a kind of a database, which stores different pieces of information on the network nodes. The manager may request for the information from agent(s) using the specified MIB address, where the related information is stored.

2.1.3 Communication Model

Communication model provides a set of messages and protocol used to exchange information between the manager and the agent. The manager requests for information from the agent using messages specified for requesting the information from manager to agent. Similarly, the agent responds with the special message type reserved for the response message containing the requested information. The communication model also set the protocol used to exchange all these messages. For example, the Simple Network Management Protocol (SNMP) is one of the widely used protocols for communication of management information.
2.1.4 Functional Model

Functional model, as the name suggests, deals with the management functions performed on the network. According to ISO, there are mainly five categories of functions performed for managing the network, which is commonly known as FCAPS: Fault, Configuration, Accounting, Performance and Security. These five functions form the basis for network management functionality.

Figure 2.1 sketches a scenario of basic communication taking place between a manager and agents. This example uses the four models defined by ISO. The manager requests for information followed by the corresponding response from agent. The agent sends unsolicited notifications in the form of alarms, if any undesirable situation occurs. Take an example of traffic utilization on links. The manager monitors the number of bytes passed through a certain interface of router. The location of counter within the router has a certain MIB address (as defined by the information model). The manager keeps polling router and requests for the value of counter representing the number of bytes passed through the interface by using that particular MIB address of byte counter. The agent responds with the value of the counter, which may then be used by manager to perform different types of analyses and present results in the user interface. However, sometimes the agent may set an alarm (without any request from manager) to notify manager when the number of bytes within an interval of time exceeds a certain pre-defined threshold. The communication between the manager and the agent takes place using SNMP protocol (as standardized by the communication model). The information regarding the counter is used by the manager for performance management function of FCAPS.

In the above example, manager requests information from agents, and process them centrally at single point. Based on the collected information, the manager derives a set of metrics representing the overall functionality of network. However, the centralized processing of the data collected from agents makes it computationally difficult for the manager to provide a better view of whole network if the network is too large, which is specially the case for the network operators today.
In order to cope with this situation, Remote MONitoring (RMON) probes are used. These probes are distributed at several locations within a network. Each probe locally monitors a certain segment of network, processes the data and sends results to the manager for visualization of monitoring information. This decentralized architecture for network monitoring and analysis brings greater productivity for network operator by shifting intelligence to the edge of network, and supporting FCAPS on local segments of network, thus, reducing management traffic load on network links.

2.2 QoE-Centric Network Management

Previous section provided a brief overview about the monitoring and analysis architecture used to support network management functions. The next important aspect of network management is the usage of appropriate metrics to provide an effective view of each network management function in the FCAPS. Particularly, the metrics representing faults and performance issues on network must be QoE-centric, i.e., the metrics should accurately represent the mentioned issues, as actually perceived by the users of a network service.

The next chapter will briefly introduce QoE and a summary of efforts made by the research community and industry to assess and improve QoE of network and application services. Subsequently, the QoE-centric network management is explained further in Chapter 4.
CHAPTER 3

QUALITY OF EXPERIENCE ASSESSMENT

This chapter will give a brief overview of the objective assessment of QoE. The QoE is fundamentally a measure of the subjective assessment of a service performance made by the user. For example, users give their feedback about service performance in the form of Mean Opinion Score (MOS). However, the subjective assessment is not always possible as it consumes a lot of time and resources to organize efforts for obtaining subjective feedback of large number of users. As an alternative, objective assessment models automatically assess QoE of a service over time. Thus, repetition of subjective assessment can be avoided by using objective assessment models.

3.1 OBJECTIVE QoE ASSESSMENT MODELS

The construction of objective assessment models requires a set of metrics or parameters, which can be modelled against the user subjective feedback. The parameters are usually the performance indicators of a service, which can be measured objectively. Hence, the first step is to determine the performance indicators of a service. Some of the widely used performance indicators in-
CHAPTER 3. QUALITY OF EXPERIENCE ASSESSMENT

clude packet loss, delays and throughput. The second step is to model the selected performance indicators against subjective feedback of users. Finally, the constructed objective assessment model is used in different usage scenarios to calculate the QoE level of a service.

The determination of parameters depend on the scenarios in which model is intended to be used practically. On a high level, there are generally two scenarios in which QoE objective estimation models are used. First, in the test environment using active tests, and second, the production environments, i.e., on the live traffic via passive observation. In the active tests, a user emulator/replicator generates traffic from a certain application on the network. Meanwhile, the required performance metrics are collected, which can be used in the objective QoE assessment models. Conversely, in passive observation on an operational network, the real users usually generate traffic, which is collected to find the values of required performance metrics. The QoE assessment models then use these values to estimate QoE.

The type of data, which can be collected in both the aforementioned scenarios differs based on the extent to which client, network and server-side instrumentation are done. In active tests, it is often easier for network operators to collect data from client-side device and application in addition to the network traffic. However, this is usually difficult in case of passive observation due to absence or lack of control on client-side device and application. In short, the amount of data, which can be accessible differs based on the probe. It also determines, which QoE assessment model can be used in a given context. This leads the discussion to the active and the passive probes. The following description further explains these probes.

3.1.1 Active Probe

In active probing or active testing, traffic or signal is sent on a network to test the quality of transmission. As described above, in this environment, the client-side instrumentation is possible. Hence, it makes the collection of network, client-side device and application level details easier. Thus, objective QoE estimation
3.1. OBJECTIVE QOE ASSESSMENT MODELS

model in this environment may take into account several different influencing factors at multiple layers to estimate QoE. The factors may include:

- the characteristics of actual received content, such as the complexity of actual played video, or the requested web page,
- the type of application used at the client side, such as, the web browser or video streaming application,
- the arrival/display times of information in the application interface, such as, the display of frames in the video on screen or the rendering of HTTP responses in the web browser.

Hence, active tests allow more control on the collection of information about the transmitted traffic. However, it may still lack the behaviour of real users and their corresponding usage scenarios. Due to this reason, the monitoring systems based on the data reported by active probes may not sketch an accurate picture about how the real users actually perceive the service in real time. Hence, the models may become inaccurate in terms of capturing the real user QoE.

3.1.2 PASSIVE PROBE

On the operational networks, passive probes monitor the traffic generated as a result of user actions at the application level. The objective quality assessment models in this case rely on the information, which may be extracted from the packets on the network.

Often, client-side instrumentation is not possible due to a number of reasons, which amongst others include user privacy constraints and extra processing load on the user device. Therefore, the performance parameters from the application-level are difficult to obtain in the production environment. The quality can only be inferred or assessed based on indicators at the packet level, if server side logs are unavailable. However, passive probes help understand the user and the usage behaviour in reality. Moreover, network operators can make timely
actions to control the quality levels by taking appropriate measures, as a result of close-to-real-time monitoring and assessment.

Objective assessment models for network-based QoE monitoring rely on the information from the payload and the header. In the standardization activities performed by the ITU-T study group 12, the above-mentioned two groups of models are usually referred to as:

- the bitstream models
- the parametric packet-layer models

### 3.1.3 Bitstream models

The bitstream models rely on payload information of streams above the transport layers [21][22][23]. The information about content characteristics may also be available to these models. However, depending on the encryption of streams, the information about payload may not be available. In this case, client-side instrumentation is required. Thus, it is difficult to implement these models in the passive probes on network.

Generally, the bitstream models are more suitable to work in the active probes. These models are often slow as there is computational complexity involved in processing the elementary streams, such as data, audio or video signals. However, the offline estimation of QoE might still be possible, depending on the privacy constraints and the availability of required information about stream.

### 3.1.4 Parametric packet-layer models

The parametric packet-layer models inspect only the packet header information and estimate the values of different performance parameters, such as, loss, throughput, delay and delay variation [11][24]. The models usually map parameter values to a QoE score. Additional information, such as, the video codec used, may also be available [25]. The parametric packet layer models are generally
3.1. OBJECTIVE QOE ASSESSMENT MODELS

considered lightweight models, as they do not require deep packet inspection. These models may work in both the active and the passive probes.

In addition to the QoE estimation at a given time, one of the ultimate objective of assessment models is to provide the diagnostic information, which helps operators to reach to the root cause of observed QoE degradations. It is called glass box approach in the ITU-T recommendation G.1011 (05/2013) [26]. A large number of QoE assessment models are based on the QoS parameters such as packet loss, delay and throughput. These QoS parameters themselves depend on a number of factors, such as, available resources (e.g., link capacity) on access or core network, network coverage, user mobility and protocol functionality etc. Hence, modelling QoE against QoS parameters may not automatically help operators to find the root causes of QoE issues. Therefore, QoE monitoring systems need to consider QoE assessment approaches, which help in pinpointing the ultimate cause of QoE degradation.

Furthermore, it is equally important to consider the appropriate time granularity while developing QoE models. As mentioned previously, the subjective tests are often performed using short audio-visual sequences or a couple of web pages. The obtained user opinions are then often modelled against the measured average values of KPIs. Consequently, the models may suffer when it comes to continuous QoE estimation of a long transmission over time. Consider a session in which user watches multiple short videos or one long video sequence. Similarly, take a task-based web browsing session spanning over several web pages as an example. In these scenarios, the user satisfaction may not be one-to-one dependent on a single performance issue or a degradation event, but is an outcome of a sequence of inter-connected events. Therefore, QoE assessment models need to consider the impact of these dynamics over time. Specifically, the models which are better representative of realistic usage scenarios (from short to long sessions) need to be designed. Undoubtedly, a multi-timescale view of QoE can help achieve this objective.

This thesis contributes towards passive network-level assessment of QoE. It proposes and evaluates a method, which can be used to monitor recurring QoE issues over time. Particularly, the results discuss the timescales on which
users react to the waiting times and how these problems can be detected using multi-timescale-resolution analysis.
To devise user-centric network management mechanisms, understanding of the communication between user (including application) and network is important. Communication works in both directions, i.e. from the network to the user, as well as, from the user to the network. The events that escalate from the network to the user in the form of performance and fault issues affect the user interaction with the service, which can then be observed in the form of traffic characteristics on network, driven by the user behavior and actions.

Moreover, in addition to monitoring fault and performance issues propagating from network to user, QoE monitoring agents on network may also collect information about user behavior indications from network traffic characteristics. The monitoring should be done as close to the user (or a set of users) as possible to minimize the impact of any additional factors affecting service, along the path between the network and the user. Based on the available information in both directions, the objective QoE models estimate MOS score at different timescales and report them to the Network Manager periodically as shown in Figure 4.1.

The agents may estimate QoE by probing network actively or passively. In
the active probing, the client-side instrumentation is often available as requests are made by the artificial client. The agents estimate QoE using objective QoE models based on the received performance at the client-side application. In the passive monitoring, it is however challenging to assess from network the performance received actually by the user. Therefore, the monitoring of user behavior in the form of user actions from network traffic complements the results from the objective QoE estimation.

The network manager polls the agents periodically for the reports about the estimated MOS scores at desired timescales. Additionally, historical reports can be compiled to estimate overall QoE over longer timescales, such as days, weeks or months using complex integrated QoE models. In the undesirable situations, such as, long outages or user-perceived fault events, the agents may alarm network manager for immediate actions.

The next step in the QoE-centric Network Management consists of dynamic resource allocation and management. In the events of performance degradation or faults, the network manager needs to take actions by scheduling resources to raise QoE levels. For example, when recurring outages (resulting in frequent video freezes) – due to the inappropriate management of resources – annoy users,
the network manager must take immediate actions to optimise the resources and thus, raise QoE levels. Similarly, the network manager need to take decisions to avoid the under-utilization of resources. For example, if a user is not using certain resources on the network, network manager may take timely actions to release resources, and allocate them to the users who need these resources. The backbone of a good resource management policy is based on the user-centric monitoring and analysis of network.

4.1 QoE Monitoring from Sessions to Packets

The another important aspect of QoE-centric monitoring and management of networks is the understanding of QoE timescales. The studies show that the overall user experience of a service or a product evolves over time, taking into account the dynamics of memory effect and expectations [15][27]. Therefore, the monitoring solutions need to provide view on QoE over multiple time-scales. Papers II and III present findings related to the studies on evolution of user subjective opinions over time.

![Monitoring scale: From Session to Packets](image)

Figure 4.2: Monitoring scale: From Session to Packets
Chapter 4. QOE-Centric Network Management

Figure 4.2 depicts granularity from the packets up to the sessions level. The finest granularity (in the figure) is based on the performance metrics estimated at the packet level. Depending on the particular performance criteria or metrics, the time interval of the calculation varies. For example, the number of lost packets can be counted over the whole file download or over smaller time intervals. Similarly, average packet throughput can be calculated per RTT or in the fixed intervals in a download depending on the designed probes.

The next (coarse) level of monitoring is at the object level. These objects refer to the web pages downloads in this case. The object could be image, text or application on a web page. The estimation can be in the form of object load time or the number of objects loaded in a certain intervals. Each object could be composed of one of more network-level packets.

On a further higher level, one or more objects form a web page or a download. The QoE monitoring could be based on the page-level performance metrics, such as, the render start or render end times of web pages. It gives a further coarse-grained view on QoE and in case of performance or fault issues, only page-level view make it difficult for the operators to reach to the root cause of the problem.

The highest-level view in Figure 4.2 is based on the complete session of usage performance by the user on the Internet. For example, a web browsing session can be based on the visit of one or more web pages by the user. The session QoE will be based on the accumulation of all the different experiences over the course of session time. A session can be a very small based on only one download/upload or it can be very long based on several downloads/uploads. While it can provide an overall view of a user QoE, it will certainly need a view on smaller timescales to localize the issues that damage QoE.

The monitoring can further move beyond the scales shown in Figure 4.2. Such as, the scales further narrowed down with monitoring on the link-level (bits and bytes) or upwards to the multiples of sessions on very long timescales. The byte-level monitoring at a very small timescale may not correspond to the user perception timescales, but may help in the detection and isolation of the root causes of problems affecting QoE.

Hence, the remote monitoring probes or agents can be implemented, which
provide flexibility to have the view about performance and quality on several different scales described above. The metrics on one or more of these scales can be requested by the manager from the agents distributed at various locations on the network. To bridge the above discussion further to the practical implementation, the next section provides an overview of the QoE estimator agents developed in Celtic project Quality of Experience Estimators in Networks (QuEEN).

4.2 QoE estimator agent in QuEEN Project

QuEEN specifies agent for the estimation of QoE [28]. The generic structure of agent allows objective QoE assessment model at layers. Moreover, the agent can be used with the existing probes using Simple Network Management Protocol (SNMP).

The organization model of agent defines two roles: Master and Slave agents. Several slave agents can be distributed within a network. The agents may acquire data from the existing probes on network and apply the respective QoE models. The slave agents then report the results of QoE estimations to the Master agent via SNMP.

QuEEN agent specifies a Management Information Base (MIB) subtree, which gives a unique identification to the objects, thus, making it suitable for QoE-specific estimations. The name of subtree is qoe-monitoring with object ID of 200. The subtree is a leaf of the experimental (3) node in iso.organization.dod.internet (1.3.6.1) MIB. The structure of this subtree is depicted in Figure 4.3.

The agent node within this subtree defines a number of objects within QoE agent. For example, inputs and output value of a particular QoE model can be accessed using the model object. Similarly, metrics node lists the QoE indicators used for estimating QoE.

New metrics for QoE estimations can be specified as child nodes of metrics node. For example, the network-based QoE estimation metrics defined and evaluated in this thesis can be added as leaves of metrics node. Moreover,
different metrics can be used to indicate QoE at various different timescales from seconds to hours to days.
Chapter 5
Problem Statements

This chapter lists the problem statements from each of the attached papers.

Paper I: Quality of Experience from User and Network Perspectives Main research questions

1.1. What is the relationship between user opinion scores and QoS parameters, such as loss, throughput and download time?

1.2. Do traffic characteristics like session volumes change with the changing network QoS?

1.3. What is the relationship between session volumes and QoE?

Research contribution

This paper presents our study on the correlation between network-level QoS and QoE perceived subjectively by the users. The study has taken two approaches to map the user behavior to network QoS.

The first approach is based on the user perspective, which takes into account the subjective ratings by the users in the test environment. Users perform a web browsing activity and then rate the service. The performance of network is shaped by introducing different loss rates on the network. The QoS parameters
such as loss and throughput are measured on the network level. The download
time of each web page is also measured on the application level. The mapping is
then performed between the user subjective responses and the QoS parameters
to extract the thresholds on the QoS parameters with regards to QoE. Finally,
the relationship between QoE and each of the QoS parameters is derived with
the help of regression analysis.

The second approach is based on the study of traffic traces, captured on
the operational network of an major telecommunication operator. Relationships
between the above-mentioned QoS parameters (losses and throughputs) and the
user session volumes are derived to observe the interest of users in the service
at different performance levels.

Finally, the relationships derived from the results of test and operational
environments are compared, in order to relate the objectively-measured user
session volumes to the subjectively-measured QoE. It was found out that the
user session volumes increase with the increasing user experience which shows
that the happy users surf more.

In this work, I made the major contribution, which includes experiments,
measurements, analysis and writing under the continuous supervision of two
co-authors.

Paper II: Back to Normal? Impact of Temporally Increasing Net-
work Disturbances on QoE Main research questions

2.1. How do users rate page load times before and after facing network distur-
bances?

2.2. Do user ratings recover immediately after network problems are resolved?

2.3. Do user segments exist with regards to tolerance level in the face of waiting
times?

Research contribution

Users often experience brief episodes of network failure and performance
issues in the form of long waiting times during the delivery of content. After a
while, when the problems resolve, the network performance gets back to normal. This paper investigates if the user satisfaction level also gets back to normal (i.e. corresponding to the pre-disturbance phase) or not?

To investigate aforementioned question, we conducted task-based subjective tests in lab. Users went through multiple shopping sessions and bought products online on a given web site. They rated the page load times in the form of MOS scores at each web page during the shopping sessions.

The findings of the paper shows that the QoE decays with recurring problems on the network. The MOS scores do not recover immediately after network performance gets back to normal. This finding applies to all the subjects participated in the tests. However, in terms of overall tolerance to disturbances, four segments of users exist. Some users are thus clearly more intolerant than the others right from the start to the end of the tests.

I lead the contribution in this work under continuous guidance of Markus Fiedler. The co-author Pangkaj Paul actively participated in the experimentation and result analysis. The discussions with last two co-authors helped me in designing and executing this study.

**Paper III: In Small Chunks or All at Once? User Preferences of Network Delays in Web Browsing Sessions Main research questions**

3.1. How do users respond to the short but frequently occurring delays in a web browsing session?

3.2. How do users respond to the long but rarely occurring delays in a web browsing session?

**Research contribution**

This subjective study investigates about the distribution of delays that users prefer during a session, given a fixed overall session waiting time. In the study, each user went through three shopping sessions based on five web pages each. The users faced the same nominal overall waiting time in each session. The only difference was in the spread of duration and frequency of delays across the webpages in a session. The longer the duration of delay, the rarely they occur
during a session. Thus, the study investigated tradeoff between duration and frequency of delays during web browsing sessions.

According to the results, users prefer small but frequently occurring delays as compared to the long but rarely occurring delays. They prefer 4 s load time occurring at every page throughout the session in comparison to the 16 s waiting time on a single page with all the other pages having only 1 s load time. The findings were consistent regardless of the sequence in which the users went through the sessions.

All the co-authors participated actively in this study, as well as, the publication writing.

**Paper IV: Modeling and Analysis of Web Usage and Experience Based on Link-Level Measurements**

**Main research questions**

4.1. What are characteristic of traffic gaps caused due to the user inactivity on the Web?

4.2. What are features of traffic gaps typically induced by network?

4.3. How to identify traffic gaps caused by network at multiple timescales using wavelet analysis?

**Research contribution**

This paper presents passive monitoring and analysis method, which assists in the identification of those traffic gaps on the network that may result in the degradation of QoE. The gaps in traffic can also be due to the inactivity of the user (the user think times) between two transactions as well as the behavior of the application as depicted by classical ON-OFF models. This paper first revises the classical ON-OFF model to cater for the OFF times reflecting the accidental traffic gaps, induced by the network. It then proposes a wavelet-based criterion to differentiate between the network-induced traffic gaps and user think times. As it doesn’t require deep packet inspection, the criterion is simple and intended to be implemented in near-real-time.

The original idea about multi-resolution analysis came up during discussions with Markus Fiedler. I executed the study from measurements to analysis un-
der continuous guidance of Markus Fiedler. I lead the paper writing as main contributor, while, the other authors actively participated in the discussions, writings and corrections.

**Paper V: Traffic Gap Quantification using Wavelets Main research questions**

5.1. How do energy of wavelet coefficients change with the change in duration and frequency of traffic gaps?

5.2. How do energy of scaling coefficients change with the change in duration and frequency of traffic gaps?

5.3. What are the characteristics of wavelet and scaling coefficients at timescales corresponding to the duration of transient outages?

**Research contribution**

Paper IV (previous paper) presents wavelet-based criterion for traffic gap detection via qualitative measurements on two different networks. This paper guides the discussion further by presenting a systematic quantitative evaluation of wavelet-based traffic gap detection. Using a variety of traffic traces with deterministic and non-deterministic (model-based) traffic gaps of nominal durations, this paper discusses how wavelets detect the timescales on which the problems occur. Thus, the results show how an ample understanding of duration and frequency of recurring traffic gaps can be acquired via the values of wavelet and scaling coefficient energy functions at various timescales. Paper IV and Paper V together assist in meeting the research objective II explained in 1.

I made the major contribution in this paper under supervision of Markus Fiedler and Patrik Arlos.

**Paper VI: Inferring User-Perceived Performance of Network by Monitoring TCP Interruptions Main research questions**

6.1. How do TCP connections terminate in the case of interrupted and uninterrupted transfers?
6.2. Does the TCP connection termination process differs due to the client side mobile web browser?

6.3. How do the content types affect TCP connection termination process?

6.4. Can we infer actions performed by the user in web browser by monitoring TCP connection termination process?

**Research contribution**

In this paper, findings obtained from a systematic study of the TCP connection termination behaviors for web transfers are discussed, which include a set of active tests conducted in an isolated environment. These tests were conducted using various mobile and desktop web browsers and content types. The objective of the study was to investigate the difference in the TCP connection termination process in the case of interrupted and uninterrupted web transfers. It was observed that the TCP connections interrupted by the user usually consisted of more than one consecutive TCP reset (RST) flags from the client-side.

I lead this study and made major contributions to the paper.
Chapter 6

Conclusions & Outlook

6.1 Conclusions

This section draws a set of conclusions from the studies conducted in this thesis work. The conclusions are made with regards to the three research objectives presented in Chapter 1. These conclusions from each research objective are discussed below:

- **Relationship between waiting times and QoE:** In response to the research objective I, Papers I–III in this thesis discuss results from detailed subjective studies. The studies conclude that the exponential relationship fits best between waiting times and user opinion scores. In single page no-task scenario, user opinion scores drop below acceptable level as waiting times approach 3 s. The limit for acceptable page load times extend above 4 s in the task-based sessions (multiple webpages). However, the user memory appeared to be a strong factor in influencing the waiting time threshold acceptable by the users. The impact of frequently recurring web page delays accumulate over time in user memory, which result in the global decay of user opinion scores. Moreover, considering a 5-page shopping session, users generally prefer short less than 4 s load time on all pages in a session in comparison to the one long 16 s page load times.
CHAPTER 6. CONCLUSIONS & OUTLOOK

• QoE-Centric monitoring of user performance issues: QoE evaluation requires continual assessment of delivered service over time. The user feedback in response to a service performance degradation event may not be the outcome of only a single event, but depends on multiple degradation events occurred over time due to the memory effect. In order to keep track of all such events, multi-timescale monitoring and visualization of service performance is important. Thus, Paper IV takes a step forward into the research objective II. It proposes a wavelet-based criterion to detect traffic gaps (transient outages) on multiple timescales via qualitative measurement of traffic streams on two networks. It concludes that the local maxima in the energy of wavelet coefficients at a certain timescale indicates the sign of recurring gaps at the corresponding timescale. Particularly, the network with bad QoE exhibits scaling in energy of wavelet coefficients at timescale ranging from 1 s and 4 s, indicating recurring gaps at the corresponding timescales.

Motivated by the results from qualitative measurements, Paper V took a deeper look at the wavelet-based traffic detection using results from detailed quantitative measurements. It showed that the streams with recurring traffic gaps at certain timescales followed by the burst of packets results in the global maxima at the corresponding timescales. The peak in the wavelet energy highlights problem timescales. Furthermore, the scaling coefficients also detect the duration of traffic gaps. The energy of scaling coefficients level off at the time scales corresponding to the duration of traffic gaps.

• Monitoring of user reactions to performance issues: The monitoring of user reactions compliments QoE-centric performance monitoring. The user annoyance indicators strengthen the understanding of problems perceived by the user. Paper VI presents and evaluates method for monitoring of connection abandonments made by the users. The results show that the TCP connection termination process also depends on the client-side platform besides user action in the web browser.

Moreover, another parallel study (not included in this thesis) analysed traffic traces captured on network operator’s network [29]. It showed
that transfers terminate abruptly when individual requests within transfers take longer time. It indicates that users abandon transfer based on the time taken by the individual request. This evidence appeared while comparing transfer times of the last requests of interrupted and uninterrupted TCP connections. The last request within interrupted TCP connections on average took longer time than the last request of uninterrupted TCP connections.

6.2 Future Work

Future work needs to devise mechanisms, which do not only detect QoE issues, but also be able to link them to their root causes. Thus, network management systems need to leverage the benefits of QoE monitoring in order to adapt resources and increase revenues for a business. The network-based QoE monitoring method presented in this thesis need to be further improved by linking it to the application-level details and user subjective opinions.

As a long-term future goal with regards to QoE monitoring and improvement, all the stakeholders in a service delivery chain need to work together. This means that device manufacturers, application developers, network operators and service providers require synchronisation to understand the changing user expectations under different usage contexts and offer a delightful user experience everywhere and at all times.


PAPER I

QUALITY OF EXPERIENCE FROM USER AND NETWORK PERSPECTIVES

PUBLISHED IN JOURNAL
ANNALS OF TELECOMMUNICATION
VOLUME 65, ISSUE 1–2, PP. 47–57, 2010
Quality of Experience From User and Network Perspectives

Junaid Shaikh, Markus Fiedler and Denis Collange*

Blekinge Institute of Technology, Karlskrona, Sweden, (junaid.junaid, markus.fiedler)@bth.se

*Orange Labs, Sophia Antipolis, France, denis.collange@orange.com

Abstract – The impact of network performance on user experience is important to know, as it determines the success or failure of a service. Unfortunately it is very difficult to assess it in real-time on an operational network. Monitoring of network-level performance criteria is easier and more usual. But the problem is then to correlate these network-level Quality of Service (QoS) to the Quality of Experience (QoE) perceived by the users. Efforts have been done in the previous years to map user behavior to traffic characteristics on the network to QoS. However, being able to successfully relate these traffic characteristics to user satisfaction is not a simple task and still requires further investigations. In this work, we try to associate on one side the correlations between various traffic characteristics measured on an operational network and on the other side the user experience tested on an experimental platform. Our aim is to observe some pronounced trends regarding relationships between both types of results. More precisely, we want to validate how and to what extent the volumes of user sessions represent the level of user satisfaction. Along this way, we need to revise classical relationships between some of the network performance indicators such as loss, download time and throughput in order to strengthen the understanding of
1 Introduction

There has always been a gap of perception between the Internet Service Providers (ISPs) and their customers when talking about the performance of network service. The reason is that providers and users use different criteria to assess the performance. Service providers often use specific network level Quality of Service (QoS) parameters like throughput, loss ratio or delay to measure service performance. These parameters are typically measured on network nodes, or between two provider’s machines. In contrast, users usually perceive the service performance in more subjective and non-technical terms. They want to be served within a reasonable response time. They are uninterested in the values of these technical network parameters. This subjective perception of the users is usually called Quality of Experience (QoE).

The common practice to estimate user perception from network-level performance criteria is to conduct out many large experiments in a controlled environment. Some performance criteria are modified in a given range and different panels of typical users give a mean opinion score (MOS). This method has more especially been applied to voice and video traffic. However such a comprehensive practice is no more applicable today on Internet: the number of applications is very high and always growing, for each application new versions are regularly released with new functions, new traffic characteristics, new performance requirements, etc. The usages of the applications may also very different depending on the users. Furthermore the expectations of the users vary a lot depending on their experience, their access to Internet, the other applications they use. So the old comprehensive practice to assess the feeling of users about the network-level performance is too expensive to be applied to all the existing applications on Internet.

A new method has then been proposed in [3] to infer automatically from passive measurements on an operational network the user perception. On a real
network, the millions of active connections observe a wide range of performance. The behaviors of the users characterized through various traffic metrics show strong correlations with the network-level performance, even if the reaction of the protocols may also have an impact. Thresholds on QoS levels can then be deduced from these measurements: from the point where some traffic characteristics begin to change, until the point where no connection succeeds. There is however no validation in paper [3] neither about the real feelings of the users, nor about the correlations of these feelings with the traffic characteristics.

The objective of our analysis is then to compare these two methods to correlate the user perception with network-level performance criteria: the classical comprehensive method based on experiments on a testbed, and automatic passive method analysing the correlations between some traffic characteristics and some performance criteria.

User perception is amongst others seen from service utility, the relative usage of a service by users. This usage might be affected by network performance. If the latter is good, the user is motivated to maintain or even increase its activity level. However, bad network behavior may make users give up and declare a service useless for them, which would reduce the service utility. Hence service performance can have a strong impact on service utilization by the users. Our aim in this paper is to investigate whether the use (in volume) of a service is a function of the perceived quality and how it correlates with the subjective ranking by the users. The results should be given in formulae which are easy to understand, interpret and applicable for threshold control.

This paper presents a comprehensive analysis on the changing user behavior at different service performance levels through both objective and subjective measurements. First, it discusses the correlation of subjective grading of the service by the users with a set of service performance parameters. In this context, the relationships between these key parameters are reviewed and compared to published work. This way, we obtained a systematic, quantitative view on the effects of data loss on both objective and subjective parameters. Furthermore, the paper discusses significant threshold values of service performance in accordance to user perception. This analysis is based on the results of web
surfing experiments on a test-bed. Second, it discusses the correlation of traffic characteristics of user sessions with several network performance metrics. This discussion is based on operational traffic generated by real users on an ADSL network. Finally, a few results from both methods are compared to show how and to what extent they complement each other. Our results are mainly divided into two parts: the results obtained from the experiments on the test-bed of Blekinge Institute of Technology (BTH) and traffic captured on the operational ADSL network of France Telecom (FT).

The remainder of the paper is organised as follows: Section 2 provides an overview of related work. Section 3 describes BTH’s measurement platform and methodology, the impact of the loss ratio on throughput and download time, and the relationship between QoS and QoE parameters. Section 4 describes first FT’s measurement platform and methodology, a selected set of general traffic characteristics on the network. It shows then an analysis of the correlations between traffic characteristics and some performance metrics. Section 5 attempts to compare the results from the two previous sections, aiming at identifying trends for how the users’ satisfaction correlates with their activity. Section 6 concludes and points out future directions of work to be done.

2 Related work

There is a wide range of factors that influences the QoE. Moreover, their relative impact depends on the application. ITU-T Recommendation G.1010 discusses several key parameters and their impact on user perception classified by different types of applications. These key parameters include delay, delay variation and information loss. Several interesting thresholds on these key parameters are discussed concerning usage of different applications [10].

ITU-T Recommendation G.1030 [11] presents experimental results regarding the subjective responses of different types of users in relation to response times of web browsing sessions [11]. The Mean Opinion Score (MOS) is approximated using the logarithm of normalised response times. This recommendation is also useful for realising the impact of user expectation and background on the user-
perceived quality of service.

Finding indicators of user satisfaction from network traffic traces is an important way of analysing user behaviour. The Transmission Control Protocol (TCP) connection termination process is a useful resource of indirectly observing the user feelings. In 2003, user experience described by the interruption probability of user Hypertext Transfer Protocol (HTTP) connections in relation to the sizes of the flows i.e. TCP connections between hosts, their average throughput and connection completion time was presented [16]. A similar type of study is carried out by the authors of [13] to present the results regarding user cancellation rate of HTTP connections in relation to response times and effective bandwidth. The authors of [3] discuss some characteristics of user’s transfers and their correlation with network performance parameters. In [8], a relationship between loss and QoE on Mean Opinion Score (MOS) scale is analysed for a voice application.

Another work [7] presents the relationship between web response times and losses in the network. It discusses the difference in effect of losses on the response times due to the difference in the size of transfers. In [15], a model of TCP throughput based on packet loss and Round Trip Time (RTT) is presented.

In all of the above works QoE estimation is done either by objectively measuring the user activity on the network or by obtaining subjective responses from the users through experiments. To the best of our knowledge, there are no studies that compare subjective, user-centred and objective network-centred points of view. This paper builds a bridge between both user and network views by presenting both types of results together; the results inferred from the traffic analysis on a service provider’s network and the subjective responses of the users during experiments in a controlled environment. This comparison constitutes a first step to establish directions for further studies in this regard. Additionally, we present user session volume distributions and relationships between some of the renowned network performance indicators. The purpose is to provide basic understanding about them and to validate to which extent these new results support (or reject) the already established relationships.
3 Active measurements on experimental platform

This section discusses the results obtained by the measurements on the test-bed of BTH. These end-to-end measurements are performed in order to observe the quality perceived by the user. We will analyse these results in the following subsections.

3.1 Measurement platform and methodology

Experiments were performed on the test-bed at campus of Blekinge Institute of Technology. This test-bed is based on Distributed Passive Measurement Infrastructure (DPMI) [1]. As represented on Figure 1, this test-bed contains a server, a client, the Linux Traffic Controller (TC) shaper [9], two measurement points (M2 and M3), a Measurement Area Controller (MArC) and the Consumer station for data collection as shown in Figure 1. The traffic shaper is located between the server and the client. One measurement point (M2) is located between the client and the traffic shaper and another measurement point (M3) is located between the server and the traffic shaper. The traffic shaper can control parameters like loss, delay and bandwidth between server and client. We limit our experiments here only to the loss. On DPMI this packet loss is generated by Netem [17] with the default loss model, applying a uniform distribution [15]. Traffic traces from both directions can be captured at the measurement points M2 and M3. This information can be filtered and analyzed later by the consumer, see Figure 1. This information consists of timestamps, payload and sender/receiver IP addresses of each packet. On one side, the average network-level throughput and the download times on the link level can be deduced from this information. On the other side, the average throughput and the download times on application level are measured with a modified Fasterfox [4] utility of the Firefox web browser that logs accessed web pages and their download times. The interest of considering both network-level and application-level is that the first one depends more on the characteristics of the network path, while the second is closer to the observations of the user. In addition to this, users are
asked to provide their subjective responses about the service on the extended MOS scale from 5 to 0 [12] with the grades 5 = excellent, 4 = good, 3 = fair, 2 = poor, 1 = bad, and 0 when the user is tired of waiting and breaks the session. A link of 10 Mbps is used between the server and the client.

Experiments were performed downloading a webpage of size $X = 1.13$ MB containing an image. The packets were sent taking advantage of the Maximum Transmission Unit (MTU) of 1500 B on IP level. The user on the client computer opens that webpage and then rates his/her surfing QoE. While downloading – which always happens from the server due to disabled caching in the client – losses with nominal intensity $L$ are introduced through a traffic shaper in the direction from server to client. Successive packet loss intensities of 0%, 2%, 4%, 8% and 10% are used. A given user performs ten consecutive downloads of the same page per loss level. Loss is introduced in the ascending order of its magnitude. It thus increases the download time $T$ and correspondingly reduces the applicative throughput $R' = X/T$. Download times and thus even perceived throughputs are prominent performance parameters from the view-

Fig. 1: Test-bed setup
point of the user [18] and are amongst others used for performance-optimised selection amongst several available networks [5]. Given this background, we will concentrate on measuring user-perceived download times $T$ and derive applicative throughput values $R'$ from these. Different relationships between QoS parameters such as $L$, $T$, $R'$ and user-perceived QoE will be analysed in terms of different regressions (linear, logarithmic, exponential and power), whose validities will be evaluated through the coefficient of correlation:

$$r = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \sqrt{n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}}$$

(1)

where $x_i$ and $y_i$ are the corresponding values on the x-axis and y-axis and $n$ is the total number of $x$ and $y$ samples, respectively.

Furthermore, the timing and size information of packets in both directions are captured on both the measurement points and stored on consumer for later analysis. In the sequel, we will focus on the throughput of one flow based on one single transfer obtained from the download of one page at different $L$ values. This is done in order to compare with the results of the passive measurements described in the next section.

### 3.2 Impact of packet loss on download time and on throughput

In this subsection, we present results obtained by end-to-end measurements that show the impact of shaper-induced loss with nominal intensity $L$ (given in %) on the download time $T$ (given in s) and the applicative throughput $R'$ (given in Mbps). For each loss level $L$, ten experiments were performed. In order to illustrate the variations of the results, the averages are accompanied by two curves, upper and lower, at the distance of the standard deviation.

Figure 2 shows the dependency of the download time on the nominal loss induced by the shaper. Download times increase with the loss ratios, which is quite understandable as TCP slows down the transmission due to the loss [15]. The download time is a non-linear convex function of the nominal loss.
The higher the loss is, the larger the growth in download time. As the loss ratio grows, the variations in the download times grow as well indicating the disturbances on the network.

![Graph showing download time as a function of nominal loss.]

**Fig. 2:** Download time (average ± standard deviation) as a function of nominal loss.

We find the following regressions for the relationship between the nominal loss and the download time shown in Table 1. The exponential fit matches best with a correlation of 99.7%, followed by linear and power regressions that also yield good correlation values. The power relationship is almost linear.

Figure 3 shows the plots between calculated applicative throughput and nominal loss. There is a significant degradation in the throughput for the loss ratio between 2 % to 4 %. The overall trend that the applicative throughput
Table 1: Regressions on download time $T$ (given in s) vs. nominal loss $L$ (given in %), rounded at three decimals and with the best fit in bold.

<table>
<thead>
<tr>
<th>Regression</th>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>0.981</td>
<td>$T = 1.4L - 0.91$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>0.895</td>
<td>$T = 5.0 \ln(L) - 0.60$</td>
</tr>
<tr>
<td>Exponential</td>
<td>0.997</td>
<td>$T = 1.1 \exp(0.26L)$</td>
</tr>
<tr>
<td>Power</td>
<td>0.969</td>
<td>$T = 1.1L^{1.0}$</td>
</tr>
</tbody>
</table>

Fig. 3: Applicative throughput (average ± standard deviation) as a function of nominal loss.

$R'$ decreases when the loss rate $L$ increases is obvious.

We find the following regressions for this relationship between $L$ and $R'$.
as shown in Table 3. The exponential curve again is the best fit for the $L$-$R'$ relationship. It resembles the best-fitted regression of Table 3, which is not surprising due to the way $R'$ is calculated. Again, the power relationship almost reduces to a $1/L$ relationship, which is clearly different from the earlier postulated $1/\sqrt{L}$ relationship and its versions [15].

![Table 2: Regressions on applicative throughput $R'$ (given in Mbps) vs. nominal loss $L$ (given in %), rounded at three decimals and with the best fit in bold.](image)

### 3.3 Relation between QoE and QoS parameters

This section discusses the relationship between QoE, captured by Opinion Scores (OS) summary statistics, and above discussed QoS parameters like loss, throughput and download times. This section presents results about how subjective grading of the users varies with varying QoS parameters.

Figure 4 shows the results between QoE and the loss ratio $L$. This is an average grading by users for ten downloads per $L$ level. The user grading decreases continuously with increasing losses on the network. This shows that user experience can be correctly predicted by looking at the estimated loss level in the network. The average OS is very good for 0 % and approaches poor while the $L$ increases above 4 %. There is no variation in the Opinion Score at 0 % of $L$ showing the consistency in grading at perfect conditions. Variations in the Opinion Score are more or less constant for $L$ between 2 % to 10 %.
According to Table 3, the linear relationship fits best between QoE and $L$ with a correlation of $-99.7\%$. This finding supports [11] where it is also postulated as a linear relationship, however with a different factor in front of $L$ ($-0.31$ instead of $-0.37$). Hence we can say the user experience decreases linearly with increasing loss ratios.

Figure 4: Quality of Experience (average ± standard deviation) as a function of nominal loss.

Figure 5 shows the plot between QoE and the download time $T$. For each value of the Opinion Score, all the corresponding download times are averaged. The trend is obvious that the Opinion Scores decrease as the download times increase. The combination of file size and link speed prevents download times to drop below 1 second, and we do not observe the Opinion Score “excellent” (grade
Table 3: Regressions on Quality of Experience QoE (given through average Opinion Scores) vs. nominal loss $L$ (given in %), rounded at three decimals and with the best fit in bold.

<table>
<thead>
<tr>
<th></th>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$-0.997$</td>
<td>$\text{QoE} = -0.31L + 4.3$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>$-0.942$</td>
<td>$\text{QoE} = -1.4 \ln(L) + 4.3$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$-0.969$</td>
<td>$\text{QoE} = 5.5 \exp(-0.2L)$</td>
</tr>
<tr>
<td>Power</td>
<td>$-0.877$</td>
<td>$\text{QoE} = 5.2 , L^{-0.72}$</td>
</tr>
</tbody>
</table>

Fig. 5: Quality of Experience as a function of download time (average ± standard deviation).

We observe a poor Opinion Score for download times between 5 seconds and 8 seconds. Then users break their sessions for download times larger than...
15 seconds.

Table 4: Regressions on Quality of Experience QoE (given through average Opinion Scores) vs. download time $T$ (given in s), excluding null Opinion Scores.

<table>
<thead>
<tr>
<th>Regression</th>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$-0.983$</td>
<td>$\text{QoE} = -0.318 , T + 4.158$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>$-0.994$</td>
<td>$\text{QoE} = -1.426 , \ln(L) + 4.469$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$-0.995$</td>
<td>$\text{QoE} = 4.836 , \exp(-0.150 , T)$</td>
</tr>
<tr>
<td>Power</td>
<td>$-0.955$</td>
<td>$\text{QoE} = 5.339 , L^{-0.638}$</td>
</tr>
</tbody>
</table>

According to Table 4, exponential fitting works best, followed by logarithmic fitting as supported by ITU-T Rec. G.1030 g1030, both with a very good correlation.

Figure 6 shows the QoE as a function of the applicative throughput $R'$. Again, we compute the average and the standard deviation of all the throughputs which received the same grade. The Opinion Score is very good for $R'$ above 6 Mbps while it is bad below 1 Mbps. The Opinion Score is null for throughputs around 0.5 Mbps showing that the user is no more interested in continuing HTTP transfer.

Table 5 shows that the logarithmic regression fits best the QoE-$R'$ relationship. The factor in front of the logarithm of $R'$ resembles the one seen from the download times (cf. Table 3). The higher the throughput, the better the Opinion Score given by the user.

For the sake of comparison, we now also show QoE as a function of network-level throughput $R$ (given in Mbps). This analysis is done for an arbitrarily selected single flow that leads to the Opinion Score from 0 to 4.
3.3 Relation between QoE and QoS parameters

Fig. 6: Quality of Experience as a function of applicative throughput (average ± standard deviation).

Table 5: Regressions on Quality of Experience QoE (given through Opinion Scores) vs. the applicative throughput $R'$ (given in Mbps), excluding null Opinion Scores.

<table>
<thead>
<tr>
<th>Regression</th>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>0.955</td>
<td>QoE = 0.44 $R'$ + 1.0</td>
</tr>
<tr>
<td>Logarithmic</td>
<td><strong>0.995</strong></td>
<td>QoE = 1.5 ln($R'$) + 1.153</td>
</tr>
<tr>
<td>Exponential</td>
<td>0.878</td>
<td>QoE = 1.175 exp(0.188 $R'$)</td>
</tr>
<tr>
<td>Power</td>
<td>0.960</td>
<td>QoE = 1.208 $R^{0.651}$</td>
</tr>
</tbody>
</table>

Figure 7 illustrates the results obtained for relationship between $R$ and QoE. We see almost similar trend between $R$ and QoE as we observed in Figure 6.
Hence it validates the results we obtained for applicative throughput.

![Graph](image)

**Fig. 7:** Quality of Experience as a function of network-level throughput for a single flow.

Table 6 lists some regressions between the OS QoE-$R$ relationship. It shows that the logarithmic regression fits best once again. Comparing with Table 6, we see similar regressions in both cases.

## 4 Passive measurements on real-users network

This section discusses the results obtained from traffic collected on the France Telecom network. In this section, we first analyse the overall traffic and then we correlate the user behaviour (through the characteristics of his traffic) to
Table 6: Regressions on Quality of Experience (given through Opinion Scores) vs. the network-level throughput $R$ (given in Mbps).

<table>
<thead>
<tr>
<th></th>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>0.956</td>
<td>QoE = 0.29 $R + 0.76$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>0.979</td>
<td>QoE = 1.2 $\ln(R) + 1.3$</td>
</tr>
<tr>
<td>Exponential</td>
<td>0.562</td>
<td>QoE = 0.048 $\exp(0.38 R)$</td>
</tr>
<tr>
<td>Power</td>
<td>0.855</td>
<td>QoE = 0.042 $R^{2.3}$</td>
</tr>
</tbody>
</table>

the performance metrics. Our aim is to extract the user perception from the detailed traffic analysis of his behaviour.

We first describe in subsection 4.1 the network where measurements are collected. In subsection 4.2, we present the relation of session volumes with performance criteria on sessions such as the mean throughput and the loss ratio.

4.1 Measurement platform and methodology

This subsection describes the setup in the ADSL backhaul network of France Telecom and how the traffic is captured from the network. Our collection infrastructure is shown in Figure 8. Traffic traces are collected on the ADSL access network on a BAS (Broadband Access Server) that collects the traffic coming from many DSLAM (Digital Subscriber Line Access Multiplexer). Each BAS multiplexes the traffic of 10 DSLAMs connecting 4000 residential and small enterprises clients in total. The probe is located between the BAS and the first router of the backbone network. The TCP/IP headers of the whole HTTP traffic are captured without any other sampling. These TCP/IP headers are then used to compute many traffic metrics for each flow (size in packets, volume in bytes...) and performance criteria (throughput, loss ratio...). The traffic of all the flows between the same source and the same destination (IP addresses) is then aggregated in sessions, as long as the silence time between two consecutive flows is less than a given threshold otherwise a new session begins for this cou-
example of IP addresses. We will analyze in the next subsection the influence of this threshold on the sizes of session.

4.2 Correlation of traffic characteristics with performance metrics

We discuss in this subsection the correlation of traffic characteristics with performance metrics. Our objective is to detect some correlations between the user behaviour and the network performance, even if traffic characteristics are also influenced by the protocols as observed in [3].

The network performance metrics we consider in this subsection are the packet loss and the throughput. The loss ratio concerns more the network operator, as it is an indication of the congestion state in its network or in peering networks. A user does not really perceive the loss ratio, only its consequences such as longer response times, or lower throughputs as observed in section 2. On the contrary, the user is more concerned by the throughput of his transfers, which conditions the time he need to get large files, and that he can compare with the capacity of his access link. The network operator is less concerned by the throughput of individual flows. These depend indeed most of the time on external factors, like the output of web servers, the number of and distances between hops in-between server and client, the user access link, etc. The network

Fig. 8: Collection infrastructure
operator is only responsible of bad throughputs in case of congestion, which may as well be detected through the loss ratio. So we first consider the correlations of traffic characteristics with the loss ratio, and then with the mean throughput of sessions.

We roughly approximate the loss ratio by the proportion of out-of-sequence packets on the network. We have seen in [3] that there are many methods to measure the loss ratio more precisely. These different approximations of the loss give similar correlations with the traffic characteristics. So we choose out-of-sequence packets as an example in the rest of this paper. A packet of a TCP connection is an out-of-sequence packet if its sequence number is below the sequence number of the last transmitted packet on this TCP connection. Even if it appears to be quite rough, this estimation of the loss ratio has the advantage to be very fast, so it can be computed in real-time for packet trace inspection on high-speed links.

Figure 9 and Figure 10 present the session sizes for downloads and for uploads in relation to the ratio of out-of-sequence packets. The different curves show average session volumes for different aggregation thresholds. As observed for flows in [3] we notice in Figure 9 for downloads a continuous decrease in the average session sizes with increasing out-of-sequence packets. This decrease is faster for ratios larger than $10^{-3}$. The power regression fits very well these curves as shown in Table 7. All the curves for the different aggregation thresholds are rather close except the curve associated to the largest threshold (1024 seconds) which shows bigger session sizes for an approximated loss ratio above 10 %. As these thresholds are larger than the usual timers of protocols, this deviation could be explained by the behaviour of users that renew a connection ten minutes later when the quality is too bad.

The curves for the upload transfers are very different. The average session volumes are quite indifferent to the out-of-sequence ratio when the latter is larger than $4 \cdot 10^{-3}$. We considered here all the flows using the TCP port 80. Most of these flows are HTTP as 80 is well-known port of this application. However So
Fig. 9: Ratio of out-of-sequence packets vs. session volumes downloaded for different silent time thresholds

Table 7: Regressions of session volumes downloaded ($V$) vs. ratio of out-of-sequence packets ($L$) in case of a 64 s silent time threshold.

<table>
<thead>
<tr>
<th>Regression</th>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$-0.313$</td>
<td>$V = -54703 \ L + 21433$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>$-0.813$</td>
<td>$V = -6287 \ ln(L) - 21897$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$-0.679$</td>
<td>$V = 8750 \ exp(-9.5L)$</td>
</tr>
<tr>
<td>Power</td>
<td>$-0.996$</td>
<td>$V = 98 \ L^{-0.62}$</td>
</tr>
</tbody>
</table>

this port may be used by other applications than HTTP, with perhaps different characteristics and different performance requirements. Moreover, the user may be probably less impatient and less worried by bad quality with uploads as long as he is not waiting for an answer. However, smaller out-of-sequence ratios than
Fig. 10: Ratio of out-of-sequence packets vs. session volumes uploaded at different hours of the day

$4 \cdot 10^{-3}$ yield a growth in the average session volumes independently of the silent time threshold.

Table 8: Regressions of session volumes uploaded ($V$) vs. ratio of out-of-sequence packets ($L$) in case of a 64 s silent time threshold.

<table>
<thead>
<tr>
<th></th>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$-0.349$</td>
<td>$V = -29240 \ L + 5870$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>$-0.801$</td>
<td>$V = -2033 \ln(L) - 5589$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$-0.391$</td>
<td>$V = 2577 \ exp(-6.1L)$</td>
</tr>
<tr>
<td>Power</td>
<td>$-0.894$</td>
<td>$V = 239 \ L^{-0.42}$</td>
</tr>
</tbody>
</table>
From the above results, we can notice the change in the session sizes depending on the out-of-sequence ratio representing the loss ratio, in particular for download transfers. As out-of-sequence packets and losses are indications for the degradation in performance, we can clearly see the user session volumes decreasing with the corresponding degradation of quality of service.

Another important performance criterion for the users is the throughput of their transfers. Throughput measurements are always vital in analysing the network conditions. Increasing or decreasing throughputs strongly affects the behaviour of users on the network. The effect of throughput can also be realized by analysing the following plot shown in Figure 11 that presents the average throughput in Mbps on the x-axis and average volumes of the sessions in packets downloaded on the y-axis. When the throughput increases, the average session size also increases. This increase shows the increased utility of the network for the increasing throughput.

In the sequel, we give the regressions for the correlations between throughput and the average volumes of download transfers on Table 9.

<table>
<thead>
<tr>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Linear</strong></td>
<td>$0.928$</td>
</tr>
<tr>
<td></td>
<td>$V = 1202 \cdot R + 208$</td>
</tr>
<tr>
<td><strong>Logarithmic</strong></td>
<td>$0.804$</td>
</tr>
<tr>
<td></td>
<td>$V = 1128 \cdot \ln(R) + 3519$</td>
</tr>
<tr>
<td><strong>Exponential</strong></td>
<td>$0.808$</td>
</tr>
<tr>
<td></td>
<td>$V = 77 \cdot \exp(0.78R)$</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td>$0.972$</td>
</tr>
<tr>
<td></td>
<td>$V = 976 \cdot R^{1.02}$</td>
</tr>
</tbody>
</table>

The power regression appears as the best approximation; however, the power as such is pretty close to one, which points at an almost proportional relationship.
ship. Considering $W = V/R$ as total average waiting time spent by the user per session, we find $W \sim R^{0.02}$, which means that the total waiting time hardly depends on the throughput.

5 Combination of user and network view

In this section, we compare the complementary results obtained from the user view, presented in Section 3, and from the network view, presented in Section 4. The explicit user view and grading are represented by regressions between user-perceived QoE in terms of opinion scores, nominal loss ratios, download times, and applicative and network-level throughput, respectively. The network view provides in particular regressions between average session volumes and approximations of the loss ratio as well as network-level throughput, which implicitly represent the user’s activities and grading. Combining these results
provide us with ideas on which impacts network performance and QoE have on the user’s activities.

5.1 Comparison by throughput

We have already presented in Sections 3.3 and 4.2 the results regarding the effect of network-level throughput on user grading and usage. In this subsection, we relate the session volumes to the subjective grading of the user by their individual relationships with network-level throughput. To this end, we recall the best-fitted equations for the QoE-\( R \) relationship from Table 6 and for the \( V-R \) relationship from Table 9, respectively:

\[
QoE = 1.2 \ln(R) + 1.3 \tag{2}
\]

\[
V = 976 R^{1.02} \tag{3}
\]

Session volumes obviously rise stronger (almost linearly) with rising throughput than the QoE (in a logarithmic way). This is illustrated in Figure 12, which plots Equation 2 and the normalized volume \( V(R)/V(1 \text{ Mbps}) \) according to Equation 3, for the purpose of a qualitative comparison between both trends.

Obviously, for small throughput values, both trends are similar. However, as the throughput rises, the growth in volume accelerates as compared to the growth in QoE. Indeed, a combination of Equation 2 and Equation 3 – if it were possible – would yield an exponential relationship \( V \sim \exp(QoE) \). From this, we can deduce that users that perceive a good QoE (which is enabled through high throughput) tend towards much more voluminous sessions, i.e. consume much more pages than users that perceive worse QoE.

5.2 Comparison by loss

We now compare session volumes in the downlink direction as function of the loss ratio, approximated by the out-of-sequence ratio, on one hand and subjective
user gradings, given by Opinion Scores, as functions of the loss ratio on the other hand. The corresponding best-fitted equations for the QoE-L relationship from Table 3 and for the \( V - L \) relationship from Table 7 read:

\[
QoE = 0.31 \, L + 4.1 \quad (4)
\]

\[
V = 98 \, L^{-0.62} \quad (5)
\]

Figure 13 plots Equation 4 and the normalized volume/\( V(10\%) \) according to Equation 5. While the loss ratio sinks to 1%, the QoE grows significantly and approaches the – for these experiments – optimal opinion score of 4. This approach continues asymptotically while loss ratios tend towards zero. The session volume, on the other hand, keeps rising as the out-of-sequence ratio
user gradings, given by Opinion Scores, as functions of the loss ratio on the other hand. The corresponding best-fitted equations for the QoE-L relationship from Table 3 and for the $V - L$ relationship from Table 7 read:

$$\text{QoE} = 0.31 \, L + 4.1 \quad (4)$$

$$V = 98 \, L^{-0.62} \quad (5)$$

Figure 13 plots Equation 4 and the normalized volume/$V(10 \%)$ according to Equation 5. While the loss ratio sinks to 1%, the QoE grows significantly and approaches the – for these experiments – optimal opinion score of 4. This approach continues asymptotically while loss ratios tend towards zero. The session volume, on the other hand, keeps rising as the out-of-sequence ratio

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig12.png}
\caption{QoE (given in Opinion Scores) and normalised session volume as functions of network-level throughput.}
\end{figure}
Conclusions and future work

Motivated by the need to draw conclusions about user satisfaction from network measurements, this paper investigates possible correlations between user-perceived Quality of Experience (QoE) and network-level traffic characteristics. In particular, we analysed on one side the quantitative relationships between Quality of Experience, expressed in Opinion Scores, and Quality of Service pa-

Fig. 13: QoE (given in Opinion Scores) and normalised session volume as functions of the (approximated) loss ratio.
rameters such as loss ratio, download times and throughput, obtained from experiments from the end user perspective. We then investigated on the other side the correlations between traffic characteristics (session volumes) and performance criteria such as loss rates, throughputs and measured in an operational network. The qualitative comparison of QoE and session volumes via throughput and loss ratios indicates growing session volumes with improved QoE. In other, simple words, happy users surf more. However, the duration of the sessions of web surfing seems less dependent on the throughput and thus on the perceived QoE.

In terms of practical applicability of above discussed results, service providers can make use of the relationships between QoE and traffic characteristics such as session volumes, throughput and loss to automatically assess the utility functions for applications. This method can be cheaply used for new applications avoiding long and expensive experiments. It can be also regularly applied on operational networks to follow the evolutions of existing applications, of their traffic characteristics and of their performance requirements. Such an estimation of QoE could help the service providers to continuously monitor the user satisfaction level, react timely and appropriately to rectify the performance problems and hence provide the services according to the user expectations.

Regarding future work, and due to the results outlined above, this study provided the inspiration for in-depth studies of user patience in view of performance problems. In particular, we are interested in measuring and modelling users pertinence to the service as function of network-level problems, which is currently done within the Special Joint Research Project “QoEWeb” within the European Network of Excellence Euro-NF (Networks of the Future).

References


[2] D. Collange, J. L. Costeux, *Correlation of Packet Losses With Some Traf-


PAPER II

BACK TO NORMAL? IMPACT OF TEMPORALLY INCREASING NETWORK DISTURBANCES ON QoE

Published in the proceedings of
IEEE Workshop on Quality of Experience for Multimedia Communication (QoEMC)
Atlanta, USA, December 2013
Back to Normal? Impact of Temporally Increasing Network Disturbances on QoE

Junaid Shaikh, Markus Fiedler, Pangkaj Paul, Sebastian Egger†, Frederic Guyard‡

Blekinge Institute of Technology, Karlskrona, Sweden, (junaid.junaid, markus.fiedler)@bth.se
papb11@student.bth.se

†Telecom Research Center
FTW, Vienna, Austria
egger@ftw.at

‡R&D Orange Labs
Sophia Antipolis, France
Email: frederic.guyard@orange.com

Abstract – Brief episodes of network faults and performance issues adversely affect the user Quality of Experience (QoE). Besides damaging the current opinions of users, these events may also shape user’s future perception of the service. Therefore, it is important to quantify the impact of such events on QoE over time. In this paper, we present our findings on the temporal aspects of user feedback to disturbances on networks. These findings are based on subjective user tests performed in the context of web browsing on an e-commerce website. The results of this study suggest that the QoE drops significantly every time the page load time grows. The after-effects of network disturbances on user QoE remain visible even when the network problems
are over, i.e., users do not immediately return to the same level of opinion scores as compared to the corresponding pre-disturbance phase. They tend to remember their recent experiences. Our results also show that there are four segments of users that exist with regards to their feedback to page load times. Network operators may customize their services according to each segment of users to raise the overall QoE. Finally, we show that the exponential relationship provides best fits of QoE and page load times for all segments of users.

1 Introduction

Increasing reliance of a wide spectrum of daily life activities on the Internet put stringent requirements on today’s data networks. They need not only be available and accessible around the clock but also capable enough in delivering distinct quality. However, unfortunately, the networks still are not vigilant enough to meet these demands. Particularly, the fast emerging mobile broadband networks are prone to failures and transient outages, mainly due to resource allocation, mobility and configuration issues, which debase Quality of Experience (QoE).

Generally network downtimes over large time windows (minutes to hours scales) are noticed and resolved well by service providers. However, the issues related to transient outages often remain unnoticed [1] [2]. Eventually, the ON and OFF phases – giving rise to delays and waiting times for users – without any perceptible follow-up by service providers, often result in dissatisfied users. Thus, there are studies needed to understand the accumulating impact of recurring short-term service disruptions (on the scale of seconds) on temporal QoE.

Previously, a number of studies has been done to quantify the impact of delays on QoE [3]– [5]. Similarly, relationships between QoE and Quality of Service (QoS) are also derived [6]– [9]. Yet, there is a lack of studies, which quantify the accumulating impact of service disruptions on QoE over time. In another paper [10], authors have presented results with regards to the time dynamics of QoE. Their investigation showed the role of user memory in user perception of waiting times on the Web. However, there are further studies needed in this regard to portray the escalating effect of short-term outages on user QoE.

In this paper, we have tried to illustrate the piling-up effect of bad memories from increasing network OFF times on the user QoE. Our study is based on task-based user
subjective tests done in the context of web browsing on an e-commerce website. This study attempts to answer the possible underlying psychological factors, which motivate users to adopt a certain opinion in case of delays. One of the important aspects related to this is the assessment of user’s tendency to return to the pre-disturbance level of opinion about service after network problems are rectified. Using clustering, we also show different segments of users in terms of their memory and response to the delays on web pages. The knowledge about such aspects can be instrumental for service providers in customizing their services according to the user types and, hence, in devising better strategies for retaining their customers.

The structure of this paper is as follows. Section 2 presents the methodology of user tests. Section 3 provides details about the experiment setup used in this study to conduct user tests. Section 4 presents results on the users’ responses to Page Load Times (PLTs) and the possible reasoning behind the obtained results. Finally, Section 5 poses a set of conclusions and an outlook of the future work.

2 Methodology

In this study, a total of 43 subjects participated in subjective tests. The mean age of participants was 24.5 years, the maximum age was 32 years and minimum age was 21 years. All the subjects were everyday users of the web browsing service and use e-commerce websites regularly for online shopping. Before starting the test, a 5-minute training session was conducted for each of the participants, in which necessary instructions were provided about the test procedure.

Each subject in this study performed task-based web browsing for an e-commerce website. The task was based on the selection and purchase of a laptop computer. Each subject went through 12 shopping sessions. Each shopping session was based on web browsing of three web pages: Product selection page, product details & purchase page and payment confirmation page. The first page (product selection) consisted of 21 objects (2 CSS, 2 JS, 17 JPEG and PNG images). The second page (product details & purchase) consisted of 3 objects (all images) and some text providing the specification of selected product. The third webpage of shopping session consisted of 2 objects (all images) and some text acknowledging the purchase.

Particular packets carrying web page content on the network were targeted and delayed in order to increase Page Load Times (PLTs). These packet-based delays
introduced OFF times resembling outages on real network [1] [2]. Figure 1 illustrates the PLTs over time faced by each subject. The x-axis of the plot in Figure 1 represents web page number and the y-axis represents the PLTs. Each shopping session (based on three web pages) is illustrated by S1 to S12 on the secondary x-axis. As illustrated in Figure 1, each user went through a variety of PLTs, from less than 1 s to around 20 s. Delays were introduced in both increasing as well as decreasing order to understand the transition in opinion scores of subjects. For instance, the PLT was increased at page 4 and then decreased at page 5 to assess how subjects react to them.

At every page, each subject was asked the following two questions:

1. Which web page is this?

The following three options were given to answer the above question:

- Product Selection
- Product Details & Purchase
• Purchase Confirmation

This question to test whether subjects are attentive to their task.

2. How do you feel about its loading time?
The answer of this question was provided in the form of the five-level ACR scale for rating quality, recommended by ITU-T.

3 Experimental setup

A client-server model was implemented to conduct experiments. When a subject requested for a web page from client machine, the server received the request and
subsequently, responded with content of the requested web page. These requests and responses were transferred via a network emulator called KauNet that allows for having impact on specific packets, thus controlling the PLTs in well-defined ways [11].

The server was installed with the Ubuntu 10.10 operating system. It was configured with Apache 2.2 to act as a web server. Apache is currently the most popular web server [12]. Web pages deployed on the web server were developed using CodeIgniter (a PHP framework) [13]. Moreover, the widely used free open-source software for Linux systems, Bind9 was installed to setup the DNS server in order to translate user-requested URL to web server IP address.

Similarly, the client machine consisted of Windows 7 operating system. The Google Chrome web browser was installed and used for web browsing on the client side. Google Chrome was chosen because currently it is far more popular than any other web browser [15]. Moreover, an open-source web debugging proxy tool called Fiddler [16] was also deployed on the client side in order to collect the HTTP (S) logs. These logs were collected in Java Script Object Notation (JSON) format and stored in HTTP ARchive (HAR) files.

In order to collect and store the network level traffic, the Distributed Passive Measurement Infrastructure (DPMI) was deployed [14]. As shown in the Figure 2, the DPMI consisted of Measurement Area Controller (MArC), Measurement Points (MPs) and Consumer machines to control, capture and store the network traffic, respectively. The MPs were equipped with two Endace Data Acquisition and Generation (DAG) 3.5E cards to capture network traffic near the server and the client sides in both directions. The choice of DPMI is motivated by the fact that it enables high accuracy measurements (up to nanosecond level) with a distributed architecture to collect packets at multiple points within a network.

A signaling script was developed and placed on the client machine in order to make the test procedure automatic. Based on the design of experiment procedure (Session ID and URL), this script signaled the desired network settings to the network emulator. Similarly, it signaled other machines on the network to collect required logs in appropriate files based on the User ID, Session ID and the URL. Furthermore, it collected the answers to the questions, mentioned in the previous section and stored them in a local data base along with web page URLs, network settings and User IDs. The automatic setup has proved helpful in preventing the interruptions for subjects during the tests that would have otherwise occurred because of manually changing network settings, creating log files and collecting opinion scores from subjects.
4 Results and Analysis

4.1 QoE over time

Each subject browsed through a total of 36 web pages and provided a rating on the MOS scale ranging from 5 (= excellent) to 1 (=bad) after a web page was completely loaded. A total of 44 ratings and PLTs per web page were obtained from the 44 participants. Subsequently, the means of opinion scores (\(MOS_j\)) and PLTs (\(\bar{t}_j\)) at web page \(j\) can be expressed as:

\[
MOS_j = \frac{1}{n_j} \sum_{i=1}^{n_j} OS_{i,j} \tag{1}
\]

\[
\bar{t}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} t_{i,j} \tag{2}
\]
Where $OS_{i,j}$ represents opinion of user $i$ at web page $j$ and $n_j$ represents total number of opinion scores received for web page $j$.

![Diagram](image)

**Fig. 4:** Opinion scores for undisturbed web page transfers (Left: Rating share for page 1 [Mean PLT: 0.7 s], Right: Rating share for page 36 [Mean PLT: 0.2 s])

As shown in Figure 3, generally user QoE plunges as soon as PLTs ascend. For example, in the first session S1, PLTs are below 1 s while MOS is above 4.5. However, when PLT increases to 3 s at page 1 of session S2, MOS drops to 4.1, immediately. This indicates how subjects notice increasing delays of various intensities across all their shopping sessions. In contrast to sharp fall of QoE in the case of increasing PLT, the QoE does not increase as sharply in the case of decreasing PLT. Although, subjects usually express their contentment in the form of higher ratings as soon as waiting times descend, they still abstain from resorting to same ratings as those observed during the pre-disturbance period. This shows that the memory or recency effect prevails among the subjects, this phenomenon becomes quite evident by noticing the values of MOS for web pages 5–6, and 17–18. Obviously, the MOS values do not return back to the same level as appears before encountering additional delays on page 4 and page 16, respectively. Moreover, the MOS values are 4.6 and 3.6 on pages 1 and 36, respectively, showing a significant loss in QoE over time, despite of similar PLTs (less than 1 s). This is further illustrated in Figure 4. The ratio of subjects giving opinion score of 5
(Excellent) decreased significantly for page 36 (coming back from disturbances). At page 1, the average PLT was around 0.7 s and the share of rating 5 was more than 75% while, at page 36 the average PLT was about 0.2 s and the share of rating 5 decreased to less than 20%. This significant drop in rating level shows the impact of accumulation of waiting time effects in the users’ working memory, manifested in the recent past. Similarly, in paper [10], Hossfeld et al also showed the impact of memory effect on user QoE.

Additionally, we also observe that the QoE recovers significantly when no network disturbances occur during the whole task (shopping session). This is depicted by MOS for pages 13–15 (session S5) and pages 25–27 (session S9). The MOS of these sessions are approximately similar to the MOS of session S1. In contrast, if users face network disturbances on one of the pages during a shopping session (task), their ratings for subsequent pages of that respective session remain significantly low. This can be witnessed, for example, by observing the MOS levels of S2, S6 and S10.

In order to further strengthen our understanding of the observed decay in MOS and underlying memory effect among users, we computed standard deviation of opinions scores for web pages, where no network disturbances were applied. The PLTs for these web pages were kept well below 1 s. The objective was to determine whether the standard deviation varies significantly over time throughout the course of the experiment. Let $\sigma_j$ represents standard deviation of opinion scores at page $j$, which is expressed by:

$$\sigma_j = \sqrt{\frac{1}{n_j} \sum_{i=1}^{n_j} (OS_{i,j} - MOS_j)^2}$$  \hspace{1cm} (3)

Where $j$ represents only those web page transfers, which are not disturbed by network emulator, yielding page load times were below 1 s.

Figure 5 shows the standard deviation of opinion scores for web pages with undisturbed network settings (PLTs below 1 s). The difference of opinion scores among subjects gradually increase as illustrated by the increasing standard deviation. This indicates that the memory effect is subject to change among individuals. Some users tend to resist more than others in reinstating to their perception about service quality. It was probably the reason why there was increasing standard deviation.
4.2 Segmentation of users

As shown, increasing standard deviation expresses the existence of dispersion in opinions about the service quality. Therefore, it is imperative to segment the subjects into different categories before interpreting their responses. Before segmentation, we performed linear regression between PLTs and opinion scores for each subject. The reason for performing linear regression is to determine how each subject adopts her opinion score with change in PLT.

Let $QoE_i$ and $t_i$ denote the opinion scores and PLTs received from the user $i$, respectively. Similarly, let $\alpha_i$ and $\beta_i$ be the intercept and slope of the equation for user $i$, respectively. Then after applying linear regression on opinion scores and PLTs of user $i$, we got the following equation:

$$QoE_i = \alpha_i + \beta_i \cdot t_i$$

(4)
Hence, we extracted 44 pairs of $\alpha$ and $\beta$, each pair representing each of the 44 subjects. In order to divide subjects into segments, we performed clustering by applying the popular k-means clustering algorithm [17] on the values of $\alpha$ and $\beta$. Before applying k-means clustering on the set of $\alpha$ and $\beta$, optimal number of clusters $k$ needs to be determined. We performed the following steps to determine the number of clusters:

1. Set $k=2$.
2. Apply k-means clustering with input $k$. Extract cluster centroids ($\mu$) for each of the clusters.
3. Compute sum of squares of differences between data points in a cluster and their respective $\mu$. Let $D_j$ be the sum of squares of differences within cluster $j$. Let the centroid of cluster $j$ be represented by $\mu_j$. Moreover, let $x_{ij}$ be a data point $i$ in cluster $j$, and $m_j$ represents the total number of data points in a cluster $j$. Then, $D_j$ can be expressed by:

$$D_j = \sum_{i=1}^{m_j} (x_{ij} - \mu_j)^2$$

(5)

4. Add values of all $D_j$ and compute the total sum of squares of differences ($D$) for all clusters $k$, as follows:

$$D = \sum_{j=1}^{k} D_j$$

(6)

5. Plot values of $D$ against corresponding values of $k$.
6. Repeat steps 2 to 4 after incrementing $k$ by 1, for an arbitrary number of times $n$ until elbow can be seen in the plot. We performed these steps by setting values of $k$ to 2,3,4,5 and 6.

The plot of $D$ versus $k$ is shown in Figure 6. The elbow can be observed for $k = 4$. The reduction in $D$ becomes marginal once $k$ goes above 4. Hence, we set $k = 4$ in our study and applied k-means clustering. We obtained 5 users in cluster one, 13 users in cluster two, 19 users in cluster 3 and 6 users in cluster 4. The values of $\alpha$ and $\beta$ for users within each cluster are depicted in Figure 7.

Figure 8 presents MOS per web page per cluster of users. From the plot, it becomes evident that the MOS of each cluster is following almost a similar pattern. As soon as PLTs increase, their respective MOS decrease steeply. However, when PLTs decrease,
all clusters show a sign of memory effect and therefore, their respective MOS grows grudgingly.

Generally, subjects in cluster one appear to be the most tolerant as compared to subjects in any other cluster. Their MOS at any stage of the experiment does not go below the opinion score of 2. This infers that in comparison, this type of users is more optimistic about service quality and hence, can be a source of positive words of mouth for the service provider. On contrary, subjects in cluster four show the most negative response to delays. Their opinion scores go already below 3 as PLTs approach 3 s. After being exposed to network disturbances, these subjects hardly return to their initial level of satisfaction. Retention of such users can be challenging for service providers. Cluster two and three show a rather moderate and stable behavior. These two clusters accumulate to form the biggest segment of subjects that participated in our experiments. The stability in the opinion scores of these subjects show a strong indication of the memory effect. They tend to stay firm about their opinions despite of variations in the PLTs. Nevertheless, a sharp contrast is evident between cluster
one and cluster four.

**Self-herding behavior:** As evident from Figure 8, the MOS of each cluster across the sessions S2 to S12 remains below their corresponding MOS of session S1. It shows that the users usually stick to the decisions taken by them in the recent past and therefore, do not rate the service quality above the level of rating provided in the first session. This behavior can be explained by a terminology called “self-herding” [18]. Self-herding refers to the tendency of a person to follow consciously or subconsciously her own decisions taken in the past.

**Type A/B behavior pattern:** In the study [19], the authors classify humans into two broad categories based on their tolerance to delays: type A and type B. Type A personality users are rather impulsive, time urgent and aggressive as opposed to Type B that are patient, focused and easy-going. In our study, we observed some shades of Type A/B personality as a contrast of impulsion versus patience encompassing cluster one to four (i.e. cluster 1 ≃ rather A and cluster 2 ≃ rather B).
Finally, we tested multiple linear regressions on average PLTs and their respective MOS values for each of the clusters, separately. Exponential regression (cf. Equation 7) appeared to fit best on the data of each cluster with Pearson Correlation ($r$) values equal to $-0.91$, $-0.90$, $-0.88$ and $-0.80$ for cluster one, cluster two, cluster three and cluster four, respectively. Figure 9 presents plots of exponential best fit for each of the clusters, with their corresponding $\alpha$ and $\beta$ values.

$$QoE = \alpha \cdot e^{\beta \cdot t} \quad (7)$$

**Vierordt’s law:** It is evident from Figure 9 that when PLTs are below 2 s, any increase in PLTs result in faster decay in MOS. However, when PLTs are higher than 6 s, the decay in MOS becomes rather slow. This observation can be explained by Vierordt’s law [20]. According to the latter, users either overestimate or underestimate the duration of the delay. Usually, they tend to overestimate durations of less than 2 seconds, accurately estimate durations between 2 to 6 seconds, and underestimate durations when delays are above 6 seconds. This law was further confirmed by an
Fig. 9: Exponential fits of QoE and PLTs of clusters

experimental study [21]. Similarly, the exponential interdependency of QoE and QoS was presented with the term IQX hypothesis by Fiedler et al. [9], which stated that, if the QoE is very high, a slight disturbance will decrease QoE strongly. On the contrary, if the QoE is already low, slight disturbance will not reduce QoE much. In this regard, one can relate the IQX hypothesis to Vierordt’s law using the results presented in this study. The small PLTs (less than 2 s) imply high QoE, where any slight increase in PLTs are overestimated by users, bringing QoE down rather strongly.

5 Conclusion

In this paper, we presented a set of observations with regards to the impact of short network outages (OFF times) on QoE. The results were obtained by performing a lab study, which involved user subjective tests in the context of web browsing. There are several conclusions extracted from this study, which are discussed below.
First, the impact of recurring OFF times on QoE accumulate with time, which can be seen in the form of decaying trends in the user opinion scores. When the waiting times grow, user satisfaction level drops immediately. Subsequently, user opinions do not witness the sign of complete recovery after the network problems are rectified. This is an evidence of the role of memory effect and time, which need to be considered as important variables in the modeling of QoE. Second, we found the existence of four different user segments with regards to their feedback against varying page load times. While all these segments witness memory effects, some users are rather impulsive than others in changing their opinions. Hence, their opinion scores drop rather steeply with increase in page load times. Nevertheless, we witness an exponential relationship between QoE and page load times in all cases, regardless of the user segment. Third, on average, the opinion scores of each segment of users remain under the level of MOS given during their corresponding first shopping sessions before the occurrence of disturbances. Specifically, their scores do not seem to rise above their respective initial MOS levels.

This is ongoing work, which includes the assessment of the reaction of users to varying frequencies and durations of network outages. We are trying to model how users experience a typical (mobile) network service over time and the associated factors that can influence their feedback about that service. Moreover, we intend to assess the impact of disturbances within different key phases of a transfer from the network perspective. Another important aspect of our work is the extension of these user tests beyond the lab to the field settings.

Acknowledgment

This research is supported by Vinnova through the Celtic-plus project Quality of Experience Estimators in Networks (QuEEN).

References


REFERENCES


PAPER III

In Small Chunks or All at Once? User Preferences of Network Delays in Web Browsing Sessions

Published in the proceedings of
IEEE Workshop on QoE-centric Network and Application (QoENAM)
Sydney, Australia, June 2014
In Small Chunks or All at Once? User Preferences of Network Delays in Web Browsing Sessions

Nazrul Islam, Vijaya John David Elepe, Junaid Shaikh, Markus Fiedler

Blekinge Institute of Technology, Karlskrona, Sweden,
(nais11, viel12)@student.bth.se
(junaid.junaid, markus.fiedler)@bth.se

Abstract – The time-critical tasks on the Internet are increasing. The delays in these tasks can have severe implications on the Quality of Experience (QoE) of a service. Therefore, networks require smart user-centric resource management mechanisms to reduce the impact of these delays on QoE. For this, a better understanding of the user preferences with regards to service performance is a prerequisite. In this paper, we present user responses to the three different distributions of delays, occurring during shopping sessions on the Web. By keeping the overall waiting time of the sessions same, we show how the users respond differently to the different set of delays. We analyzed the user responses and found that, the users prefer small frequently occurring delays as compared to the long rarely occurring delays within a task-based session.

1 Introduction

People rely heavily on the wide domain of applications and services running on the Internet. Large number of these applications are mainly accessed via World Wide Web (WWW). Generally, the users expect a faster delivery of response for any request they make, without any disturbance. However, in the context of web browsing, disturbance
in the form of end-user waiting time is a common occurrence and also the key determinant factor of Quality of Experience (QoE) [1]. The waiting time is defined as the time between a client sending a request to a server and the response to that particular request is fully visible to the client.

Despite of the dramatic increase in network bandwidth over the years, networks are still not smart enough to serve these web objects immediately, according to the user expectations. Particularly, the cellular networks may even take multiple of seconds to fetch small web objects from the web servers. Several reasons describe these delays.

First, the cellular channel quality varies significantly over time. The link rates change dramatically, which makes downloads bursty and thus, produces many short outages during the transmission of packets on the network.

Second, resources are shared among multiple users on the network. The scheduler managing these resources may sometimes take significantly long time to assign resources to certain transfers. As a consequence, packets may suffer from long waiting times in the queue.

Third, multiple transfers launched by a user simultaneously may suffer from short-term outages due to the self-inflicted delays. The traffic from multiple transfers may compete with each other. Consider a scenario when a user downloads a long file and at the same time performs web browsing. The short transfers of web browsing application may suffer long delays, as packets get stuck in potentially long queues at the gateways, due to the heavy traffic generated by the file download.

For the above reasons, QoE-based management of networks is gaining central importance in the success of services provided by a network operator. Network operators need to deploy efficient and user-centric resource management mechanisms. Obviously, they need to share the resources and due to the resource sharing, delays may occur. However, they need to be aware of the user preferences and thus, share these resources in a way that minimizes the impact of delays on the user QoE.

On this background, we evaluate the impact of duration and frequency of disturbances on the web browsing QoE of online shoppers. These disturbances appear in the form of packet delays. We create situations analogous to the cellular networks, and put certain packets in a queue at an intermediate node on the network. Either long delays appear all at once and then, problems get resolved, or short delays appear continuously for a long period of time. The long delays for a short period of time means, users suffer from a considerably high Page Load Time (∼16 seconds) on a single web
In this paper, we will evaluate how users respond to the above situations. The results of this study will propose a set of principles for the QoE-based performance management, which is an integral part of the functional dimension of network management, i.e., FCAPS (Fault, Configuration, Accounting, Performance and Security Management) [2].

Previously, a several number of papers reported the impact of delays on the end-user QoE [3, 4]. Another study showed that the user-QoE drops significantly over time when the Page Load Time grows but it does not recover completely after the network problem is resolved [5]. In [6], authors show that the user satisfaction level breaks when waiting times exceed 10 seconds in a single session. Similarly, another study illustrated the importance of short waiting times in the case of e-commerce services [7]. The users cancel download of images when the waiting time exceeds 10 to 20 seconds [1]. The study [8] that estimates for user tolerance of Quality of Service (QoS) for an e-commerce website states that, the delay between 2–6 seconds can be estimated accurately by the users. All the related studies draw some thresholds on tolerable waiting times, based on the user QoE. According to the best of our knowledge, there is no study, which describes the trade-off between the duration and frequency of delays in a systematic manner. The next section describes our methodology further in detail.

The remainder of this paper is structured as follows. Section II provides details about the research methodology. Section III presents experiment setup used in this study to conduct user tests. Section IV analysis and results from the experiment. Finally, Section V poses a set of summaries and concludes with future work.

2 Methodology

In this study a total of 42 participants took part in the experiment. The mean age of all participants was 26 years. The maximum age was 33 and a minimum age was 19 years. All these subjects were regular users of Internet and use e-commerce website for online shopping. We provided a primary training session of 5 minutes for
each subject. During the training all necessary instructions required to perform the
test are provided. A task-driven process was provided. These tasks were selecting a
category, then a product in the category and purchasing a product in it. We provided
three shopping sessions for each subject. Each session was defined with a browsing of
five web pages: Category selection page, product selection page, product details page,
payment details and payment confirmation page.

Figure 1 represents one complete experiment of a subject, where 1st, 2nd and 3rd
session represents three individual shopping sessions with the peak delays of 4 seconds,
10 seconds and 16 seconds, respectively at any web page of the session. In this paper,
the term peak delay represents the highest Page Load Time faced with a subject during
a shopping session of five web pages. In Figure 1, the y-axis represents the Page Load
Time for each individual web page, which indicates the amount of delay perceived by
the subject for each web page. Despite the difference in the delay pattern, we kept
the total waiting time of every shopping session approximately 20 seconds. We want
to see whether the users report their experience differently due to the different peak
delays within a session?

The applied delay patterns can be viewed from Figure 1. The first session repre-
sents 4 s session, in which we put a continuous delay where users perceived continuously
4 seconds of delay on all pages in the session. Second session represents 10 s session,
in which we put a delay in the second and the fourth pages in which user perceived a
delay of 10 seconds on both these pages. The third session represents the 16 s session,
where we put a large delay on the third page, where all users perceived a delay of
16 seconds on this page in the web session.

Additionally, we randomized the order in which the above mentioned sessions
appear to each subject. Some users had 4 seconds of peak delay in the first session,
followed by 10 seconds of delay in the second and then the 16 seconds delay at the end.
While some other users experienced 16 seconds of delay at the beginning, followed by
4 seconds of delay and then 10 seconds of delay at the end. All the delay patterns are
mentioned in Table 1. Hence, every subject experienced these sessions with a random
occurrence.

At the end of each purchase session, subjects were asked to answer these following
questions:

1. *How do you feel about the overall loading time?*

The options for this question were provided based on the five points ACR scale
for rating quality, which is recommended by ITU-T [9].

2. *Would you be willing to use this internet service again?*

The options for answering this question were given as follows:

- Yes
- No

Based on the results obtained from these questions from each of the subjects, a detailed analysis is made to find out the impact of these different peak delays.

### 3 Experiment Setup

In order to find out the effect of the network disturbances on a web browsing session, an experimental setup was established having a server, a client and a network emulator. The network emulator was placed between the server and the client to generate desired network environment (Figure 2). The KauNet [10] was installed and configured in Linux environment (Ubuntu 10.04) as the network emulator. All traffic passes through the network emulator and the bandwidth of 10 Mbps link.
We used popular web server called Apache web server (Apache 2.2) [11] configured on the server machine. The application Bind9 [?] is installed and configured on the server for Domain Name System (DNS) service. Server machine was setup with Ubuntu 10.10 and client machine with Windows 7 operating systems. All the web pages on web server were deployed using a well known PHP framework called CodeIgniter [13]. CodeIgniter is an open-source, lightweight, powerful web application framework. These web pages were accessed by the client side web browser (Google Chrome) based on the request. Where Google Chrome browser was set to a incognito mode [14]. The abduction of all HTTP(s) traffic between the client computer and the Internet over the Windows platform was done by an open-source web debugging proxy tool known as Fiddler [15].

We used fiddler to collect the logs on the client side. These logs were stored in HTTP ARchive (HAR) files. A PHP script was used to extract all the required information presented in the 'har' file. This script fetched timestamps based on the first request from the client and the last response from the server. To collect the network level trace, we used network protocol analyzer T-shark [16], which was used on the client machine. It captured all the packets from the client-server communication on the network-level. All files are stored locally in a ‘pcap’ format. A Perl script was developed to extract the timestamp of the request from the client and the last response being sent by the server to the client.

We developed an automated tool to manage the entire experiment and placed on the client machine. The end-user can have a continuous flow of a real life web browsing experience without interruption. Based on the design of the experiment process (User ID, Session ID and URL), this script fetches the desired network settings specified by the user for that session and signaled the network settings to the network emulator.
Furthermore, it collects answers to the question, mentioned in the earlier section. This information was stored in the local database (MySQL) [17] based on the User ID, Session ID, network settings and answers to the questions.

4 Analysis and Results

4.1 Impact of peak delays on end-user QoE

This section discusses the overall impact of peak delays in a session. As we mentioned in the previous section, peak delay refers to the highest Page Load Time faced by a user within a shopping session. The users performed three shopping sessions and thus, went through three different peak delays, which we induced systematically in a controlled manner. However, the total waiting time, i.e., the accumulated Page Load Times of five pages for each of the three sessions was the same, i.e., 20 seconds.

Figure 3 illustrates the Mean Opinion Score (MOS) values given by the users in each of the three sessions. Clearly, the users prefer delays to be short, even if these delays continue to occur for many consecutive web pages during a session. They gave better ratings in the session in which all the five web pages loaded in 4 seconds (approx. MOS value of 3.2), in comparison to the session in which one web page took 16 seconds to load, while, all the other four web pages loaded within 1 second of time (approx. MOS value of 2.5).

Additionally, we also report the share of user ratings for each of the sessions. Figure 4 displays bar chart of ratings for each session. The session with 4 seconds of peak delays leads to more positive percentage ratings, as compared to the other two sessions. Note that, none of the users reports “Excellent” rating. Obviously, the peak delay was not less than 4 seconds in any of the sessions. Therefore, the users noticed delay and their flow of thoughts got interrupted during the task, which motivated them to provide ratings below “Excellent”. The “Good” and the “Fair” ratings constitute of more than 80% of the total ratings for 4 seconds of peak delay. This share reduces significantly for the sessions with peak delays of 10 seconds and 16 seconds, and is replaced mostly by the climbing “Poor” ratings.

The above findings suggest that users do not like long network disturbances, occurring all at once. The users are more tolerant towards low intensity network disturbances continuing over a long period of time. Although, the accumulated waiting
time in every session is approximately the same, but the amount of peak delay makes the difference in the user given MOS. Obviously, the users avoid situations where they have to wait for too long without any response. The network resources management mechanisms should ensure that, the end users keep getting at least some piece of response, without waiting for too long at a time. The end-user waiting times in one chunk should therefore be small and distributed over time.

4.2 Impact of peak Delay on End-User Acceptability

In this subsection, we present the impact of peak delay on the end-user acceptability of service. We provided users the option to express their acceptability of the delays by answering “Yes” or “No” to a close-ended question, which is mentioned in the previous section. It is interesting to see a suggestive difference between their responses for different sessions, which are displayed in Figure 5.

Around 80% of the users find a continuous 4 seconds of Page Load Time acceptable to them. However, more than half of the users find the service unacceptable, if any one web page during a shopping session takes more than 10 seconds of delay to load.
4.3 Impact of delay sequence on QoE

This section discusses the impact of the order in which sessions appear. As mentioned previously, we randomized the sequence in which users faced delays. In the experiment, a user could go through one of the six possible sequences of sessions, which are mentioned in the Table 1. For example, shopping session with the peak delay of 4 seconds may appear at any of the three possible positions: the first session (start), the second session (mid) or the third session (end). This randomization of the order makes sure that our results do not get biased by the sequence in which delays occur.
4 ANALYSIS AND RESULTS Paper III

In Figure 6, the x-axis represents a session’s position of appearance to a user during the experiment and y-axis represents the user given MOS. The “start”, the “mid” and the “end” indicate the location of each session. We observe that the MOS for 4 seconds peak delay session remain significantly higher than the other sessions regardless of the position. We do not find a clear trend due to the location, but it is clear that, the users prefer small disturbances appearing frequently, as compared to a long disturbance appearing without any response.

From these observations, we can point out that, though the overall waiting times are approximately same for each session, the user given MOS is different depending on the position where delay has been perceived. The later scores for a same amount of delay do not seem to rise above the respective initial MOS score.

4.4 Network-level analysis

In this subsection, we provide a brief look at events, occurring at the packet-level. It is interesting to look at the reactions of the protocols, when we induce certain delays on the packets. We targeted packets transmitted from the server to the client, carrying
Table 1: Combinations of sequences for different peak delay sessions

<table>
<thead>
<tr>
<th>Sequence of order</th>
<th>Session Appearance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4 s, 10 s, 16 s</td>
</tr>
<tr>
<td>2</td>
<td>4 s, 16 s, 10 s</td>
</tr>
<tr>
<td>3</td>
<td>10 s, 4 s, 16 s</td>
</tr>
<tr>
<td>4</td>
<td>10 s, 16 s, 4 s</td>
</tr>
<tr>
<td>5</td>
<td>16 s, 4 s, 10 s</td>
</tr>
<tr>
<td>6</td>
<td>16 s, 10 s, 4 s</td>
</tr>
</tbody>
</table>

Fig. 6: MOS at different positions for all three sessions

response of the object requested by the client. Thus, we created conditions in which the packets get stuck in a queue at the intermediate node.

Generally, on average, two TCP connections open at the first page of a session.
These connections do not usually terminate at the end of page download. They continue over the next web pages of the session and entertain further subsequent requests from the client. Therefore, we do not observe terminations of TCP connections, until the end of the shopping session in normal scenario. However, when hold packets in a queue, we start to see the abnormalities, which are described below.

When we apply delay on the packet carrying the response from the server, we notice that the client keeps initiating new TCP connections with “SYN” packets, frequently, until it receives packets from the server. The higher the amount of delay, the greater the number of TCP connection requests from the client. For example, when we applied approximately 10 seconds of delay on a packet from the server side, we found 9 additional SYN requests from the client side, as mentioned in Table 2. Similarly, we observe a large number of TCP terminations initiated by the server, when we apply delays higher than 10 seconds. The exact number of TCP connections termination are listed in Table 3. When the server does not get ACKs of the sent packets, it starts terminating the existing TCP connections, and keep on retransmitting the FIN packets.

<table>
<thead>
<tr>
<th>Delay</th>
<th>Mean</th>
<th>STD</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>No delay</td>
<td>3.41 s</td>
<td>1.17 s</td>
<td>10.40%</td>
</tr>
<tr>
<td>4 s</td>
<td>5.43 s</td>
<td>1.85 s</td>
<td>10.31%</td>
</tr>
<tr>
<td>10 s</td>
<td>12.33 s</td>
<td>1.43 s</td>
<td>3.50%</td>
</tr>
</tbody>
</table>

Similarly, the client also retransmits the object requests by sending GET requests for the same object recursively. We observe a significant increase in the amount of GET requests from the client side when we apply delay on packets.

The opening of many sockets for a single web page download wastes lot of resources at both end systems, and also produce unwanted traffic on the network. Such delay events do not only waste resources but also produce a multiplicative impact on the end-to-end performance of data transfers as well as on the QoE.
Table 3: *TCP connection terminations observed at one of the web pages.*

*Large number of Terminations for 16 s delay*

<table>
<thead>
<tr>
<th>Delay</th>
<th>Mean</th>
<th>STD</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>No delay</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4 s</td>
<td>0.15 s</td>
<td>0.48 s</td>
<td>100%</td>
</tr>
<tr>
<td>16 s</td>
<td>12.03 s</td>
<td>4.24 s</td>
<td>10.65%</td>
</tr>
</tbody>
</table>

5 QoE-based Network Management

A user session on the network is a continuous flow of experience, where a user comes across a series of waiting times. These waiting times are caused by the end-to-end packet delays. At times, the packet delays are so high that they damage badly the QoE of whole session. Over the years, it has been witnessed through many studies that the major reason behind such delays is the inappropriate management of network resources, and more specifically, the inefficient adaptation of link capacity. The assignment of link capacity need to be user-centric and takes into account the strategies, which maximize the overall QoE of a user session.

The outcomes of this study show that the users prefer small waiting times over sudden long delays within a session. The results provide a guideline to network operators for the user-centric management of link capacity. The link capacity adaptation mechanisms must ensure that a user at least gets a minimal level of service without long interruptions, such that, a constant flow of requested data is received by the user. Specifically, network operators should avoid the assignment of link capacity, which is much higher than the amount of anticipated consumption, followed by multiple seconds of outages. Instead, they need to shape link capacity based on user QoE, such that, it ensures constant flow of response. The user-centric network management will thus help network operators to improve customer satisfaction and reduce underutilization of the available link capacity.

We observed in this study that the users do tolerate waiting times in a session, if they are distributed in small chunks over times. Our results show that a moderate link capacity resulting in constantly occurring waiting times around 4 s are more acceptable to the users in comparison to the high link capacity (waiting times below...
1 s), followed by occasional long waiting times. Moreover, our results also showed how the inefficient assignment of resources result in the abrupt initiation and terminations of TCP connections, which further result in the degradation of performance as well wastage of network and system-level resources. Hence, the results in this study signifies the importance of user-centric adaptation of link capacity and more generally, user-centric, QoE-based management of networks.

6 Conclusion

In this paper, we presented the results of our study on the user responses to delay distributions during task-based shopping sessions. Our study was based on the subjective experiments performed in lab environment with users. In this study, we designed three different test conditions for each user. Each condition refers to a specific type of peak delay induced during page loading process. The main question of this study was to determine whether users like long delays occurring all at once or they prefer short delays, occurring continuously on many consecutive pages. In the experiment, each user went through three web browsing sessions to perform shopping tasks. Each session was based on 5 web pages. In one session, they faced 4 s of additional delay on each of the five pages. In the another one, they went through 10 s of additional delay on two pages, while no delay at the remaining three pages. In the third session, they faced a long delay of 16 s all at once on one of the web pages, while the other four web pages had no additional delay. We found out that the users prefer short delays (4 s) occurring continuously, in comparison to the rarely occurring long delays. Their MOS score was above 3 for 4 s of peak delay, which reduced to 2.5 for 16 s of peak delay. Users also found the service unacceptable, when they faced 16 s of delay on one of the web pages during a session. Although, the cumulative waiting time of each session was 20 s, the session with low peak delay obtained better ratings from the users.

Network operators may use these results to devise better user-centric strategies for resource management. In particular, they need to take into account that the interactive activities like web browsing, and particularly the shopping tasks on Web, require immediate responses from the server. Any additional delays due to channel scheduling, long packet queues or link quality changes may produce significant impact on the user QoE. Therefore, the appropriate resources should be assigned to this type of application in order to keep the delays short. One of the considerations is to schedule the traffic such that the short time slots be assigned frequently in order to avoid long
interruptions during the flow of data.

7 Future Work

In order to understand the impact of variety of delay distributions on packets during a usage session, this research work needs to be extended to the real-life scenario using field trials or crowd sourcing experiments. The real-life scenario will allow more freedom to test the impact of variety of network conditions on QoE. It is crucial to identify the impact of time-varying network performance on the overall session QoE of a user. Specifically, the studies need to quantify and model the impact of duration and frequency of extra network delays on user session QoE. This can be achieved by creating controlled time-varying network conditions using model-based network emulations. The future work also needs to extend this study to other popular applications and content types. The aim is to come up with a generalized model, which enable network operators to carry out QoE-based network management tasks, and thus, improve the customer satisfaction.

Acknowledgment

This research is supported by Vinnova through the Celtic-plus project Quality of Experience Estimators in Networks (QuEEN).

References


REFERENCES


[16] “T-shark: The Wireshark Network Analyzer 1.10.0,”

PAPER IV

MODELING AND ANALYSIS OF WEB USAGE AND EXPERIENCE BASED ON LINK-LEVEL MEASUREMENTS

PUBLISHED IN THE PROCEEDINGS OF
INTERNATIONAL TELETRAFFIC CONGRESS (ITC)
KRAKOW, POLAND, SEPTEMBER 2012
Modeling and Analysis of Web Usage and Experience Based on Link-Level Measurements

Junaid Shaikh, Markus Fiedler, Patrik Arlos, Denis Collange*

Blekinge Institute of Technology, Karlskrona, Sweden,
(junaid.junaid, markus.fiedler, patrik.arlos)@bth.se

*Orange Labs, Sophia Antipolis, France,
denis.collange@orange.com

Abstract – Internet traffic monitoring and analysis have been playing a crucial role in understanding and characterizing user behavior on the web. In particular, ON-OFF models capture the essential phases of user communication with web servers. The OFF phases reflect both deliberate and accidental gaps in the traffic flow. In this paper, we present a passive monitoring and analysis method devised to assist in the identification of such traffic gaps that may result in the degradation of Quality of Experience (QoE). Our first contribution consists in a revised ON-OFF model to cater for OFF times reflecting accidental gaps which are induced by the network. Second, a wavelet-based criterion is proposed to differentiate between the network-induced traffic gaps and user think times. The proposed method is intended to be implemented in near-real-time as it does not require any deep packet inspection. Both web service providers and network operators may use this method to obtain objective evidence of the appearance of QoE problems from link-level measurements.
1 Introduction

There has been an enormous growth in the deployment and usage of wireless networks during the recent years. The performance of these networks is highly varying due to their availability and coverage issues. Particularly, the outages in the traffic in wireless networks are quite frequent and they result in longer waiting times on the web. We use the term ”outage” here to refer to the events causing short-term temporary disruptions in the data transfer due to problems in the network. Outages and competition for resources lead to gaps in the network traffic, and they are perceived badly once their consequences show up in the user interface.

Moreover, the random appearances of such outages result in bursty traffic and sudden degradation in the Quality of Service (QoS). This sudden degradation in the QoS affects the user Quality of Experience (QoE) significantly. On the link level, these outages could be seen in the form of gaps in the traffic flows. For the Internet Service Providers (ISPs), it is important to monitor these traffic gaps resulting from the network outages in order to obtain hints on how to improve their services. However, the outages due to the network problems are not the only reasons for gaps in the traffic. The gaps in traffic may just be due to the inactivity of the user, which we call the user think time between two transactions. Hence, it is important to distinguish between both reasons that lead to gaps in the traffic.

Successful differentiation between these two types of gaps enables the ISPs to identify the network outages by monitoring the traffic flow on the network. We propose a fast approach to keep track of the user think times that doesn’t require any deep inspection of packets for identifying the end of a user transfer.

In this paper, we provide the following contributions. First, we discuss the characteristics of the gaps caused due to the user inactivity on the web. Second, we present an analysis of the features of the gaps induced by the network, and how they could be used to differentiate between a smooth and a disturbed live video streaming transfer on the web. Third, we present a wavelet-based criterion to identify the traffic gaps caused by the problems in the network. We have targeted live video streaming on the web because the consequences of problems in the network can be experienced immediately in the form of freezes in the video. Every freeze results in the loss of information in the case of live video streaming and hence, in user dissatisfaction.

In [1], the authors identified gaps and user-perceived problems, but they didn’t quantify the boundary towards the think times. They decoded the stream and sim-
ulated the buffer content afterwards. Moreover, their study was based on Youtube video. According to the best of our knowledge, this study is first in provoking the discussion on the network-based criterion that differentiates between the user think times and the network outages and relates them to user-perceived video delivery issues on web. We also derive a rather simple criterion for discerning network outages from think time that can be evaluated in near-real-time.

The remainder of this paper is structured as follows. Section 2 describes the methodology of this work, Section 3 describes the ON-OFF models for the Web and presents the findings on the quantification of the user think times, and Section 5 quantifies the traffic gaps along with the comparison of smooth and disturbed web-based video transfers. Section 6 presents the wavelet analysis of the captures traces and Section 7 proposes the criteria for identifying the network-induced traffic gaps. Finally, Section 8 concludes the paper along with a short description of our short-term future work.

## 2 Methodology

This section presents the methodology used for conducting this study. There are three sides of this study. First, it outlines the user think times on the web based on the previous studies. Second, it presents an analysis on the properties of two live video streaming transfers on web. And third, it proposes a criterion to monitor and detect the presence of the traffic gaps on different timescales.

![Fig. 1: A web session with serial transfers: One client, one server](image-url)
To outline user think times, a literature review is done to understand how users launch their requests on the web. Our findings are summarized in Section 4.

We then analyzed the inter-packet times in order to be able to differentiate between a smooth and a disturbed web transfer. For this purpose, we did live web-based video streaming from a distant server via two different Wi-Fi networks: one at a home in Sweden and another at a hotel in Germany. The Wi-Fi at hotel was chosen because, usually, networks at hotel show signs of capacity shortage when many uncoordinated users are active at the same time, e.g. during the evenings. Video streaming was done using the Firefox web browser with embedded Flash player. At the same time, the traffic was captured via the Wireshark [2] traffic capturing tool. The Macintosh operating system version 10.6.3 was used on the computer with processor speed 2.53 GHz and 4 GB of memory. Each transfer is 180 s duration long. The captured traffic from the server to the client direction was used for the analysis of inter-packet times.

Subsequently, the user think times were compared to the inter-packet times of smooth and disturbed transfers. This allowed us to draw a borderline between (1) the duration of the gaps generated due to potential user think times between two transfers, and (2) the gaps due to inter-packet times within the same transfer. Finally, the wavelet analysis was performed on both the transfers to visualize the ON and OFF phases along with their frequency and duration on different timescales.

Fig. 2: Parallel transfers in a web session: One client, multiple server
3 The ON-OFF model

The user web session could be characterized by an ON-OFF model as illustrated in Figure 1. Each web session may consist of the transfer of one or more web pages. Similarly, every web page may consist of one or more objects. To retrieve each of these objects, a request is sent from the client to the server. In this section, we will first describe the terms used in the ON-OFF model. Later, we will classify ON-OFF models based on the nature of the web pages and the user behavior.

3.1 ON times

An ON time during a session is defined by the time elapsed from the arrival of a request from the client side to the end of the corresponding response from the server side. The ON times are illustrated by the black boxes in the ON-OFF model shown in Figure 1.

3.2 OFF times

An OFF time in the ON-OFF model represents the silent time between two subsequent transactions. For example, the client, after receiving the last object from the server, may take some time referred to as an OFF time before launching the next request. During the OFF time, there is no packet containing the data seen on link-level. The OFF times are further classified into two categories: The active OFF times and the inactive OFF times.

3.2.1 Active OFF times

When a user requests a web page, which consists of multiple objects, the client-side web browser may retrieve those objects by sending automatic requests to the server. The time elapsed from the end of the previous response from the server to the arrival of the next automatically-generated request from the client-side web browser is called the active OFF time. The active OFF times are shown as grey boxes in Figure 1.
3.2.2 Inactive OFF times

An inactive OFF time is the time a user spends on viewing or reading the contents of the page when one or more objects are already retrieved from the server. Inactive OFF times are also called the user think times and are shown by the white boxes in Figure 1.

3.3 Base file

When a new webpage is requested, the first request which is generated by the client is for the object which is commonly referred to as the base file of the page.

3.4 Embedded file

The files that are retrieved subsequently after the base file of the page are called embedded files. An embedded file could be an inline image, a link to the another page and a video or a video player etc.

4 A web session

A simple web session could be defined as a sequence of requests made by a single client to a particular web server. It starts when a user requests for a page by typing a URL in the address bar, clicking on a hyperlink or clicking a bookmark. Either of these user actions generate a request for the base file from the server. After the base file is retrieved, the subsequent requests for the embedded objects are made automatically by the client-side web browser. The request for every embedded object is qualified by the active OFF time.

After the page is displayed, the user reads or views the contents of the page. For instance, the user may be reading a newspaper, watching a video or filling a form etc. This time is called the inactive OFF time or the user think time.

The structure of websites has changed significantly during the last few years. The emergence of Web 2.0 has fueled the popularity of mashups. A webpage is made up of tens and hundreds of objects, which are hosted by one or more servers. According
to [3], on average each webpage is composed of more than 50 objects. When a user requests for a web page, the objects of the pages may get retrieved from several servers. Usually, these objects are not transferred serially, but in parallel to each other.

Figure 2 presents a web page transfer involving a parallel transfer of objects. There can be any of the two possible reasons behind such parallel transfers. First, the client may generate multiple requests to the same server, each from a different TCP port to allow the parallel transfers. Second, when the embedded objects on a page are hosted by more than one servers, the client may generate multiple requests to the multiple servers in parallel to each other. Hence, multiple objects on the same page could be retrieved from multiple servers at the same time.

In both the above cases, multiple TCP connections could be observed on the network level, each carrying a different object at the same time. However, in the latter case, objects retrieved from the different servers are not considered as part of the same web session, as a session is based on the pair of client and server IP addresses. The change in source or destination IP addresses marks a different session.

In [4] authors presented two regions of OFF times, (1) 1 ms to 1 s for the active OFF times and, (2) 30 s to 3000 s for inactive OFF times. In [5], the median value of user think times is given as 15 s based on the silent time threshold between two documents requested from the client side. According to [7], most of the requests from the same user are launched with an inter-request time of less than 64 s. The appropriate value for the silent time threshold for most user sessions were shown to be between 100 s to 1000 s. It means that, on the average, a single user makes a sequence of requests to a particular server with the think times less than 100 s during a single visit. More than half of these requests were generated automatically by the client-side application since their think times were less than 1 s, which we call the active OFF times. User think times during a session were defined based on the inter-request times for the base files. Half of these inter-request times were above 8 s, with a large ratio of these times between 16 s and 64 s. More than 30% of the inter-requests times were less than 1 s and remaining 20% were between 1 and 8 s. The user sessions of the Youtube website were further characterized in another study [9]. There, the user think times were found to be increased to 30 s, possibly because of the video streaming. The users take some time watching the video before making the next request. In the same study, authors found out the value of active OFF times to be between 1 s to 30 s with only 14% of these values exceeding 1 s.

Summarizing the above observations from the literature, we find that the active
OFF times are usually less than 1 s, while inactive OFF times vary quite a lot. We observe that their values are generally above 8 s. These observations are further listed in Table 1. This draws a borderline that the gaps above 8 s in the traffic between a particular pair of the user and the server potentially shows the user think time between two transfers.

<table>
<thead>
<tr>
<th>Lit</th>
<th>Active OFF</th>
<th>Inactive OFF</th>
<th>Silent time threshold</th>
<th>Additional info</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4]</td>
<td>&lt; 1 s</td>
<td>30 – 3000 s</td>
<td>Part of Inactive OFF times</td>
<td>-</td>
</tr>
<tr>
<td>[5]</td>
<td>-</td>
<td>~ 15 s</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[7]</td>
<td>&lt; 1 s</td>
<td>&gt; 8 s</td>
<td>100 – 1000 s</td>
<td>Football website</td>
</tr>
<tr>
<td>[9]</td>
<td>&lt; 1 s</td>
<td>~ 30 s</td>
<td>&gt; 1000 s</td>
<td>Youtube</td>
</tr>
</tbody>
</table>

5 Network-induced gaps

In the previous section, we saw that the user think times are usually above 8 s, as a user normally takes at least this much time to view the content and clicking the button for the next request. In this section, we discuss the gaps that are induced by the network, which can be due to weak signals in the wireless networks, congestion problems or poor performance of network protocols etc, and are the potential candidates for destroying the user experience. For this purpose, we did some experiments of live video streaming in the Web browser via two different networks. Experiments were conducted in Wi-Fi environment at a home in Sweden and a hotel in Germany. The video streaming was quite smooth in the home environment while it was freezing quite frequently in the hotel environment.

Figure 3 shows the inter-packet times that we observed from the transfer performed in the home environment. Obviously, typical inter-packet times are well below 1 s with an average of 72 ms. This characterizes the gaps when the video streaming is working fine without freezes. We see that these inter-packet gaps rise and approach 1 s occasionally. However, such occasional increases in the inter-packet times did not
hurt the video streaming and we didn’t observe any freezes. This further shows that frequency of such inter-packet times around 1 s also matter. Once the buffer gets empty, then the freezes occur and hence affects the user experience [10].

<table>
<thead>
<tr>
<th>Network</th>
<th>Duration</th>
<th>Total OFFs</th>
<th>1 s – 2 s</th>
<th>2 s – 3 s</th>
<th>3 s – 4 s</th>
<th>Above 4 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home Wi-Fi</td>
<td>180 s</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hotel Wi-Fi</td>
<td>180 s</td>
<td>37</td>
<td>24</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 4 illustrates the inter-packet times observed from the trace captured in the

Fig. 3: *Inter-packet times of video stream captured via Home Wi-Fi*
hotel environment. The difference is quite obvious. There are frequently occurring inter-packet gaps above 1 s and some of them may result in the freezes in video streaming depending on the buffer size, and hence delay the delivery of the transfer. Users are often very sensitive to the delays while watching the live video streaming, which is probably due the nature of content they are watching e.g. sport events or news. They lose a piece of information each time the video is frozen. Hence, such gaps above 1 s are very much threatening the user experience in the live video streaming context. Most of these inter-packet gaps were found between 1 s and 4 s, occurring more or less with the regular intervals of time during the first 100 s of video streaming. We did observe a considerable number of freezes in the video during this period. Finally, after 100 s of video streaming, there were even longer gaps above 4 s. The inter-packet gaps between 1 s and 4 s repeating in cycles are alarming and should be given special attention while monitoring.

Table 2 summarizes the results from the two transfers. Here, we use the term OFF
times for the gaps (i.e. inter-packet times) above 1 s. Clearly, the video streamed on the hotel Wi-Fi illustrates the bad transfer full of OFF times. Most of these OFF times are less than 4 s. These frequently occurring OFF times between 1 s to 4 s characterize very well the existence of bad transfers and should be given special attention. Although, 1 s to 4 s OFF times during a transfer are alarming, it does not mean that the good transfers do not consist of such gaps at all. Therefore, the frequency of such OFF times is also a major point of consideration before declaring a transfer “good” or “bad”. So we also need to keep track of the duration of the ON times between two subsequent OFF times. For this reason, we present in Figure 5 the CCDF of the durations of OFF times, ON times and ON+OFF times from the bad transfer. ON+OFF times represent the cycle and illustrate the frequency of the occurrence of gaps.

Obviously, the OFF times are greater than the ON times. It shows that the time the user can really enjoy the video before the video freezes is less than the duration of freeze time. This is itself a clear indication of the bad transfer. The frequency of freezes sinks as their duration increases. We also observe freezes that are in the range of think times. A buffer of some seconds takes most of the problems away, given the traffic catches up again after the freeze time for which we see evidence in [11]. The curve of ON+OFF times in the figure shows the cycle time i.e. how frequently one ON and OFF times phase finishes. In Figure 5, 80% of the ON+OFF times are still less than 5 s, showing only a slight difference as compared to the OFF times. This further illustrates the frequency of gaps which is high in this case. Here, we have not presented the distribution plot of ON times and OFF times for the transfer done via home Wi-Fi. The reason is that, there is only one OFF time of duration above 1 s during the whole transfer, that follows an ON time of around 170 s. Actually, occasional OFF times of slightly above 1 s after considerably long ON times characterize a good quality transfer, assuming that the buffer size is long enough to keep up with such occasional short outages. Hence, the relative difference between the OFF times and the ON times is also one of the major factors in declaring the quality of a transfer. In the next section, we will further show how this relative difference could be visualized and quantified on the different timescales.

Further, for validation purposes, the direction of the transfer could also be considered. Usually, the user think times are followed by the request in upward direction i.e. from the user to the server. Conversely, the outage in the video streaming download is followed by the data packet (containing a request) in the downward direction. Therefore, if a gap in the traffic is followed by the packet with payload from user side
then the gap could be considered as the user think time, while if it is followed by the packet from server side, then the gap was potentially a result of an outage.

6 Wavelet analysis

In order to visualize the quality problems at different scales, we have performed the Haar wavelet analysis of both the traffic transfers. It allows us to identify the time scale at which the problem occurs. We can view both the time and the frequency components together, for instance, how long the gaps are and how frequently they occur. Hence, the wavelet analysis performs the localization allow us to locate those time instances where the problems occur. The wavelet analysis also allows us to test the trend and the burstiness of a transfer on the fly, passively from the measurements.

![CCDF distribution of ON and OFF times for the transfer done via hotel Wi-Fi](image)

Fig. 5: CCDF distribution of ON and OFF times for the transfer done via hotel Wi-Fi
on the link-level. To check the burstiness at different time-scales, the $d$ coefficients are used; and to view the trend (like moving average) of a transfer, the $c$ coefficients are utilized. The $d$ coefficients are henceforth called the wavelet coefficients and the $c$ coefficients are called the scaling coefficients.

6.1 Calculation of Wavelet and Scaling coefficients

The $d$ coefficients extract the detail in the time series (traffic trace) at different scales and different locations. In other words, the $d$ coefficients display the degree of difference between the data points at different locations in a time series.

![Wavelet and scaling coefficients](image_url)

**Fig. 6:** Wavelet and scaling coefficients
Let \( y \) be the vector that represents data points in a time series: 
\[
y = \{y_1, y_2, y_3, y_4, \ldots, y_n\}.
\]
Let \( n \) be the length of the vector \( y \), which must be a power of 2 such that, \( n = 2^j \).
Thus, on the finest scale \( J - 1 \), the wavelet coefficients \( d \) between the two successive points can be calculated as:
\[
d_{j,k} = y_{2k} - y_{2k-1},
\]
where \( k = 1, 2, 3, \ldots, n/2 \) and \( j = J - 1 \).

Let’s assume that there are \( n = 8 \) data points in the vector \( y \), then on the finest scale \( J - 1 = 2 \), there will be four wavelet coefficients: 
\[
d_{2,1} = y_2 - y_1, \quad d_{2,2} = y_4 - y_3, \quad d_{2,3} = y_6 - y_5, \quad d_{2,4} = y_8 - y_7.
\]
Hence, the values of wavelet coefficients demonstrate the variation between the immediate neighbors at a particular scale.

Moreover, a smoothing operation can be performed on the time series by obtaining \( c \) coefficients. These coefficients give us information about a time series on the coarser scale. The operation of scaling coefficients is similar to the moving average smoothing operation. Thus, the \( c \) coefficients at the finest scale \( J - 1 \) can be calculated as:
\[
c_{j,k} = y_{2k} + y_{2k-1},
\]
where \( k = 1, 2, 3, \ldots, n/2 \) and \( j = J - 1 \).

In order to obtain the detail coefficients \( d \) at the coarser levels \( J - 2, J - 3, \ldots, 0 \), the differencing between the two non-overlapping consecutive pairs of \( c_k \) is performed at each level, as mentioned in the Figure 6. Hence, the \( y_{2k} \) and \( y_{2k-1} \) in the equation will be replaced by \( c_{2k} \) and \( c_{2k-1} \), respectively:
\[
d_{j,k} = c_{2k} - c_{2k-1}
\]
Similarly, the \( c \) coefficients at coarser scales can be calculated as:
\[
c_{j,k} = c_{2k} + c_{2k-1},
\]
where \( k = 1, 2, 3, \ldots, n/2 \) and \( j = J - 2, \ldots, 0 \).

Furthermore, for the purpose of normalization, all the obtained \( c \) and \( d \) coefficients are divided by \( \sqrt{2} \) before using them in the spectrum analysis.

We performed the wavelet analysis on the throughput of both the web transfers (collected at home and hotel) mentioned in the previous section. The throughput was
calculated as the number of packets received at the client side during each time window. The time window was set to 125 ms. Based on the obtained time series of throughput, we calculated the $c$ and $d$ coefficients from the finest to the coarsest scales. Finally, we computed the power spectrum of $c$ and $d$ coefficients at each scale from $j$ to 0 to observe the scaling behavior. Investigating different series of coefficients allowed us to pinpoint those locations in the transfer, where the change in the perceived performance occurred. The Equations 5 and 6 were used to calculate the power spectrum for $c$ and $d$ coefficients, respectively:

$$\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} c(j, k)^2 \quad (5)$$

$$\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} d(j, k)^2 \quad (6)$$

where $n_j$ is the number of coefficients at band $j$. Furthermore, the $\log_2$ of each power spectrum ($\mu_j$) is calculated, which is then plotted against the respective band ($j$), as depicted in the Figures 7 and 8.

**Table 3: Linear regressions between bands and the spectrum values of wavelet coefficients**

<table>
<thead>
<tr>
<th>Band</th>
<th>Hotel</th>
<th>Home</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 to 4</td>
<td>$\log_2(\mu_j) = -1.83j + 8.02, r = -0.96$</td>
<td>$\log_2(\mu_j) = 0.22j - 0.55, r = 0.54$</td>
</tr>
<tr>
<td>4 to 6</td>
<td>$\log_2(\mu_j) = 0.9j - 2.80, r = 0.99$</td>
<td>$\log_2(\mu_j) = 0.6j - 2.28, r = 0.85$</td>
</tr>
<tr>
<td>6 to 9</td>
<td>$\log_2(\mu_j) = -0.72j + 7.08, r = -0.98$</td>
<td>$\log_2(\mu_j) = 0.41j -1.56, r = 0.83$</td>
</tr>
</tbody>
</table>

To perform the wavelet analysis, we considered the throughput of the first 128 seconds of each transfer. As we calculated throughput in the time windows of 125 ms, therefore, the number of data points ($n$) becomes 1024, yielding scales $J = 10$ (0 to 9). The finest scale is $J − 1 = 9$ (at $0.125$ s) and the coarsest scale is 0 (at $64$ s).

In Figure 7, we present the plots of the spectrum of the $d$ coefficients from band 0 to 9. Each wavelet spectrum plot illustrates very well the different properties of a transfer...
and can thus be used to unveil the perceived performance at different timescales. For the hotel transfer, we observe three different scaling behaviors. We divide these three scaling behaviors into three different regimes within one transfer, i.e. less than 1 s (bands above 6), between 1 s to 4 s (bands 4 to 6), and above 4 s (bands below 4). The scaling behavior between 1 s and 4 s is of particular importance, as it characterizes the frequent OFF periods at a scale of 1 s to 4 s. Hence, it results in increased burstiness while going from band 4 to 6 (timescales: 4 s to 1 s). Since such scaling is absent on the timescales less than 1 s, it indicates that the frequent traffic gaps of around 1 s exist in the network traffic. However, at lower bands (higher timescales), we observe high burstiness indicating a difference in the quality at the different locations in the transfer. For instance, at band 1 (32 s timescale), we observe highest spectrum values, indicating the shift in the quality of the transfer every 32 s. In contrast, we observe a relatively stable behavior in the home transfer where the spectrum values increase with the decreasing timescales. The wavelet spectrum analysis of home transfer indicates usual network traffic behavior without many outages, which implies that the quality
of transfer at higher timescales appear smoother, while it appears bursty as we go towards lower time scale, i.e. higher bands.

The above-mentioned behavior is illustrated very well by the linear regressions (fitted on the spectrum data of wavelet coefficients) mentioned in the Table 3. For the hotel Wi-Fi network, three regimes are clearly visible. The linear regression of band 4 to 6 (timescales: 4 s to 1 s) shows strong positive correlation proving the existence of scaling on these scales. However, the other two regimes show a negative correlation. It indicates that much of the variation in the hotel transfer is present on the 1 s to 4 s scale. Conversely, from the linear regression, we observe less burstiness in the home transfer on the higher timescales, indicating a rather smooth transfer with much of the scaling at very small timescales. However, the existence of scaling at shorter timescales is the sign of activity (higher ON times).

Figure 8 displays the power spectrum of the $c$ coefficients for the home and the hotel transfers. The trend is very clear that the transfer done at home network gives higher values across different scales as compared to the transfer done at the hotel. At higher bands, i.e. shorter time scales, many of the time windows are empty which leads to a slow decay. In contrast, stability of the home transfer is evident from the consistent decay in the spectrum of scaling coefficients. Table 4 list the linear regressions for the hotel and the home transfers. There is a strong negative correlation for both transfers. However, the value of $\alpha$ is indicating a faster decay in the case of the home transfer.

7 Criteria for alarming gaps

In this section, we will propose a criterion to monitor those outages or traffic gaps, which can be helpful in prompting the service providers to take proactive actions for improving the QoE. As mentioned in the previous sections, there could be multiple reasons behind the occurrence of traffic gaps during a web session between a client and a server such as:

- The user think times.
- The server could be heavily loaded and may result in the bursty traffic.
- The client-side web browser causing the active OFF times.
The problems in the network such as the signal problems in wireless networks, scheduling on the base stations, congestion in the network or the dynamics of the network protocols.

To confirm if a traffic gap was from the server side and not due the user inactivity, the direction of the data packet after the traffic gap needs to be observed. If the next packet containing the data is from the server side and not from the client side, then the
last gap was not due to the user inactivity but due to the network behavior. However, it requires packet inspection to detect the direction.

All the above causes produce scaling in the traffic that might be different on different timescales. In order to identify the traffic gaps induced by a badly behaved network, it is important to observe the duration and the frequency of the gaps. The wavelet analysis of traffic generated in a session is an important tool to visualize on the fly both the duration and the frequency of traffic gaps. We propose the following step-by-step procedure to identify the network-induced traffic gaps:

1. A spectrum analysis of wavelet and scaling coefficients should be performed, and change point separating multiple scaling behaviors in the spectrum plot of wavelet coefficients should be identified.

2. If the time scales between 1 s and 4 s show a different scaling of the wavelet coefficients than their neighboring timescales (above 4 s and below 1 s) – for example if the corresponding slope changes sign – one can deduce that the traffic gaps of 1 s to 4 s are recurring frequently.

3. The scaling behavior on the long (>4 s) and shorter (<1 s) timescales suggest the shift in the quality of transfer at different times and the amount of variation in the traffic during the ON times, respectively. The negative slope is the sign of increasing inactivity.

8 Conclusions and future work

This paper proposed a simple wavelet-based criterion that can be useful for the service providers to monitor the user transfers. The criterion is fast as it does not require any deep packet header information and hence enables the service providers to take immediate appropriate measures based on the pure observation of the flow of data associated to the stream.

In this paper, we outlined the difference between the duration of traffic gaps generated due to the user think times and the network outages during a transfer. We found that the network outages that result in freezes in video transfers on web often constitute of duration between 1 s to 4 s, while user think times are usually above 8 s. This implies that, a gap above 8 s after a smooth transfer is likely to characterize the user think time. Therefore, such gaps can be ignored by the service providers.
Conversely, the gaps of duration between 1 s to 4 s, occurring with frequent intervals are a sign of poor quality transfer, as shown by the longer durations of OFF times as compared to the ON times and small ON+OFF time durations.

All these properties at different timescales could be visualized with the help of wavelet spectrum analysis. The presence of scaling at timescales below 1 s indicates the ON time and hence, the signs of activity. However, the presence of scaling at timescales between 1 s – 4 s, and the absence of scaling at shorter and longer time scales characterize the frequent OFF times with shorter ON times.

This is an ongoing work. Our short-term future work includes the investigation of traffic gaps with the other types of web traffic along with the investigation of wavelet spectra as a function of time. We further intend to validate this wavelet-based criterion with the help of experiments with real users on our test-bed, in order to differentiate between the user think times and the network-induced traffic gaps.

Acknowledgment

This work had been partly supported by the FP7 Network of Excellence Euro-NF and the CELTIC project QuEEN.

References


PAPER V

Quantitative Evaluation of Wavelet-based Traffic Gap Detection

To be submitted
Quantitative Evaluation of Wavelet-based Traffic Gap Detection

Junaid Shaikh, Markus Fiedler, Patrik Arlos
Blekinge Institute of Technology, Karlskrona, Sweden,
(junaid.junaid, markus.fiedler, patrik.arlos)@bth.se

Abstract – The time-varying behavior of mobile channels induces disturbances within traffic streams. These disturbances appear in the form of traffic gaps, which typically occur at multiple timescales. Network monitoring systems require smart data analysis methods to detect these disturbance events and raise alarms whenever they degrade QoE. Wavelet transforms provide important means for multi-resolution analysis of traffic streams and help monitor dominant frequencies at the relevant timescales. This work systematically evaluates wavelet’s ability of detecting traffic gaps at targeted timescales. Using a variety of time series exhibiting deterministic and non-deterministic traffic gaps of nominal durations, we discuss how wavelets detect the timescales on which the problems occur. It is continuation of an earlier work, which presented a qualitative evaluation of wavelet-based traffic gap detection.

1 Introduction

Wavelet analysis is a powerful tool for monitoring trends, changes, periodicities and abnormalities from the collected data [5] [10]. One of the major advantages of using wavelet analysis tool with regards to Quality of Experience (QoE) is that it can capture performance problems over time and show them at various timescales.
QoE is based on a continuous flow of user experience, evolving from various interactions with one or more services. Particularly, the impact of negative events, such as outages, accumulate over time. The performance levels observed throughout the course of usage are all interconnected from the QoE perspective, particularly due to the user memory [11] [20]. In order to monitor networks from user perspective, it is therefore important to devise mechanisms that present network performance at various different timescales.

The monitoring tools based on wavelet transforms can strengthen the understanding and visualisation of timescales at which performance problems repeat [13] [4]. The ability to detect duration and frequency of negative performance events enable monitoring systems to view and assess the overall impact made by these events on user experience. Hence, a better assessment of QoE becomes possible using metrics extracted by wavelet analysis.

In past, several papers reported studies on wavelets w.r.t. its usage in the network performance domain. Particularly, the studies used wavelet transforms to understand traffic patterns on the Internet [7] [8] [14] [2] [15]. Similarly, the authors in [3] [19] [17] used wavelets to model network traffic. Moreover, several works also reported the use of wavelet transforms to detect unusual traffic patterns caused by the different anomalies, intrusions and Denial of Service (DoS) attacks [12] [16] [6] [18]. These past studies have shown the viability of multi-resolution analysis using wavelets. However, the majority of literature on wavelets w.r.t. its application in the network traffic analysis was written more than a decade ago, when the area of network Quality of Experience (QoE) had not drawn enough attention from the research community. However, now with its improved understanding, QoE monitoring and modelling (particularly for bulky traffic streams related to long video sessions) can benefit a great deal from the application of multi-resolution property of wavelets.

This work is continuation of [1], where we presented qualitative measurements to show wavelet’s ability of monitoring QoE. In this work, we evaluate wavelets quantitatively by considering various different cases from deterministic to non-deterministic occurrence of performance issues in data streams. The non-deterministic cases are better representative of a real network. However, the deterministic cases are simple and easy to understand. Therefore, we use the deterministic cases to illustrate the impact of change in performance on the values of wavelet coefficients. We expect that this systematic evaluation of wavelets in terms of traffic gap detection will help improve its application in the area of QoE monitoring in general, and the detection of duration...
and frequency of user waitings times in particular.

This paper is organised as follows. Section 2 presents the terminologies used in this paper. Section 3 introduces wavelet transforms and the mechanism to calculate discrete wavelet transform coefficients. Section 4 briefly describes the methodology used in this paper for the data collection and analysis. Sections 5 and 6 presents results on the wavelet analysis of deterministic and non-deterministic ON-OFF traffic patterns. In the end, Section 7 discusses the application of wavelets on traffic streams constituting packet losses or originating from Variable Bit Rate (VBR) sources, followed by Section 8, which presents conclusions.

2 Background

This section describes basic terminologies that will be used in this paper.

2.1 Traffic Gap

This section briefly introduces the term “traffic gap” with regards to its usage in this paper. A traffic gap \( T_{\text{off}} \) is defined as the time duration elapsed between the arrival of two subsequent packets at an observation point on the network, such that, the total duration of this inter-arrival time is above a pre-defined threshold value. For example, if the threshold value is set to \( x \) seconds, any inter-arrival time duration equal to or greater than \( x \) seconds is referred to as a traffic gap.

Figure 1 depicts the definition of traffic gap. As soon as the time duration between two packets exceeds \( x \) seconds, i.e., the pre-defined threshold value, we call it a traffic gap. A traffic gap ends as soon as the next packet arrives. In this study, the terms traffic gap and OFF time are used interchangeably to refer to the same phenomenon.

![Fig. 1: Traffic Gap, ON time and Cycle time](image1)

![Fig. 2: Two state ON-OFF model](image2)
2 BACKGROUND Paper V

2.2 ON Time

We define ON time ($T_{on}$) as the time duration elapsed from the end of a traffic gap to the start of next subsequent traffic gap. Broadly, ON time refers to the normal connectivity when packets arrive without any disruption caused by the network, as shown in Figure 1. Note that the inter-arrival time exists between two packets during ON time, however, the duration inter-arrival time is below the pre-defined threshold value mentioned in the previous section.

2.3 Cycle Time

Cycle time ($T_{cycle}$) refers to the sum of a pair of two adjacent ON and OFF periods. It is a measure of the tendency of a network returning back to a certain state. In this paper, we use the term cycle time to express how soon a network returns back to the ON or the OFF state. In Figure 1, a cycle time is illustrated, which begins from the start of ON time and ends at the end of next traffic gap. Furthermore, a cycle time $T_{cycle}$ can be expressed as:

$$T_{cycle} = T_{on} + T_{off} \quad (1)$$

From the perspective of users, duration of a freeze or time waiting for the content, matters. Additionally, how frequently these disturbances repeat determine the satisfaction of users with a service. Thus, cycle time is a measure, which can used to monitor the frequency of such disturbances. For example, on average, the larger the $\frac{T_{off}}{T_{cycle}}$, the worse the performance from the user perspective.

2.4 ON-OFF model

A two state model summarizes the overall performance of a traffic stream from the user perspective. Figure 2 depicts a two state ON-OFF model. The time durations a network spends in ON and OFF states on each visit are represented by $T_{on}$ and $T_{off}$, respectively. The average ON and OFF times during a traffic stream are the expectations of the respective ON and OFF times, represented by $E(T_{on})$ and $E(T_{off})$, respectively.

Finally, the transition rates of moving from OFF to ON ($\lambda$) and ON to OFF ($\mu$)
can be estimated as:

$$\lambda = \frac{1}{E(T_{off})}$$  \hspace{1cm} (2)

$$\mu = \frac{1}{E(T_{on})}$$  \hspace{1cm} (3)

In this paper, we classify the two-state ON-OFF into two main categories: deterministic model and non-deterministic model. Briefly, the deterministic model requires the ON time ($T_{on}$) and the OFF time ($T_{off}$) to be fixed throughout a traffic stream. This implies that we have the definite information when a network will transit to the another state i.e. from ON to OFF state and vice versa. Conversely, a non-deterministic model requires that the values of $T_{on}$ and $T_{off}$ are randomly distributed around their respective mean values $E(T_{on})$ and $E[T_{off}]$.

3 Wavelet Transform

Wavelet transform provides an important means for multi-resolution analysis of a non-stationary signal or a time series [2] [9]. It means that the frequencies of a signal can not only be analyzed at a single timescale, but across multiple timescales. Thus, it allows localization of dominant frequencies at a certain timescale among several considered timescales, and hence, enables localization of timing information of changes occurring in a time series. On contrary, the other transforms, such as Fourier transform and its variants allow the analysis of periodicities (frequencies) in a signal at only a single resolution (timescale).

The property of multi-resolution analysis of Discrete Wavelet Transforms (DWT) is particularly useful for quality monitoring of traffic streams on mobile networks. The typical time-varying nature of mobile channels induce burstiness in the streams across various timescales, which appear in the form of ON and OFF times of varying durations [21]. Particularly, the behavior of traffic streams at small (sub-second) timescales on a mobile network can be drastically different from the behavior at large (multiple of seconds to minutes) timescales [14]. Consequently, these behaviors may produce very different impacts on the QoE. For example, burstiness at a sub-second timescale may produce completely different impact in the quality of a video streaming as compared to the transient outages followed by bursts at a multi-second timescale.
Similarly, the quality within one temporal locality of a video can be significantly different from the quality within another temporal locality. Therefore, network monitoring systems require methods to detect and classify events that particularly result in QoE degradation, based on the multi-resolution analysis of dominant frequencies at relevant timescales.

Given an original time series as input, DWT decomposes it into different components, corresponding to a set of frequencies at various timescales. There are different types of DWTs. The simplest one is the Haar wavelet transform [22], which fits well to the analysis of discrete time series corresponding to the network traffic streams. Therefore, in this study, we keep our focus on the Haar wavelet transform.

In Haar wavelet, the original time series (used as input) is successively decomposed into a set of scaling and detail coefficients. The scaling coefficients can be considered the outcome of low-pass filtering, while, detail coefficients result from the high-pass filtering of the input signal. We also refer to the scaling coefficients by $c$ coefficients, and detail coefficients by difference or $d$ coefficients.

### 3.1 Wavelet Coefficients

Let $y$ be a vector representing an original time series with $n$ data points, such that, $y = \{y_1, y_2, y_3, y_4, ..., y_n\}$. Let $n$ be the length of a time series, which must be a multiple of 2, i.e., $n = 2^N$. It means that the time series can be decomposed on a total of $N$ scales, starting from the finest scale $j = 0$, and incrementing up to the coarsest scale $j = N - 1$.

The scaling or $c$ coefficients on the finest scale $j = 0$ can be calculated by taking pairwise sums of adjacent but non-overlapping data points in the original time series, and dividing each coefficient by a normalization factor of $\sqrt{2}$. Conversely, detail or $d$ coefficients can be calculated by taking the pairwise difference of two non-overlapping data points in the time series and dividing each of them by $\sqrt{2}$. The equations for calculating scaling (Eq. 4) coefficients at the finest scale are the following:

$$c_{0,k} = \frac{1}{\sqrt{2}} \cdot (y_{2k} + y_{(2k-1)})$$
$$d_{0,k} = \frac{1}{\sqrt{2}} \cdot (y_{2k} - y_{(2k-1)})$$

(4)
Similarly, scaling and detail coefficients on the coarser scales, i.e., from scale $j = 1$ to $j = N - 1$ are calculated recursively using the following equations:

\[
  c_{j,k} = \frac{1}{\sqrt{2}} \cdot (c_{(j-1),2k} + c_{(j-1),2k-1}) \\
  d_{j,k} = \frac{1}{\sqrt{2}} \cdot (c_{(j-1),2k} - c_{(j-1),2k-1})
\]  

(5)

Consider an example illustrated by Figure 3. A time series consists of $n = 2^3 = 8$ data points, i.e., $y = \{5, 7, 0, 10, 6, 4, 0, 11\}$. Thus, there can be a total of $N = 3$ possible scales, i.e., $j = 0, 1, 2$. At the finest scale $j = 0$, scaling coefficients are calculated by summing up the two adjacent and non-overlapping pairs of data points, whereas, detail coefficients are calculated similarly by taking the differences instead. Successively, at the coarser scales, $j = 1$ and $j = 2$, scaling and detail coefficients are calculated by taking the pairwise sums and differences of scaling coefficients at their respective previous levels.

![Wavelet decomposition of a discrete timeseries](image)

**Fig. 3:** Wavelet decomposition of a discrete timeseries

The computational complexity of wavelets transformation of a time series is $O(n)$ where $n$ is the total number of data points in time series. In contrast, the complexity for the fast Fourier transform is $O(n \cdot \log_2(n))$. 

143
3.2 Energy of scaling and detail coefficients

The energy of scaling and detail coefficients is calculated by taking the sum of squares of $c$ and $d$ coefficients respectively at each scale [13]. The energy of scaling coefficients is similar to the second moment of a time series at a certain time scale. Similarly, the energy of detail coefficients implies variability and periodicity at a particular scale.

The energy of scaling coefficients at a certain scale $j$ is calculated using the Eq. 6, where $n_j$ is the total number of coefficients at scale $j$.

$$c \mu_j = \log_2 \left( \frac{1}{n_j} \sum_{k=1}^{n_j} c(j,k)^2 \right)$$

$$d \mu_j = \log_2 \left( \frac{1}{n_j} \sum_{k=1}^{n_j} d(j,k)^2 \right)$$

The energy function provides suitable representation by summarizing all the values of coefficients at a certain timescale to only one value. The one value per timescale will reduce the amount of information, which requires interpretation by the monitoring systems.

4 Methodology

This section briefly describes the methodology used in this paper to generate data used for the analysis. We developed a simulation tool to generate ON-OFF patterns of nominal durations. Thus, the output of simulation is in the form of a time series with ON and OFF phases. The ON phases contains a constant number of packet arrivals in each time slot, while, the OFF phases contains zero packet arrivals. The length of each time slot is fixed to 0.125 s.

We considered two different scenarios in the simulation: Deterministic and Non-deterministic scenario. In the deterministic scenario, the durations of ON and OFF periods are fixed, therefore, we generated only one time series for each combination of ON and OFF time duration. Section 5.1 provides further details. However, in the case of non-deterministic scenario, we used the random number generator. It accepts average ON time ($E[T_{on}]$) and average OFF time ($E[T_{off}]$) to generate geometrically distributed ON and OFF durations, respectively. We performed 40 tests with each
combination of $E[T_{on}]$ and $E[T_{off}]$ values. Follow Section 6 for further details.

Subsequently, we performed wavelet analysis on each time series. In the non-deterministic case, our analysis produced 40 values of energy functions $c_{\mu_j}$ and $d_{\mu_j}$ at each scale $j$. Thus, we used mean and confidence intervals of energy values $c_{\mu_j}$ and $d_{\mu_j}$ at each time scale, which we present in this paper.

5 Deterministic Traffic Gaps

We describe deterministic traffic gaps as the gaps of a fixed time duration repeating after a fixed interval of time throughout the life of a traffic stream. Consider the two state ON-OFF model explained previously in Section 2.4, where $T_{on}$ and $T_{off}$ are the times a network spends in ON and OFF states, respectively, on each visit. In the case of deterministic traffic gaps, we consider both $T_{on}$ and $T_{off}$ to be fixed throughout a traffic stream. Thus, there is no randomness involved in transition between the states. This case may not be a very realistic case, considering the practical behavior of network performance, which often involves varying behavior [2]. Nevertheless, our objective here is to show how wavelets interpret the traffic gaps, therefore, we start our analysis with the simplest cases in the form of fixed ON and OFF periods.

A traffic stream with deterministic traffic gaps can be characterized by static ON, OFF and cycle time values. Thus, if $T_{on} = a$ and $T_{off} = b$ in seconds, then a network always spends $a$ number of seconds in ON state and $b$ number of seconds in OFF state on each visit to the corresponding state. The time interval for which network is in ON state, the packets are supposed to arrive with a constant inter-arrival time between them. In the OFF state, there are no packet arrivals at the observation point.

In this study, we assume that a traffic gap or OFF time implies that a number of packets are delayed due to a bottleneck link or scheduling issues at the base station in mobile networks. Consequently, the packets stuck in a queue (buffer) at a certain node along the path. We assume that the size of the buffer at the intermediate node is long enough, such that, it never fills up completely and hence, the packets are not dropped from the queue. As soon as the packets are released, they arrive in the form of burst at the observation point.
5.1 Time series of deterministic packet arrivals

Consider a traffic stream with a constant number of packet arrivals. For example, let a source generate 1 packet every 0.125 s interval. Similarly, we count the number of arriving packets at the observation point every 0.125 s interval. Thus, during ON time, we find 1 packet in every 0.125 s slot. Similarly, when the network goes to OFF state, we find no packet in the following intervals of 0.125 s until network makes the transition back to the ON state. However, the traffic source meanwhile keeps on generating 1 packet every 0.125 s. These generated packets accumulate in a queue at a certain intermediate node on the network. As soon as network returns back to the ON state, all the accumulated packets arrive together in the form of a burst. For example, if a network goes in OFF state for 1 s, 8 packets accumulate in the queue. After OFF time, these 8 packets along with one additional packet of the next interval arrive at the observation point, making the number of arriving packets equal to 9. In the following intervals in the ON state, the normal behavior resumes and we find 1 packet every interval of 0.125 s for rest of the ON time duration.

Figures 4(a) and 4(b) present the number of packet arrivals for data transfers on network with fixed ON and OFF periods of network connectivity. In the multi-plot shown in Figure 4(a), duration of ON times are fixed to 1 s, 3 s, 7 s and 15 s in the

(a) ON times equal to or greater than OFF (b) OFF times equal to or greater than ON times.

**Fig. 4:** Number of packet arrivals in intervals of 0.125 s. Fixed $T_{on}$ and $T_{off}$ times.
four plots from the top to the bottom. The duration of OFF time is fixed to 1 s in all the four cases. Conversely, in Figure 4(b), durations of OFF times are fixed to 1 s, 3 s, 7 s and 15 s in the plots starting from the bottom to the top, and the ON time remains the same in all plots. In each case, the ON and the OFF intervals are thus pre-determined, and hence, repeat after fixed intervals of time. In the mentioned figures, there are mainly three types of transfers. First, the transfers with equal ON and OFF periods. Second, the transfers with longer ON periods and shorter OFF periods. Third, the transfers with OFF time durations exceeding ON time durations.

A peak in packet arrivals after each OFF period indicates the burst of packets delivered as soon as network returns back to the ON state. It indicates situations in which packets accumulate in a queue at a certain node within a network, and are released once the outage ends. The network connectivity thus restores and transmission of packets continues thereafter.

5.2 Wavelet analysis of deterministic traffic gaps

In this section, we report the wavelet analysis of time series with fixed gaps repeating after fixed interval of time shown in the Section packetarrivalsdeterministic. This case is the simplest case. It helps us describe the changes in wavelet coefficients with the changes in duration and frequency of ON and OFF periods in traffic flows.

Figure 5 shows the energy of difference coefficients calculated at timescales ranging from 0.125 s to 32 s. The energy of difference coefficients is calculated using Equation 5. The curves in figure show peak at timescales corresponding to the half of $T_{cycle}$ (cycle time). Similarly, each curve shows a dip at timescale equal to the corresponding cycle time. In the solid curve, the cycle time is equal to 2 s, therefore, the peak in energy occurs at the 1 s timescale ($\frac{T_{cycle}}{2}$) followed by a dip at 2 s timescale ($T_{cycle}$). Similarly, when the cycle time is equal to 16 s, the energy peaks at 8 s and then dips at 16 s. This shows that the dominant frequency in a trace can be represented by the average cycle time, and thus, wavelets express it in the form of a peak followed by a dip in the energy.

Timescales on the left side of peak (i.e., below $\frac{T_{cycle}}{2}$) are also noteworthy. These timescales present an important indication about the share of ON and OFF times within cycle times. In order to understand it, consider energy of difference coefficient for the transfers, in which, the share of OFF times is higher than the share of ON times within cycle times. If a curve on the left side of peak levels, it indicates that the
energy on lower timescale is high due to the greater burstiness stemming from longer OFF times. The number of packets accumulated in the queue are higher with higher OFF times. The bursts grow when the OFF times grow. All these packets clump together and arrive at receiver within a very small interval of time. Hence, it produces high variability at smaller timescales, which further increases the level of energy at the corresponding timescales. The higher the burstiness, the more horizontal the curve of energy function becomes on the left side of peak energy.

The duration of a particular network disturbance in the form of traffic gap or OFF time within a cycle time is useful in understanding the QoE perceived by the user. The higher the share of OFF time, the more the degradation in the QoE. For example, with a certain cycle time, the share of OFF time points to the amount of degradation in the QoE due to the waiting time faced by a user. Conversely, the ON time durations expresses the recovery in QoE level.

Subsequently, in order to acquire further understanding about the timescales over which the disturbances occur, we present the energy function of scaling coefficients. Figure 6 presents the curves with energy of scaling coefficients. The energy of scaling coefficients grow almost linearly with the increasing timescales as the share of OFF times reduces. The dent in energy curve appears over the timescales corresponding to the duration of OFF times. The energy of scaling coefficients appears to be levelling off at the timescale from 0.5 s to 1 s (when \( T_{on} \geq T_{off} \)). It is an indication of recurring OFF times of the duration 1 s. The dent in energy at 1 s timescale persists in all these curves. However, when the share of ON times increases, the curves show a constant growth across all timescales.

The curves of scaling coefficient energy for transfer with the OFF times exceeding the ON times become horizontally straight over certain timescales (0.125 s to 8 s), indicating the occurrence of OFF time durations on comparable timescales. As the duration of OFF time decreases, the horizontally flat part of the curves also reduces. It allows energy to increase with the increasing timescales.

The curves of scaling coefficient energy functions are suitable representation of the share of frequent OFF times within cycle time. Hence, monitoring and visualization of duration of transient outages can be expressed by energy function of scaling coefficients. The proportional increment in the energy with the increasing timescales represent cases with smooth communication without occasional abrupt outages over timescales of interest.
5.2 Wavelet analysis of deterministic traffic gaps

Fig. 5: Energy of Difference Coefficients. Fixed $T_{on}$ and $T_{off}$ times.

Fig. 6: Energy of Scaling coefficients. Fixed $T_{on}$ and $T_{off}$ times.
In Figure 7, we present wavelet energy of transfers that spend a fixed amount of time in the OFF state after spending random amount of time in the ON state. Once the network visits OFF state, it stays there for a duration of 1 s. The duration of time spent in the ON state is random and exponentially distributed with a certain mean duration. Due to the randomly distributed ON times, the cycle times also vary depending on the amount of duration network stays in ON state on every visit. We have taken 40 runs with different seeds to generate randomly distributed ON time durations.

Figure 7 shows the energy curves of mean difference coefficients (based on 40 runs) along with the confidence intervals. The global maxima, i.e. the peak, occurs at 0.5 s timescale. However, the peak is not as prominent as the one observed previously in Figure 5. The cycle times vary during a transfer, however, they remain above 1 s due to the 1 s of fixed OFF time. The ON times are randomly distributed, therefore, the cycle times vary over a range of timescales from around 1 s to the longer timescales.

Figure 8 presents energy of scaling coefficients for random ON and fixed OFF period transfers with the confidence intervals. The scaling coefficient curves appear to deviate at timescales corresponding to the duration of OFF periods. The energy levels
Wavelet analysis of deterministic traffic gaps

off strongly if the ON time to OFF time ratio becomes smaller. For example, compare
the solid (Mean ON time = 1 s, OFF time = 1 s) with the dotted dash curve (Mean
ON time = 15 s, OFF time = 1 s) at timescale of 1 s. The former starts to level off,
moving from higher to lower timescale between 0.125 s to 1 s timescales, while, the
latter continues almost the same trend throughout from 0.125 s to 32 s. The almost
15 times longer mean ON time duration describes this behavior.

In order to strengthen the understanding further about the quantification of net-
work performance using wavelets, we present another metric. This metric is composed
of the number of scaling coefficient values, which are equal to zero. We take the ratio
\( \gamma_j \) of the number of zero scaling coefficients \( m_j \) to the total number of scaling
coefficients \( n_j \) obtained at a certain scale \( j \). The following equation expresses the
metric:

\[
\gamma_j = \frac{m_j}{n_j}
\]

Figure 9 presents the share of zero scaling coefficients among the total number
of coefficients. The curves in figure show that the higher the ratio of zero scaling
coefficients to the total number of coefficients, the higher the number of empty slots
(i.e., zero packet arrivals) at the corresponding timescale. The solid curve shows that
there are around 40% of coefficients, which are equal to zero at timescale 0.125 s. It
shows that the sum of non-overlapping neighboring pairs are zero for 40% of the total
number of pairs. In other words, there are traffic gaps of at least 0.25 s duration for
40% of the time. The reduction in the value of this metric in Figure 9 for the other
curves indicates that the length of the ON time durations get longer, which further
increases total length of the transfer. However, the OFF times are fixed to 1 s, and
therefore, the ratio of zero to total number of coefficients decreases.

Typically, for the case, with equal ON and OFF time, the ratio of zero to total
coefficients at base timescale (i.e. 0.125 s) should be 50%. However, the ON times
are exponentially distributed and therefore, the share of traffic gaps in the transfer
is below 50% in the solid curve at 0.125 s timescale. It becomes 0 at 1 s timescale
showing that there are no empty slots above 1 s timescale. The metric is useful in
visualizing the maximum duration of traffic within a transfer and the timescale on
which a large number of traffic gaps are present.

The curves in Figure 9 are very handy as they convey several different aspects of
network performance and QoE. First, they provide information about the existence
Fig. 8: *Energy of Scaling coefficients. Exponentially distributed random $E(T_{on})$ and fixed $T_{off}$ times.*

Fig. 9: *Share of zero Scaling coefficients. Exponentially distributed random $E(T_{on})$ and fixed $T_{off}$ times.*
of the length of traffic gaps (OFF times) by exactly quantifying the gaps at several
timescales. Second, they approximate the longest OFF time within a transfer. The
timescale at which the curves level off and become zero points to the maximum length
of a gap within a transfer. Third, trend of the curves indicate the number of OFF times
of duration corresponding to the timescales. For example, the curve with $E(T_{on}) = T_{off}$
in Figure 9 shows that, among 40% traffic gaps, there are 10% of the gaps which are
at the timescale of 0.125 s and rest of the 30% gaps are above 0.125 s timescale. Thus,
the network monitoring systems can use this metric to quantify the share of OFF times
among cycle times in combination with the energy function curves of difference and
scaling coefficients.

6 Non-Deterministic Traffic gaps

This section discusses the wavelet-based monitoring of networks that spend random
time in both the ON and the OFF states. As described in the Section 2.4, $T_{on}$ and $T_{off}$
are the time duration spent in each of the ON and OFF states, respectively. Similarly,
we assume that $T_{on}$ and $T_{off}$ are exponentially distributed [23] around their respec-
tive mean durations of $E(T_{on})$ and $E(T_{off})$, respectively. Since, we count the number
of packet arrivals in every 0.125 s slot, therefore, we approximate the exponential
distribution with its discrete counter part, i.e., the geometric distribution.

Let’s assume that $E(T_{on})$ is 3 s and $E(T_{off})$ is 1 s. The average number of slots of
duration 0.125 s in ON state thus becomes 24. Similarly, the average number of 0.125 s
slots a network spends in the OFF state is equal to 8. Now we calculate transition
probabilities $\lambda$ and $\mu$ (as shown in the Equations 2 and 3), based on the number of
time slots in each state. In this example, the value $\lambda$ is 1/9, i.e. on average, a network
is expected to go from OFF to ON state every 9th slot. Similarly, the probability of
transition from the ON to the OFF state ($\mu$) is 1/25.

6.1 Time series of non-deterministic packet arrivals

Figure 10(a) presents the number of packet arrivals every 0.125 s interval. In Figure
10(a), we choose mean ON times of 1 s, 3 s, 7 s and 15 s. Similarly the mean OFF time
is 1 s. The duration a network spends in ON and OFF state on each visit is random and
follows geometrical distribution. Hence, the figure shows random traffic gaps followed
Fig. 10: Number of packet arrivals in intervals of 0.125 s. Exponentially distributed random $E(T_{on})$ and $E(T_{off})$ times.

by bursts depicting randomness seen in the performance of real networks. The impulse plot at the bottom is the most stable one with 16 times loner durations of ON times. However, the occasional traffic gaps occur during the transfer. These occasional bursts may result in small rarely occurring disturbances for the users. Conversely, the top most time plot is the most disturbed transfer toggling between ON and OFF periods of almost similar durations, resulting in frequent burst.

Conversely, 10(b) shows the time series with $E(T_{off}) \geq E(T_{on})$. For example, these are perhaps the types of transfers which users may not be able to tolerate due to $E(T_{off})$ frequently exceeding $E(T_{on})$. It means that the users may have to wait for the video freeze to end for longer duration of time as compared to the time duration for which they actually can enjoy the smooth video streaming, of course if the application-level buffer has not accumulated a very large amount of content.

6.2 Wavelet analysis of non-deterministic traffic gaps

Figure 11 shows curves of mean energy energy values with confidence intervals for difference coefficients corresponding to the time series shown in Figures 10(a) and 10(b). The results show that the energy is generally high and spread over a range of timescales. Particularly, from the sub-second up to 1 s timescales exhibit almost
similar but high energy, with peak occurring at 1 s. These curves are different from the ones shown in Figure 5, where ON and OFF times were fixed and deterministic. In this case, since, the traffic gaps are not fixed and spread over a range of timescales, the frequency at any one timescale is not clearly dominant in comparison to the other timescales. Particularly, at small timescales (0.125 s to 1 s), the frequencies are high. The reason for this behavior is the existence of significantly high number of traffic gaps at small timescales due to the exponential distribution with 1 s of mean duration. The bursts at small timescales raise energy at these timescales. Hence, by looking at these curves, one may safely assume that the duration of most recurring traffic gaps are between 0.125 s to 2 s timescale.

Figure 11 also exhibits energy of difference coefficients for the transfers shown in Figure 10(b). The energy curves are interesting manifestation of the traffic gaps. We observe that the peak of energy in all cases extends almost to the duration of OFF time. In the case of mean OFF time duration is 15 s, we observe that the energy is high up till 16 s timescale. Later, it starts to descend. Similarly, in other cases, the energy remains high but stable up to the duration of OFF time.

To understand the difference in values of scaling coefficients in the case of non-deterministic traffic gaps, we present energy of scaling coefficients in Figure 12. The curves clearly start to deviate at the timescale corresponding to the mean duration of
Fig. 12: Energy of Scaling coefficients. Exponentially distributed random $E(T_{on})$ and $E(T_{off})$ times.

Fig. 13: Share of zero Scaling coefficients. Exponentially distributed random $E(T_{on})$ and $E(T_{off})$ times.
traffic gaps, i.e., 1 s. Hence, there is no significant difference between the deterministic and non-deterministic cases, when it comes to the energy of scaling coefficients. This common observation implies that when the energy of scaling coefficients starts to level off at a certain timescale, it indicates that the traffic gaps are recurring at the corresponding timescale.

To quantify the traffic gaps of durations corresponding to the given timescales, we present the ratio of zero to total number of scaling coefficients (see Figure 13). With 1 s mean OFF time duration, we observe that the traffic gaps, as long as 4 s, exist in the transfers, as shown by the green curve. However, most of the traffic gaps are of duration less than 2 s. This observation is in conformance to the exponential distribution with mean 1 s.

Summarising the above findings:

- We presented three wavelet-based metrics: energy of different/detail coefficients ($\mu_j^d$), energy of scaling coefficients ($\mu_j^c$) and the ratio of zero to total number of coefficients ($\gamma_j$).
- The energy peak of difference coefficients extends from the base timescales up to the timescales almost around $E(T_{\text{off}})$.
- The cycle time ($T_{\text{cycle}}$) can be quantified by the timescale corresponding to the descending energy values of difference coefficients.
- The more the share of ON times in a cycle time, the more linearly the energy of scaling coefficients grows across short to long timescales.
- The share of zero scaling coefficients at each time scale ($\gamma_j$) can be used to approximate the percentage of traffic gap durations of the corresponding time scale ($j$).

7 Discussion

In this work, we kept our focus on non-lossy links, i.e., we presented results with regards to the situation in which packets are not dropped from the queue during the OFF times. However, in reality, the losses may occur in certain situations. In scenarios, when source sends traffic at a constant rate, the packet loss may reduce burstiness. Specifically, the peak of packet arrivals after each OFF period, which we observed in our results, will reduce. However, the exact reduction in the peak depends
on the amount of packet loss. In such a context, the energy of difference coefficients is expected to reduce across several timescales. Particularly, the energy is expected to reduce at the lower time scales, i.e., from the base time scale up to the time scale, which exhibits peak in the wavelet energy.

Moreover, another scenario involves a situation in which a traffic source generates packets at a variable rate. In this scenario, the time scales that we consider (starting from the base time scale) may exhibit short to long bursts. Consequently, these bursts will also contribute to the change in behaviours of wavelet coefficient values. To mitigate such situations and quantify the contribution of a traffic source(s), the traffic can be captured at the inlet and outlet of monitored segment of network. Thus, it will also involve estimation wavelet energy values at both locations, which need to compared further to understand the behaviour of network.

Finally, the understanding of traffic behaviour at different timescales using wavelets greatly improves the capability of managing networks, beyond the monitoring of network traffic gap detection. Particularly, the management systems can efficiently share and manage resources among several users by utilising the information w.r.t traffic behaviour at different timescales.

8 Conclusions and outlook

In this paper, we evaluated the ability of wavelets to analyze streams and detect timescales at which periodicities are dominant. With multi-timescale view, the wavelets open new opportunities in the QoE monitoring scenario. The user opinions typically evolve over time and are dependent on the events repeating throughout the lifetime of a session. We showed that the wavelet energy functions provide a quick summary and visualization of timescales at which certain problems repeat. Specifically, the energy of detail coefficients surge at the timescale on which there is high variation in the performance. Similarly, the scaling coefficients assist in the quantification of the length of traffic gaps within a stream. By comparing these timescales with the timescales on which the users typically react, an efficient QoE monitoring mechanism can be implemented practically on network and application layers. Moreover, the practical application of wavelets can be modified in the form of different combination of data streams and configurations to suit a given scenario.

The future studies may extend the cases presented in this paper. Particularly, the
studies should evaluate wavelets quantitatively under packet loss and bursty source scenarios. The subjective opinions of users should be modelled against the wavelet coefficients. We envision a holistic wavelet-based QoE assessment mechanism, based on the passive monitoring of traffic streams. We believe that this type of passive mechanisms will allow continuous QoE assessment over time, and thus, strengthen the monitoring systems particularly in the situations when deep packet inspection may not be possible due to the emerging encrypted traffic on network.

References

REFERENCES


PAPER VI

INFERRING USER-PERCEIVED PERFORMANCE OF NETWORK BY MONITORING TCP INTERRUPTIONS

Published in the journal of
Network Protocols and Algorithms
Inferring User-Perceived Performance of Network by Monitoring TCP Interruptions

Junaid Shaikh, Markus Fiedler, Patrik Arlos, Tahir Minhas, Denis Collange*

Blekinge Institute of Technology, Karlskrona, Sweden,
(junaid.junaid, markus.fiedler, patrik.arlos,tahir.nawaz.minhas)@bth.se

*Orange Labs, Sophia Antipolis, France,
denis.collange@orange.com

Abstract – The fluctuating performance of wireless and mobile networks has triggered the need for smart algorithms to assess the user perception, resulting from the quality of network services. While efforts have been done to model the user experience resulting from the network performance, there is still the need for practical methods to assess the user-perceived performance, in the real environment. In this work, we present a set of criteria to observe the user behavior on the Web, passively from the network-level. The criteria are based on the monitoring of TCP control flags and HTTP requests. Thus, information about user actions performed in the web browser can be inferred by monitoring the TCP termination flags and by keeping track of the HTTP requests. Along the way, we also present some anomalies observed in the TCP connection termination process, which may result in performance degradation of Web transfers.
1 Introduction

In the recent years, the Internet has witnessed the mushrooming of networks and applications. Among all the applications, the Web application is the most dominant one. The popularity of Web is further fueled by the migration of video on the Web. According to study [1], Web traffic accounts for the major part of the traffic volume on the Internet.

With the increasing usage of the Internet, the expectations of users are also evolving. While the computational power of the devices, intelligence of applications and speed of networks are increasing with time, following Moore’s law, expectations of users are following “the More’s law” [2]: Users want more in less time. They are becoming increasingly strict and intolerant about the quality of network and application services. This is because the users now have to rely on the Internet for their everyday tasks. Due to these growing expectations, the margin of error is getting smaller and the network protocols and algorithms need to perform smartly and accurately.

For an Internet Service Provider (ISP), it is extremely important to monitor and keep track of the service quality as perceived by the user. There are several competitors in the market and a user may easily switch to another service provider as a result of dissatisfaction, taking several other users with her. Hence, there is need of a mechanism in order to learn about the user experience over time, in order to provide better services. However, being certain about user experience is a complex task for the following reasons.

The quality perceived by the users is mainly affected by the several network-dependent and application-specific factors. However, there are many other factors that may influence the quality as perceived by the users such as, the prior experiences, expectations, the context of use, etc. Users from different geographical backgrounds may have different expectations regarding the service quality, based on previous experiences. A user surfing at work could probably be more intolerant about the bad quality of service, as compared to a user using the service on leisure.

Monitoring of the user-perceived performance has two main requirements. First, a model is required, that takes into account all the parameters that influence user-perceived performance of a service; second, a method, which estimates the performance by measuring the above parameters in a fast and scalable fashion [3]. Unfortunately, it is not so easy to measure all these parameters online in real-time, from the network-level.
For ISPs, it is easier to measure the network-dependent factors than asking the each user about her experience subjectively. However, retrieving the application-specific and user-related factors is not simple. It is because ISPs do not have control on the user-end devices. Therefore, conducting subjective experiments in a lab environment, with real users, has been a common practice to model the user-perceived performance. Although, they provide control at the user-end, subjective lab experiments have proven to be away from reality [4].

Another method of estimating user-perceived performance is to investigate those parameters that represent user actions on the application level and provide indications about user behavior. Utility of a service could be one of the ways to have indications about the interest of users in a service [5]. In [6], user session volumes were shown as the indicator of user behavior and were compared against the service performance. While session volumes may provide an overall picture of the interest of all the users in a network service, it is not easy to infer the interest of a single user. Second, it is also difficult to deduce thresholds on the performance, beyond which a user stops using the service.

Monitoring of the Transmission Control Protocol (TCP) connection terminations on the Web is one of the ways to monitor user-perceived performance degradation of a service [7]. Normally, users press the Stop or Reload button in the web browser to abort an on-going transfer, when it is much slower than their expectations. These interruptions result in early termination of the TCP connections with a Reset (RST) flag from the client side. These RST flags can be monitored passively on the network-level to observe the user behavior.

Before considering the TCP RST flags as being the indication of user behavior, it is important to make sure that a TCP RST flag is generated only when the user interrupts a TCP connection. Therefore, it calls for a detailed classification of TCP end flags, which allows identification of those TCP RST flags that are generated as a result of user interruptions.

In [8], TCP connections are, based on the type of termination, classified as normal connections, abnormal connections, unfinished connections and interrupted connections.

In this work, we have performed a systematic study to show in detail the sequence of termination flags, in order to identify the transfers aborted by the users. The sequence of these termination flags occurred as a result of different actions as listed in
Table 1, performed in the web browser. Hence, monitoring and classification of the termination type in a real scenario may provide indications about a user’s behavior. To ensure a fair and representative comparison, we have conducted a number of controlled experiments with various web browsers. These experiments are done on both smart phones and laptops. The results of experiments on smart phones were presented in [?]. A more detailed study along with a set of termination criteria is presented in this paper.

Network operators and Web service providers can use this knowledge to passively monitor the behavior of users over time, and manage their resources accordingly to guarantee high-quality user experience. The research community working on network Quality of Experience can use this study to validate it against the subjective experiments with real users. Finally, Web users can also use results to choose web browsers that are operating according to the rules defined by the standards.

Hence, our contribution in this paper is two-fold. First, we present the different sequences of the TCP termination flags observed with a number of web browsers. With the help of these results, cooperation between web browsers and Web servers could be improved to raise the performance of Web transfers. Second, we develop a set of criteria, which could be used to identify the user action performed in the web browser. It may help service providers to monitor the user-perceived performance.

The remainder of this paper is organized as follows. Section 2 provides some background information on the Web transfers and the TCP protocol. Section 3 describes the related work. Section 4 gives the methodology used. Section 5 shows the results and discussion on the sequences of the TCP termination flags. Section 7 presents results from passive measurements. Section 8 proposes a set of user-interruption criteria. Section 9 concludes the paper.

2 TCP and Web transfer

A Web transfer consists of the request of one or more objects from the client to the server and the transfer of requested objects from the server to the client. Figure 1 depicts a Web session between a pair of the client and the server machines.

A Web session is a combination of one or more Web transfers. A Web transfer
starts when a user requests for Webpage or a file. A Web page may consist of multiple embedded objects. Each embedded object is retrieved after the client-side web browser requests for the respective object, automatically. The transfer of each object is shown as the ON time in the Figure 1. Two objects in a Web transfer are separated by the active OFF time; the time taken by the client-side web browser to launch the next request automatically, after the transfer of the previous object.

Two subsequent Web transfers in a session are separated by the inactive OFF time, which is also called the user think time. This is the time taken by the user before launching the request for the next page or a file from the same server.

![Fig. 1: ON-OFF model for Web transfer](image)

The Web traffic is carried by the TCP on the transport layer, hence making TCP the most widely used protocol on the Internet for almost two decades. A TCP connection goes through several states from the connection establishment to the data transfer.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uninterrupted</td>
<td>User allows the page to load completely</td>
</tr>
<tr>
<td>Kill-browser</td>
<td>The user kills the web browser before the page has been loaded completely</td>
</tr>
<tr>
<td>Stop/Reload</td>
<td>The user presses the stop or reload button before the completion of transfer</td>
</tr>
<tr>
<td>Link-follow</td>
<td>The user clicks another link or follows a bookmark before the completion of previous transfer</td>
</tr>
</tbody>
</table>
TCP is a reliable stream-oriented protocol [10]. Before the transfer of the stream of bytes, a virtual connection is established. After the connection establishment between two TCPS, a request is made by the client to the server, followed by data transfer from the server. After the end of the data transfer, the connection is closed by going through a proper termination handshake.

TCP connection termination handshake employs the control field in the TCP header to flag the end of a connection. To signal the end of the connection, a segment is sent from either side, with the finish (FIN) flag set in the control field of the TCP header. The other side then responds with a FIN segment to confirm the receipt of the FIN segment. This handshake confirms that the data transfer is completed, and the connection could be closed. Sometimes, the FIN segments also contain last chunk of data in it.

To signal the error conditions, a segment with reset (RST) flag is sent. The RST segment can be sent from one of the sides to deny a connection, if a connection was requested to a nonexistent TCP port. It is also sent when one of the sides aborts an existing TCP to signal an abnormal situation.

Figure 2 illustrates a TCP flow carrying a Web transfer. It starts with a SYN handshake between the client and the server. The client then requests for the file with a HTTP GET request. If the requested file is available on the server, then the server responds with the file in the form of stream of data bytes on the TCP level. It is shown by the TCP flow carrying DATA for a Web transfer in Figure 2. The client acknowledges one or more data segments from the server with an ACK segment. Once the transfer of requested file is completed, the client may request for another file with another HTTP GET request in the same TCP connection, if the client and the server are both supporting persistent TCP connections. Once all the requests from the client are served, the client or server then starts the connection termination handshake procedure, which is shown by FIN segment. The other end then responds with a FIN to terminate the connection.

3 Related Work

Evaluation of the user-perceived performance of network services is a complicated task. User perception is very much subjective, which vary heavily from one person
to the another. There have been efforts made in order to evaluate and model the user perception. Many of these efforts are limited to the user perception of voice and video applications [3] [11] [12]. Literature on QoE shows that it has either been drawn as qualitative nature subjective evaluation or quantified as a function of the QoS parameters.

The most common way to understand the user perception of a service performance is to conduct large number of expensive experiments in a lab environment. Many works reported these subjective experiments. The problem with these experiments is that they do not represent user perception that arises in the real environment. Users are too conscious about the performance in the lab environment than the real environment. They do not have freedom to do their tasks in natural way [4]. Secondly, the numerical scales on which the ratings are taken may not map well with the real emotions of the users.

In [13], a framework is proposed to capture users’ perception while they are using network applications. It requires a subject to click a key whenever she feels dissatisfied with the service. While their framework could be useful for the test environment, it is not easy to have their tool installed on user machines in the operational environment. In [13], authors presented their results on the data, which was collected by an end-host data collection tool. It was concluded that it is very much challenging to get the user feedback over time, particularly when the performance of the network degrades. Authors of [14] collected the user-centric data by a similar tool. The tool allows users to report their irritation with the help of the tool.

Most of the above work required the use of additional tools for the self-reporting of opinions given by the users. We do not find many works where existing infrastructure is used to infer the user perception. Another major point we observe in the literature is that the most of work on user experience has been directed towards the improvement of a particular application rather than the network services. In [16], a wavelet-based criterion is presented for identifying the user-perceived problems passively, based on the link-level measurements.

This work presents a set of criteria to infer the user perception from the client-side TCP RST flags. In [17] [18] [19], it is shown that the TCP RSTs are generated due to several causes. Therefore, it may not be appropriate to consider TCP RST flags as the straightforward indication of user perception. In [20], authors presented a heuristic to identify the TCP RSTs generated as a result of the interruption from the client. We will present these criteria and its shortcomings in the Section 8.
In [13], a framework is proposed to capture users’ perception while they are using network applications. It requires a subject to click a key whenever she feels dissatisfied with the service. While their framework could be useful for the test environment, it is not easy to have their tool installed on user machines in the operational environment. In [14], authors presented their results on the data, which was collected by an end-host data collection tool. It was concluded that it is very much challenging to get the user feedback over time, particularly when the performance of the network degrades. Authors of [15] collected the user-centric data by a similar tool. The tool allows users to report their irritation with the help of the tool.

Most of the above work required the use of additional tools for the self-reporting of opinions given by the users. We do not find many works where existing infrastructure is used to infer the user perception. Another major point we observe in the literature is that the most of work on user experience has been directed towards the improvement of a particular application rather than the network services. In [16], a wavelet-based criterion is presented for identifying the user-perceived problems passively, based on the link-level measurements.

This work presents a set of criteria to infer the user perception from the client-side
TCP RST flags. In [17] [18] [19], it is shown that the TCP RSTs are generated due to several causes. Therefore, it may not be appropriate to consider TCP RST flags as the straightforward indication of user perception. In [20], authors presented a heuristic to identify the TCP RSTs generated as a result of the interruption from the client. We will present these criteria and its shortcomings in the Section 8.

4 Methodology

We conducted a set of active tests to observe the sequence of TCP termination flags exchanged in both directions. To execute the tests, we established an isolated environment. These tests were conducted by accessing a Webpage on a smart phone or a laptop. The Webpage was located on a local Web server.

Two types of tests were performed: Uninterrupted and interrupted. In the uninterrupted tests, the user issues a Webpage request and then allows the transfer of the Webpage to finish completely. In the interrupted tests, the user aborts an ongoing transfer of Webpage by performing some action in the web browser. The user action could be either pressing the Stop or the Reload button, exiting the web browser or clicking a hyperlink on the Webpage. These actions are further mentioned in Table 1.

In order to study the impact of the content type, three Web pages were developed. One Webpage had simple text, the second one had an image and the third one had a flash video, played in a shockwave player on the Webpage. Since the results of the tests with text and image Web pages were almost similar to each other, we only present the results related to text and to video in the remainder of this paper.

Moreover, tests were first performed on three popular mobile platforms: Windows 6.5 (HTC HD2), Android 2.2 (HTC Desire HD) and Symbian 3.0 (Nokia N8). Built-in web browsers were used on each of these platforms, as external browsers were not supporting the video content. The web browser used by Windows 6.5 is Microsoft Internet Explorer 6.0. The user agent string in the HTML header reports AndroidÔs web browser as Mobile Safari and Symbian’s web browser as Browser NG, which is used on the Nokia mobile phones. On Android and Symbian platforms, the built-in web browsers use Webkit as the HTML rendering engine, which is an open-source web browser engine [21]. Each of the tests on mobile platforms was conducted with 40 repetitions.

Subsequently, other popular web browsers were also tested. Tests were performed
on a laptop equipped with Windows XP operating system. Four Web browsers were tested: Internet Explorer 8, Firefox 3.6, Google Chrome 4.1 and Opera 10.51. All of these web browsers support persistent TCP connections.

Finally, passive measurements were performed on an operational network of an ISP with real users. We identified different browsers from the traffic traces with the help of their user agent strings. After identifying the Web browsers, we carried out further analysis to extract their TCP terminations connection sequences.

## 5 Active tests

Table 3 summarizes the different termination types seen from all the experiments. The type of termination here refers to the sequence of terminating flags that were seen at the end of a TCP connection. Five different types of terminations are observed. Sequences of these termination flags occurred as a result of different actions as listed in 1, performed in the web browser.

<table>
<thead>
<tr>
<th>Termination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_s F_c R_c$</td>
<td>A FIN from the server followed by a FIN and then one or more RSTs from the client</td>
</tr>
<tr>
<td>$R_c F_c R_s$</td>
<td>One or more RSTs from the client followed by a FIN from the client and a RST from the server</td>
</tr>
<tr>
<td>$F_s F_c$</td>
<td>A FIN from the server followed by a FIN the client</td>
</tr>
<tr>
<td>$F_c R_c$</td>
<td>A FIN from the client followed by one or more RSTs from the client</td>
</tr>
<tr>
<td>$R_c$</td>
<td>One or more RSTs from the client</td>
</tr>
</tbody>
</table>
5.1 Uninterrupted transfers

The bar charts in Figures 3–8 highlight the number of each terminating sequence observed as a consequence of different user actions performed in different web browsers. On Symbian and Android platforms, all the TCP connections ended with a proper FIN handshake. After the data transfer, a FIN from the server is sent which is followed by a FIN from the client-side to end the connection. This type of termination follows the rules as described by the standards [10].

**Client-side RST flag:** On the Windows platform, however, the text-based Web-
page transfer is finished with a FIN from server, followed by a FIN and then a RST flag from the client. This RST flag appears to be a reaction of the client to the ACK received from the server, which triggers the client to immediately shutdown the connection by sending a RST flag. This behavior is found to be consistent in all the transfers.

**Multiple TCP connections per transfer:** Subsequently, when the video-based Webpage is downloaded from the Windows platform, there is another interesting pattern seen in the connection termination process. After receiving the base file, the client makes a GET request for the video player. It then immediately terminates the
5.1 Uninterrupted transfers

Fig. 7: Termination flags on Windows with text content

Fig. 8: Termination flags on Windows with video content

collection with a RST flag and initiates the new connection with a SYN handshake. The GET request for the previous file is thus repeated once again and then the video is played in the web browser. The second connection is terminated similarly as was observed in the case of text-based Webpage. Hence, two connections are opened for playing video in the web browser. The connection establishment procedure creates extra overhead, which affects badly the overall speed of the transfer. The TCP connection also goes into the slow start phase once again. The client-side software should avoid this kind of behavior as the opening of multiple TCP connections per transfer
may degrade the performance of the transfer.

5.2 Interrupted Transfers

Interrupted transfers are those in which a user aborts an on-going transfer by manually performing any of the three actions (before the end of the download) in web browser: Pressing the stop or reload button, exiting the browser or clicking a hyperlink on the Webpage. The results in Figures 3–8 illustrate that the connection termination pattern is similar when the interruptions are made from Android and Windows platform, while it is slightly different in the case of the Symbian platform.

Server-side RST flag: While using the Symbian platform, a large ratio of TCP connections were terminated with one or more RST flags from the client, followed by a FIN flag from the client and then a RST flag from server. The reason why the server responded with a RST flag is that when it received a RST flag from client, it assumed the connection was already closed and therefore, when it received an additional packet from client (containing a FIN flag) on the same port, it responded with a RST flag to once again signal the end of the connection.

Retransmissions: In a few interrupted transfers on Symbian and Windows, and in the majority of transfers on Android, connections were terminated with a FIN flag followed by one or more RST flags from the client. By looking at the interrupted traffic traces, we found out that, when the client starts the termination process with a FIN flag, then the server responds with the retransmission of previous unacknowledged segments. Upon receiving the retransmitted segments the client tears down the connection by sending one or more RST flags. This kind of anomaly may result in the wrong estimation of loss rates on the network. On the Windows platform, the majority of the connections were terminated with one or more RST flags from the client without any FIN flag. Indication of user-interrupted transfers: Generally, from all the above observations of interrupted connections, one thing is common that at least one RST flag from the client-side is seen regardless of the platform. The other important evidence about the user-generated interruption is that more than one consecutive RST flags were seen in most of the cases as soon as the user performs an interruption in the web browser.

Indication of transfers not interrupted by the user: On the other hand, a single RST flag is seen if the transfer is not interrupted by user, on which the RST flag is sent automatically by the client-side software.
6 Active tests with other popular web browsers

To test other popular web browsers, we further continued our tests on a laptop equipped with the Windows XP operating system. Tests were performed with four web browsers: Internet Explorer, Firefox, Opera and Google Chrome. Figure 9 presents the total number of RST flags observed from all our tests. This figure gives an overall picture of the web browser generating the highest number of RST flags.

Figure 9 confirms the results we retrieved from mobile web browsers. The Internet Explorer web browser generates more TCP RST flags than any other web browser. The RST flags on Internet Explorer are not only generated when the user aborts a transfer, but also in the cases when a transfer is not aborted by the user. Another observation we got from the experiments is that, the generation of TCP flags is not dependent on the operating system, but on the web browser. We performed these tests on the same operating system and we observed different behavior in terms of TCP termination flags.

**Proper FIN handshake:** For uninterrupted tests, the connections were terminated with proper FIN handshake in the case of Firefox and Opera web browsers. The server initiates the FIN handshake by sending a segment with FIN flag after the completion of data transfer. The client then responds with a FIN/ACK segment. It tears down the connection along with the acknowledgement of the last segment of data from the server.

**Termination initiation from the client side:** In the case of Google Chrome web browser, connections termination starts with a FIN from the client instead of a FIN from the server for uninterrupted tests. After the data transfer, the client initiates termination handshake, by sending a FIN segment. The server then responds with a FIN flag. Hence, the client does not wait for the server time-out but starts tearing down the connection proactively.

**RST flag from the client side:** The Internet Explorer replies with a RST flag instead of a FIN flag, after receiving a FIN flag from the server. After the data transfer, the server sends a FIN flag, and the client then responds with a RST flag. This behavior indicates the abnormal condition according to the TCP standards. These results indicate that the Internet Explorer does not seem to follow the standards. These tests with the Internet Explorer also confirm our observations we discussed in the case of mobile web browsers. The Internet Explorer in the case of video transfer opens two TCP connections. The first connection is terminated as soon as the video
player is requested. The request for the video player is then repeated in the second connection. Due to such behavior of Internet Explorer, we observe a larger number of RST flags in the case of Internet Explorer as compared to other web browsers as displayed in Figure 9.

**Interrupted tests:** In the case of interrupted tests, we observe a similar behavior in the case of all web browsers. The client terminates a connection with a RST flag as soon as the user aborts a connection in the middle of the transfer. If the client still receives a data segment from the server, after sending a RST flag, then it repeatedly sends RST flags to enforce the termination of the connection.

![RST flags](image)

Fig. 9: *Total number of RST flags*

7 Passive measurements on Operational Network

In this section we describe the connection terminations that we observed on the operational network of an ISP. We only present the web browsers that we considered in our active tests, i.e. Internet Explorer, Firefox, Google Chrome and Opera. We observed a large variety of connection terminations, i.e. the connections with different sequences of the termination flags. However, we have listed the most common termination flag sequences with each of the web browsers, which account for more than 75% of the connections.

**The most common termination type:** The server sends a FIN flag after transferring the data, which is followed by a FIN flag from the client side. Generally, this is the most common termination flag sequence observed on the TCP connections.
Transfers on Internet Explorer: The most common termination flag sequence observed with the Internet Explorer is the following: The server sends the data, which is immediately followed by one or more FINs from the server. The client then responds with one or more RST flags to tear down the connection. This is quite similar to what we observed in our active tests. There are also significant numbers of connections that are terminated with one or more RSTs from the client side. We infer that these connections are the mix of both terminated by the client-side web browser (in the case of video transfers), as well as transfers to abort a transfer.

Connection terminations with Firefox and Google Chrome: We see similar termination flag sequences observed with Firefox and Google Chrome. The majority of the connections are terminated with a FIN from the server, followed by a FIN from the client. However, in many cases, the server sends a RST flag after sending a FIN to flag the end of the connection. In these cases, we do not observe any termination flag from the client side.

Connection termination initiation by the client: There is also a large number of cases in which the connection termination handshake starts with a FIN flag from the client side followed by a FIN flag from the server side. This is the most common in the case of Opera and second most common in the cases of Firefox and Google Chrome. The client in such cases detects the end of the data transfer and hence, it sends a FIN immediately along with an ACK of last data segments from the server.

8 The user–interruption criteria

In [20], the authors proposed a user-interruption criterion. This criterion is based on a heuristic to determine those connections, which are interrupted by the users before a transfer is completed. To test this criterion, we applied it on our collected traces with the help of Tstat [22]. Tstat is a TCP statistics and analysis tool that implements the user interruption criterion. We will first explain this criterion and then discuss the outcomes that we got after the application of this criterion on our traces.

According to the criterion mentioned in [20], a connection is terminated by the client if the server sent data but didn’t send a FIN or an RST segment, and the client...
<table>
<thead>
<tr>
<th>Web browser</th>
<th>Most common terminations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>$F_s R_c$</td>
<td>One or more FINs from the server followed by one or more RSTs from the client</td>
</tr>
<tr>
<td>Explorer</td>
<td>$F_s F_c$</td>
<td>One or more FINs from the server followed by one or more FINs from the client</td>
</tr>
<tr>
<td>Firefox</td>
<td>$F_s F_c$</td>
<td>One or more FINs from the server followed by one or more FINs from the client</td>
</tr>
<tr>
<td></td>
<td>$F_c F_s$</td>
<td>One or more FINs from the client followed by one or more FINs from the server</td>
</tr>
<tr>
<td></td>
<td>$F_s R_s$</td>
<td>One or more FINs from the server followed by one or more RSTs from the server</td>
</tr>
<tr>
<td>Google</td>
<td>$F_s F_c$</td>
<td>One or more FINs from the server followed by one or more FINs from the client</td>
</tr>
<tr>
<td>Chrome</td>
<td>$F_s F_c$</td>
<td>One or more FINs from the server followed by one or more FINs from the client</td>
</tr>
<tr>
<td></td>
<td>$F_c F_s$</td>
<td>One or more FINs from the client followed by one or more FINs from the server</td>
</tr>
<tr>
<td></td>
<td>$F_s R_s$</td>
<td>One or more FINs from the server followed by one or more RSTs from the server</td>
</tr>
<tr>
<td>Opera</td>
<td>$F_c F_s$</td>
<td>One or more FINs from the client followed by one or more FINs from the server</td>
</tr>
<tr>
<td></td>
<td>$F_s F_c$</td>
<td>One or more FINs from the server followed by one or more FINs from the client</td>
</tr>
<tr>
<td></td>
<td>$F_c F_s R_s$</td>
<td>One or more FINs from the client followed by one or more FINs from the server followed by one or more RSTs from the server</td>
</tr>
</tbody>
</table>
sent an RST segment. These connections were named as eligible connections, and are expressed by:

\[
Eligible := \neg(FIN_s \lor RST_s) \land DATA_s \land RST_c
\]  

(1)

Where \(FIN_s\) is a FIN flag from the server and \(RST_s\) is the RST flag from the server. \(DATA_s\) is the data transfer from the server and \(RST_c\) is the RST flag from the client. \(\neg\) represents “NOT”, \(\lor\) represents “OR” and \(\land\) represents “AND” as logical signs. Hence, the equation says: “A connection is eligible, if NO FIN OR RST from the server is seen, AND DATA from the server AND the RST flag from the client is observed.”

Although \(Eligible\) connections represent those terminated by the client, they do not tell whether the connection is terminated by the user or not. The \(Eligible\) connections could be terminated by the client-side software when the data is already transferred, and the server is idle, waiting for the time-out. In order to identify the connections interrupted by the users during a data transfer, the authors also consider the connection termination time. The interruption criterion is thus expressed by:

\[
Interrupted := Eligible \land \frac{t_{\text{gap}}}{\alpha \cdot \mu_{\text{RTT}} + \beta \cdot \sigma_{\text{RTT}}} \leq 1
\]  

(2)

Where \(t_{\text{gap}}\) is the time elapsed between the last data segment from the server and the actual flow end. \(\mu_{\text{RTT}}\) and \(\sigma_{\text{RTT}}\) are the mean and the standard deviation of the RTT per connection, respectively. Hence, if a connection is \(Eligible\) and \(t_{\text{gap}}\) is less than one RTT time, then the connection is said to be interrupted by the user.

We executed Tstat on our interrupted and uninterrupted transfers. We found out that the user interruption criterion was not working accurately for those video transfers, in which Internet Explorer was used as client-side Web browser.

While this interruption criterion determines the connections interrupted within \(t_{\text{gap}}\), it does not identify the user action performed in the Web browser that resulted in the interruption of ongoing transfer. The identification of the user action in the Web browser is also important to know, as there could be different motivation behind each user action. For example, users usually press the stop or reload button when they are annoyed. However, they might also follow a link before the completion of a page when they have already seen enough information on the page, which may not be the result of anger or dissatisfaction.
To address the aforementioned shortcomings, we propose a set of criteria. The set of criteria are specified by a finite state machine diagram in Figure 10. There are two types of transitions shown in the diagram in Figure 10. One is the TCP flow transition triggered mainly by the TCP control flags. It is shown with the solid black arrows in the diagram. Another type of transition is the HTTP request transition, which is initiated each time by the arrival of a new HTTP request from the client side. HTTP request transitions are represented with dotted arrows in the diagram. Additionally, there is one more transition shown in the diagram, which takes both TCP and HTTP into the idle state. This transition is shown by dashed arrow. The machine comes out of this state when the user requests for the next web page from the same Web server. We will first describe the TCP flow states in detail, followed by the description of the HTTP transitions.

**Fig. 10: State diagram of a Web transfer**

**State 0. Start:** The state machine starts with this state as soon as a new request...
from the user is received. It refers to the user action of opening a new URL, clicking a link on a page or starting a completely new Web session.

**State 1. Handshake:** When the user requests for a Web page, the handshake process for the connection establishment starts. The Client-side TCP initiates this handshake with a SYN flag, shown by the transition number 1a in the Figure 10.

**State 2. Connection established/data transfer:** The server responds to the client with a SYN/ACK shown by the transition 2a, which is further acknowledged by the client, and the connection is established. The first HTTP transition is then made with the request for the base file $H_B$ of the page, represented by the dotted arrow 2b. After the connection establishment and $H_B$ request from the client, the data transfer starts. TCP and HTTP both stay in this state until a termination flag (FIN or RST) is seen from any side (client or server). During or after the data transfer, there could be any of the following three more events, which result in transition out of state 2:

**State 3. Eligible:** If a RST flag is seen from the client before any FIN or RST from the server as represented by 3a, then the connection becomes eligible, as defined by Equation 1. This state could be reached before or after the completion of the data transfer (state 2). Sometimes, client-side browsers send a RST flag after the completion of the data transfer. Hence, merely seeing a RST flag from the client does not confirm that a data transfer is interrupted. To identify those client-side RST flags indicating interruption of the data transfer, HTTP transactions (request and response) and $t_{gap}$ (see Equation 2) should be taken into account.

**State 5. Interrupted:** If the RST flag from the client is seen when the data transfer from the server is still going on (shown by transition 5a), then the connection is called interrupted. However, there are two types of such interruptions: One made by the client-side web browser and another one by the user. It is not possible to differentiate between both of them only by observing the TCP flow. Let’s recall the video page download case, which was performed on the IE web browser. We observed that the client terminates the data transfer each time after it requests for the video player. In this case, although the TCP connection termination met the interruption criteria, the transfer is not interrupted by the user. In order to identify the TCP connections interrupted by the user and not by the browser, we need to take into account the HTTP request and response messages. For instance, if a TCP connection interruption is followed by the arrival of the last HTTP request, then it shows that the TCP connection is interrupted automatically by the web browser (see Internet Explorer case with the video transfer). Conversely, if the TCP interruption is followed
by either of the transitions 2e, 2f and 6b then the connection was interrupted by the user and not by the browser. When a transfer is reloaded by user, request for the last base file $H_B$ is repeated. The transition 2e shows that the Web transfer is reloaded by the user. Similarly, when a user follows a new link or a bookmark (link-follow), before the completion of the previous transfer, clients sends the HTTP message containing the request for a completely new file. The transition 2f shows this link-follow behavior. Finally, if the user interrupts a transfer by pressing the stop button or by killing the web browser, then it does not immediately trigger any new HTTP request message but followed by the user think time (the time user takes before launching the new request for a page). It is illustrated by the transition 6b in the state diagram.

**State 6. Idle/User think time:** Let’s recall the ON-OFF model of the Web transfer described previously in Figure 1, where each web transfer is followed by inactive OFF time. This state could be reached by either of the two possibilities. First, if a transfer is completed without any interruption by the user, the user takes some time for reading the page or thinking about the next link before launching the request for the new page. Second, when the user interrupts the previous transfer by pressing the stop button or killing the web browser, then it takes a period of silence time before the user requests for a new page.

Summing up the above discussion, we observed from the active and passive tests that the web browsers often do not follow the given TCP standards. We classified several types of TCP connection terminations between the client and the server. The authors in [20] proposed a criterion, which is quite helpful in detecting those TCP connections that are interrupted by the client before the end of data transfer from the server. However, there could be two causes of such client-side interruptions: the client web browser or the user. To identify the interruptions done by the user and not by the web browser, we presented a set of criteria. According to the criteria, the HTTP request and response messages need to be taken into account beside the TCP flags in order to identify the transfers interrupted by the users.

9 Conclusions

In this paper, we proposed a set of criteria to monitor the user actions in the Web browser. The monitoring of these actions can provide indications about user reaction to the network performance. These are based on the TCP interruptions and HTTP
requests. In order to study whether TCP RST flags could be used to monitor the user behavior on the Web, we conducted several experiments with different web browsers. We found out that some web browsers send TCP RST flags without any interruption by the user. Therefore, TCP RST flags alone could not be used to monitor the user actions in the web browser. However, TCP RST flags along with the knowledge of HTTP request and response messages can allow us to passively monitor the user-perceived performance.

Additionally, we also showed some of the abnormal behaviors by the web browsers. We believe that there is a need of a better mechanism for communication between web browsers and the web servers, in order to improve the performance of TCP connections and raise the user experience.

Acknowledgment

This work has been sponsored by the Network of Excellence Euro-NF (FP7 ICT N_i 216366) and was started within Euro-NF’s Specific Joint Research Project QoEWeb. We would also like to thank our master students Adeel Ashfaq and Umer Bilal for their work on the active tests in the lab environment.

References


REFERENCES


http://dx.doi.org/10.1145/2018602.2018613


http://dx.doi.org/10.1145/1052812.1052823

http://dx.doi.org/10.1007/978-3-540-79232-1_20

http://dx.doi.org/10.1007/978-3-642-20305-3_3

http://dx.doi.org/10.1109/GLOCOM.2003.1259011
