Non-Intrusive Network-Based Estimation of Web Quality of Experience Indicators

Junaid Shaikh
Abstract

Quality of Experience (QoE) deals with the acceptance of a service quality by the users and has evolved significantly as an important concept over the past 10 years. Network operators and service providers have gained interest in QoE-aware management of networks, in order to better fulfill end-user demands and gain a competitive edge in the market. While this growth promises new business opportunities, it also presents several challenges to the networking researchers, which are mainly related to the assessment of user experience.

Several QoE assessment models have been proposed to estimate the user satisfaction for a given service quality. Most of them are intrusive and require knowledge of the content reference. In contrast, the network operators require non-intrusive methods, which allow models to be implementable on the network-level without having much knowledge about that reference. The methods should be able to monitor QoE passively in real-time, based on the information readily available on network level.

This thesis investigates indicators, which are intended to be used in the development of non-intrusive network-based methods for the real-time QoE assessment and monitoring. First, a bridge is made between the user and network perspectives by correlating the user traffic characteristics measured on an operational network and user subjective experience tested on an experimental platform. It is shown that the user session volume appears to be an indicator of users’ interest in the service. Second, the TCP connection interruptions are investigated as an indicator to infer the user experience. It is found out that the request-level performance metrics show stronger correlations between the interruption rates and the network Quality of Service (QoS). Third, a wavelet-based criterion is devised to assist in the identification of those traffic gaps, which may result in the degradation of QoE. It can be implemented on the network-level in quasi-real-time to quickly identify the user-perceived performance issues.
Acknowledgement

With respect and gratitude, I acknowledge my supervisor Prof. Markus Fiedler, who has been an excellent mentor and friend throughout my stay at Blekinge Institute of Technology (BTH). He has been a great inspiration and under his leadership, I have known how to meet the targets with ambition, consistency and hard work. His trust in my ability made me work even harder and better. Working with him has been a rewarding experience, which I will always treasure.

I would also like to show my gratitude towards Dr. Patrik Arlos. His guidance and instructions helped me in gaining hand-on experience with the equipment used for the network measurements. His feedback on my thesis report assisted me in improving my mistakes.

I would like to thank Mr. Denis Collange (R&D Orange labs) for his guidance throughout my Licentiate. He always welcomed the chance to collaborate and work together for investigating different research problems. I truly enjoyed my stay at R&D Orange labs and learned to look at the problem from a network operator’s perspective.

Along this journey, I have met and worked with a number of excellent people. A special thanks goes out to Prof. Adrian Popescu for his teaching and discussions during Ph.D. courses. I must acknowledge Dr. Khalil Laghari (INRS Canada) for initiating various interesting discussions on the research topic and providing critical input to my work.

I am thankful to all my fellow Ph.D. students who have been extremely helpful throughout my stay at BTH. I would also like to thank Eva-Lotta Runesson, the unofficial “mom” of all Ph.D. students at the department and Camilla Ericsson for making life easier at the department. I also acknowledge the efforts of those master thesis students who worked under my supervision with hard work and consistency. They kept me on my toes as a mentor.

With profound love and respect, I thank all my family members who have al-
ways supported me. I would like to make a special mention of my uncle Abdul Khaliq Shaikh; he has always been a source of inspiration. Without his support and encouragement, I may not have completed higher studies. I am grateful to my parents, grandparents, brother Naveed and my sisters for boosting up my confidence in achieving my goals in life. At last but not the least, I would like to acknowledge my wife Rabail for making special difference to my life. She has been my best friend, and a great source of happiness in my life.

In the end, I would like to thank my best friends, Irfan and Kamran for bringing in moments of immense joy and laughter. Several debates with them over tea – at Cafe Clifton Karachi – have become memories to cherish for lifetime.
# I Quality of Experience from User and Network Perspectives

1 Introduction ................................................. 23
2 Related work ................................................. 26
3 Active measurements on experimental platform ................. 27
   3.1 Measurement platform and methodology ................. 27
   3.2 Impact of packet loss on download time and on throughput 29
   3.3 Relation between QoE and QoS parameters .............. 32
4 Passive measurements on real-users network ...................... 37
   4.1 Measurement platform and methodology ............... 38
   4.2 Correlation of traffic characteristics with performance metrics ................. 38
5 Combination of user and network view ........................... 44
   5.1 Comparison by throughput ................................. 44
   5.2 Comparison by loss ....................................... 45
6 Conclusions and future work .................................. 47

# II Passive Methods for the Assessment of User-perceived Quality of Delivery

1 Introduction ................................................. 55
2 Monitoring of user interruptions ............................... 57
3 Monitoring of traffic gaps on the network ...................... 59
4 Conclusion and outlook .................................... 63

# III Classification of TCP Connection Termination Behaviors for Mobile Web

1 Introduction ................................................. 71
2 Methodology .................................................. 73
3 Test-bed setup ................................................. 74
4 Termination flags ............................................. 75
   4.1 Uninterrupted transfers .................................. 76
   4.2 Interrupted transfers ...................................... 79
5 Time-to-termination ........................................ 81
2 Future work .......................................................... 136

BIBLIOGRAPHY ............................................. 137
Chapter 1: Introduction

Future telecommunication networks are faced with increased demand for resources due to continuous growth in data traffic. The usage of bandwidth-intensive applications is increasing, leading to congestion in the access networks. On the other hand, the user expectations are growing over time. Any slight disturbances in the network result in a decay in the user satisfaction level, which is an undesirable situation for the network operators. Due to the availability of several competitors in the market, user satisfaction has become the most critical component for gaining competitive edge in the market.

Quality of Experience (QoE) has emerged as an important concept during the last decade with the focus on understanding of the user acceptability of a service or an application. Specifically, it is the user-centric evaluation of a service or application performance. Hence, in order to prevent user churn and save revenue, real-time assessment of Quality of Experience (QoE) is becoming the primary tool for network operators in managing the networks.

1 Background

Since 1990s, there has been an enormous growth observed in the usage of Internet. This growth in the usage is further fueled by the increasing number of multimedia applications and widespread availability of World Wide Web (WWW). Furthermore, advances in network bandwidth and computational power of end-user devices have also contributed significantly in making the Internet a platform for performing daily life activities. This reliance on the Internet for the wide spectrum of activities has resulted in the increasing importance of QoE.

Furthermore, the popularity of QoE can be attributed to the following factors:

1. The deployment of packet switching technology (traditionally the best-effort
services) in telecommunication networks brought more emphasis on QoE in the packet-switched networks. The processes in the telecommunication management networks are customer-oriented. The Enhanced Telecom Operations Map (eTOM) Business Process Framework amongst others specifies service assurance operation processes. These processes have to ensure that the service quality is delivered according to the customer demands and Service Level Agreements (SLAs).

2. The deregulation of telecommunication sector triggered a fierce competition among the network operators. There are now several network operators available in the market and QoE is a distinctive factor between their success.

3. While the bandwidth of the physical links is increasing, at the same time there is also an exponential growth observed in the traffic in the network [1]. According to the Cisco forecast, the global Mobile Broadband traffic volume will roughly double every year [2], posing a serious challenge to the Mobile Broadband service providers. According to [3], 7% of the subscribers consume nearly 90% of the traffic, leading to a shortage of network resources for most other subscribers, and thus creating QoE issues for them. Network operators have to make smart decisions to manage this traffic dynamically using available network resources, while keeping user QoE to the acceptable level.

4. The number and type of applications is growing. Each category of application has its own set of performance requirements and criteria to work smoothly on the network. For instance, on a network with the same amount of resources, the same user can have different experiences based on the category of application used. Dimensioning a network for all the categories of applications on the same criterion may result in under- or overestimation of resources. For example, in the case of underestimation, a user might be using an application which requires more than the available network capacity. In this case, the user may get dissatisfied with the perceived network quality due to possible congestion in the network. Conversely, in the case of overestimation, a user may be using an application that works well at the low capacity. But since the available network capacity is high, it may lead
to a waste of the resources. Hence, there is an increasing realization that the design of networks should be user-centric and networks should dynamically adapt to the needs of the users.

5. At times, the user gets annoyed and dissatisfied with a network operator, even though the network performance is adequate. In such cases, there could be other reasons behind the user annoyance, such as the performance of the server (which is serving content to the user), performance of the user-end device, application interface or the quality of the content itself. The user blames network operator for all such problems, as it is often the nearest point of contact for a user. Hence, for a network operator it is important to evaluate the user experience automatically, in order to be aware of all these issues, and to keep the users informed about the reasons of degradation in perceived performance in order to prevent the user churn.

6. Socio-economic factors such as cost and social background of the user may also result in the different expectations and hence, lead to the different experiences. For example, users with different backgrounds and previous experiences may have different perception about the same quality of a network service. Similarly, a user paying higher cost for a service may expect better quality than a user using the service at the low cost. These issues need to be addressed by QoE, by mapping the effect of all these factors on the service performance.

2 Definition of QoE

Before going further into the engineering of QoE, this section gives a brief picture about the definitions of QoE given in the literature. There has been lot of divide seen in the definition of QoE, depending on the field of profession it comes from. ITU-T defines QoE as:

“The overall acceptability of an application or a service, as perceived subjectively by the end user.” [4]
The above definition suggests that the QoE is subjective in nature which reflects the overall acceptability of a service by a user. The questions then arises: What level of service quality is acceptable to the user? Does the perceived service or application quality is same for all the users or does it change across the users? If it changes, then how to quantify that change? It will relatively be easier to practice and assess QoE if the definition is more specific and provides an interface to the practical implementation of the QoE.

In [5], QoE is defined as:

“QoE is how a user perceives the usability of a service when in use – how satisfied he or she is with a service.”

The definition provides a Human Computer Interaction (HCI) perspective of QoE. The end-to-end service involves several components and stakeholders. It is possible that the usability of a service is affected by the factors which are hidden to the users. Therefore, there is further need of a definition that comprehensively cover all such factors.

Another definition of QoE found in the literature is:

“The degree of delight of the user of a service, influenced by content, network, device, application, user expectations and goals, and context of use.” [6]

The above definition mentions those factors that may affect the user perception of a service quality. With the help of this definition, a conceptual model could be devised, which further breaks down these factors into sub-factors. For example in the case of network, these sub-factors may include the end-to-end delay, throughput, outages and losses etc. Hence, by understanding all the factors that effect user perception of a service, it will make it easier to build those systems that assess all these factors and gives a picture about the user QoE. It also provides a platform to adopt an interdisciplinary approach for provisioning QoE.
3 Quality of Service vs. Quality of Experience

In [5], Quality of Service (QoS) is defined as:

“The ability of a network to provide a service with an assured service level.”

QoS is usually described in terms of the individual technical parameters such as loss, delay, jitter and throughput. These parameters are usually referred to as Key Performance Indicators (KPIs). However, it is argued that the KPIs alone cannot reveal the end-user’s experience [3]. These indicators may characterize the performance of a segment of the Internet or an individual system. However, they don’t necessarily represent the end-to-end service as perceived by the user. For example, the users may care about the download time of a webpage but they don’t usually bother about the Round Trip Times (RTTs) of individual packets measured in a network operator’s domain.

In [3], the indicators of user experience are referred to as Key Quality Indicators (KQI). The user experience is dependent on the several other factors besides the QoS parameters, depending on the application. Those factors need to be identified, measured and translated to the user experience in order to assess the QoE over time. Network operators need to monitor QoE in order to ensure that the users are happy with their services. In [7], the function of QoE is defined as:

“The function of quality of experience (QoE) evaluation includes two aspects: to monitor the experience of user on-line, then to control and justify the service based on the QoE to ensure that the quality of service can highly meet the requirements of the user.”

4 Assessment of QoE

For network operators, in order to deliver their services in the best possible way, it is important to assess how their users perceive the service quality. The assessment method should be based on the parameters that could be measured
and managed practically. These parameters should be viable enough to provide realistic view of the service as perceived by the end-user. There are mainly two types of methods for the assessment of QoE:

1. Subjective assessment
2. Objective assessment

### 4.1 Subjective assessment of QoE

The user experience is subjective in nature and is often expressed by users in the terms which are qualitative in nature such as excellent, good, bad, etc. The feelings and the opinions of users do not just vary widely across the different users, but they also vary for the same user over time. It is a challenge for the network operators to adopt and manage their services according to the wide spectrum of user feelings.

The subjective evaluation of a service is obtained by various means, which are further described below:

**User complaints**

Users provide their feedbacks about the perceived quality of service by contacting their respective network operators. These feedback are usually conveyed in the form of user complaints. However, this mechanism of obtaining user perception of a service is not considered very effective. Users often express their opinions after several bad experiences with the service. While some users log their complaints, many do not even do so and simply switch to the another network operators. Moreover, on average, one frustrated customer will tell 13 other people about their bad experiences [5]. This behavior leads to a negative word-of-mouth about a network operator.

**User surveys**

Another method used for obtaining the user opinions about a service is to perform surveys. In this case, the network operators contact their customers and inquire them about their experience with the service. Although, this is a proactive way of gathering the user responses about a service, it is not very appropriate way of evaluating a service due to the following reasons. First, many users do not even
participate in the survey. Second, the delivery of a service is dependent on several factors. Variation in any one of them may affect the user perception. Hence, it will be too hard for network operators to find the optimal level of each of these parameters based on a survey. Third, user experience varies over time. Conducting these surveys frequently may disturb the users.

The user subjective tests

Conducting subjective experiments in a test environment is one of the most widely used method for the subjective evaluation of QoE. The test environment allows control on the parameters used in the experiments. Due to the controllable parameters, the effect of variation in each of these parameters on the user experience could be assessed and compared to the other similar studies. Finally, the user experience could be modeled as a function of each of the parameters. Since these tests are repeatable, the findings can be confirmed by conducting experiments several times, thereby providing higher degree of confidence on the obtained results. However, there are also certain drawbacks observed in this approach.

- The self-reporting of opinions by the users in the test environment may not reflect those feelings, that are normally surfaced as a result of the usage done in the real life.

- The time duration of the tests is often kept short, so that the participants do not get bored by the tasks during the tests. However, it is very difficult to capture the temporal aspects of the user experience in the test environment, which normally could be seen on the longer time scale in the real life.

- Conducting subjective tests in a lab environment is an expensive and time-consuming task. It is often seen that convincing users to attend the test session requires lot of effort. A possible alternative to this issue is to delegate tasks to several users around the world by conducting crowd-sourcing experiments. A study conducted using crowd-sourcing platform is recently reported by Tobias et al. in [8].

- The numeric scales are often used to translate the opinion scores of users for quantification of the user experience. The Mean Opinion Score (MOS) [9]
scale is widely used to obtain the user responses about a service quality, based on the ordinal scales such as the Absolute Category Rating (ACR) and the Degradation Category Rating (DCR) [10]. However, there are certain problems with these scales. First, it is possible that the participants in the tests have different interpretations of an opinions score and they provide the same rating for different experiences or vice versa. For example, different people can have different interpretations of the word “Fair”. Second, the rating from the user is usually taken after the test, which makes it hard to relate the user opinion to a particular event, that influenced the user rating during the test. It is possible that their experiences vary over time during the test, from the first second to the last second. Third, the difference between the subjective opinions for instance, “excellent” and “good” may not be the same as the difference between the numbers 5 and 4 on the numeric scale.

4.2 Objective assessment of QoE

In the objective assessment of QoE, computational models consisting of measurable parameters are used to assess the user experience. It is different from the QoS assessment in that it takes into account several additional parameters besides network performance indicators (such as loss, delay, etc.), in order to assess the user-perceived performance of a service.

The objective assessment of QoE from the network requires two major steps. First, the computational models are required, which take into account several parameters in order to approximate the perceived quality. Second, the methods are required, which allow those models to be implemented practically in the real-time scenario.

Computational models

As mentioned above, the computational models require measurable parameters for the objective assessment of QoE. Ideally, these models should take into account:
• user expectations;
• prior experiences.

The existing computational models for the objective assessment of QoE are mainly directed towards the improvement of a particular application. Most of the existing QoE models are formulated for the quality assessment of voice, image and video applications. These models vary in their computational complexity and accuracy.

The speech quality planning model which is also known as the E-model, was given by ITU-T recommendation G.107 [11]. The model provides prediction for the experienced speech quality of a conversation. The prediction of speech quality is done based on the packet loss rate, coding scheme, bit rate, loudness and echo.

In the ITU-T recommendation G.1070 [12], an opinion model for the video telephony is described in detail. The model is based on three building blocks. First, the speech quality is assessed using the E-model. Second, the video quality is estimated using a video quality estimation model. Third, the results of both the speech quality as well as video quality are given as input to the multimedia quality integration model, which further combines them and produce a final estimated quality of a video telephony conversation.

Recently, there has been lot of work done on the quality estimation of audio and video streaming applications. However, there is no final standardized model available for the streaming applications.

The models which are presented in the previous work mostly focused on the quality estimation for the planning and design of networks, applications and end-terminals. This is suitable for the traditional circuit switching networks, where the conditions were not so dynamic and resources used to be reserved for the end-to-end transmission. However, these models are not very much appropriate for the planning of IP networks. IP networks are very dynamic and heterogeneous, which are traditionally designed for the best-effort services. Often, the traffic from the large number of sources and applications passes through a single network. Hence, the planning of networks based on the results obtained from the quality estimation of a single application has proved to be unscalable in today’s scenario.

Considering the dynamicity of the operational IP networks, the real-time quality monitoring models should rather be designed to assess the QoE in real-time.
The models should take into account easily measurable parameters, which could be retrieved by the operators of networks and servers in the real-time. The networks and applications should adapt themselves dynamically based on this real-time quality assessment.

**Practical methods**

The another requirement of the objective assessment of QoE is to devise those methods, that allow the real-time assessment of QoE in the operational environment. The methods should be efficient and practically implementable. The methods should be capable enough to:

- measure and retrieve all those parameters required by the computational model;
- communicate the results of assessment with the network operators.

There are several computational models available today. However, there are lack of practical methods for the implementation of these computational models on the networks. While there has been a lot of focus on the modeling of QoE during the previous years, the engineering of methods to implement such models still remains a challenge. The models usually are implemented within the application at the user-end. There are two main drawbacks of this approach. First, the computation of these models requires processing which is done at the user end, hence creating overhead at the user-end devices. Second, the results of QoE assessment are need to be transferred to the network operators, which consequently may create extra traffic on the network. Although, the implementation of these models at the user-end may produce more accurate results. However, these methods can be non-scalable, considering the scenarios where each user is using several applications and each application is doing its QoE assessment separately. Hence, there are more efficient methods required to implement computational models for the real-time monitoring of QoE from network-level.
5 Network-based methods for assessing QoE

In this thesis, network-based methods refer to those methods that allow the real-time monitoring of QoE from network-level. The objective of these methods should be to compute QoE in real-time and provide feedback to the network operators. Subsequently, the information from the QoE assessment should be used to manage the networks dynamically.

In the past, there has been a significant amount of work done for the designing of network monitoring methods by the network performance community. The focus of these works have been to assess the performance from the network perspective through technology-centric performance indicators. However, the work on QoE indicators is still missing.

QoE monitoring methods should be designed in way so that they:

- consider the user-perceived performance indicators;
- are implementable on the network level.
Today, the user-centric performance monitoring face many challenges. Several users with different backgrounds are connected to the networks. Their usage on a network is not limited to a single application. Figure 1 depicts a scenario where N number of users are connected to a network. Each user is using one or more applications as represented by X, Y and Z. The current QoE computational models consider parameters for example, content characteristics, codecs, coding bitrate. It is not easy to obtain these parameters from network-level measurements. Network measurements are usually performed between the access and the core networks as mentioned in Figure 1. Therefore, there is a need of identifying those indicators of user experience, which can be obtained easily from the network traffic and allow QoE monitoring methods to be implementable in the real-time without creating an extra load on the network. In short, the methods should be lightweight and assess QoE on the fly in order to be implemented in the real-time in high-speed networks.
Chapter 2: Web Quality of Experience

1 Popularity of web

The World Wide Web (WWW) has been the most popular application on the Internet since its commercialization in the year 1995. Currently, it is not only limited to the information sharing of a single domain but has become the integral part of the daily life activities related to education, business, commerce, science, social networking and entertainment etc. Recently, the arrival of video on web further fueled its popularity. According to [13], Web traffic accounts for major volume of the Internet traffic. Its share has increased from 41% in 2007 to 52% in 2009.

2 User experience on web

The user experience on web is often described as a function of response times. Since late 1990s, the World Wide Web has also been called the World Wide Wait [14]. The reason is that the users have to wait for the amount of time, which is sometimes beyond their expectations. Their flow of thoughts is often interrupted due to the longer waiting times. The situation got even worse with the usage of web browsing on the mobile networks, as the waiting times grew further due to the limited capacity in mobile networks.

In 1968, Miller introduced guidelines for the human-computer conversational transactions [15]. He presented three limits for the response times which are:

- 0.1–0.2 s: If the computer responds within this limit then the response is perceived as instantaneous by the user.
1– 5 s: If the response is provided within this time limit, then users feel that there is some delay and the computer is working on their command. Their flow of thoughts however, remain uninterrupted during this time limit.

5–10 s: If the response time is in this range, then the flow of thoughts is interrupted and users start abandoning their tasks, as the waiting times proceed towards the 10 s limit.

Similarly, Steven Seow defines four classes of responsiveness [16]:

0.1–0.2 s: Instantaneous.

0.5–1 s: Immediate.

2–5 s: Continuous.

7–10 s: Captive.

The middle threshold is thus broken into two further classes: Immediate and Continuous. The users perceive some delay when the response times go above 0.5 s. However, the processing is still perceived as immediate, as far as the system responds within 1 s time limit. The flow of thoughts is interrupted with the response times going above 2 s.

The user experience on web has serious implications on the businesses. The users are getting more intolerant and may click away if the content is not served in the expected amount of time. Any small increment in the page load time (in the fractions of a second) affects strongly the user interests in the webpage [17].

A study conducted in 2006 showed that the online shoppers expect a maximum of 4 s of page load times. The similar study was also conducted in 2009 and the expected page load times were reduced to 2 s [18] [19]. In 1999, the value of expected page load time was reported as 8 s. It indicates that the user expectations are growing with the time. Users are becoming more impatient. On the other hand, the networks (in particular the mobile networks) are not yet ready to cater such expectations in order to improve the user experience.
3 Assessment of Web QoE

Many studies on web user experience assessment were done in the past. Most of these studies were done in the context of HCI, with the objective to improve the user interface and design of web pages [20] [21] [22]. There have also been many studies done on the web user behavior, mainly for the workload characterization of the web servers [23] [24] [25] [26]. Their scope is usually limited to a single web site and these studies do not quantify the effect of network performance on the user behavior.

During the last few years, there is a lot of attention given to the QoE assessments for the voice and the video applications. It is also evident from the ITU-T standardization activities for these applications, well documented in the ITU-T recommendations. However, we do not see many comprehensive studies on the QoE assessment of web browsing activities, done by the network performance community.

In the ITU-T recommendation G.1030 [27], an opinion model for the web browsing applications is provided. The model maps response and download times to the perceived quality of a web-browsing session. It observes the impact of waiting time of a single page. It also takes a context-dependent approach, based on the expected page load times i.e., 6, 15 and 60 seconds, corresponding to a fast, medium and slow network. The study considered two types of users, based on their training: naive users and expert users. The study concludes that the relationship between QoE and session time varies based on the expectations of the users and therefore, different models are derived for the different contexts.

Similarly, other studies also touched upon some of the aspects of QoE assessment of web browsing activities. In [28], the impact of latency on the QoE of online shoppers was investigated. It was found that the type of task, the length of time of the user interaction with the website and the method of page loading affect the impact of latency on QoE.
4 Thesis overview and contributions

In this thesis, we have presented our work on the network-based indicators of QoE, with focus on the web application. The indicators are based on the user behavior (actions on the web) and traffic characteristics. Our main objective is to provide such indicators, that can help operators to assess QoE passively from the network-level. Part I of this thesis bridges the gaps between the user and the operator by providing relationships between the user traffic characteristics analyzed on the operational network and the user subjective ratings taken in the test environment. Part II touches upon two methods for the passive assessment of QoE. Part III further goes deeper into one of the passive methods described in part II, which is based on the TCP control flags. Part IV, presents some results related to the performance of interrupted TCP connections, based on the traffic collected on an operational network. Part V proposes a wavelet-based criterion to identify the network-induced traffic gaps. Finally, Chapter 3 in the end will conclude this thesis with a set of conclusions and the future work.

4.1 PART I: Quality of Experience from User and Network Perspectives

This part 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 regressions.

The second approach is based on the study of traffic traces, captured on the operational network of an ISP. 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.

4.2 PART II: Passive Methods for the Assessment of User-perceived Quality of Delivery

This part introduces two network-based methods to objectively assess the user-perceived network performance. One method is based on the monitoring of TCP control flags incorporated with the flow-level performance, to assess the user’s interest in the service. The other straightforward method inspects the traffic gaps in user sessions to assess the user-perceived performance. This method doesn’t require any deep packet inspection, and therefore, it is more suitable for the implementation in real-time to assess the user-perceived performance on the fly.

4.3 PART III: Classification of TCP Termination Behaviors on Mobile Web

In this part, 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 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. However, it was concluded that the TCP connections termination behavior is heavily dependent on the client-side application.

4.4 PART IV: User impatience and network performance

The discussion from the previous part is further extended in this part to analyze the correlation between the interrupted TCP connections and different end-to-end to
performance metrics. The study is performed on the traffic traces collected from the operational network of an ISP. It is found out that the request-level performance metrics show stronger correlations between the interruption rates and the network QoS as compared to the connection-level performance metrics. These interruptions are also dependent on the application type.

4.5 PART V: Monitoring and Analysis of Web Usage and Experience Based on Link-level Measurements

This part presents another passive monitoring and analysis method, which assists in the identification of those traffic gaps on the network that may result in the degradation of Quality of Experience (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 part 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 any deep packet inspection, the criterion is simple and intended to be implemented in near-real-time.
Part I
Part I

Quality of Experience from User and Network Perspectives
Part I is published as:


1Slight formatting adjustments are made in tables, figures, equations and references.
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 behaviour 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 this impact on each other and on user satisfaction. This preliminary study is based on the application web.

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 techni-
cal 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 behaviours 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 behaviour 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 behaviour 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.

![Test-bed setup](image)

**Figure 1: Test-bed setup**

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 $R' = \frac{X}{T}$. Download times and thus even perceived throughputs are prominent performance parameters from the viewpoint 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.

![Download time (average ± standard deviation) as a function of nominal loss.](image)

Figure 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.
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><strong>Exponential</strong></td>
<td><strong>0.997</strong></td>
<td>$T = 1.1 \exp(0.26L)$</td>
</tr>
<tr>
<td>Power</td>
<td>0.969</td>
<td>$T = 1.1 L^{1.0}$</td>
</tr>
</tbody>
</table>

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

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 $R'$ 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.

<table>
<thead>
<tr>
<th>Coefficient of correlation $r$</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear $-0.964$</td>
<td>$R' = -0.67L + 6.9$</td>
</tr>
<tr>
<td>Logarithmic $-0.989$</td>
<td>$R' = -2.8ln(L) + 7.2$</td>
</tr>
<tr>
<td><strong>Exponential</strong> $-0.998$</td>
<td>$R' = 8.9 \exp(-0.25L)$</td>
</tr>
<tr>
<td>Power $-0.963$</td>
<td>$R' = 9.0L^{-0.97}$</td>
</tr>
</tbody>
</table>

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.](image)

Figure 4 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 5). 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 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>$QoE = -0.31 L + 4.3$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>$-0.942$</td>
<td>$QoE = -1.4 \ln(L) + 4.3$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$-0.969$</td>
<td>$QoE = 5.5 \exp(-0.2L)$</td>
</tr>
<tr>
<td>Power</td>
<td>$-0.877$</td>
<td>$QoE = 5.2 L^{-0.72}$</td>
</tr>
</tbody>
</table>

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

According to Table 4, exponential fitting works best, followed by logarith-
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><strong>$-0.995$</strong></td>
<td>$\text{QoE} = 4.836 \exp(-0.150 , T)$</td>
</tr>
<tr>
<td>Power</td>
<td>$-0.955$</td>
<td>$\text{QoE} = 5.339 , L^{-0.658}$</td>
</tr>
</tbody>
</table>

mic 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.

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>$\text{QoE} = 0.44 , R' + 1.0$</td>
</tr>
<tr>
<td><strong>Logarithmic</strong></td>
<td><strong>0.995</strong></td>
<td>$\text{QoE} = 1.5 \ln(R') + 1.153$</td>
</tr>
<tr>
<td>Exponential</td>
<td>$0.878$</td>
<td>$\text{QoE} = 1.175 \exp(0.188 , R')$</td>
</tr>
<tr>
<td>Power</td>
<td>$0.960$</td>
<td>$\text{QoE} = 1.208 , R'^{0.651}$</td>
</tr>
</tbody>
</table>
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.

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.

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

Table 6: Regressions on Quality of Experience (given through Opinion Scores) vs. the network-level throughput $R$ (given in Mbps).

<table>
<thead>
<tr>
<th>Regression</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><strong>0.979</strong></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>

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 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 couple 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 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.

![Figure 9: Ratio of out-of-sequence packets vs. session volumes downloaded for different silent time thresholds](image)

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
Figure 10: Ratio of out-of-sequence packets vs. session volumes uploaded at different hours of the day

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><strong>-0.996</strong></td>
<td>V = 98 L^{-0.62}</td>
</tr>
</tbody>
</table>

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 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 \(4 \cdot 10^{-3}\) 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.
Figure 11: Throughput (Mbps) vs. session volumes downloaded (packets)

Table 9: Regressions of session volumes downloaded ($V$) vs. throughput ($R$) 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>0.928</td>
<td>$V = 1202 R + 208$</td>
</tr>
<tr>
<td>Logarithmic</td>
<td>0.804</td>
<td>$V = 1128 \ln(R) + 3519$</td>
</tr>
<tr>
<td>Exponential</td>
<td>0.808</td>
<td>$V = 77 \exp(0.78R)$</td>
</tr>
<tr>
<td><strong>Power</strong></td>
<td><strong>0.972</strong></td>
<td><strong>$V = 976 R^{1.02}$</strong></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. 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:

\[
\text{QoE} = 1.2 \ln(R) + 1.3
\]  
\[
V = 976 R^{1.02}
\]

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 \(V \sim \exp(\text{QoE})\). From this, we can deduce that users that perceive a good QoE (which is enabled through high throughput)
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:

\[
\text{QoE} = 0.31 \, L + 4.1 \tag{4}
\]

\[
V = 98 \, L^{-0.62} \tag{5}
\]
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 decreases, and keeps doing so even beyond 1%. While these trends differ in shape, they point in the same direction: Decreasing loss ratios correlate with both increased session volumes and improvements of QoE; the latter however get marginal for small session sizes.
6 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 parameters 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


Part II

Passive Methods for the Assessment of User-perceived Quality of Delivery
Part II is published as:

Abstract

Passive monitoring of user-perceived performance degradation is an important tool for service providers to improve customer loyalty. In this paper, we discuss our on-going work on the development of two network-based methods to objectively assess the user-perceived network performance. One method is based on the observation of TCP connections interrupted by the users. This method allows us to detect user’s interest in the service in relation to the network performance. Another method is simple and based on the identification of traffic gaps in the user transfers that may hurt the user perception. This work, amongst others, provokes a discussion on the impact of the frequency and duration of such gaps.

1 Introduction

The automatic detection of user-perceived performance degradation in networks is an important challenge for network operators. Over the years, network operators have been assessing network performance based on the Quality of Service (QoS) parameters such as throughput, loss ratio and delay of the overall traffic, measured within their domain. However, there is an increasing realization that novel methods are needed to assess the network performance as perceived by the users. This is because, the way those QoS parameters are measured and evaluated employing the classical techniques, do not present the true picture of user-perceived performance of networks. Moreover, the user perception of network performance, i.e. Quality of Experience (QoE), is not only dependent on the QoS parameters.

QoE is a rather subjective notion based on several factors such as user expectations, prior experiences, personal context and usage context etc. besides end-to-end network performance. Therefore, measuring the QoE is a complex task. Subjective user ratings on the MOS scale [1] are often used in order to assess the user perception of the network performance. However, the self-reporting of opinions may be annoying for users under real conditions, since they have to inform about the performance degradation themselves when they already lose their patience after a significant number of trials. In a test environment, self-reporting of
opinions by users may lead to unreliable results [2]. Additionally, the validity of
the MOS scale has also been questioned due to several reasons mentioned in [3].

Previously, works have been done in order to assess the QoE. However, the
majority of those works are limited to the performance of particular applica-
tions and are directed towards the improvement of application performance [3–6].
In [3], authors propose a framework for the evaluation of user-perceived perfor-
mance of a particular application in the test environment. In [4], inter-dependency
of loss and delay, and its effect on the quality of multimedia applications is pre-
sented. An approach for real-time monitoring of video quality is presented in [5],
without taking user perception into account. In [6], authors have shown how the
online gamers are sensitive to the network QoS. Many previous studies focused
on the quantification of network-level performance parameters such as loss, delay
or throughput in relation to the MOS scale or application-level performance. In
contrast, very little has been said about the development of practical methods to
infer the user perception of performance in real-time. Hence, this stimulates the
need for the automatic passive estimation methods to assess the user-perceived
performance variations over time.

In this work, we focus our discussion on the development of simple methods
to assess the user-perceived performance. The methods should be practically im-
plementable by the service providers in real-time. For this purpose, we present
two methods that – upon further refinement – are expected to be very efficient and
helpful in detecting user churn and the user-perceived performance degradations
over time. One method is based on the TCP control flags incorporated with flow-
level performance to assess the user annoyance and another method inspects the
traffic gaps in user sessions to assess the user-perceived performance.

The remainder of this paper is structured as follows. Section 2 discusses how
the TCP resets could be helpful in understanding the user behavior, Section 3
proposes a network-based criterion to identify the traffic gaps damaging for user-
perception of network performance and finally, Section 5 presents the conclusion
and outlook.
2 Monitoring of user interruptions

The Transmission Control Protocol (TCP) is a connection-oriented transport layer protocol. Every normal TCP connection starts with a three-way SYN handshake and ends with a three-way FIN handshake procedure [7]. However, there are conditions when a TCP connection is reset: One of the hosts may either deny the connection establishment due to a nonexistent port or abort an established connection. In both cases, a TCP RST flag is seen in the control flag field of the TCP header.

The major volume of data on the Internet is carried by the TCP connections [3, 9]. For example, Web browsing which is the most popular activity on the Internet [10], usually employs the TCP connections for the data transfer. Normally, users press 'Stop' or 'Reset' button in the Web browser to abort a connection when a web transfer is slower than their expectations, generating one or more TCP RST flags from the user side. Therefore, monitoring of TCP reset flags generated from the user side could be one of the potential ways to observe the user behavior.

However, some of the studies in past revealed that there are significant number of TCP connections on the Internet that consist of one or more TCP RST flags, and client-side applications are the major contributor of these flags [2–4]. Some of the client-side Web browsers terminate the TCP connections with a reset instead of proper FIN handshake, this abiding from the rules mentioned in the standards [5]. Therefore, a criterion to identify the user-generated TCP RST flags is required.

Figure 1 illustrates an interrupted Web transfer. After SYN handshake, user requests for an object with HTTP GET message. The server then starts sending data to the client, denoted by $D_s$. The interruption from the user, i.e. the TCP RST flag from the client side, is represented by $R_c$, which is observed before the completion of data transfer from the server side. $t_s$ is the time when the last data packet from the server side is seen before the generation of $R_c$, while $t_e$ is the time when the end of connection is observed.

In [6], the authors presented a criterion to identify the TCP resets generated due to user interruptions. According to this criterion, a connection is called interrupted if the data is sent from the server followed by no server-generated FIN or RST and an RST from client is observed within the time roughly equal to the
value of mean RTT of the connection after the server sent the last segment containing data. Let $F_s$ be the FIN flag from server and $R_s$ the RST flag from server. If $\mu_R$ and $\sigma_R$ represent the mean and the standard deviation of RTT, respectively, then the criterion presented in [6] can be represented by the following notation:

$$I := \{ \neg \text{obs}(F_s \lor R_s) \land \text{obs}(D_s \land R_c) \} \land \left\{ \frac{t_e - t_s}{\alpha \cdot \mu_R + \beta \cdot \sigma_R} < 1 \right\}, \quad (1)$$

where $\alpha$ and $\beta$ are the constants and their values are TCP implementation-dependent however, normally $\alpha$ is equal to $1/8$ while $\beta$ is equal to $1/4$ [7].

In the case of Web transfers, connections are usually interrupted by the following user actions in the Web browser: 1) Pressing the ‘STOP’ button; 2) pressing
the 'RELOAD' button; 3) killing the Web browser; 4) clicking on another link on a Web page before a transfer is finished. The first three actions may potentially be result of the user dissatisfaction, but clicking on another link indicates that user has got the information she is interested in and therefore wants to proceed to the next page. Although the heuristic mentioned in equation 1 defines a criterion for identifying the connections interrupted by the users, it does not indicate the user action behind the interruption. Therefore, it is important to make a distinction regarding the user action that resulted in the connection interruption.

Moreover, our recent active tests suggest that, accessing video content in the Internet Explorer (IE) Web browser resulted in connection interruption without abortion made by the user. Our on-going work will further address these shortcomings by proposing a refined user interruption criterion. The degree of user annoyance during a certain time period will then be defined as the ratio of interrupted connections to the total number of TCP connections during that time period.

3 Monitoring of traffic gaps on the network

One of the ways to observe the user-perceived performance of the network is to monitor the traffic gaps on the network. Particularly, the wireless networks are prone to frequent outages due to availability issues. These outages result in traffic gaps that can be observed by monitoring the user transfers. This approach
is fast and simple as it does not require the deep inspection of packets and hence enables the service providers to take immediate actions on the discovery of such traffic gaps. However, network problems are not the only cause of such gaps. These gaps may well be due to the inter-transaction times between two subsequent transfers from the users. After the end of one transfer, a user may take some time before starting the next transfer. This silent time between two subsequent transfers of a user is often referred to as the user think time. It is therefore important to distinguish between the gaps resulting from the user think times and the network outages. In [1], authors identified the gaps and user-perceived problems but they didn’t quantify the boundary towards the think times, and they decoded the stream and simulated the buffer content afterwards. Moreover, their study was based on Youtube video.

In [16], we discussed about the user think times in detail based on the observations from past works [4–7, 9]. In this paper, we will just give an overview of these times. User think times are generally of two types: user think times during a session as shown by the inactive OFF time in figure 2 and the user think times between two sessions shown by silent time threshold [8] and inter-session time. A session is defined as a series of requests issued from a particular client to a particular server in a single visit to the server. These requests may comprise of one or more transfers. The start and the end of a session is determined with the help of a preset timeout value i.e. the silent time threshold. If the arrival of a new transfer is reported from the same client after a time-interval equal to or greater than the silent time threshold value then the start of a new session is marked.

It was found from previous works that user think times are usually above 8 s. After the end of one transfer, users usually take 8 s before launching the next transfer. In contrast, the gaps generated due to the network outages are normally between 1 to 4 s. Hence, frequently occurring gaps between 1 and 4 s characterize the bad transfers. Such gaps should be taken seriously by the service providers as they leave adverse effects on user perception.

In this paper, we further demonstrate the effect of gaps by presenting the throughputs of two web-based live video streaming transfers captured at two different locations. One transfer was done via a residential network connection in Sweden with a 10 Mbps shared bandwidth. Another transfer was done via a hotel network connection in Germany. The traffic was captured via Wireshark 1.2.7 net-
work protocol analyzer [2] on a machine with processor speed 2.53 GHz, memory of 4 GB and Macintosh operating system.

Live Video streaming was done for 180 s duration each. The perceived quality of both transfers was different from each other. No freezes were observed in the video streaming done via residential network, while considerably long freezes were seen in the video streaming performed via hotel network.

Figure 3 and 4 report throughput of a smooth and disturbed video streaming transfer respectively. Throughput is averaged over intervals of one second. The difference between both figures is clearly visible. Smooth transfer shows considerably longer gaps above 1 s as compared to the disturbed transfer. Moreover, there is also higher variation seen in the throughput of disturbed transfer as compared to smooth transfer

For smooth transfer, throughput is mostly between 9 to 20 KB/s. However, there are occasions when it rises and falls below these upper and lower limits. We occasionally observe smaller throughput as well. For instance, initially it took few seconds to start the video streaming and later at the end, we again observe a gap with no throughput for one second. However, we did not notice any freeze in the
video due to such performance degradation probably because of sufficiently large jitter buffer. 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]. Once the buffer gets empty, freeze occurs, which destroys the user experience [10].

Conversely, the disturbed video streaming as depicted in Figure 4 shows less throughput. It can easily be confirmed from the figure 4 that there are frequent gaps of 1 to 4 s, when we see no data at all. These gaps in traffic resulted in considerable freezes in the video. The low throughput and frequent gaps between 1 s to 4 s are the sign of outages and poor network conditions that differentiate these gaps from the user think times, which are usually larger than 8 s. Such low throughputs with frequent gaps are alarming for service providers and hence timely control measures should be taken based on the observation of such gaps. We also observe traffic gaps in range of the user think times, i.e. above 4 s, however such gaps are not so frequent and are followed by smaller gaps (1 to 4 s) and high throughput variations.

Figure 4: Throughput of a disturbed video streaming transfer
4 Conclusion and outlook

This paper presented an overview on our ongoing work on the development of passive methods for the monitoring of user-performance performance of networks. The methods are simple and doesn’t require any additional infrastructure for the implementation. A criterion for the identification of TCP resets generated as a result of user dissatisfaction will help in receiving user feedback automatically from the network traffic. Identification of traffic gaps that result in waiting time for users will assist network operators to quickly detect the user-perceived performance of networks.

This work is one of the initial steps towards the development of a fast and simple network-based criterion for monitoring any QoE-relevant delivery issue. Our short-term future work involves the quantification of the impact of these gaps on user perception. In particular, it will be of interest to investigate the impact of the frequency and duration of freezes the user is ready to accept. We are also interested in the relation of such gaps with buffer sizes. Finally, the recency effect of such gaps need to be quantified as well i.e. for how long a user remembers about the occurrence of previous outage. We believe that further development in this area can provide promising results in the area.

Acknowledgment

This work has been partly sponsored by the European 7th Framework Networks of Excellence EuroNF (216366) and the Swedish Agency for Innovation Systems VINNOVA for the project EViMonA d-nr 2007/02505.


Part III
Part III

Classification of TCP Connection Termination Behaviors for Mobile Web
Part III is published as:

Abstract

Many emerging smart applications and services employ Web technology, and users nowadays surf the Web from any device via any kind of access network. Typically, high page latencies trigger users to abort ongoing transfers, resulting in the abrupt terminations of the TCP connections. This paper presents a systematic study of the termination process of the TCP connections and identifies the reasons behind the observed sequences of termination flags. Monitoring and classification of the termination behavior of the TCP connections can provide indications about the user-perceived performance of Web transfers. From the results, it is observed that the TCP termination behavior is heavily-dependent on the client-side application. Therefore, a set of criteria is required to identify the abortions made by the user.

1 Introduction

There has been an enormous growth in usage of the Internet on mobile phones during the recent years. According to the study [1], Web usage on mobile phones is expected to represent 50% of the total Internet usage by 2014. Each mobile device is now equipped with multiple wireless network interfaces, but the performance of these networks is highly varying, mainly due to availability and coverage issues. These performance problems in the wireless networks result in high page latencies and hence affects badly the user experience on the Web. Therefore, the automatic detection of user-perceived performance degradation is an important challenge for network operators. Such a monitoring approach may allow network operators to take user-centric decisions for the switching between multiple wireless technologies when the performance gets poor.

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. 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 abortions result in early termination of the TCP connections with
Reset (RST) flag from the client side. These RST flags can be monitored passively on the network-level to observe the user behavior.

However, some of the studies in the past revealed that there are a significant number of TCP connections on the Internet that consist of one or more TCP RST flags. Client-side applications are the major contributor of these flags [2–4]. Some of the client-side Web browsers terminate the TCP connections with a RST flag instead of proper FIN handshake (TCP FIN flags from both sides) which is against the TCP standards mentioned in [5]. But these studies did not explain how to identify the client-side RST flags that result in an early termination of TCP connections (i.e. the RST flags that are sent while the data transfer is still going on).

In [6], the authors presented a criterion to identify the TCP interruptions done by the client. While this criterion is helpful in identifying those TCP connections which are interrupted, it does not guarantee whether interruptions are due to the abortion made by the user or automatically by the Web browser. It stimulates the need for an improved criterion that identifies the interruptions done by the user and also gives an idea about the type of user action performed in the Web browser.

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. 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 real scenario may provide rough indications whether a connection was interrupted by the user or not. To ensure a fair and representative comparison, we have conducted a number of controlled experiments with various Web browsers. These experiments are done on smartphones to test the mobile Web browsers. 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 a quality user experience. The research community working on the 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.

The remainder of this paper is organized as follows. Section 2 describes the methodology used to conduct this study and Section 3 presents the experimental setup. Section 4 discusses the sequence of termination flags observed with dif-
for Mobile Web

2 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 smartphone. 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 test, 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 Webpages 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 Webpages were almost similar to each other, we only present the results related to text and to video in the rest of this paper.

Moreover, tests were 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 Android’s 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 [7].

Summarizing the above description, each experiment was differentiated from other based on either the user action, content type or mobile platform. Hence, a
full factorial experimental design with $4 \times 2 \times 3 = 24$ experiments were established. Each experiment was repeated 40 times that made total number of runs to be $24 \times 40 = 960$.

3 Test-bed setup

The setup used for experiments is based on Distributed Passive Measurement Infrastructure (DPMI) [8], as shown in Figure 1. For capturing of traffic traces, a Measurement Point (MP) equipped with two Data Acquisition and Generation (DAG) cards was used [9]. It captures traffic in both directions. The time of MP is synchronized with Network Time Protocol (NTP) [10] and an input pulse of GPS is used for the clock ticks. The network, transport and application headers are captured and then forwarded to the Consumer, where they are stored for further analysis. A Measurement Area Controller (MArC) controls the MP and the Consumer. Apache 2.2 is chosen as Webserver in these experiments as it is the most popular Webserver in use [11]. It is installed on a machine with Windows XP operating system. The mobile client in the Figure 1 is a smartphone that is used for requesting the Webpage by user via the access point using IEEE 802.11b WiFi technology. A full duplex link of 100 Mbps is used between Web server and access point. The TCP version used by the Web server is TCP New Reno.

<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>User kills the browser before the page is completely loaded</td>
</tr>
<tr>
<td>Stop/Reload</td>
<td>User presses the stop or reload button before the page is completely loaded</td>
</tr>
<tr>
<td>Link-follow</td>
<td>User clicks another link before the page is completely loaded</td>
</tr>
</tbody>
</table>
4 Termination flags

Table 2 summarizes the different termination types seen from all the experiments. Termination type here refers to the sequence of terminating flags that were seen at the end of a TCP connection. Five different types of terminations

<table>
<thead>
<tr>
<th>Termination</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_sF_cR_c$</td>
<td>A FIN from the server followed by a FIN and then one or more RSTs from the client</td>
</tr>
<tr>
<td>$R_cF_cR_s$</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>$F_sF_c$</td>
<td>A FIN from the server followed by a FIN the client</td>
</tr>
<tr>
<td>$F_cR_c$</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>
are observed. Sequence of these termination flags occurred as a result of different actions as listed in Table 1, performed in the Web browser.

4.1 Uninterrupted transfers

The bar charts in Figures 2–7 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
Figure 4: Termination flags with text Webpage on Android

Figure 5: Termination flags with video Webpage on Android

of termination follows the rules as described by the standards [5].

**Client-side RST flag** On the Windows platform however, the text-based Webpage 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 in-
Figure 6: Termination flags with text Webpage on Windows

Figure 7: Termination flags with video Webpage on Windows

An interesting pattern seen in the connection termination process, as shown in Figure 8. After receiving the base file, the client makes a GET request for the video player. It then immediately terminates the connection 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.

![Figure 8: Video transfer on the Windows platform](image)

### 4.2 Interrupted transfers

Interrupted transfers are those in which a user aborts an on-going transfer by manually performing either 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 2–7 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
Table 3: Statistics of Time-to-Termination

<table>
<thead>
<tr>
<th>Platform/Content</th>
<th>Uninterrupted $\tilde{t}$</th>
<th>$\sigma_t$</th>
<th>Interrupted $\tilde{t}$</th>
<th>$\sigma_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android/Text</td>
<td>5.52 s</td>
<td>0.004 s</td>
<td>0.10 s</td>
<td>0.16 s</td>
</tr>
<tr>
<td>Android/Video</td>
<td>5.52 s</td>
<td>0.004 s</td>
<td>0.01 s</td>
<td>0.002 s</td>
</tr>
<tr>
<td>Symbian/Text</td>
<td>5.65 s</td>
<td>0.09 s</td>
<td>0.78 s</td>
<td>0.36 s</td>
</tr>
<tr>
<td>Symbian/Video</td>
<td>5.52 s</td>
<td>0.03 s</td>
<td>0.01 s</td>
<td>0.02 s</td>
</tr>
<tr>
<td>Windows/Text</td>
<td>5.58 s</td>
<td>0.04 s</td>
<td>0.19 s</td>
<td>0.02 s</td>
</tr>
<tr>
<td>Windows/Video</td>
<td>0.01 s</td>
<td>0.02 s</td>
<td>0.01 s</td>
<td>0.003 s</td>
</tr>
</tbody>
</table>

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 termination process with a FIN flag, then the server responds with the retransmission of previous unacknowledged segments. On 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. Another important evidence about the user-generated interruption is that more than one consecutive RST flags was seen in most of the cases as soon as the user performs an interruption in the Web browser.

**Indication of non-user-interrupted transfers:** 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.
5 Time-to-termination

To further understand the difference between user-interrupted and client-side application-interrupted connections, we present the time-to-termination $t_t$, which refers to the time elapsed from the last data packet from server $t_s$ to the first termination flag $t_f$ observed:

$$t_t = t_f - t_s$$  \hspace{1cm} (1)

Table 3 lists the mean time-to-termination ($\bar{t}_t$) and standard deviation ($\sigma_t$) observed from our traces for interrupted and uninterrupted experiments. The mean time-to-termination $\bar{t}_t$ is taken by averaging the time-to-terminations $t_t$ of 40 repetitions of each experiment, based on which the standard deviation $\sigma_t$ of an experiment is calculated.

In the case of uninterrupted connections, $\bar{t}_t$ is around 5.5 s. These terminations mainly occurred as a result of FIN flag from the server. It shows the time-out value of the Apache Web server that closes a persistent connection roughly after 5 s of idle time. When the video-based Webpage is requested on the Windows platform, $\bar{t}_t$ reduces to 0.01 s. This is because two connections are opened when the video-based Webpage is called. The first connection terminates with $\bar{t}_t$ of 0.01 s and the next connection with $\bar{t}_t$ of around 5.5 s. Hence, in this case it becomes difficult to differentiate between user-interrupted and client-side application-interrupted connections based on the time-to-termination.

For interrupted connections, we observe a notable difference between the values of time-to-termination due to the requested content type. When text pages are requested, the first terminating flag is seen 0.1 – 0.8 s after the last data packet from the server. In contrast, these values are reduced to 0.01 s, when video pages are requested. This reduction in $\bar{t}_t$ is because the inter-arrival time between packets is smaller than the ones observed in the case of text page and therefore, more packets are in flight when the user interrupts the transfer.

The value of standard deviation in most of the cases is very small i.e. below 100 ms. However, it gets higher when the Web pages containing the text are
aborted on the Android and the Symbian platforms. It is because the download
time in the case of text pages is very small (i.e. almost around 1 s) therefore, the
time required for the user to interrupt the transfer is also very small. Sometimes
the page is interrupted at the middle of the transfer (when many data packets are
in flight), and sometimes it is interrupted at the tail of the transfer (when not many
packets are in flight). Due to this varying user behavior, we observe the higher
standard deviation in the time-to-terminations.

6 Conclusion

In this paper, we studied the TCP connection termination process resulting
from different user actions in various Web browsers. We found out that the con-
nection termination behavior is heavily dependent on the type of Web browser
used. Therefore, the TCP RST flag alone, cannot be used to detect the user ac-
tion performed in the Web browser. The results presented in this study provide
a baseline for the development of a set of criteria for monitoring user-perceived
performance of networks.

Along the way, we also deduct from our observations that a better manage-
ment approach is needed between client-side Web browsers and servers to improve
the performance of TCP connections. Hence, the client-side implementation of
TCP connections according to the standards and close cooperation between Web
browsers and servers can be helpful in raising the Web browsing experience of
users.

7 Future work

Our short-term future work includes the study of the TCP termination process
with other popular Web browsers. We will then propose a set of criteria for the
passive monitoring of the user-perceived performance of Web transfers. Tests
will subsequently be conducted with real users for the validation of the criteria.
Finally, the criteria will be used to model the network performance as perceived
by the users.
Acknowledgment

This work has been supported by the FP7 Network of Excellence Euro-NF (contract number 216366). We would like to thank Prasanna Amburu, Anil Verma, Anil Kumar and Dileesh Kunpuru for their continuous help in conducting these experiments.


Part IV
Part IV

User Impatience and Network Performance
Part IV is submitted as:


²Slight formatting adjustments are made in tables, figures and references.
Abstract

In this work, we analyze from passive measurements the correlations between the user-induced interruptions of TCP connections and different end-to-end performance metrics. The aim of this study is to assess the possibility for a network operator to take into account the customers’ experience for network monitoring. We first observe that the usual connection-level performance metrics of the interrupted connections are not very different, and sometimes better than those of normal connections. However, the request-level performance metrics show stronger correlations between the interruption rates and the network quality-of-service. Furthermore, we show that the user impatience could also be used to characterize the relative sensitivity of data applications to various network performance metrics.

1 Introduction

During the recent years, several subjective user studies have been conducted, in order to correlate the user satisfaction with the network Quality of Service (QoS). Unfortunately, those classical methods are expensive and consume a lot of time. Moreover, they are unable to cope with the fast evolving nature of the Internet usage. Our objective therefore is to define a relatively quicker and cheaper way to correlate the network performance and user impatience, considered as an insight into the actual users’ satisfaction. Specifically, the objective of this paper is to show that the user-driven interruptions of TCP connections are actually correlated with some bad end-to-end performance metrics.

The notion of Quality of Experience (QoE) emerged during the last few years, with the intention to involve the subjective view of the end-users into the evaluation of the quality of telecommunication services. ITU-T SG12 defines QoE as the overall acceptability of a service by the end-user. Moreover, it can also be described as the measure of the satisfaction of the customers [13].

Usually, methods of assessing user satisfaction are based on the periodic polls sent to the customers, which incur delays and cost to acquire the user feedback. Moreover, this feedback is not instantaneous, it varies due to customer mood, and changes over time. So the correlations with highly varying network performance
are not very precise. Another faster method is to consider the user complaints about the network services, made in the form of calls to the hotline. However, relying on this approach of acquiring the user feedback is not very effective as the disappointed users may already churn, i.e. choose another competitor in the market, instead of reporting the problems. Consequently, there is a need for a proactive method to assess QoE.

Additionally, there are other methods, which use statistical “known correlations” between objectively measurable metrics. Such methods have been applied and standardized for voice and then for video: with MOS (Mean Opinion Score) [12], PESQ (Perceptual Evaluation of Speech Quality) [9], PEVQ (Perceptual Evaluation of Video Quality) [10], and many other similar approaches. Such correlations are typically evaluated on representative panels of customers, in well-defined environments that offer full control over experimental conditions. However, such costly lab experiments do neglect the multitude of influence factors on QoE [4]. They are mainly designed for the improvement of a particular application or to predict the quality during the network planning phase [15]. The execution of these experiments takes time and meanwhile, new applications, and versions of applications, appear all the time, with new traffic characteristics and new performance requirements. Moreover, the behavior of the customers and their sensitivity to the QoS evolves with the time. Hence, there is a need for a cheap and fast method to assess in real-time the satisfaction level of the users, being good enough to detect problems, in a proactive manner.

In this paper, we propose to consider the client interruptions of TCP connections, as users bothered by end-to-end performance are more likely to “interrupt” their transfers. Some recent papers [3] [5] [6] require an explicit signal from the user to launch punctual traffic measurements and compare the performance with usual conditions. Such methods need the installation of a tool on the users’ host, and only few (expert) users actually implement it.

Considering the last flags of TCP connections to detect interruptions gives a similar signal for all the connected users without installing any tool on customer’s side. We must nevertheless consider TCP interrupts cautiously keeping in mind some facts:

- the algorithm [2] we use to detect interruptions is only a heuristic, based
on TCP flags exchanged in the last packets of the connection, as detailed in subsection 2.2;

- users may abort a transfer for other reasons besides of the network performance. For example, some studies in the past reported that most of the streaming sessions are interrupted due to the lack of interest in the (end of the) content [16] [17] [18]. Nevertheless, the detection of such interruptions are still important for the network operators, as the amount of content downloaded but not watched is an overhead for the network [19];

- bad TCP implementation of the application on the client or on the server side may also appear as client TCP interruptions [11] [12].

The objective of this document is then to validate the possibility of using TCP interrupts to detect bad user experience due to the degrading network performance. In the next sections, we look at the correlation between aborted TCP connections and their end-to-end performance metrics. We expect to observe an increase of the interruption rate in case of bad performance.

We first describe in Section 2 the capture and the method we used to detect the interrupted TCP connections. We then analyze in Section 3.1 the differences between the TCP connections according to their type of end. Subsequently, in Section 3.1 we apply a similar analysis separately on the main application types. We then analyze in Section 4 the differences between normal and interrupted connections on a smaller timescale closer to user perception. We conclude on the interest of TCP interruptions for network monitoring in the last section.

2 Methodology

2.1 Traffic Measurement

We analyze in this document a traffic trace taken with DAG$^TM$ cards during a busy hour, at 5pm on a weekday, Tuesday 27th April 2011. The probe captures the whole data traffic of a BAS (Broadband Access Server), between DSLAM (Digital Subscriber Line Access Multiplexers) and the first routers towards the
Internet as shown on Figure 1. Data, TV and Voice over IP traffic are transmitted on different VLANs (Virtual Local Area Networks). Our traffic trace is limited to the data traffic. The main traffic characteristics of this capture are shown in the last line in Table 1.

Figure 1: The architecture of the ADSL access network

### 2.2 Detection of interrupted TCP connections

Rossi et al. have proposed in [2] the following method based on the last TCP flags to infer whether the client has interrupted a connection.

**Eligible connections:** The “eligible” connections are by definition those respecting the three following conditions:

- the client has sent at least one packet with the RST (ReSeT) flag;
- non-empty packets have been exchanged;
- and the server has not sent any packet with the flag FIN (FINish) or the flag RST.

**Interrupted connections:** The connections classified as “interrupted” are by definition the eligible connections such that the delay between the last non-empty
packets sent respectively by the client and by the server is smaller than the average round-trip time.

**Normal connections:** The normal connections are those ending with a FIN flag in each direction.

**Unfinished connections:** The connections without any FIN or RST flag are considered unfinished at the end of the traffic trace.

**Abnormal connections:** The connections that do not fall in any of the above category are classified as “abnormal” connections.

We list in Table 1 the relative part of each of the above mentioned type of termination, found in the captured traffic.

### Table 1: TRAFFIC BY TYPE OF TERMINATION

<table>
<thead>
<tr>
<th></th>
<th>Packets (x10⁶)</th>
<th>Traffic (GB)</th>
<th>Connections (x10³)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Down</td>
<td>up</td>
<td>Down</td>
</tr>
<tr>
<td>Normal</td>
<td>44</td>
<td>29</td>
<td>42</td>
</tr>
<tr>
<td>Eligible</td>
<td>6.8</td>
<td>7%</td>
<td>4.3</td>
</tr>
<tr>
<td>Interrupted</td>
<td>13</td>
<td>13%</td>
<td>7.4</td>
</tr>
<tr>
<td>Abnormal</td>
<td>17</td>
<td>17%</td>
<td>12</td>
</tr>
<tr>
<td>Unfinished</td>
<td>8.9</td>
<td>9%</td>
<td>6.2</td>
</tr>
<tr>
<td>Non TCP</td>
<td>11</td>
<td>11%</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>101</td>
<td>69</td>
<td>115</td>
</tr>
</tbody>
</table>

### 3 Connection-Level Analysis

#### 3.1 Global analysis

We consider in this section the difference in the performance between the normal and the interrupted connections. We first compare the mean throughput of the connections classified according to their type of termination. The effective throughput is obtained from the total amount of data in the connection divided by the connection duration, reduced by silence periods larger than five seconds.
The *instantaneous throughput* is calculated by counting the number of bits transferred during one measured Round Trip Time (RTT), as reported by the number of bytes in the acknowledgement. We observe from Table 2 that the upward throughputs are slightly lower for the interrupted connections. However, the downward throughputs that are expected to influence the user satisfaction in a stronger way are surprisingly much larger for the interrupted connections.

<table>
<thead>
<tr>
<th>Type of termination</th>
<th>Mean Throughput Up</th>
<th>Effective Throughput Up</th>
<th>Instantaneous Throughput Up</th>
<th>Mean Throughput Down</th>
<th>Effective Throughput Down</th>
<th>Instantaneous Throughput Down</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>24</td>
<td>28</td>
<td>1612</td>
<td>67</td>
<td>90</td>
<td>431</td>
</tr>
<tr>
<td>Interrupted</td>
<td>22</td>
<td>28</td>
<td>1006</td>
<td>515</td>
<td>536</td>
<td>1158</td>
</tr>
</tbody>
</table>

**H1)** A first hypothesis is that the throughput of the interrupted connections is good, but that other performance criteria bother the customers.

**H2)** A second hypothesis is that the termination behavior of the TCP connections depends on the application and its implementation on the client and/or on the server side. Some applications may have higher throughputs and for some reasons higher interruption rates, which may impact these global measurements. We consider this hypothesis in the next section. Similarly, some customers or some servers with high throughputs may be subjected to more TCP interruptions.

**H3)** A third hypothesis is that, during high throughput transfers, the users may be more sensitive to the performance degradation. We look at this hypothesis in section 4.

We consider now the influence of the end-to-end packet losses on the TCP interruptions. More precisely, we consider a rough estimate of the packets lost by the TCP connections: the number of out-of-sequence packets divided by the total number of data packets. Figure fig:lossRate shows the upward and downward loss rates of TCP connections according to their type of termination. We remark that
the interrupted connections observe much more packet losses: 17% of interrupted downward connections observed loss vs. 1.6% of normal ones. Interrupted connections also observe higher loss rates than normal connections as the curves are shifted to the right. We however note that most of the interrupted connections (83%) do not observe any loss. So, a high loss rate may explain TCP interruptions for some, but certainly not in most cases.

![Figure 2: Mean loss rate of TCP associations](image)

Figure 3 presents the local and the distant round-trip times for TCP connections, according to their type of termination. We define the local (resp. distant) round-trip time as the delay seen by the probe on downward (resp. upward) connections between a data packet and its acknowledgment. Retransmissions are not taken into account. We remark in Figure 3 that the local round-trip times of the interrupted connections are a bit larger, as their distributions are shifted to the right. The distant round-trip times of interrupted and normal connections appear quite similar. The fact that the main difference between the interrupted and normal connections is on the local loop might indicate that a possible cause of interruptions could be local congestion due to the presence of multiple parallel TCP connections on the access link of the customer.

From the global analysis presented in this section and considering some criteria like local round-trip times or packet loss, we observe that interrupted connec-
tions globally observe lower performance than normal connections. But we also notice more surprising results: the interrupted connections have usually larger throughput, at different timescales, and even packet losses and round-trip times are not sufficient to explain the TCP interruptions. We have then considered many hypotheses to explain these surprising results. The first possible explanation (H1) should be rejected as the most interrupted connections do not observe packet losses, and the distributions of the local and the distant round-trip times of interrupted and normal TCP connections are not dramatically different. A first conclusion, which we draw from the above findings, is that the global performance metrics are not sufficient to explain why the TCP connections are interrupted. We will check the other hypotheses in the following sections.

3.2 Analysis per application

We consider more precisely in this subsection the differences between the main applications, especially their interruption rate and their performance. Connections are classified in applications by an Orange-internal DPI (Deep Packet Inspection) tool. This tool has been compared with Tstat in [7]. The list of identified applications detected by this tool is shown in Table 3.
Table 3: APPLICATIONS CLASSES

<table>
<thead>
<tr>
<th>Class</th>
<th>Application / Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB</td>
<td>HTTP and HTTPS browsing</td>
</tr>
<tr>
<td>EDONKEY</td>
<td>eDonkey, eMule obfuscated</td>
</tr>
<tr>
<td>MAIL</td>
<td>SMTP, POP3, IMAP, IMAPs, POP3s, HTTP Mail</td>
</tr>
<tr>
<td>CHAT</td>
<td>MSN, IRC, Jabber, Yahoo Msn, HTTP Chat</td>
</tr>
<tr>
<td>OTHERS</td>
<td>NBS, Ms-ds, Epmap, Attacks</td>
</tr>
<tr>
<td>DB</td>
<td>LDAP, Microsoft SQL, Oracle SQL, mySQL</td>
</tr>
<tr>
<td>BITTORRENT</td>
<td>Bittorrent</td>
</tr>
<tr>
<td>DOWNLOAD</td>
<td>FTP data, FTP control, HTTP file transfer</td>
</tr>
<tr>
<td>GAMES</td>
<td>Blizzard Battlenet, Quake II/III, Counter Strike, HTTP Games</td>
</tr>
<tr>
<td>STREAMING</td>
<td>MS Media Server, Real Player, iTunes, Quick Time</td>
</tr>
<tr>
<td>GNUTELLA</td>
<td>Gnutella</td>
</tr>
<tr>
<td>ARES</td>
<td>Ares</td>
</tr>
<tr>
<td>TRIBALL</td>
<td>Triball</td>
</tr>
<tr>
<td>P2P-REST</td>
<td>Kazaa, SoulSeek, Filetopia, Others</td>
</tr>
<tr>
<td>NEWS</td>
<td>Ntp</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td></td>
</tr>
</tbody>
</table>

We mainly observe in our traffic traces the eight types of applications listed in Table 4. Some connections are not recognized and therefore, denoted as “unknown”. The relative part of each application, according to the number of connections, of up and down packets and bytes, and the interruption rate are given in Table 4. We notice that these various metrics display rather different proportions. For example, Streaming represents only 1.6% of connections, but 42% of bytes downward. Download represents 0.4% of connections but 14% of bytes downward. Conversely, recognized P2P represents (similarly to the unknown connections) 31% of connections and only 10% of the bytes downward. Hence, the Streaming and the Download connections are much larger in size than the connections of the other applications.
Table 4: TRAFFIC PER APPLICATION

<table>
<thead>
<tr>
<th>Application</th>
<th>Packets</th>
<th>Traffic (GB)</th>
<th>Connections</th>
<th>Interruptions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>down</td>
<td>up</td>
<td>down</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td>(\times 10^6)</td>
<td>%</td>
<td>(\times 10^6)</td>
<td>%</td>
</tr>
<tr>
<td>Streaming</td>
<td>34</td>
<td>33</td>
<td>18</td>
<td>25</td>
</tr>
<tr>
<td>Web</td>
<td>27</td>
<td>27</td>
<td>20</td>
<td>29</td>
</tr>
<tr>
<td>P2P</td>
<td>15</td>
<td>15</td>
<td>14</td>
<td>20</td>
</tr>
<tr>
<td>Download</td>
<td>11</td>
<td>11</td>
<td>5.9</td>
<td>8.5</td>
</tr>
<tr>
<td>Unknown</td>
<td>7.1</td>
<td>7.0</td>
<td>6.8</td>
<td>9.8</td>
</tr>
<tr>
<td>Games</td>
<td>2.8</td>
<td>2.7</td>
<td>1.9</td>
<td>2.7</td>
</tr>
<tr>
<td>Mail</td>
<td>1.5</td>
<td>1.5</td>
<td>1.3</td>
<td>1.9</td>
</tr>
<tr>
<td>VOIP</td>
<td>0.6</td>
<td>0.6</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>Chat</td>
<td>0.3</td>
<td>0.3</td>
<td>0.2</td>
<td>0.3</td>
</tr>
<tr>
<td>total</td>
<td>100.7</td>
<td>68.8</td>
<td>114.9</td>
<td>15.2</td>
</tr>
</tbody>
</table>

We remark in the last column that the interruption rate is highly correlated with the application: three applications have much larger interruption rates than the other ones: Chat (15%), Streaming (10%) and Download (5%). However, in general, the mean interruption rate for all the TCP connections is 0.6%, much smaller than the interruption rate of the individual applications: Chat, Streaming and Download.

Table 5 provides for each application, the mean performance metrics for the normal and the interrupted TCP connections. A common observation to almost all of the applications is that, the mean downward throughput of the interrupted connections is usually larger than that of normal connections. The other results are more specific to each application.

1. **Chat**

   The interrupted chat connections have higher upward and downward throughputs, lower local and distant round-trip times, and higher loss rates. So, the
Table 5: **MEAN VALUES OF TRAFFIC AND PERFORMANCE METRICS OF CONNECTIONS ACCORDING TO THE APPLICATION AND THE TERMINATION OF THE CONNECTION.**

<table>
<thead>
<tr>
<th>application</th>
<th>termination</th>
<th>Mean throughput (kbits/s)</th>
<th>Round trip time (ms)</th>
<th>Packet loss rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>down</td>
<td>up</td>
<td>distant</td>
</tr>
<tr>
<td>Chat</td>
<td>normal</td>
<td>15</td>
<td>6</td>
<td>125</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>68</td>
<td>26</td>
<td>113</td>
</tr>
<tr>
<td>Download</td>
<td>normal</td>
<td>376</td>
<td>21</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>538</td>
<td>15</td>
<td>69</td>
</tr>
<tr>
<td>Games</td>
<td>normal</td>
<td>55</td>
<td>18</td>
<td>126</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>326</td>
<td>8</td>
<td>957</td>
</tr>
<tr>
<td>Mail</td>
<td>normal</td>
<td>56</td>
<td>25</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>144</td>
<td>37</td>
<td>26</td>
</tr>
<tr>
<td>P2P</td>
<td>normal</td>
<td>4</td>
<td>4</td>
<td>271</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>12</td>
<td>8</td>
<td>572</td>
</tr>
<tr>
<td>Streaming</td>
<td>normal</td>
<td>257</td>
<td>13</td>
<td>77</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>1501</td>
<td>30</td>
<td>29</td>
</tr>
<tr>
<td>VOIP</td>
<td>normal</td>
<td>23</td>
<td>9</td>
<td>134</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>97</td>
<td>21</td>
<td>160</td>
</tr>
<tr>
<td>Web</td>
<td>normal</td>
<td>72</td>
<td>30</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>interrupted</td>
<td>180</td>
<td>26</td>
<td>50</td>
</tr>
</tbody>
</table>

Loss rate per application is the sole performance criteria which may explain an interruption of the TCP connections. As not all the interrupted chat connections observe packet loss, and as the interruption rate of chat connections is much higher than for the other applications, we should assume some other reasons behind the interrupted chat connections. Maybe chat user usually close the window once they are done with the chatting, thereby interrupting a TCP connection in the end.

2. **Download**

We further notice that the upward throughputs of interrupted download connections are 30% below those of the normal ones. In contrast, the downward
throughputs, which should expectedly be more important for the users, are higher in the case of interrupted connections. Therefore, the throughput degradation should not be the primary reason for the interruption. On the other hand, we observe that the local round-trip times of the interrupted connections are much larger. Although their mean distant round-trip time is relatively a bit smaller, their combined sum – the end-to-end round-trip time – is however 70% larger. The loss rates of the interrupted download connections in both directions are also slightly higher than those of the normal download connections.

3. *Games*

Similarly, the downward throughputs of interrupted Games connections are larger than those of the normal ones, but the ratio is largely below the one for upward throughputs which may be a significant criterion for this application. Both up and down loss rates are slightly higher for interrupted Games connections, but local and distant round-trip times are much larger. Hence, interruption of Games connection seems then to be correlated with large end-to-end round-trip times, low upward throughputs and high up and downward loss rates.

4. *Mail*

All average up and down throughputs of the interrupted Mail connections are larger than those of the normal Mail connections. The mean round-trip times are slightly larger, with the loss rates in both directions almost doubled. Hence, the interruptions of Mail connections appear to be correlated with the increasing end-to-end round-trip times and loss rates.

5. *P2P*

The conclusions for P2P are similar to those of Mail, with the supplementary observation of considerably low instantaneous upward throughput of the interrupted connections.

6. *Streaming*
For streaming transfers, all the average up- and downward throughputs of interrupted connections are larger than those of the normal connections. The local mean round-trip time of the interrupted connections is slightly higher, while the distant one is significantly smaller as compared to the normal connections. The upward loss rate is exactly similar while the downward loss rate is slightly (30%) higher. The interruption of streaming can not be clearly correlated with the network performance. There appear to be two possible reasons behind this observation. First, interruptions could be due to the implementation of streaming clients. We observed in [11] that one of the client-side web browser interrupts a TCP connection as soon as the request for the video container is made, henceforth, initiating a new TCP connection to play the video. Second, users could interrupt the connections, which may be for example due to the lack of their interest in the content of the video.

7. **VOIP**

Round-trip times of the interrupted VOIP connections are a bit (20%) larger, but loss rates are smaller, and most of the average throughputs are larger, except the upward instantaneous throughput. Hence, the VOIP users appear to be more sensitive to the round-trip times.

8. **Web**

Finally, downward throughputs of the interrupted Web connections are larger, but upward throughputs are slightly smaller than those of the normal connections. Both up and downward loss rates appear to be almost similar. However, the round-trip times are significantly larger for the interrupted connections. Consequently, the round-trip times seem to be the most important user-perceived performance criterion for this kind of interactive application, as it influences directly the response time of a webpage.

Summarizing the results listed in Table 5, the download and the Streaming connections have the highest throughputs and also high interruption rates. This explains the surprising observation that we discussed in the previous section. Another obvious conclusion that further confirms the findings of the previous works
is that the sensitivity of the user to the end-to-end performance actually depends on the application.

4 TCP connection breakdown

The objective of the analysis presented in this section is to get a look closer to the user perception. Instead of considering connection-level metrics assessed on the whole duration of TCP connections, we consider transfer-level metrics for the successive alternate exchanges between the end-hosts. Those exchanges are usually due to the requests from the local client followed by answers from the distant server. We have indeed noticed in [1] that roughly 90% of the connections observe an alternate exchange of groups of packets (blocks, trains...) from one end of the connection and then from the other. We consider only this kind of connections in the rest of this subsection. For these connections, we propose in [1] to differentiate on each side of the transfer the delay to prepare the data from the delay to transfer it.

More precisely, we distinguish:

- The think-time, or warm-up, which is the silence between the last data packet from the previous train in the other direction and the sending of the first data packet of the next train. On the server side, this silence may be used to prepare the data to send, to request information from other servers, etc. On the user side, it may by used to read the information received and think about a next eventual request. The server-side warm-up times are referred to as the distant warm-up times in this paper.

- The transmission delay, during which packets are on-line, which elapse from the date of departure of the first packet of the train to the date of arrival of the last packet of the train.

Assuming that the user should not be bothered by its own think-time, Table 6 considers the distant warm-up time. In particular, it presents the mean and the standard deviation, of all the distant warm-up times. It also lists the mean of the ratio per connection of the last to the mean of all distant warm-up times. We only
consider the distant warm-up assuming that the user should not be bothered by its own think-time. We observe from this table that the distant warm-up of the interrupted connections is usually not significantly larger than the distant warm-up of the normal connections. It is 10% larger for Web and Mail, 40% larger for P2P, but it is slightly smaller for Games and VOIP, and much more for Download, Streaming and Chat. We observe similarly that the variations of the warm-up delays are also much smaller for the interrupted connections, except for P2P for which they are 50% larger. On the third part of the table, we notice that the last distant warm-up of the interrupted connections is roughly larger than for the normal connections for P2P and Games (roughly twice), and especially for Mail (more than five times).

We observe similar differences between the normal and the interrupted connections when looking more rigorously at the distributions of the warm-ups in—
stead of simply looking at the mean and the standard deviations. Unfortunately, due to the lack of space, we cannot include all the distributions in this paper. As examples, we display only the distributions for the upward and the downward transfer durations of Web and Games connections Figure 5.

Table 6: MEAN, STANDARD DEVIATION AND RATIO OF LAST TO MEAN DISTANT WARM-UPS ACCORDING TO THE TERMINATION.

<table>
<thead>
<tr>
<th>Distance</th>
<th>Normal mean (s)</th>
<th>Normal standard deviation (s)</th>
<th>Interrupted mean (s)</th>
<th>Interrupted standard deviation (s)</th>
<th>Last/mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web</td>
<td>0.7</td>
<td>0.7</td>
<td>4.2</td>
<td>2.0</td>
<td>3.6/2.7</td>
</tr>
<tr>
<td>P2P</td>
<td>0.5</td>
<td>0.7</td>
<td>0.8</td>
<td>1.2</td>
<td>5.6/11.0</td>
</tr>
<tr>
<td>Download</td>
<td>1.0</td>
<td>0.3</td>
<td>2.6</td>
<td>1.0</td>
<td>0.8/0.7</td>
</tr>
<tr>
<td>Mail</td>
<td>0.6</td>
<td>0.7</td>
<td>1.9</td>
<td>1.8</td>
<td>2.6/14.5</td>
</tr>
<tr>
<td>Games</td>
<td>0.3</td>
<td>0.2</td>
<td>1.6</td>
<td>0.5</td>
<td>1.2/3.0</td>
</tr>
<tr>
<td>Streaming</td>
<td>1.3</td>
<td>0.3</td>
<td>6.1</td>
<td>0.8</td>
<td>3.2/1.3</td>
</tr>
<tr>
<td>Chat</td>
<td>4.1</td>
<td>0.4</td>
<td>10.7</td>
<td>0.7</td>
<td>11.1/1.1</td>
</tr>
<tr>
<td>VOIP</td>
<td>0.8</td>
<td>0.6</td>
<td>2.7</td>
<td>1.1</td>
<td>19.8/1.2</td>
</tr>
</tbody>
</table>

Table 7 shows the mean, the standard deviation of the transfer delays of up and downward trains, and the ratio of the last transfer delay to the mean transfer delay. We first remark that for all the applications, the warm-up delays and the transfer delays are found in the same order of magnitude, namely from a few hundreds of milliseconds to a few seconds. The transfer durations in both directions are also found within the same magnitude. We clearly notice in Table 7 that for all the applications (except of VOIP and Chat), the interrupted connections observe much larger and much more varying transfer durations in both directions than the normal connections. Their last transfer duration is also many times larger, which points at some kind of starvation problem.

Finally, we define the response time as the delay between the time of the first packet of a request sent by a user and the last packet of the answer to this request, received on the client-side. The response time is similarly defined as the sum of the up and downward transfer delays and of the distant warm-up delay. Table 8
User Impatience and Network Performance

Figure 5: Compared distributions transfer durations of normal and interrupted Web and Games connections

presents the mean, the standard deviation and the ratio of the last to the mean response times, according to the termination and to the application. We notice in Table 8 that the interrupted connections of all the applications except of chat and VOIP, experience larger response times as compared the normal ones. We observe that for these applications, the part of response time due to the network delays as well as the ratio of the last response time to the mean are also larger. We observe different results considering the standard deviations of the response time. In this case, we remark that VOIP interrupted connections observe larger variations than normal connections. On the contrary, the variations of the response times of Web and Streaming interrupted connections are not significantly different from their normal connections. These observations may indicate that Web and Streaming users are more bothered by the large response times, while VOIP users are more disturbed by the varying response times. For most of the applications, the users appear to be very sensitive to the response times. Hence, the interrupted connections in these cases can be taken as an indication of the longer response times. It reflects that the users are more critical about the response times of the requested content. They become impatient and click away, if the transfers take slightly longer, as mentioned in [14]. They spend a constant time on the network,
expecting the content to be delivered during this time [21]. They abandon the transfers, if it takes more than their expected time.

Table 7: MEAN, STANDARD DEVIATION AND RATIO OF LAST TO MEAN DISTANT WARM-UPS ACCORDING TO THE TERMINATION.

<table>
<thead>
<tr>
<th>Downward transfer durations</th>
<th>mean (s)</th>
<th>standard deviation (s)</th>
<th>last/mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>normal</td>
<td>interrupted</td>
<td>normal</td>
</tr>
<tr>
<td>Web</td>
<td>0.3</td>
<td>1.7</td>
<td>0.7</td>
</tr>
<tr>
<td>P2P</td>
<td>1.0</td>
<td>6.1</td>
<td>2.8</td>
</tr>
<tr>
<td>Download</td>
<td>1.2</td>
<td>21.7</td>
<td>2.8</td>
</tr>
<tr>
<td>Mail</td>
<td>0.1</td>
<td>2.7</td>
<td>0.4</td>
</tr>
<tr>
<td>Games</td>
<td>0.2</td>
<td>2.3</td>
<td>0.7</td>
</tr>
<tr>
<td>Streaming</td>
<td>1.2</td>
<td>5.4</td>
<td>2.1</td>
</tr>
<tr>
<td>Chat</td>
<td>3.0</td>
<td>0.3</td>
<td>8.0</td>
</tr>
<tr>
<td>VOIP</td>
<td>1.0</td>
<td>1.1</td>
<td>2.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Upward transfer durations</th>
<th>mean (s)</th>
<th>standard deviation (s)</th>
<th>last/mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>normal</td>
<td>interrupted</td>
<td>normal</td>
</tr>
<tr>
<td>Web</td>
<td>0.2</td>
<td>1.2</td>
<td>0.5</td>
</tr>
<tr>
<td>P2P</td>
<td>1.2</td>
<td>2.9</td>
<td>3.4</td>
</tr>
<tr>
<td>Download</td>
<td>2.5</td>
<td>18.9</td>
<td>5.0</td>
</tr>
<tr>
<td>Mail</td>
<td>0.3</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td>Games</td>
<td>0.3</td>
<td>6.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Streaming</td>
<td>4.0</td>
<td>9.3</td>
<td>7.0</td>
</tr>
<tr>
<td>Chat</td>
<td>1.7</td>
<td>0.2</td>
<td>5.7</td>
</tr>
<tr>
<td>VOIP</td>
<td>1.2</td>
<td>0.3</td>
<td>3.6</td>
</tr>
</tbody>
</table>

5 Conclusion

The objective of this study was to check if the interruption of TCP connections is actually correlated with the bad network performance. Our main conclusion is
that this is actually true. Moreover, a second and the important conclusion is that the breakdown of TCP connections in alternate packet trains is critical in detecting this correlation for the most applications. We have indeed observed that the usual connection-level performance metrics of the interrupted connections are not very different, and sometimes better than those of the normal connections. However, considering separately the main applications and especially the request-level performance metrics show much better correlations between the interruption rate and the network QoS, and it also explains some paradoxical results observed earlier. A third conclusion is that diverse applications have different sensitivity to the various performance criteria.

Further studies should try to point out the root causes [20] [1] of the performance problems disturbing users according to these various applications, in order to propose some ways to reduce their impact and maximize the customer experience. The influence of the various performance criteria on the QoE considered in this document could also be more precisely characterized to define the utility functions, which may further be used to design and optimize the network and the services, taking into account the customer experience. The results observed should also be validated by comparison with similar traffic analyses in other conditions, such as others hours-of-day or in the context of wireless access networks.

Table 8: MEAN, STANDARD DEVIATION AND RATIO OF LAST TO MEAN RESPONSE TIMES ACCORDING TO THE TERMINATION.

<table>
<thead>
<tr>
<th>Response time</th>
<th>mean (s)</th>
<th>standard deviation (s)</th>
<th>last/mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>normal</td>
<td>interrupted</td>
<td>normal</td>
</tr>
<tr>
<td>Web</td>
<td>1.3</td>
<td>3.6</td>
<td>4.3</td>
</tr>
<tr>
<td>P2P</td>
<td>2.7</td>
<td>9.7</td>
<td>2.9</td>
</tr>
<tr>
<td>Download</td>
<td>4.7</td>
<td>40.9</td>
<td>3.9</td>
</tr>
<tr>
<td>Mail</td>
<td>1.1</td>
<td>3.6</td>
<td>1.9</td>
</tr>
<tr>
<td>Games</td>
<td>0.8</td>
<td>8.5</td>
<td>1.8</td>
</tr>
<tr>
<td>Streaming</td>
<td>6.4</td>
<td>15.0</td>
<td>6.5</td>
</tr>
<tr>
<td>Chat</td>
<td>8.8</td>
<td>0.8</td>
<td>13.3</td>
</tr>
<tr>
<td>VOIP</td>
<td>3.0</td>
<td>2.1</td>
<td>3.8</td>
</tr>
</tbody>
</table>
Acknowledgements

This work has been partly sponsored by the European 7th Framework Networks of Excellence EuroNF (216366), and the COST project TMA (IC0703).

References


Part V

Modeling and Analysis of Web Usage and Experience Based on Link-Level Measurements
Part V is submitted as:

Abstract

Internet traffic monitoring and analysis has 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 make the following contributions. First, we will discuss the characteristics of the gaps caused due to the user inactivity on the web. Second, we will 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 the gaps and user-perceived problems but they didn’t quantify the boundary towards the think times. They decoded the stream and simulated 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 the 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 the web and third, it proposes a criterion to monitor and detect the presence of the traffic gaps on different timescales.

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 within 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.

Figure 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.

Active OFF times

When a user requests for 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 by the grey boxes in Figure 1.

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 the inter-request time less than 64 s. The appropriate value for the silent time threshold for the 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 1: User think times

<table>
<thead>
<tr>
<th>Literature</th>
<th>Active OFF times</th>
<th>Inactive OFF times</th>
<th>Silent time threshold</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>
</tr>
<tr>
<td>[5]</td>
<td>-</td>
<td>~ 15 s</td>
<td>-</td>
</tr>
<tr>
<td>[7]</td>
<td>&lt; 1 s</td>
<td>&gt; 8 s</td>
<td>100 – 1000 s</td>
</tr>
<tr>
<td>[9]</td>
<td>&lt; 1 s</td>
<td>~ 30 s</td>
<td>&gt; 1000 s</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 observe 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>Home Wi-Fi</th>
<th>Hotel Wi-Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total duration</td>
<td>180 s</td>
<td>180 s</td>
</tr>
<tr>
<td>Total OFF times</td>
<td>1</td>
<td>37</td>
</tr>
<tr>
<td>1 s – 2 s</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>2 s – 3 s</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3 s – 4 s</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Above 4 s</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>
Figure 3: Inter-packet times of video stream captured via Home Wi-Fi

Figure 4: Inter-packet times of video stream captured via Hotel Wi-Fi
Figure 4 illustrates the inter-packet times observed from the trace captured in the 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 because 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 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 don’t 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 on the link-level. To check the burstiness at dif-
ferent 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.

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:

![Figure 5: CCDF distribution of ON and OFF times for the transfer done via hotel Wi-Fi](image)

---

Note: The image contains a graph showing the CCDF distribution of ON and OFF times for the transfer done via hotel Wi-Fi.
Figure 6: Wavelet and scaling coefficients

\[ d_{j,k} = y_{2k} - y_{2k-1}, \]  

(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: \( d_{2,1} = y_2 - y_1 \), \( d_{2,2} = y_4 - y_3 \), \( d_{2,3} = y_6 - y_5 \) and \( d_{2,4} = y_8 - y_7 \). 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:

\[ c_{j,k} = y_{2k} + y_{2k-1}, \]  

(2)

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 normalization purposes, 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:

\[
\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} c(j,k)^2
\]

\[
\mu_j = \frac{1}{n_j} \sum_{k=1}^{n_j} d(j,k)^2
\]

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. 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).

![Wavelet Coefficient Spectrum](image.png)

**Figure 7: Spectrum of wavelet coefficients**

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 the 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.

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

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

The above-mentioned behavior is illustrated very well by the linear regres-
sions (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 re-
gression 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. How-
ever, 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 the
faster decay in the case of home transfer.
Figure 8: Spectrum of scaling coefficients

Table 4: Linear regressions between bands and the spectrum values of scaling coefficients

<table>
<thead>
<tr>
<th>Access Network</th>
<th>Linear regression</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotel Wi-Fi</td>
<td>$\mu = -0.796j + 9.36$, $r = -0.99$</td>
<td></td>
</tr>
<tr>
<td>Home Wi-Fi</td>
<td>$\mu = -0.954j + 11.48$, $r = -0.99$</td>
<td></td>
</tr>
</tbody>
</table>

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. The 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 the freezes in video transfers on web are 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 characterizes 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 the 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.


Chapter 3: Conclusions and Outlook

1 Conclusions

In this thesis, we analyzed objective indicators of QoE, which could be used in the development of non-intrusive network-based methods for the estimation of QoE. Acquiring subjective feedback of users is time-consuming, and difficult to be implemented in real-time. The objective indicators allow real-time estimation of user satisfaction through constant monitoring of user traffic characteristics. There are set of conclusions drawn from the results obtained in this thesis. These conclusions are described below.

First, the relative change in the user session volumes could be associated to the change in QoE. Indications were found that the users usually spend a constant time on the Internet. During this time, they surf more if they are happy with the service, which is seen from their increased session volumes. Hence, session volumes can be used as one of the indicators of QoE.

Second, the TCP termination behavior is heavily dependent on both the user actions as well as the type of web browser used. It was found that the users abort their requests on the Web more frequently if the performance of a network is low, which can be seen in the form of interrupted TCP connections. Hence, TCP termination behavior can also be used as an indicator of request-level QoE.

Third, the wavelet analysis can be used to quickly points towards those locations in the transfers, where degradation in the performance occur. The analysis assists in the visualization of ON and OFF phases at different timescales, which can further be used to differentiate between the user think times and the network-induced traffic gaps. Hence, the quick identification of network-induced gaps can be helpful in determining the degradation of QoE in real-time. From the qualita-
tive analysis of transfers, it was found that the presence of scaling, i.e., fluctuations in the network traffic, at the timescales above 1 s and its absence at the timescales below 1 s reflects frequent network-induced gaps of duration above 1 s.

2 Future work

In future, bridges need to be built between the discussed indicators and the subjective opinions of users. To achieve this, a comprehensive test strategy will be designed for conducting the subjective user experiments. Network-based indicators will be correlated with the subjective opinions of the users.

On one hand, TCP connection termination behavior will be compared to the obtained user ratings. The aim will to be to determine those TCP interruptions which provide true evidence about the degrading user experience.

On the other hand, traffic characteristics from the experiments will be analyzed on different timescales. Scaling behavior in traffic will subsequently be used to differentiate between the user, the server and the network behaviors. Based on these behaviors, the ON-OFF models will be further refined. The impact of duration and frequency of network OFF times will particularly be quantified against the user ratings and its effect on the user behavior will be analyzed based on the resulting usage patterns. Moreover, this work will also serve in bridging gaps between the classical teletraffic models and the QoE.

Finally, a hybrid mechanism based on TCP interruptions and ON-OFF models need be devised to assist in the passive monitoring of QoE in real-time on network level.
BIBLIOGRAPHY


