

Bridging Between Quality of Experience and Quality of Service Through TCP Flag Ratios

Bamshad Shirmohamadi
Blekinge Institute of Technology
Karlskrona, Sweden
bamshad.shm@gmail.com

Markus Fiedler
Blekinge Institute of Technology
Karlskrona, Sweden
markus.fiedler@bth.se

Abstract—Since the early days of Quality of Experience (QoE) research, relationships to Quality of Service (QoS) and corresponding QoE-QoS models have been of great interest to the research community. However, the manifold quality transformations throughout the communication stack, together with accessibility issues of QoS parameters inside networks, hamper availability, applicability and generality of such models. On this background, the paper proposes the use of specific transport layer information, in particular the ratios of selected flags used by the TCP protocol, as QoE indicators. In our work, we show that such flag ratios help identifying, characterizing and quantifying user annoyance from observations anywhere along the TCP connection. As use case, we address video streaming from a TCP CUBIC web server in a wireless setting.

Keywords—*Quality of Experience (QoE), Web server, TCP flags, Quality of Service (QoS), video streaming*

I. INTRODUCTION

We witness an ever-increasing number of web sites and services as well as web-based applications with big numbers of users, where “Web services are client and server applications that communicate over the World Wide Web’s (WWW) Hyper Text Transfer Protocol (HTTP)” [1]. We also witness an ever-increasing share and role of HTTP- and WWW-based video delivery. Obviously, understanding the user’s *Quality of Experience* (QoE) in terms of “degree of delight or annoyance” [2] is an essential key of success for these services. However, asking users to evaluate QoE directly through pop-up windows, questionnaires etc. may not be the best way to go, as the corresponding tool or procedure as such may cause user annoyance [3]. Indeed, there are proposals regarding alternative ways of QoE evaluation. One of the methodologies used in this case is *linking QoE to user actions*. In [4], the correlation between user happiness and their reactions to low video streaming quality is explained. According to their survey, “68.97% of users will pause the video and wait for a short time to expect video play more fluently when video is in buffering state” [4]. Another methodology consists in *linking user actions to observations of transport protocol activities*. In the article called “Classification of TCP Connection Termination Behaviors for Mobile Web” [5], the effect of user actions on Transport

Control Protocol (TCP) flags is described. For example, if a Windows user closes the browser (Microsoft Internet Explorer 6.0) before the page (with video content) is completely loaded one or more TCP packets with the RST flag set will be sent from the client, indicating some abnormal termination.

The combination of the two above approaches leads to our new method for indirect QoE assessment, based on observing the changes in the frequencies of TCP flags instead of asking the users directly. As TCP is the transport protocol for HTTP-based video delivery, its protocol messages provide a yet unexploited and rich source of information about potential delivery problems that may affect QoE. Indeed, this work is a first step to exploit that source of information in order to allow for a quantitative deduction of QoE issues by observing a TCP connection anywhere along its path, be it at the access or close to the server. Furthermore, we investigate the capability of the TCP flags to provide hints at potential reasons behind the possible user dissatisfaction. To sum up, our attention and modeling effort concentrates on quantitative relationships between QoE and TCP flag ratios. The latter are obtained through a combination of user surveys and traffic analyses respectively.

Thus, the contribution of this work is to outline a new means to discover QoE issues from observing control traffic properties at the transport level. By the usage of our approach, service and/or network management can obtain indications of QoE decrements and potential root causes. The approach works based on the web server’s captured traffic during specific time intervals anywhere along the network path that the video traffic takes. At the end of each time interval, by calculating the percentages of the specific TCP flags and comparing them to reference values, the approach can hint at potential loss of QoE and reasons behind it, and trigger counter-actions such as migrating content, choosing different network paths, and upgrade resources (as far as possible).

The remainder of the paper is structured as follows. Section II points the reader to related work, and Section III provides a brief description of candidate TCP flags. Section IV describes the methodology, and Section V explains the process for the experiments, including setup and specifications. Section VI illustrates the results, and Section VII interpretes and discusses them. Section VIII concludes the paper and identifies a set of issues for future work.

II. RELATED WORK

The results of [6] indicate that the evaluations of metrics of QoE parameter's variations are very much dependent on the variations in the Quality of Service (QoS) parameters. However, considering the definition of web services' QoS parameters provided by W3C[7], it is possible to link QoS to QoE from different perspectives. To exemplify the case in [8], the effect of page load duration, total session duration and task completion in web-QoE has been evaluated. In [9], the work is focused on response time, availability, package loss, delay and price. In this context the price is defined as the cost to access the Web Service. So these parameters can be changed based on the service type or aim of the research. In this paper we are going to find a QoE evaluation method for video streaming server from network perspective. Reference [10] points out that "multimedia services are typically much more sensitive to throughput, delay, and packet loss than traditional services".

As shown in [11], the concept and definitions of QoE have changed throughout history, i.e. the last two decades. We employ the most recent definition for our approach, with its particular focus on user delight and annoyance,

"Quality of Experience (QoE) is the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user's personality and current state." [12]

The evaluation approaches for the user perceived quality are divided into two major categories: (1) *Direct participation* of the user in the quality rating questionnaire forms which is the most common modus operandi. For instance, at the end of a session, Skype asks the user to score the quality of conversation. Related to this way of assessment, the International Telecommunication Union (ITU) has defined the concept of opinion score as the "value on a predefined scale that a subject assigns to his opinion of the performance of a system" [13]. (2) The second category in this aspect is *indirect QoE estimation* e.g. connecting the user delight and annoyance to their actions, and observing the latter in order to derive indications of QoE. Our approach is built using both approaches, while the very goal is a method building on (2)

There are lots of factors involved in the process of bridging QoE to QoS through TCP flags. To consider all of them and to organize the approach in an appropriate way, the work is structured based on a particular part of related work, the *QoE Hourglass Model*, which is based on the well-known IP Hourglass Model and is described and motivated in detail in reference [14]. The model includes four layers that each of them has a role towards the user perceived quality. Quality of Experience (QoE), Quality of Presentation (QoP), Quality of Delivery (QoD) and Quality of Service (QoS) are the four layers that are matched to the classical Internet Hourglass Model as shown in Figure 1.

In the IP Hourglass Model, the wide parts at the top and the bottom indicate the plethora of Internet-based applications (top) and access networks (bottom). The applications rely upon a comparably small set of transport protocols, while the universal and "all-uniting" Internet Protocol (IP) with its two

versions (IPv4, IPv6) is found at the narrow waist. Likewise, in the QoE Hourglass Model, there is a plethora of applications, context factors etc. contributing to QoE. From there on, the model gradually narrows down to the level where the corresponding IP packets are handled with a certain QoS, and may basically be delayed or lost. Below the thin waist, the model grows wider again to cover the effects of all different access methods and network technologies on the QoS and thus finally on QoE. In the sequel, we will discuss the transition from QoS to QoE throughout the network stack according to the QoE Hourglass Model in more depth.

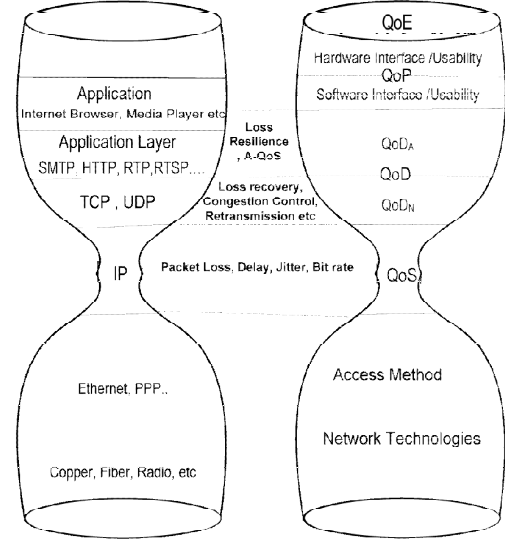


Fig. 1. The QoE Hourglass Model (from [14])

A. Quality of Service (QoS)

Throughput, packet loss, delay and jitter are four main web-server's QoS metrics from the network perspective. In order not to intervene with the underlying throughput shaping by the underlying transport protocol (TCP), our work focuses on variations in packet loss and delay. Packet loss is the ratio of undelivered packets, typically expressed in a percentage. Delay is the end-to-end transmission time, typically given in milliseconds.

B. Quality of Delivery (QoD)

QoD has an essential role in estimation of the QoE in our evaluation approach. It focuses on delivery issues such as delays, freezes, losses etc.. As such, it stretches from the transport layer to the application, cf. Figure 1. The network-related part of QoD, denoted by QoD_N, describes the impact of the network and specifically the transport protocol, on the quality perception. For instance, TCP turns loss into delays and throughput reductions – and signals its control actions using its flags, cf. Section III – while UDP transparently hands over any QoE issue towards the application. The application-related part of QoD, denoted by QoD_A, describes potential impacts of the application on quality perception, e.g. the reduction of freezing through buffering.

C. Quality of Presentation (QoP) and QoE

QoP is the level that captures presentation issues. It is directly connected to the QoE, as the users rate the appearance, i.e. the presentation of the video through the corresponding software and hardware interface. For video in particular, QoP is affected by QoD issues through (1) data delays and delay jitter, leading to freezes; (2) data losses, leading to errors in one or several picture matrices; or (3) throughput reductions, leading into slow-downs or freezes. Actually, TCP avoids (2) at the expense of (1) and (3).

III. TCP FLAGS

This section provides an overview of TCP flags of potential interest for our approach. The flags occupy the seventh field in the TCP header. This field is 9 bits and includes 9 flags that can be set to 0 or 1. The flags determine the connection states and the way of packet transfer management. A brief explanation of the flags' roles is given in Table 1.

Flag Name	Description
Nonce Sum (NS)	An optional addition to ECN that protects against accidental or malicious concealment of marked packets from the TCP sender.
Congestion Window Reduced (CWR)	Indicate that the receiver node received a TCP segment with the ECE flag set and had responded in congestion control mechanism
ECN-Echo (ECE)	Allows notification of network congestion
Urgent (URG)	Indicates a segment is urgent and should be prioritized.
Acknowledgment (ACK)	Acknowledges received data
Push (PSH)	Asks to push the buffered data to the receiving application
Reset (RST)	Resets the connection
Synchronize (SYN)	Initiates a connection
Finish (FIN)	Terminates a connection

Table 1. TCP flags description

A. The Flags' Roles in Connection Establishment

The series of steps involved in the process of connection establishment, called "three-way handshake". The simplest TCP connecting procedure is illustrated as follow.

Step	State TCP A	State TCP B
0	Closed	Listen
1	SYN-Sent → <SYN Segment>	→SYN-Received
2	Established ← <SYN,ACK Segment>	←SYN-Received
3	Established → <ACK Segment>	→Established
4	Established → <Data>	→Established

Fig.2. Basic 3-Way Handshake for connection synchronization [15]

The arrows indicate the direction of packets transition and the terms under TCP A and B are their states. The flow is started by TCP A that sends a SYN segment in step 1. Next, TCP B sends a SYN and acknowledges the SYN it received from TCP A. In step 3, TCP A sends an empty segment with ACK flag up to acknowledge the TCP B' SYN segment, and finally TCP A starts sending data. Besides of the ACK flag, which is

common in the TCP segments, SYN is the most used flag, and is a clear indicator for connection establishment.

B. The Flags' Roles in Connection Closure

If one of the connection peers has no more data to send, it initiates the CLOSE operation by sending a FIN segment. Then the other TCP first acknowledges the FIN it received and then will send a FIN segment. Finally the local TCP ends the closing procedure by sending the ACK segment to the other side. No matter the initiator, apart from ACK, FIN is the essential indicator for the closing process.

C. The Flags' Roles in Connection Break

Usually, an unexpected break terminates a TCP connection. Such interruptions can be result of user actions or automatically be triggered by a web browser. For example, in the context of video streaming, pressing the reload or stop button results in early connection termination. Some of the web browsers end the connection by the usage of RST flag instead of regular TCP close handshake. However, regardless of the reasons behind the interruption, FIN and RST are the flags for the breaks in an ongoing traffic [5], [16].

IV. METHODOLOGY

The method used in this paper consists of a subjective study of the user-perceived QoE, and measurements and statistical analysis on the TCP flag ratios with and without users.

For this purpose, a web server for video streaming, with the additional capability of varying the QoS parameters on the network level, is provided. The videos are located in the Apache HTTP server memory and are accessible via its assigned domain name.

The web site provides a scoring questionnaire for evaluation of the videos. The QoS parameters of the web server are varied gradually, users are asked to score the corresponding video, and in parallel, the server traffic is captured. All the server traffic (without any filtering) is captured in both directions through Wireshark [17]. Then the captured traffic is filtered for only TCP and HTTP packets and analyzed regarding variations in the percentage of TCP flags.

In this process, the users faced with some QoD_N issues in the form of lags in the video, and they were allowed to react to them, e.g. (1) clicking on refresh, (2) stopping the stream. Obviously, such reactions may impact the transport layer, and thus, they reflect in the flag states. Therefore, the captured traffic was examined for variations in the percentages of the flags, stemming from user actions. Finally, by linking the traffic measurements (TCP flag ratios) and the survey results (MOS), the relationships between TCP flag ratios and QoE are indicated.

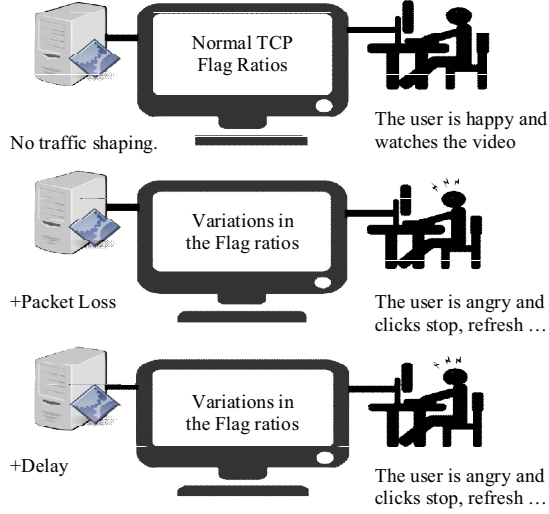


Fig. 3. Overview of the survey.

Survey and Dummy-User Cases

The TCP flag status is associated with a lot of happenings, the most important of which are TCP (control) behavior and user actions. In our work, the TCP flag changes are linked to user actions, and in particular to reactions to QoE problems. Therefore, it is required to determine to which extent the flag ratios belongs to user actions, or whether they reflect TCP behavior.

To address this issue a test module called dummy-user is considered. The aim of the dummy-user test is to identify the differences between effect of user actions and TCP behavior on the flags. To this objective, a similar test to the survey with a dummy user is considered. This trial is identical to the survey (e.g. same network conditions) except of the involvement of users. Therefore, we can conclude that the variations in the flag ratios in these tests belong to all the events in the network, with exception of the user actions. If one of the flags has similar changes in both experiments (survey and dummy-user), it is not possible to specify if it is the result from user reactions to low quality, so it is removed from the set of QoE indicators. In other words, only the flags that have different trends in the survey case as compared to the dummy-user case are used for estimation of QoE.

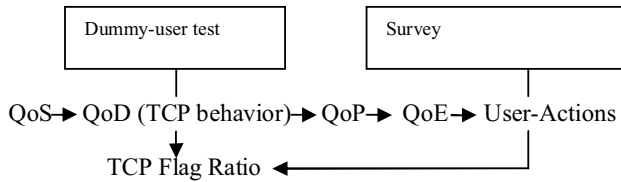


Fig. 4. Overview of two series of tests based on the terminology of the QoE Hourglass Model [14].

V. EXPERIMENTS

A. Setup and Specification

The first step is to provide a web server with sufficient capacity. The B Model of the first version of an embedded board called Raspberry Pi has been used in this case. This model is the higher-spec variant of the Raspberry Pi. It has 512 MB of RAM, two USB ports and a 100 Mbps Ethernet port. As it is a web-server for video streaming, 32 GB memory is used to have enough space. The server has not shown any limitations in performing this task in the context of our experiments.

In order to link QoE with TCP flags, we need different levels of QoE. These are provoked through decrementing the QoS parameters on the server side. In fact, the server is enabled to introduce packet delay and packet loss on the packet stream leaving the server. Considering the comparison of traffic shapers in [18], NetEm is used for applying packet loss and delay. The accuracy of purposeful variations in the metrics is verified through ping, which also determined the smallest units to vary QoS parameters.

The fast transfer of large amounts of data is a crucial property of video streaming web servers. Earlier versions of TCP face difficulties to meet this demand [19]. Thus, TCP CUBIC [19] has been selected for the web server. Its congestion control and the corresponding control messages (such as flags) both manage and reflect the QoD_N. The web server, running HTTP, is located at the application layer.

For both the tests (Survey and dummy-user), a “webm” video with duration of 1’33” is used, whose metadata is specified in Table 2.

Video	
Dimensions	640 × 360
Codec	On2 VP8
Frame rate	25 fps

Table2. Specification of the video.

To access the video, a specific video streaming web site has been set up. The HTML5 video element is used for the presentation of the video and the users watched it via the web browser Chrome.

A specific task consists in identifying the proper range for each of the QoS parameters, such that the users get to experience impairments through QoD issues. Indeed, we are interested in quantifying the result of QoS variations on temporal issues encountered in the video delivery. The latter was done by counting the number of freezes in the stream in the response to changes in the parameters. For each QoS level, e.g. 9% packet loss, the tests were repeated ten times. There are no pauses in the “no traffic shaping” condition. On the other hand, when 14% packet loss was set, typically, six freezes of more than five seconds were observed. Also, the minimum relevant disturbance levels were identified; in our case, 7 % loss or 500 ms delay were needed to make users see any freezes at all. These rather big amounts of packet loss and delay in the first place are due to the short duration of the video used in the tests; the lags in the video happened due to delays in the buffering process. For lower loss ratios and

delays, TCP was obviously able to repair the damage in time before the users got to see it, i.e. the QoD was not affected. Based on these observations, eight loss levels (from 7 % to 14 % in steps of 1 %) and eight delay levels (from 500 ms to 1200 ms in steps of 100 ms) were specified beyond the baseline case (0 % loss and 0 ms delay), yielding 17 tests cases in total.

Traffic capturing is the next essential tool that has an important role in the experiments. Wireshark [17] as one of the most used and trusted tracking tools is chosen for this purpose. TCP CUBIC with its optimized congestion control algorithm is used in the server, and HTTP is acting on top of TCP. So the capture files are filtered for these two protocols.

B. Survey

The formalism introduced in [14] shows that QoE can be related to QoD_N and to QoS by freezing the impact of the delivery on application level and of the presentation facilities. We will apply this approach in the way that, in the surveys, all the application settings presentation facilities (in our case laptop and browser) are kept unchanged for the participants. This way, QoS issues that impact QoD_(N) will be transformed in similar ways throughout the network and application stack.

All the contributors in the survey are BTH students, and the experiments were done in the same place at the University. As outlined above, QoS metrics are applied in 17 levels (baseline, eight packet loss levels and eight delay levels). Mostly five (in some cases four) different students scored the video qualities for each of the 17 levels. Their individual scores are averaged into Mean Opinion Scores (MOS). Totally 81 persons participated in the survey.

The survey is situated in a real network. The laptop is connected to the university' Wi-Fi (Eduroam), which showed to have a negligible impact on the QoS and thus on QoE. By choosing proper QoS disturbances on the server, yielding different levels QoD problems (throughput reduction), QoP problems (freezing) and user ratings (OS) and reactions, we were able to control the QoE in the desired manner.

C. Dummy-User Test

In this experiment, the video is played and the TCP flags are recorded under the same network conditions as in the survey case, however without users watching and reacting to the lags. Instead, one dummy-user process starts the video and the traffic captures, and the video plays all the way regardless of lags or any other problem. As it was mentioned before, five people used to watch the video for each grade of QoS level (e.g. 9% packet loss) in the survey. Therefore, in the dummy-user test, also the process is repeated five times for each level of QoS, in order to have the same amount of traces and traffic as in the survey. So, it is ensured that the only difference between this test and the survey is about user reactions to the lags. The results of these dummy-user tests are used to select the final set of QoE indicators.

VI. RESULTS

There are nine TCP flags, two QoS parameters (each with eight grades) and one test without traffic shaping, and two series of tests (dummy-user and survey). Therefore 34 cases are obtained. It is not possible to describe all the outputs due to space limitations. We focus on the survey case. The statistics belonging to the NS, CWR, ECE, URG and RST flags were constantly zero, while PSH has very big fluctuations, so none of them is suitable as QoE indicator. The ratios of ACK flags in case of delay are hardly distinguishable in the survey and the dummy-user cases; therefore, it is not possible to conclude that variations in the ACK flag ratios indicate user reactions to low quality. So, the ACK flag is also removed from the set of potential candidate. This leaves us with the SYN and FIN flags to be investigated in detail.

Tables 3 and 4 show the MOS provided by N users and the corresponding standard error E_{MOS} – obtained from the standard deviation of the opinion scores (SOS) as $SOS/N^{0.5}$ – and the relevant flag ratios (overall averages) as observed for the given levels of loss and delay. (As point of reference with regards to the user ratings, the undisturbed case yielded a MOS = 9.8 for $N = 5$ and $E_{MOS} = 0.2$, respectively).

Loss disturbance	MOS	N	E_{MOS}	SYN flag ratio	FIN flag ratio
7 %	9.0	5	0.32	0.849	0.592
8 %	8.0	5	0.00	0.617	0.429
9 %	6.8	5	0.37	0.588	0.372
10 %	7.2	5	1.07	0.510	0.353
11 %	4.8	5	0.86	0.665	0.427
12 %	4.2	5	1.02	0.302	0.248
13 %	3.4	5	1.21	0.189	0.163
14 %	2.8	5	0.73	0.188	0.154

Table 3. OS statistics and flag ratios for various loss disturbance levels.

Delay disturbance	MOS	N	E_{MOS}	SYN flag ratio
500 ms	8.6	5	0.51	0.514
600 ms	7.3	4	0.25	0.988
700 ms	7.6	5	0.40	0.665
800 ms	7.0	5	0.58	0.665
900 ms	6.8	5	0.63	0.729
1000 ms	5.5	4	0.65	1.320
1100 ms	4.8	4	0.85	1.019
1200 ms	3.0	4	0.41	1.060

Table 4. OS statistics and flag ratios for various delay disturbance levels.

Tables 3 and 4 reveal non-monotonic behaviors in both MOS and flag ratios as well as varying standard error values, which is due to the rather limited number of user evaluations per disturbance level. However, for growing disturbances, the following trends can be identified: (1) the MOS decreases (which is expected); (2) in case of loss, flag ratios decrease; (3) in case of delay, the SYN flag ratio increase.

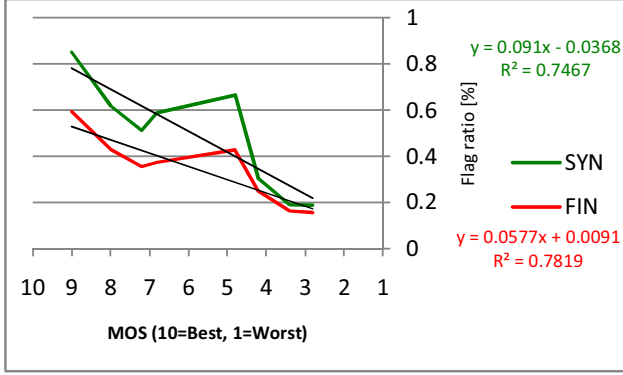


Fig. 5. TCP flag ratios versus QoE in case of loss.

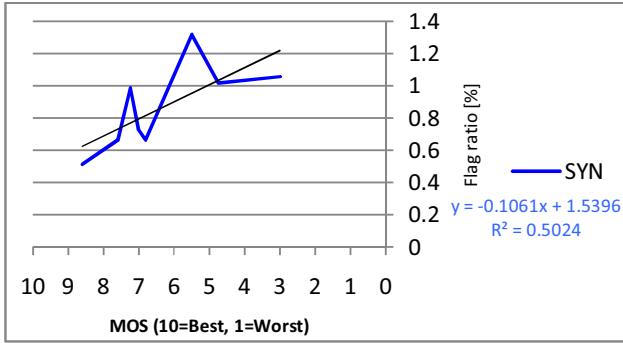


Fig. 6. TCP flag ratios versus QoE in case of delay.

Figures 5 and 6 illustrate further the trends how TCP flag ratios change with the user satisfaction in terms of MOS values; the corresponding trend lines and their coefficients of determination R^2 were generated by MS Excel. The MOS values are shown from best to worst to facilitate the interpretation the correlation between the QoE and the flag ratios. Again, we observe non-monotonic behaviors, but the trends can be clearly identified. Figure 5 shows that simultaneous downward trends in SYN and FIN at the same time might be a bad signal for increasing packet loss. In Figure 6, the percentage of TCP packets with SYN flag up is increasing; it might signal user dissatisfaction in response of excessive delay.

We now turn our attention to the trend lines shown in Figures 5 and 6, which are rescaled in a way that the approximated SYN and FIN intensities at the maximal MOS value act as reference values (100 %) and shown in Figure 7. We observe that in the loss case, SYN and FIN intensities decrease in similar manners as the MOS decreases, down to approximately one-tenth of their original value. On the other hand, in the delay case, the SYN intensity grows by factor 3 as the MOS approaches its minimal value. These quite drastic changes point at a good potential to discover the corresponding QoE issues in a real-network scenario.

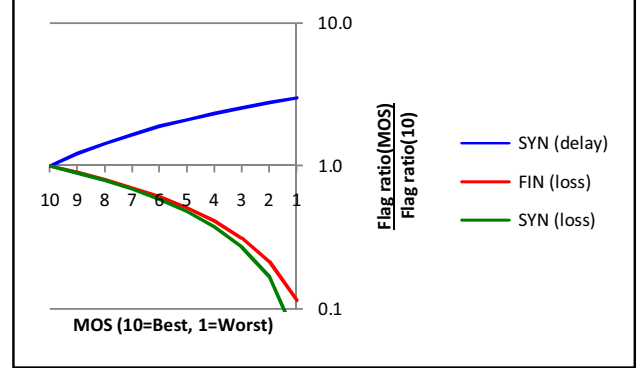


Fig. 7. Relative changes of flag ratio trend-lines versus MOS.

VII. INTERPRETATIONS

In this section, we discuss how the various observed TCP flags can be interpreted in light of the given QoS disturbances and QoE challenges. We are aware of the fact that the trends could be smoother, which is one of the targets for future work.

A. Packet Loss

A FIN flag indicates that there is no more data from the sender, while the SYN flag is only set when communication is about to start. Therefore, any decrement in both of them is a signal for less connection establishments. On the other hand, the total number of packets increases due to the recovery of packet loss by TCP, which implies increased session lengths. In the captured traffic, there are periods with big amounts of data transferred in short periods of time, called bursts, whereas the remaining time is dominated by silence. Indeed, it has been observed that by adding packet loss, the lengths of the bursts also increased.

B. Packet Delay

The SYN ratio increases with delay. Unlike packet loss, there is no growth in the total number of packets. There are a larger number of sessions, but they are shorter in duration relative to normal condition. We suspect that delays caused failed, i.e. abnormally terminated connections without FIN flags, and the TCP CUBIC reaction is restarting the handshake that results in an increment in the SYN ratio.

VIII. CONCLUSION AND FUTURE WORK

In this paper we have presented an approach to link QoE to transport-layer flags of TCP video streaming traffic from a web-server running TCP CUBIC. The links between both were established by statistical analysis of the frequencies of TCP flags, and of user ratings, which were brought together via common underlying QoS values. Actually, our work shows that a network and/or service administrator can estimate changes in the overall user satisfaction by monitoring changes in the TCP flag ratios, which can be done anywhere along the path taken by the TCP segments. Among the flags, SYN and FIN have passed the criteria to be selected as indicators. According to our results, simultaneous downward trends in SYN and FIN signal user dissatisfaction and corresponding

reactions, as a consequence of significant amounts of packet loss. The next indicator applies when the percentage of TCP packets with SYN flag up is increasing. In this condition, excessive delay is the reason behind the users' displeasure and reactions. In all the conditions, the rest of flags wave around a constant number. Figure 8 summarizes those findings in a decision tree.

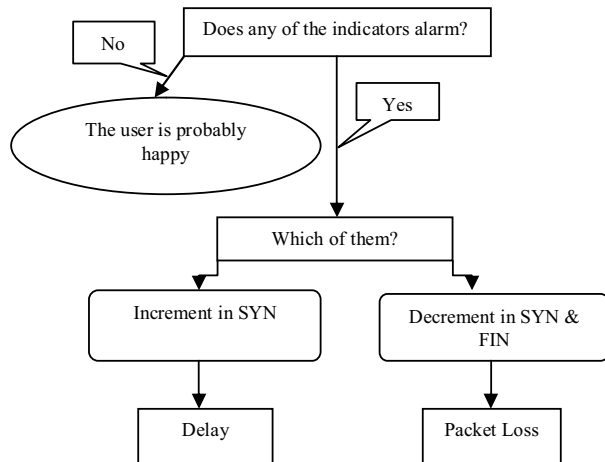


Fig. 8. Proposed decision tree.

Future work might include a larger study, yielding more quantitative data that increases the reliability and accuracy of the trends, and thus of the derived QoE models, in various mobile, wireless and fixed settings, and for specific patterns and combinations of disturbances (loss and delay). Furthermore, the approach does not yet provide detailed hints at specific problems. Therefore, future work may consist in a study on troubleshooting approaches for the web server. Moreover, the outputs of this research are dependent on the web server specification. Therefore testing the impact of changes in the server on the algorithm, as well as other versions of TCP, are candidates for future work. Last but not least, the results can be used to assess and quantify the risks associated to user reactions, in particular those that are potentially leading to churn.

REFERENCES

- [1] Java EE 6 Tutorial, <https://docs.oracle.com/javaee/6/tutorial/doc/gjvvh.html> (last seen May 4, 2016).
- [2] Dagstuhl Seminar 12181 "Quality of Experience: From user perception to instrumental metrics," <http://www.dagstuhl.de/en/program/calendar/semhp/?seminr=12181> (last seen May 4, 2016).
- [3] F. Fotrousi, S. Fricker, and M. Fiedler, "The effect of requests for user feedback on Quality of Experience," Submitted.
- [4] Y. Huang, W. Zhou, and Y. Du, "Research on the user behavior-based QoE evaluation method for HTTP mobile streaming," in Proc. Ninth International Conference on Broadband and Wireless Computing, Communication and Applications, 2014, ISBN 978-1-4799-4174-2, pp 47–51.
- [5] J. Shaikh, M. Fiedler, P. Arlos, and T. Minhas, "Classification of TCP connection termination behaviors for mobile web," GLOBECOM Workshops (GC Wkshps), 2011 IEEE, ISBN 978-1-4673-0039-1, pp 1111–1115.
- [6] V.A. Machado et al., "A new proposal to provide estimation of QoS and QoE over WiMAX networks," in Proc. 2011 IEEE Third Latin-American Conference on Communications, ISBN 1467302775.
- [7] World Wide Web Consortium (W3C), "QoS," <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/> (last seen May 4, 2016).
- [8] D. Strohmeier, M. Mikkola, and A. Raake, "The importance of task completion times for modeling web-QoE of consecutive web page requests," In Proc. 5th Int. Workshop on Quality of Multimedia Experience (QoMEX), 2013.
- [9] R. Krüger Tavares and C. Becker Westphall, "An architecture to provide QoS in web services," in Proc. 2006 IEEE International Conference on Communications (ICC'06), Vol. 2, Istanbul, ISSN: 8164-9547, pp 921–925.
- [10] W.-H. Hsu, S.-C. Yeh, Y.-P. Shieh, and C.-H. Hsieh, "Web-based QoE measurement framework," in Proc. 2013 International Conference on Signal-Image Technology & Internet-Based Systems.
- [11] S. Möller and A. Raake, "Quality of Experience: Terminology, methods and applications," PIK – Praxis der Informationsverarbeitung und Kommunikation. Dec. 2014, Vol. 37, Issue 4, pp 255–263.
- [12] P. Callet, S. Möller, and A. Perkins, "Qualinet White Paper on definitions of Quality of Experience" 2012. European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne, Switzerland, Version 1.2, March 2013.
- [13] ITU-T Recommendation P.10: "Vocabulary for performance and Quality of Service," 2006.
- [14] T. Minhas and M. Fiedler, "Quality of Experience Hourglass Model," in Proc. 2013 International Conference on Computing, Management and Telecommunications (ComManTel), ISBN 978-1-4673-2087-0, pp 87–92.
- [15] J. Postel, editor, "Transmission Control Protocol", RFC 793, September 1981.
- [16] M. Arlitt, C. Williamson, "An analysis of TCP reset behaviour on the Internet," ACM SIGCOMM Computer Communication Review, Volume 35, Number 1, January 2005.
- [17] L. Chappell, "Wireshark Network Analysis: The Official Wireshark Certified Network Analyst Study Guide," Protocol Analysis Institute, dba "Chappell University": 800.March 31, 2010, ISBN 1-893939-99-5.
- [18] T. Minhas, "Network impact on Quality of Experience of mobile video," Licentiate Thesis, School of Computing, Blekinge Institute of Technology, 2012, ISBN 9789172952270.
- [19] Networking Research Lab, "BIC and CUBIC," <http://research.csc.ncsu.edu/netsrv/?q=content/bic-and-cubic> (last seen May 4, 2016).