Quality of Signaling (QoSg) Metrics for Evaluating SIP Transaction Performance

Marco Happenhofer
Vienna University of Technology,
Institute of Broadband Communication
Vienna, Austria
E-mail: marco.happenhofer@tuwien.ac.at

Peter Reichl
Telecommunication Research Center Vienna
(FTW)
Vienna, Austria
E-mail: reichl@ftw.at

Abstract: **Service quality and user-perceived experience has been a networking research topic for many years. Whereas most related work concentrates on the quality of media, only very few papers, however, discuss the performance of the signaling for those media. This paper addresses the performance of the Session Initiation Protocol (SIP) which is very popular with Next Generation Networks (NGN). A collection of metrics, the so-called Quality of Signalling (QoSg), is introduced for measuring SIP signaling delays. In contrast to traditional end-to-end performance metrics, QoSg aims at evaluating individual SIP transactions. Therefore, this approach can be applied anywhere in the SIP network and allows for additional insight, for instance regarding congestion detection or SLA monitoring.**

1. INTRODUCTION

Due to low latencies and high bandwidths, modern IP networks offer exciting new opportunities for services over IP, like, for instance, Voice over IP (VoIP) or IPTV. Such services put certain requirements to the network in terms of delay, jitter and bulk transfer capacity, which, if not met appropriately, may cause a decrease of the perceived quality at the user side. There are different technologies for realizing the required Quality of Service (QoS), for example the IETF introduced the Integrated Services [3] and Differentiated Services [2] architectures for allocating the necessary bandwidth for a dedicated service. Additionally, the IETF has standardized IP Performance Metrics [4] which specify metrics for bulk IP performance.

However, first users have to signal the system or server that they wish to use a service, using dedicated signaling protocols. The corresponding session setup phase may introduce delays which are recognized by the customers and might frustrate them, especially it lasts so long that the caller gives up before any media has actually been transmitted. Thus, user experience does not only depend on the performance of the media, but also low signaling delays are considered a prerequisite for delivering good service quality to the customer. Especially in the case of professional and large scaled systems, it is therefore worth to investigate the impact of signaling on the perceived quality of the end user.

For VoIP systems, the IETF has designed a signaling protocol that allows establishing, modifying and tearing down multimedia sessions, i.e. the Session Initiation Protocol (SIP) [1]. So far, the performance of this protocol has been analyzed only in rather few publications. In [6], the authors measure the call setup delay between several cities in the US. Other publications, like [7], concentrate on the performance of SIP servers. Recently, the IETF has started the Performance Metric Other Layers (PMOL) working group for discussing the performance of other layers than IP. In its first draft [5], this working group has defined performance metrics for SIP, however from an end to end perspective. These metrics are measured at the calling and the called party, and are defined for specific services like call or instant messaging.

Although end to end performance has a major impact on user-perceived quality experience, it does not contribute to identifying components which introduce artificial delays. For this specific purpose, we introduce a collection of metrics, the so-called Quality of Signaling (QoSg), which target the performance of individual SIP transactions and can be applied to any arbitrary SIP element. The results can then be used for congestion detection or monitoring of Service Level Agreements (SLA). Moreover, in any case we consider it essential to understand the performance of single transactions before even start to discuss the end to end performance of complex signaling networks.

The rest of this paper is organized as follows: Section 2 briefly introduces SIP. In Section 3 we introduce the essential metrics of QoSg and discuss their relations. Based on these metrics, in Section 4 we present some measurements of SIP transactions. Section 5 describes three possible application scenarios, where service providers might benefit from the knowledge about QoSg. Section 6 concludes the paper.

2. SESSION INITIATION PROTOCOL

SIP is a signaling protocol for setting up, modifying and tearing down multimedia sessions. It uses so-called transactions to deliver reliably a request to the destination and to ensure that a response is received, irrespectively of whether the used transport protocol is the Transmission Control Protocol (TCP) [9], the User Datagram Protocol (UDP) [8] or the Stream Control Transmission Protocol (SCTP) [10]. Transactions are specified by means of state machines for the client side and the server side. SIP specifies two different transaction types, the INVITE transaction for INVITE requests and the Non-INVITE transaction for all other request types. This distinction is required because INVITE requests might get forked, and human interaction has to be taken into account.

Part of this research has been funded by the Austrian Government and the City of Vienna within the competence center program COMET.

The SIP architecture introduces the concept of SIP proxies which realize different tasks like authentication, accounting, authorization, routing etc. Usually, SIP requests have to traverse several such proxies for realizing a requested service. Between each of these proxies exists a unique transaction. Note, that only stateful SIP proxies maintain SIP transactions, stateless SIP proxies do not. Figure 1 depicts a scenario where a request passes two different SIP proxies.

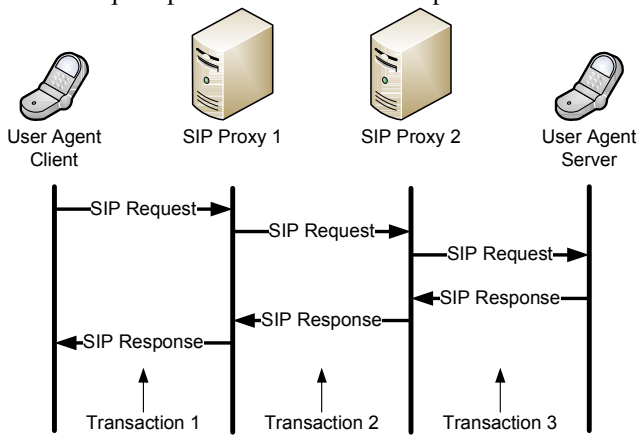

Figure 1 - SIP message flow over two SIP proxies

3. QUALITY OF SIGNALING

As already mentioned in the introduction, the IETF is currently working on performance metrics for non-IP layers and specific services, starting with the Basic Telephony SIP End-to-End performance metrics [5] as an example. Our concept follows a quite different approach: We aim at reducing SIP signaling to the smallest building block, i.e. SIP transactions, and define performance metrics for those. The resulting collection of metrics for single transactions is called QoSg.

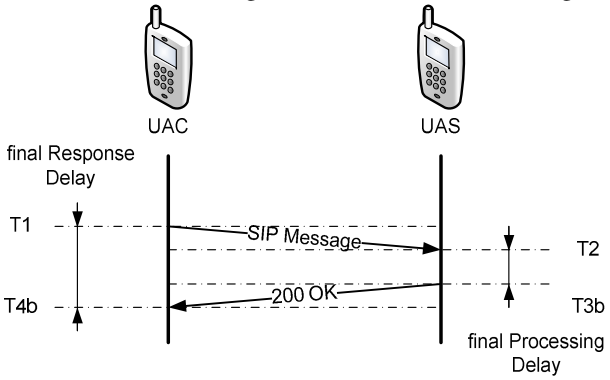

Figure 2 - Metrics for Non-INVITE transaction

It is characteristic for QoSg, which covers of course only a small fraction of the overall performance, that these metrics depend only on a few external parameters like link quality and processing delay of the server or proxy, whereas an end to end metric depends on the quality of the links between all proxies and the processing delays of these proxies.

As first key QoSg metric, we introduce the **final Response Delay (fRD)**, corresponding to the time interval from sending the request until the corresponding response arrives, see Figure 2 for the case of Non-INVITE transactions. Similarly, the **final Processing Delay (fPD)** reflects the delay from receiving a request until sending a final response message, measured at the server side.

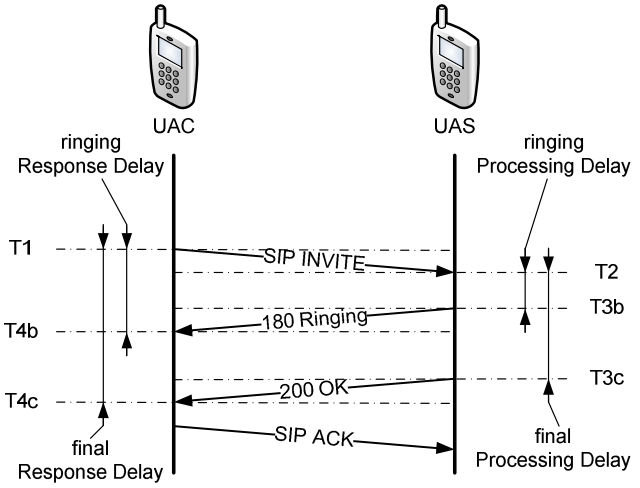

Figure 3 - Metrics for INVITE transaction

In the case of INVITE transactions (see Figure 3), the final Response Delay would include also customer-induced delays (*T3c–T3b*). Hence, further metrics are required for the INVITE transaction. The **ringing Response Delay (rRD)** represents the time from sending the INVITE request until the 180 Ringing is received. Note that this metric does not include the human interaction delay, which would tamper the results. We also introduce the **ringing Processing Delay (rPD)** for the processing delay at the server for an INVITE transaction**.**

In order to consider any delays caused by the transport protocol we apply all measurements at the application layer and not at the network layer. We further summarize any transport protocol, transmission and SIP transaction retransmission delays by the term **Transport Protocol Delay (TPD)** and derive with the help of Figure 2 following formula:

$$fRD = fPD + TPD \quad (1)$$

Coming back to Figure 2, the *TPD* for Non-INVITE transactions can be divided in an uplink (*T2–T1*) and a downlink part (*T4b–T3b*). The *TPD* depends on the link quality, more specifically loss rates, delay and jitter, and the used transport protocol. On the other hand, the *fPD* is influenced by the queuing and processing delay at the server. Note that for this initial discussion we do not consider flow control mechanisms.

Figure 4 depicts the metrics introduced so far for a scenario with two SIP proxies, where three transactions are required for passing a request from a client to the server.

In the case where the transaction is terminated at a proxy which forwards the request to another SIP entity the *fPD* is composed by the **Queuing & Processing Delay (QPD)** of

the proxy and the *fRD* of the following transaction (see Figure 4). Therefore we can derive the following formula:

$$fRD_i = TPD_i + QPD_i + fRD_{i+1} \quad (2)$$

Note that the *QPD* might be different for each system, because it depends on the current load, queuing configuration, processing tasks and CPU speed, whereas the *TPD* delay depends only on link quality and protocol selection (as mentioned before, ignoring flow control). Because it is quite complicated to estimate the *QPD*, we will provide in the next section an estimation of the *TPD*, so that we can derive the *fPD* if the *fRD* is known (due to measurements), or vice versa.

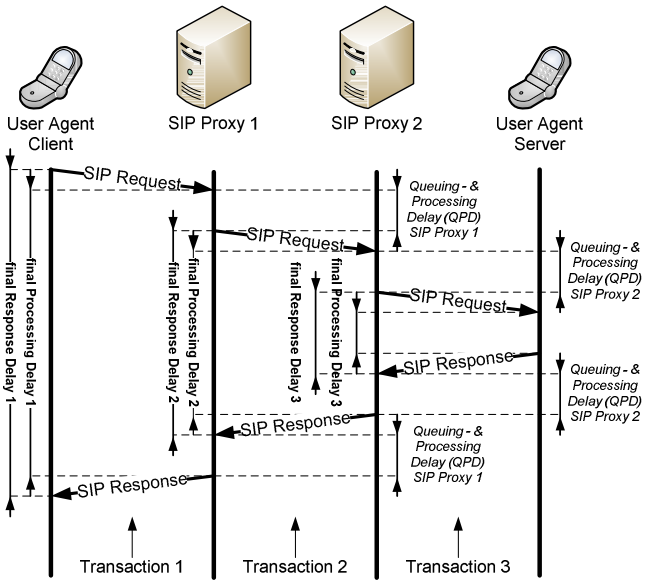

Figure 4 - Time budget for a single SIP transaction

Note, finally, that for the case of an INVITE transaction (see Figure 3), equations (1) and (2) become

$$rRD = rPD + TPD \quad (3)$$

$$rRD_i = TPD_i + QPD_i + rRD_{i+1} \quad (4)$$

4. MEASUREMENTS

This section provides some measurement results that support equations (1) and (2) above. The measurement was performed with two instances of SIPp and intermediate network emulation based on Linux traffic control, which introduces delay, losses and jitter between the user agent client and user agent server (for further details on the general measurement setup we refer to [11][12]).

We have measured the *fRD* of 180 million Non-INVITE transactions for three different transport protocols. Loss rates vary from 0% to 1% (in steps of 0.25%), delay from 0 to 45 msec (15 msec steps), and jitter from 0 to 20 msec (10 msec steps). All parameters have been modified in both uplink and downlink directions. Additional delay has been introduced at the user agent server after receiving the request and before sending the response in order to emulate artificial *fPD*. Eventually, a multiple linear regression analysis [13] for the mean value of the *fRD* has been performed to survey the linear impact of the external parameters due to the *fRD*. Equation (5) denotes the relation to be tested by regression analysis between the mean *fRD*, the *fPD* and all external parameters:

$$\begin{aligned} fRD = {} & b_0 + b_1 \cdot loss_{uplink} + b_2 \cdot loss_{downlink} + \\ & + b_3 \cdot delay_{uplink} + b_4 \cdot delay_{downlink} + \\ & + b_5 \cdot jitter_{uplink} + b_6 \cdot jitter_{uplink} + b_7 \cdot fPD \end{aligned} \quad (5)$$

	UDP	TCP	SCTP
offset (b_0)	0.001	-0.009	0.000
loss uplink (b_1)	0.519	0.866	0.224
loss downlink (b_2)	0.420	0.900	0.196
delay uplink (b_3)	0.997	1.132	1.014
delay downlink (b_4)	0.996	1.131	1.005
jitter uplink (b_5)	0.033	0.157	0.069
jitter downlink (b_6)	0.031	0.199	0.085
fPD (b_7)	**0.995**	**1.003**	**1.000**
correlation coefficient	0.999	0.999	0.999

Table 1 - Multiple linear regression

Table 1 depicts the values of the coefficients b_0 to b_7 of formula (5) as result from the multiple linear regression anlysis. Whereas the regression is quite an approximation of the mean *fRD* however, we can express how good this approximation fits to our measured values. A correlation coefficient of 1 would mean that our regression fits exactly the measured values and our correlation coefficient is 0.999, so this regression is pretty perfect.

The b_7 coefficient is the most interesting one for us, because he explains the impact due to *fPD* for the *fRD*. This value is for all transport protocols almost the same 1. So we could modify the formula (5) and remove the coefficient b_7 because he is independent of the transport protocol 1. Therefore and that fact that our approximation is almost perfect we can derive formula (1) from formula (5) by replacing the terms b_0 up to b_6 by *TPD*. We can further conclude that the external parameters, as loss, delay and jitter influence only the *TPD*, as we assumed in the previous section and not the *fPD*.

Figures 5 to 7 illustrate the *fRD* for UDP, SCTP and TCP, respectively, and for three different *fPD* values (0 sec, 0.1 sec and 0.2 sec), using a link with 1% loss and 45 ± 20 msec one-way delay in each direction. On the left hand side, the y-axis refers to the density, and on the right hand side to the distribution function. The left density and distribution (red) depicts the case with 0 msec *fPD*, the curves in the middle (green) with 100 msec, and the right ones (blue) with 200 msec.

The triangle shape of the density functions goes back to the delay emulation, which is uniformly distributed for one direction. Hence, the roundtrip distribution is a convolution of these two uniform distributions, resulting in a triangle shape.

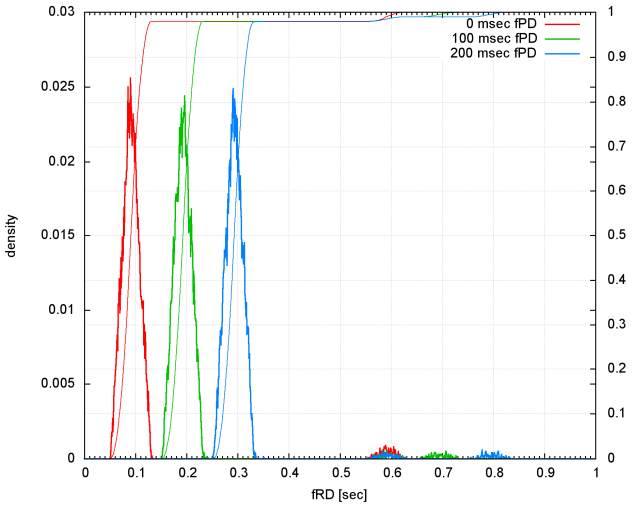

Figure 5 - fRD distribution for UDP

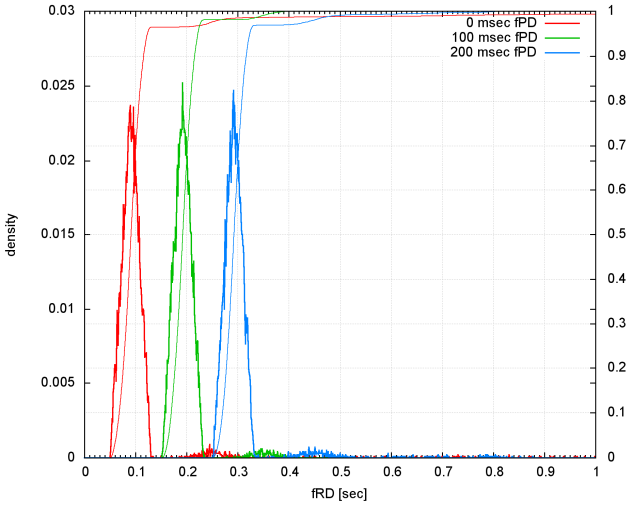

Figure 6 - fRD distribution for SCTP

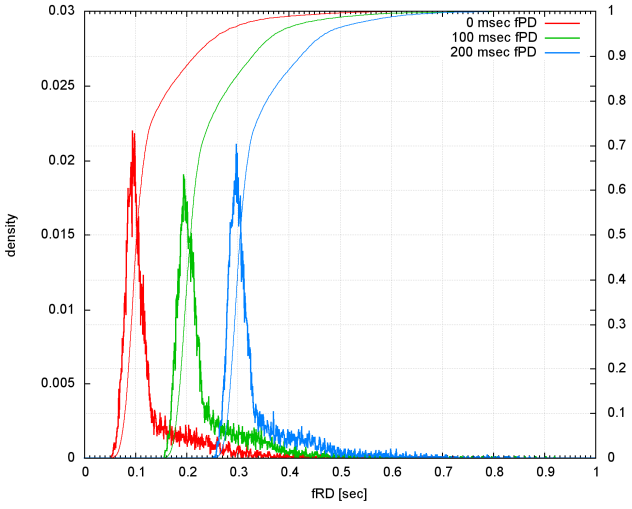

Figure 7 - fRD distribution for TCP

All three figures depict retransmissions due to SIP or due to transport protocol by spikes (e.g. at 0.6, 0.7 and 0.8 seconds in Figure 5). In the case of UDP the retransmission is predefined by SIP at exactly 500 msec and this goes in line with the 500 msec delayed spikes in Figure 5. In Figure 6, the SCTP stack estimates the roundtrip time more precisely and resends the lost message after the SCTP timer expires which might be in our case about 130 msec. Compared to SCTP and UDP, at TCP (Figure 7) we observe delays between the first transmission and the retransmissions. We attribute this effect to Head-of-Line Blocking (see [11]). Irrespectively of the used transport protocol, we can derive that a delay at the server (*fPD*) always introduces the same amount to our *fRD* (vertical shift of the density and distribution function).

Summarizing, the results of the regression analyses together with the presented diagrams confirm the analytical expressions (1) – (4), which explain the linear dependency of *fRD* on the delay at the server side.

5. APPLICATION FIELDS FOR QoSg

In this section, we put forward some directions where the measurement of QoSg proves to be useful, and how service providers may profit from such measurements.

5.1 How to select an appropriate transport protocol?

As already mentioned, SIP allows the usage of TCP, UDP and SCTP as transport protocols. For a service provider, this poses the natural challenge to select an appropriate transport protocol for a specific link characteristic. Here, the results of our measurements and regression analysis might help to identify the most efficient transport protocol, based on expected traffic and delay. Table 1 describes in detail how external parameters influence the *fRD* for the different transport protocols. Based on that, we may conclude that TCP introduces the largest *TPD*, because all weights b_1 to b_7 are much higher for TCP than for the other protocols (except for the offset b_0). We further argue that SCTP performs better than UDP for the case of losses, as SCTP detects loss much faster due to missing ACKs, and resends the lost SIP message earlier. Note, however, that we did not provide any results concerning the amount of sent bytes by these transport protocols, which is a significant aspect from an economical point of view.

Of course we are well aware of the fact that performance may not be the only criterion motivating the decision to use a certain transport protocol; further aspects like security (TLS), NAT traversal and message size play an important role, too.

5.2 How to identify overloaded components in the system?

SIP introduces no suitable mechanism for overload control or protection. Because SIP servers cannot give feedback about the current load, they can become overloaded by incoming SIP messages in busy hours, or due to some telephone voting event etc. There is some standardization going on in the IETF and ETSI to deploy explicit congestion notification, however, we would like to argue rather for an implicit congestion notification like with TCP, where network over-

load is assumed if no ACK arrives in a given time interval (depending on past round time measurements).

Hence, we propose to adapt TCP's congestion control mechanism towards a flow control for SIP by continuously measuring the *fRD*. Equation (2) demonstrates that the *fRD* depends on three terms, of whom we can estimate the first one, the *TPD*, with the help of our regression analysis presented in Table 1, if we are aware of the link quality. The second summand in equation (2) reflects the current load on the system caused by queuing and processing delay at the server, and the third summand is the *fRD* of the next transaction. Altogether, if we assume more or less constant link quality and observe an increasing in *fRD*, we may conclude that there is at least one SIP server whose queuing delay is increasing and which therefore is prone to be overloaded.

If, however, the quality of the link changes, we will observe changes of the *fRD* as well, which might lead us to the (wrong) conclusion that congestion appears. Table 1 shows that the increase of one-way delay introduces significant additional delay for any transport protocol. Further studies on this topic are required to get more comprehensive results regarding the question whether this SIP congestion detection mechanism is appropriate or needs further refinement. Especially the impact of flow control of TCP and SCTP should be discussed and analyzed in life systems.

5.3 How to ensure SLAs between Service Providers?

Complex telecommunication services might invoke several application servers potentially belonging to different authorities, and calls might be routed from one domain to another. On the other hand, each service provider would like to ensure that all of its subscribers receive a certain signaling performance, independently of whether the communication passes other authorities or not. To realize such quality, a service provider has to ensure that all other authorities involved process requests within given timeframes, which can be described in a Service Level Agreement (SLA). Our QoSg metrics are very good candidates for such common SLA metrics, and thus could be applied for the agreement and monitoring of actual performance. If, for instance, each party monitors the *fRD* and *rRD* metric and knows the link quality of the intermediate connection, it can estimate the *TPD* and derive the *fPD* or *rPD* at the other side using equations (1) and (2). These results could be used to verify if an SLA is fulfilled or not.

6. CONCLUSIONS AND OUTLOOK

In this paper, we present the concept of Quality of Signaling, which comprises a collection of metrics for individual SIP transactions. The idea behind QoSg is quite different from the concept of end-to-end performance, however, we argue that it is essential to understand the performance of single transactions before discussing end-to-end performance metrics over complex signaling frameworks. In the course of the paper, we have concentrated on the composition of the delay metrics and have isolated the *TPD*, which can be estimated through regression analysis. As the overall delay for a single transaction depends only on the *TPD* and the *fPD*, we can estimate the *fPD* as soon as the *fRD* is known, e.g. by measurements.

We have further proposed three potential application fields for QoSg. First of all, the results of the SIP transaction measurements can be used to identify which transport protocol is optimal for a certain link characteristic. Secondly, continuous measurements of the *fRD* allow detecting sporadic increase of the delay, which we can assume to be caused by an overload situation, if the quality of all links remains largely constant. Finally, we highlight the usefulness of these metrics also for SLA verification between service and application providers.

Current and future research will focus on analyzing in depth the suitability of the introduced metrics are for the all three application fields; especially the scenario of using QoSg for detecting overloaded SIP proxies and servers deserves special interest.

REFERENCES

[1] J. Rosenberg et al.: "SIP: Session Initiation Protocol". IETF, RFC 3261, June 2002.

[2] K. Nichols, S. Blake, F. Baker, D. Black: "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers". IETF, RFC2474, December 1998.

[3] J. Wroclawski: "The Use of RSVP with IETF Integrated Services". IETF, RFC 2210, September 1997.

[4] V. Paxson et al.: "Framework for IP Performance Metrics". IETF, RFC 2330, May 1998.

[5] D. Malas, A. Morton: "Basic Telephony SIP End-to-End Performance Metrics". IETF, draft-ietf-pmol-sip-perf-metrics-05, May 2010.

[6] T. Eyers, H. Schulzrinne: "Predicting Internet Telephony Call Setup Delay". IPTEL 2000 (First IP Telephony Workshop), 2000.

[7] K. K. Ram, I. C. Fedeli, A. L. Cox, S. Rixner: "Explaining the Impact of Network Transport Protocols on SIP Proxy Performance". IEEE ISPASS'08, pp 75–84, 2008.

[8] J. Postel: "User Datagram Protocol", IETF, RFC768, August 1980.

[9] J. Postel: "Transmission Control Protocol". IETF, RFC 793, September 1981.

[10] R. Stewart et al.: "Stream Control Transmission Protocol". IETF, RFC 2960, October 2000.

[11] M. Happenhofer, P. Reichl: "Measurement Based Analysis of Head-of-Line Blocking for SIP over TCP". Proc. 18th MASCOTS, Miami Beach, FL, U.S.A., Aug. 2010.

[12] M. Happenhofer, C. Egger, P. Reichl: "Quality of Signalling: A New Concept for Evaluating the Performance of Non-INVITE SIP Transactions". Proc. 22nd International Teletraffic Congress (ITC-22), September 2010.

[13] N. Draper, H. Smith: Applied Regression Analysis, New York, Wiley, 1998.