Computer Communications 33 (2010) S50–S57

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

TCP and web browsing performance in case of bi-directional packet loss

R.E. Kooij [a,b], R.D. van der Mei [c,d,*], R. Yang [d,c]

[a] TNO Information and Communication Technology, Department of Planning, Performance and Quality of Service, Delft, The Netherlands
[b] Delft University of Technology, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft, The Netherlands
[c] Center for Mathematics and Computer Science, Department of Probability and Stochastic Networks, Science Park 123, 1098XG Amsterdam, The Netherlands
[d] VU University, Faculty of Exact Sciences, Department of Mathematics, Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Available online 18 April 2010

Keywords:
Quality of service
TCP
Download times
Response times
Correlated packet loss

ABSTRACT

Performance modeling of the transport control protocol (TCP) has received a lot of attention over the past few years. The most commonly quoted results are approximate formulas for TCP throughput (Padhye et al. (2000) [1]) and document download times (Cardwell et al. (2000) [2]) which are used for dimensioning of IP networks. However, the existing modeling approaches unanimously assume that packet loss only occurs for packets from the server to the client, whereas in reality the packets in the direction from the client to the server may also be dropped. Our simulations with NS-2 show that this bi-directional packet loss indeed may have a strong impact on TCP performance. Motivated by this, we refine the models in Padhye et al. (2000) [1] and Cardwell et al. (2000) [2] by including bi-directional packet loss, also including correlations between packet loss occurrences. Simulations show that the proposed model leads to strong improvements of the accuracy of the TCP performance predictions. In addition we show how our model can be used to predict quality of experience for web browsing sessions.

1. Introduction

Performance problems in IP networks are well-recognized, and many studies have been investigating performance models for transport control protocol (TCP). Pioneering work in this field was done by Mathis et al. [3] and Ott et al. [4], who derive simple square-root formulas for the throughput of infinitely long TCP traffic flows under idealized periodic behaviour of the TCP congestion window, including the impact of the packet loss rate and the round trip time. More recent work in this direction is reported in [11,12]. Padhye et al. [1] propose a refinement of the models in [3,4] by taking into account detailed packet-level dynamics of the TCP window mechanism, and show that this refined model is able to more accurately predict TCP throughput and is accurate over a wider range of loss rates. However, the Padhye model does not take into account the behaviour of the TCP slow start and fast recovery mechanisms. Based on the model in [1], Cardwell et al. [2] propose a model for the mean total data download time to transfer limited amounts of data, explicitly taking into account TCP slow start and fast recovery. This model is quite accurate in predicting the mean transfer download time (TDT) under the assumption that losses happen only in the direction from the server to the client and losses in the successive rounds are independent; here, a round is the period of time between the departure of the first segment of the current window and the arrival of its acknowledgment. A limiting factor of the model in [2] is that it assumes that packet loss only occurs for data packets (i.e. for packets from the server to the client), whereas in reality the packets in the direction from the client to the server (e.g., ACKs) may also be dropped, which can have a significant impact on the TDTs experienced by the end-users. Although in the current Internet loss of ACK's is quite rare, it can be anticipated that in the near future loss of ACKs may become more common. The first reason for this is the increase of wireless and ad hoc networks in the access, which exhibit higher loss ratios than wireline networks. A second reason is the still increasing penetration of peer-to-peer applications which, due to their high uplink bandwidth consumption, may also lead to an increase of ACK losses.

In this paper, we study the performance of TCP in the presence of bi-directional packet loss. Extensive NS-2 [5] simulations demonstrate that bi-directional packet loss may indeed have a strong impact on the performance of TCP. Motivated by this, we extend the Cardwell model [2] by considering packet losses occurring in both directions. Since in practice packet losses are known to be correlated, we also include a class of correlated loss patterns into the model. Our main focus is on the interactive Web browsing type of applications. For this type of applications, the two main factors that determine the user-perceived performance of TCP-based applications are the response time, i.e. the time from clocking on a link until the first packet arrives and something appears on the

* Corresponding author at: Center for Mathematics and Computer Science, Department of Probability and Stochastic Networks, Science Park 123, 1098XG Amsterdam, The Netherlands. Tel.: +31 205924129.
E-mail address: mei@cwi.nl (R.D. van der Mei).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.04.011

screen, and the total download time, i.e. the time between clocking on a link and the arrival of the last packet. Hence, TDT can be decomposed into two parts: (1) the response time, and (2) the data transfer time. For this reason, our model consists of two sub-models: a sub-model for the response time and a sub-model for the data transfer time. First, we develop a new model for the RT, that gives an approximation for the mean response time as a function the loss probability, the round-trip time (RTT), the timer granularity G and the initial value of the retransmission timer T_0, in the presence of bi-directional packet loss. An important feature of the model is that it includes a dynamic scheme for the Retransmission Timer, which depends on the time granularity and the RTT. Second, we develop a new model for the data transfer time is the presence of bi-directional packet loss. To this end, we use the model in [2], which in turn uses the model in [1], as the basis and extend this to include the possibility of bi-directional packet losses. Third, we evaluate the accuracy of the model extensions by NS-2 simulations for a wide range of parameter settings. The results show that the models are indeed highly accurate for a wide range of parameter settings, and outperform the performance predictions based on [2] in many situations, particularly in situations where the packet loss ratio is significant (typically 4% or more) for sustained periods of time.

TCP performance modeling is notoriously difficult, due to the bursty nature of IP networks. To this end, we need to balance accuracy and complexity: the model to be proposed in this paper is developed in such a way that on the one hand it predicts the performance of TCP quite accurately, while on the other hand the model is still simple enough to provide insight in the impact of the model parameters on the performance. To this end, the model presented in this paper is explicitly built upon the models in [1,2] extending these models to include the specifics of bi-directional packet loss.

The contribution of this paper is threefold. First, we demonstrate via simulations that bi-directional packet loss may indeed have a strong impact on the performance of TCP (see for example Fig. 8). Second, the model for the RT in the case of bi-directional packet loss, including the dynamics of the Retransmission Timer, is new. Third, the extension of the Cardwell model [2] for the DTT by explicitly including the impact of packet losses occurring in both directions, and correlations between successive packet loss occurrences, has not been presented before. As such, the model proposed in this paper, which is explicitly built upon the most widely used models [1,2] is a significant step forward in understanding the performance of TCP.

The remainder of this paper is organized as follows. In Section 2, we discuss the metrics that determine the end-user's perception the Quality of Service of Web browsing applications, viz., the response time (RT) and the transfer download time (TDT). These results are used for the development of simple yet accurate models for the RT and the data transfer time (DTT), which relates to the TDT via Eq. (12) discussed below; these models explicitly capture the impact of bi-directional packet loss. In Section 3, we evaluate the accuracy of these models by comparing the performance predictions based on the models with simulations with NS-2. In Section 4, we discuss a simple model to include the impact of correlated loss patterns, and evaluate the accuracy of the model by simulations. The results demonstrate that the proposed model significantly outperforms the model in [2]. In Section 5, we apply our model to predict the quality of experience for web browsing. Finally, in Section 6, we address a number of topics for further research.

2. Modelling

In [6] extensive research has been to done to investigate the main factors that determine the end-user's perception of Quality of Service (QoS) for web browsing applications. The results of this research shows that the following two factors dominate the user perception of web browsing quality:

1. *Response time (RT)*: time from clicking on a link until first packet arrives and something appears on the screen, and
2. *Total download time (TDT)*: time between clicking on a link and the arrival of the last packet.

In Section 2.1, we develop an analytical expression for the RT, and in Section 2.2, we develop a model for the DTT. The models are then combined to obtain an expression for the TDT (see Eq. (12)).

2.1. Response time

In this section, we develop a model for the RT, i.e. the time it takes to establish a TCP connection and the additional time it takes to send the first packet containing data. From a user's point of view, the RT is simply the time from clicking on a URL link until the first packet arrives and something appears on the screen. Each TCP connection starts with a three-way handshake, in which the client and server exchange initial sequence numbers. The RT is determined by the time it takes to send four packets successfully; here, the first three packets are related to the three-way handshake while the fourth packet contains the first data.

Failures can be classified as two kinds. The first kind is due to a packet loss from the client to the server. The second kind is caused by a packet loss from the server to the client. For both kinds of failures, packets will be retransmitted from the client side. The forward data packet loss rate is denoted by p_f, and the backward packet loss rate by p_b. Moreover, suppose there are n packet losses before the first packet containing data are received. Let P_n be the probability of receiving the first data packet after exactly n packet losses, and define RT_n to be the conditional mean response time when the first data packet is received after exactly n packet losses.

Lemma 1. *Assuming that the packet loss occurrences are independent, we have: for $n = 0, 1, \ldots$,*

$$P_n = (n+1)(p_f + (1-p_f)p_b)^n(1-p_f)^2(1-p_b)^2. \quad (1)$$

Proof. It is convenient to define the concept of a cycle. It is simply the RTT if both a packet sent by the client and its ACK are sent successfully. We call this a successful cycle. Based on the assumed independence of packet loss occurrences, the probability that this occurs is $p_s = (1-p_f)(1-p_b)$. If either the packet sent by the client is lost or its ACK then the cycle is defined as the time between sending the packet and the time it is retransmitted. This will be denoted as an unsuccessful cycle. Assuming independence, the probability that this occurs is $p_u = p_f + (1-p_f)p_b$. If exactly n losses occur before the first data are received successfully then this implies that exactly $n + 2$ cycles have passed of which n are unsuccessful and two are successful. Obviously the last cycle needs to be successful. Therefore, for the other successful cycle there are $(n + 1)$ possible locations. Hence, $P_n = (n+1)p_u^n p_s^2$. Substitution of the values of p_u and p_s yields the result. □

For the situation described in this section, loss is detected through a Retransmission Time Out (RTO). If a packet sent by the client is not acknowledged before the Retransmission Timer expires then the packet is retransmitted. With each retransmission the Retransmission Timer is doubled, up to a maximum value 64 times its original value. Denote by T_0 the initial value of the Retransmission Timer. The simulation package NS-2, see [5], uses $T_0 = 6$ s, although RFC2988 [7] recommends $T_0 = 3$ s. Note that

TCP uses sample values of the RTT to adjust the Retransmission Timer. However, according to Karn's algorithm, this only occurs for packets that are not being retransmitted. Hence, in our situation, such an adjustment can only occur if the first two packets are sent successfully. According to [7], upon its first update the Retransmission Timer becomes

$$T_u = max\{1, RTT + max\{G, 2RTT\}\}, \tag{2}$$

where G denotes the TCP timer granularity. In many TCP implementations G is set to 500 ms. Let T_n be the average value of the Retransmission Timer, given the first data packet is received after exactly n losses. Then based on [1], it can be shown that the conditional mean response time is given by the following expression: for $n \leqslant 6$,

$$RT_n = (2^n - 1)T_n + 2RTT, \tag{3}$$

and for $n \geqslant 7$,

$$RT_n = [63 + 64(n - 6)]T_n + 2RTT, \tag{4}$$

where

$$T_n = \frac{1}{n+1}T_u + \frac{n}{n+1}T_0. \tag{5}$$

Eq. (5) follows from the fact that, given exactly n packet losses, $n + 1$ attempts were needed to successfully transmit a packet. Consequently, with probability $n/(n + 1)$ the Retransmission Timer was set to its default T_0, whereas the timer was set to T_u with probability $1/(n + 1)$.

Lemma 2. *Assuming the packet loss occurrences are independent, the mean response time RT is given by the following expression:*

$$RT = 2RTT + \frac{A}{1-A}\sum_{k=0}^{7} b_k A^k, \tag{6}$$

where $A = p_f + (1 - p_f)p_b$, $b_0 = T_0 + T_u$, $b_1 = 3T_0$, $b_2 = 6T_0 + T_u$, $b_3 = 14T_0 + 2T_u$, $b_4 = 32T_0 + 4T_u$, $b_5 = 72T_0 + 8T_u$, $b_6 = 160T_0 + 16T_u$ *and* $b_7 = -160T_0 - 32T_u$.

Proof. By conditioning on the value of n, it is readily seen from Eqs. (3)–(5) that RT satisfies

$$\begin{aligned} RT &= \sum_{n=0}^{\infty} P_n RT_n = \sum_{n=0}^{6} P_n\left[(2^n - 1)T_n + 2RTT\right] \\ &\quad + \sum_{n=7}^{\infty} P_n[63 + 64(n-6)T_n + 2RTT]. \end{aligned} \tag{7}$$

After a tedious but straightforward calculation, which was performed with the help of the Computer Algebra software Maple, this expression can be simplified to (6). □

2.2. Data transfer time

The Data Transfer Time (DTT) is the time between sending the first data packet and receiving the last data packet. From the previous section, we know the TCP connection establishment time is the time taken to send three packets successfully. Therefore, *we approximate the connection establishment time by 3/4 times the RT*, discussed in Section 2.1. In [2], a model is proposed under the assumption that packet loss happens only in the direction from sender to receiver. This model directly depends on this one-way packet loss, whereas in reality not only data segments can be lost during a TCP data transmission but also the ACKs of data packets can be dropped in the direction from the receiver to the sender. Therefore, we extend the Cardwell model by including the impact of the loss of ACKs.

Packet loss may occur in one of the following two cases:

Case 1 (*Packet is lost during transmission from the sender to the receiver*). As a first step, we discuss in which situation a data packet is considered lost by TCP. We focus on a single data packet. We assign to this data packet an index k indicating the position of the data packet within the current window. Obviously, if k is lost during transmission from the sender to the receiver, then this packet is considered lost by the sender TCP. The probability of the occurrence of Case 1 is equal to p_f.

Case 2 (*Packet is sent successfully, but the ACK is lost*). Case 2 occurs when packet k is sent successfully, but the ACK for packet k is lost. The probability of the occurrence of Case 2 is $(1 - p_f)p_b$. For Case 2, we cannot determine immediately whether data packet k is considered lost by the sender TCP or not. In fact, if packet $k + 1$ and its ACK are sent successfully then the sender does not notice that the ACK of packet k was lost, hence packet k is not considered lost. If either packet $k + 1$ or its ACK is lost then we have to take into account packet $k + 2$ and its ACK. We have to repeat this analysis until we have considered all packets in the current window. In conclusion, in Case 2, if there exists $m \in \{1,\ldots,w - k\}$, the ACK of packet $k + m$ is received by the sender, then the sender is ensured that packet k is sent successfully. Here, w denoted the window size. For this situation, Fig. 1 (left) illustrates an example. If for all $m \in \{1,\ldots,w - k\}$ the ACK of packet $k + m$ is not received by the sender, packet k is considered lost by the sender TCP, see Fig. 1.

Denote p_k as the probability that packet k is considered lost by the sender. Then we have,

$$p_k = \left[p_f + (1 - p_f)p_b\right]^{w-k}. \tag{8}$$

Therefore, in Case 2 the expected probability that a packet is considered lost by the sender can be expressed as follows:

$$\sum_{k=1}^{w}\frac{p_k}{w} = \frac{1}{w}\sum_{k=1}^{w}\left[p_f + (1 - p_f)p_b\right]^{w-k}. \tag{9}$$

Combining the results for Case 1 and Case 2, the complete expression for p, denoting the probability that a packet is considered lost by the sender, can be deduced to:

$$\begin{aligned} p &= p_f + \frac{(1-p_f)p_b}{w}\sum_{k=1}^{w}\left[p_f + (1-p_f)p_b\right]^{w-k} \\ &= p_f + \frac{p_b}{w(1-p_b)}\left[1 - \left(p_f + p_b - p_f p_b\right)^w\right]. \end{aligned} \tag{10}$$

Expression (10) includes the unknown variable w, the current window size. In order to complete the DTT model for bi-directional packet loss we substitute w by the minimum of W_{max} and the average window size given in [1]. Thus, we approximate w as:

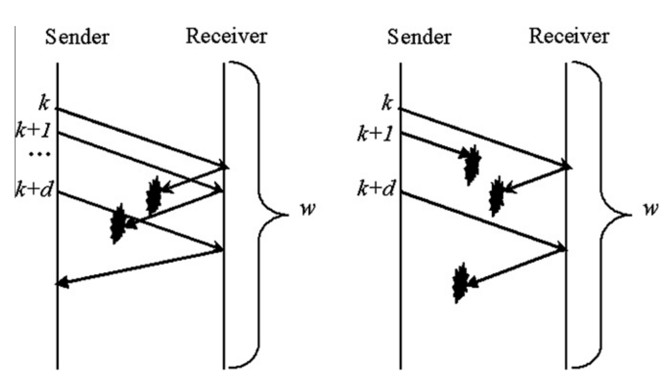

Fig. 1. Packet k considered sent (left) and lost (right).

$$w = \min\left\{W_{max}, \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2}\right\}. \quad (11)$$

Our model for the TDT is now complete:

$$\mathrm{TDT} = (3/4)\mathrm{RT} + \mathrm{DTT}(p), \quad (12)$$

where RT denotes the mean response time derived in Section 2 and DTT (p) is the Cardwell formula for the DTT where we use the packet loss probability p given in (10) where w satisfies (11). The Cardwell formula for DTT (p), extensively discussed in [2], is briefly described in Appendix A.

3. Validation

To assess the accuracy of the models developed in Section 2, we have performed extensive simulations. The results are outlined below. The topology of our simulation is depicted in Fig. 2. In Section 3.1, we assess the accuracy of the model for the RT developed in Section 2.1, and in Section 3.2 we validate the model for the TDT developed in Section 2.2.

3.1. Response times

For the validation of the model for the mean RT, the packet loss probability at the aggregation link n1-n2 has been varied between 0% and 10%. It is assumed that the packet loss is random (i.e. without correlation between consecutive lost packets) In our simulation packets in both up-and downlink directions suffer from this packet loss. For ease of the discussion, we assume that the packet loss rate in both directions is the same, i.e. $p_f = p_b$. The (minimum) RTT is set to be 600 ms. The NS-2 script has recorded actual values of the packet loss, RTT and the mean RT. Fig. 3 shows the mean RT as a function of the loss probability, for the simulations and for the model.

We conclude from Fig. 3, and many additional simulation results not shown here, that our model for the mean RT is highly accurate over the whole range of loss rates considered. In fact, analytical result for the mean RT always falls within the 95%-confidence interval of the simulated values.

3.2. Total download times

In this section, we report NS-2 simulation experiments that have been run as a validation for our model for the mean TDT. The simulation topology is given by Fig. 2. We ran several simulations where we varied the access link rate, the RTT, the maximum window size W_{max} and the file size. For all simulations we have set the number of data packets acknowledged by one ACK equal to 1 while the initial slow-start window size is 1 packet. In Figs. 4 and 5 the Maximum Segment Size (MSS) is 1640 Bytes and the link capacity equals 200 Mbps. The file size was taken to be 1000 packets. The forward loss probability is set equal to the backward loss probability (i.e. $p_f = p_b$), and the delayed acknowledgment option is disabled.

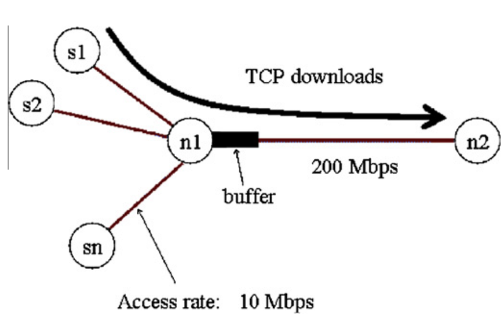

Fig. 2. Simulation topology.

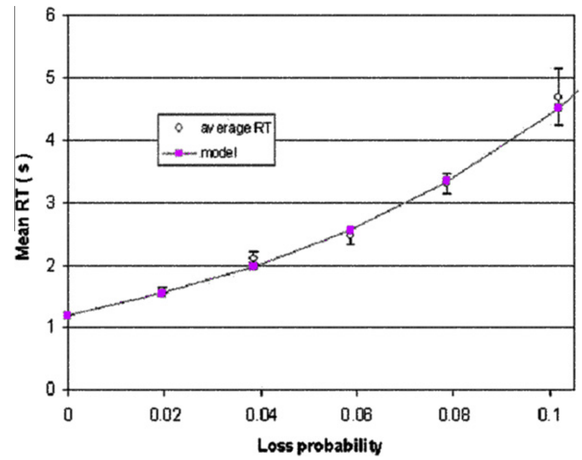

Fig. 3. Comparison between model and simulation for mean response time.

Figs. 4 and 5 and additional simulations show that our model works very well for predicting the mean TDT for large files. The relative errors under these new situations are very low; in most cases they are within ±6%. For small files our model is very accurate under the condition that the forward and backward loss probabilities are at most 5%.

The experiments mentioned above are all under the assumption that the forward loss probability is equal to the backward loss probability. Obviously, in a realistic network environment this is not necessarily the case. Therefore, we simulated another scenario in which the loss probabilities in both directions are not necessarily equal. For this experiment we set MSS = 1000 Bytes with a link capacity of 100 Mbps. The files size is set to 500 packets, and the delayed acknowledgment option is disabled. Furthermore, we assume that RTT = 0.3 s, access-link capacity = 2 Mbps, $W_{max} = 32$ packets, while the backward loss probability is fixed at $p_b = 2\%$. The forward loss probability p_f is varied between 0% and 10%.

From Fig. 6, and similar simulation results not presented here, we conclude that our model is also accurate for predicting the mean TDT under the situation that the forward packet loss probability is unequal to the backward packet loss probability.

4. Correlated packet loss

According to [8], the correlation structure of the packet loss process can be modeled with an underlying Markov chain. In particular, the two-state Gilbert model was found to be an accurate model in many studies, see for instance [9,10]. Therefore, we use the Gilbert model to simulate packet loss patterns over links. For simplification, we only consider packet losses in the forward direction, thus it is assumed that ACK's are not lost, i.e. $p_b = 0$, and moreover, the delayed acknowledgment option is disabled.

In the Gilbert model, one state represents a lost packet which is called state B and the other state represents the situation when a packet is successfully delivered to the destination which is referred to as state G. Let m denote the probability of going from state G to state B, and n is that of staying in state B. On the link between the sender and the receiver, the average packet drop rate then satisfies

$$\pi_B = \frac{m}{1+m-n}. \quad (13)$$

Substituting this packet loss probability defined in our model for TDT (replace p_f by π_b), we get the model for predicting the mean

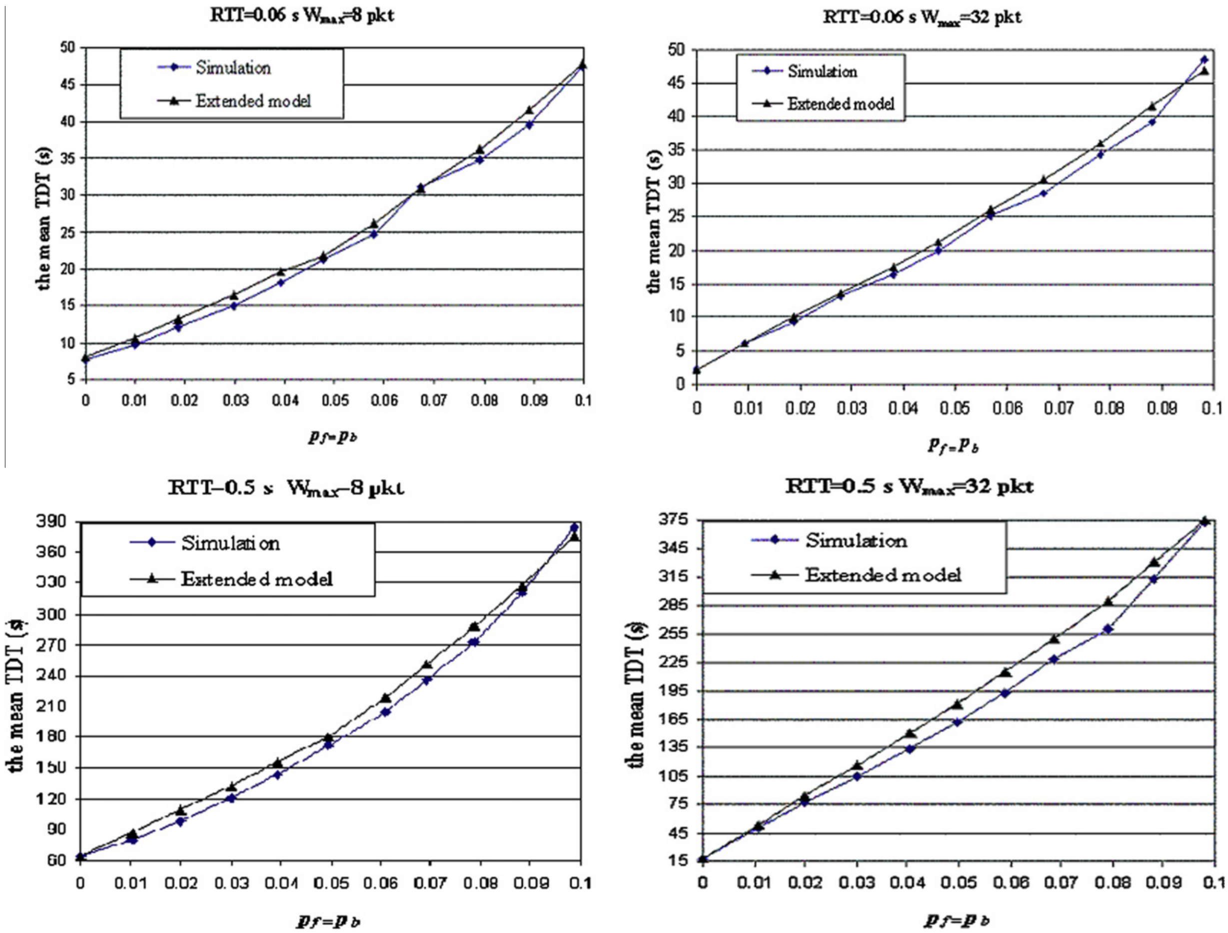

Fig. 4. Simulation vs. model results for bi-directional packet loss: access link = 30 Mbps.

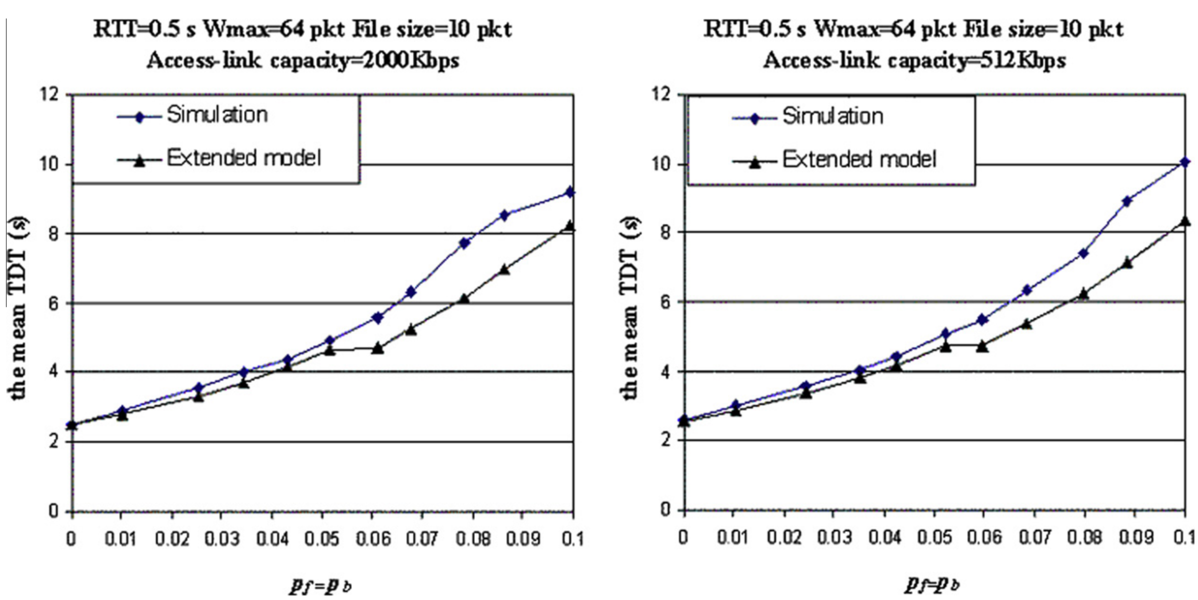

Fig. 5. Simulation vs. model results for bi-directional packet loss: access link = 2 Mbps (left), access link = 512 kbps (right).

TDT in the situation when packet losses are correlated according to the two-state Gilbert loss model. To validate the accuracy of the model for the case of correlated packet loss, we ran a variety of additional NS-2 simulations for the simulation topology depicted in Fig. 2. For this experiment, MSS = 1000 Bytes while the link capacity was set to 200 Mbps. Furthermore, we assume that RTT = 0.2 s, access-link capacity = 10 Mbps, W_{max} = 32 packets. In the forward direction, the probability m of going from state G to

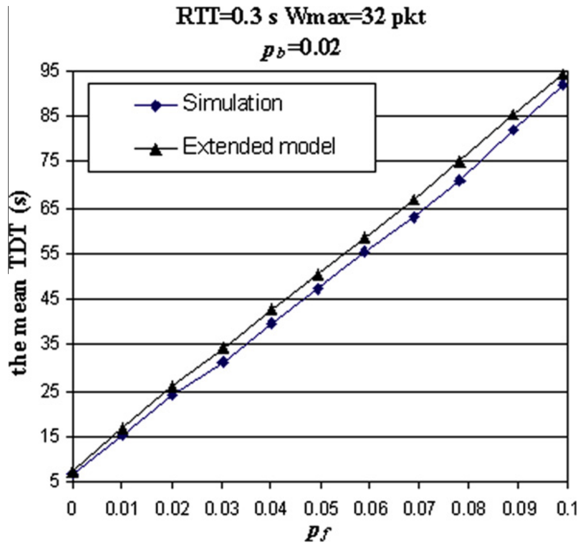

Fig. 6. Simulation vs. model results for bi-directional packet loss with fixed ACKs loss rate.

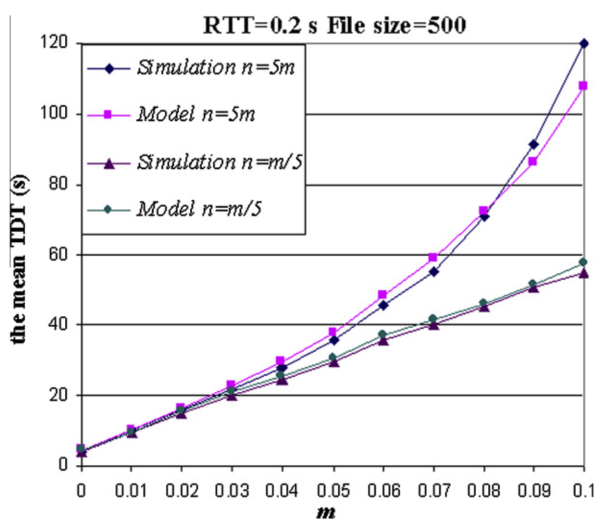

Fig. 7. Simulation vs. model results for correlated packet loss.

state B: $\{0, 0.01, \ldots, 0.1\}$, while the probability n of going from state B to state G is in $\{m/5, 5m\}$. Fig. 7 shows the mean TDT as a function of m.

Fig. 7 demonstrates that when packet loss on links is correlated and the correlation of packet losses is known, we can apply our model to predict the mean TDT by substituting the packet loss probability p_f by π_B.

To justify the relevance of the inclusion of bi-directional packet loss in the model, we again have performed various simulations. Fig. 8 shows the mean TDT as a function of the packet loss ratio for the following two scenarios. In Scenario 1 (left-hand side), the RTT = 60 ms and the maximum TCP window size 8 packets, while in Scenario 2 (right-hand side) we have the RTT = 500 ms and the maximum window size 32 packets. In both cases, the file size was set to 1000 packets.

The results shown in Fig. 8 demonstrate that our extended model outperforms the Cardwell model [2], especially when the packet loss becomes significant. Moreover, it shows that the inclusion of bi-directional packet loss is indeed justified, and leads to more accurate performance predictions. As such the model extension discussed in the paper leads to more accurate TCP performance predictions.

5. Web browsing performance as perceived by users: mean opinion scores

The aim of this section is to apply our model to determine the quality of a web browsing session, as perceived by users. An important observation in modeling the perceived quality in web browsing (cf. [6]) is the fact that the expected maximal session time will dominate the perceived quality. If one expects a session time of 100 s, the perceived quality of a 10-s session will be much higher than if one expects a session time of 1 s. In general, quality perception related to response time can be classified according to the following three perceptual regions [13]:

- *Instantaneous experience:* 0.1 s is about the limit for having the feel that the system is reacting instantaneously, an important limit for conversational services (e.g., chatting).
- *Uninterrupted experience:* 1.0 s is about the limit for the user's flow of thought to stay uninterrupted, even though the user does lose the feeling that the service is operating directly, an important limit for interactive services (e.g., gaming).

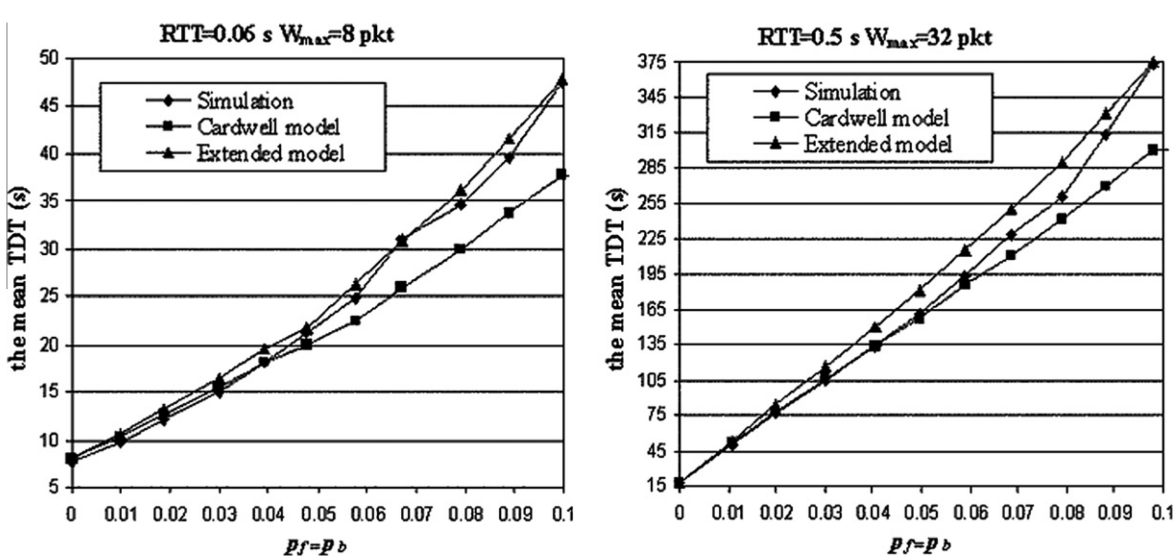

Fig. 8. Simulation vs. model results for correlated packet loss.

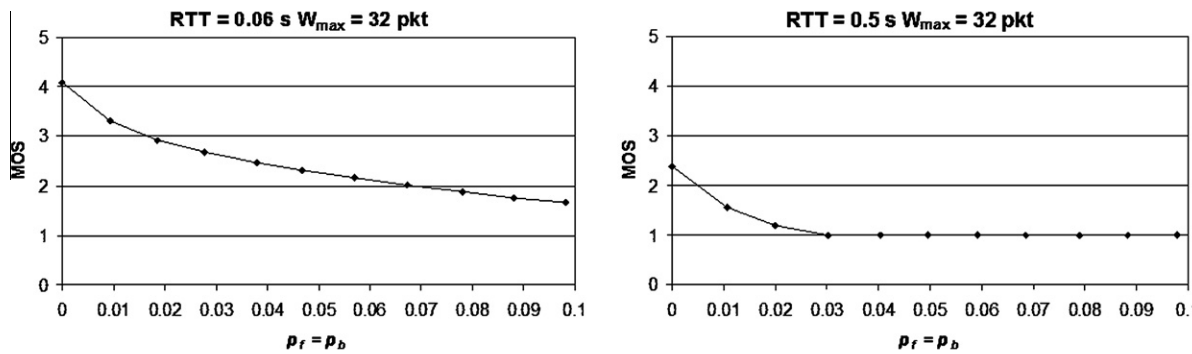

Fig. 9. Perceived quality of web browsing for bi-directional packet loss.

- *Loss of attention:* 10 s is about the limit for keeping the user's attention focused on the dialogue. For longer delays, users want to perform other tasks while waiting for the computer to finish, so they should be given feedback indicating when the computer expects to be done. Feedback during the delay is especially important if the response time is likely to be highly variable, since users will then not know what to expect.

Regarding download times, users tend to adapt their quality judgment towards the expected download time [14]. When users are informed about the expected download time, they are willing to accept large download times.

Recommendation ITU-T G.1030 [6] describes the explicit relation between the Response Time and the Total Download Time, and the quality as experienced by users. The quality of experience is expressed in terms of a Mean Opinion Score (MOS), a five-point scale based upon ITU-T Rec. P.800 [14]: 5: excellent, 4: good, 3: fair, 2: poor, and 1: bad.

According to [6] for a web browse session characterized by a Response Time RT and Total Download Time TDT, the MOS score satisfies

$$\mathrm{MOS} = \frac{4}{\ln((0.005Max + 0.24)/Max)} \times (\ln(TDT) - \ln(0.005Max + 0.24) + 5), \tag{14}$$

where *Max* denotes the expected maximum Total Download Time in seconds.

Note that the Response Time only appears implicitly in (14) as it is part of the Total Download Time (TDT), see also (12). It is claimed in [6] that (14) is applicable for TDTs between 5 and 100 s. Therefore, we will apply (14) under the assumption that *Max*=100 s, which leads to the following relation:

$$\mathrm{MOS} = 4.75 - 0.81\ln(\mathrm{TDT}). \tag{15}$$

Next, we will apply (15) to two scenarios introduced in Section 3 (i.e. the topology is given by Fig. 2), where for each scenario the access link rate equals 30 Mbps. Furthermore, we assume that the number of data packets acknowledged by one ACK equals one, while the initial slow-start window is one packet. The MSS is again 1640 Bytes and the link capacity is 200 Mbps. The file size is set to 1000 packets and Wmax is fixed at 32 packets. The forward loss probability is set equal to the backward loss probability (i.e. $p_f = p_b$) and the delayed acknowledgment option is assumed to be disabled. We will consider two values for the RTT: 0.06 s and 0.5 s. Note that these two scenarios correspond to the two right-most figures depicted in Fig. 4. Fig. 9 shows the web browsing performance as experienced by users for the two scenarios.

According to [14] in order to obtain an acceptable perceived quality, the MOS score should be at least 3.5. Therefore, we conclude from Fig. 9 that for the scenario with RTT = 0.5 s we can never realize acceptable perceived quality. On the other hand, for RTT = 0.06 s, we can deduce that the perceived quality is acceptable as long as $p_f < 0.68\%$.

The above two examples illustrate that our TCP performance model can also be used to assess the perceived quality for web browsing in terms of MOS. It would be interesting to validate the perceived quality model (15) by means of subjective experiments that are based upon real TCP traffic.

6. Conclusions and topics for further research

In this paper, we have extended the commonly used TCP performance model for the download times by Cardwell et al. [2] by including the impact of bi-directional packet loss. Extensive simulations with NS-2 show that this bi-directional packet loss may have a strong impact on TCP performance, and that the proposed model refinement accurately captures the impact of bi-directional packet loss, and correlations between the loss occurrences. As such the model refinement is highly valuable for IP link dimensioning purposes. In addition we have applied our model to predict the quality of experience for web browsing.

The results lead to a number of challenges for further research. First, the model considered in this paper is focused on TCP performance over a single network domain. However, next-generation communication services (e.g., online consumer services, E-commerce applications) will typically have highly distributed architecture, crossing multiple administrative domains. Extension of the model towards the inclusion of multiple domains is a challenging topic for further research. Second, in the present paper we considered a simple correlation structures between packet loss occurrences, whereas in reality the loss patterns may be far more complicated. Extension of the model including more realistic packet loss patterns is a challenging area for further research. Finally, in many communication networks TCP-based applications and UDP-based applications will be integrated. Inclusion of the impact of UDP streams on the performance of TCP is also a challenging area for further research. Finally, it would be interesting to validate the perceived quality model (15) by means of subjective experiments that are based upon real TCP traffic.

Appendix A. The Cardwell formula

Below we briefly describe the Cardwell formula for the Data Transfer Time (DTT), considered as a function of the packet loss rate. The reader is referred to [2] for a more in-depth discussion. Adopting the terminology introduced in [2], the Cardwell formula for the DTT can be decomposed as follows:

$$\mathrm{DTT}(p) = E[T_{ss}] + E[T_{loss}] + E[T_{ca}] + E[T_{delack}], \tag{16}$$

where T_{ss} is the latency for the initial slow start, T_{loss} is the delay due to any Retransmission Time Out (RTO) or fast recovery that happens at the end of the initial slow-start, T_{ca} is the time to send the remaining data (i.e. the amount of data after slow start or any following loss recovery), and T_{delack} is the delay between the reception of a single segment and the delayed ACK for that segment. Expressions for each of the four terms in (16) are detailed out below.

First, $E[T_{ss}]$ is approximated by the following expression:

$$E[T_{ss}] = \begin{cases} RTT\left[\log_\gamma\left(\frac{W_{max}}{w_1}\right) + 1 + \frac{1}{W_{max}}\left(E[d_{ss}] - \frac{\gamma W_{max} - w_1}{\gamma - 1}\right)\right] & \text{when } E[W_{ss}] > W_{max} \\ RTT \log_\gamma\left(\frac{E[d_{ss}](\gamma-1)}{w_1}\right) & \text{otherwise,} \end{cases} \tag{17}$$

where $E[d_{ss}]$, the number of data segments we expect the sender to send before losing a segment, is given by the following expression:

$$E[d_{ss}] = \frac{(1-(1-p)^d)(1-p)}{p} + 1, \tag{18}$$

with p the data segment loss rate and d the number of data segments to be transmitted. Moreover, W_{max} is the maximum window size, γ is the rate of exponential growth of the congestion window during slow start, and w_1 is the number of segments in the initial congestion window.

Second, $E[T_{loss}]$ can be expressed as follows:

$$E[T_{loss}] = l_{ss}\left(Q(p, E[W_{ss}])E[Z^{TO}] + (1 - Q(p, E[W_{ss}]))RTT\right), \tag{19}$$

where

$$l_{ss} = 1 - (1-p)^d \tag{20}$$

is the probability that initial slow start phase ends with the detection of a packet loss, which can occur either via retransmission timeouts (RTOs) or triple duplicate ACKs. The probability that a sender in congestion avoidance will detect a packet loss with an RTO can be expressed in terms of the packet loss rate p and the window size w as follows (cf. [1]):

$$Q(p, w) = \min\left(1, \frac{1 + (1-p)^3(1-(1-p)^{w-3})}{(1-(1-p)^w)/1-(1-p)^3}\right). \tag{21}$$

Moreover, the expected cost of an RTO, $E[Z^{TO}]$ is given by (cf. [1]):

$$E[Z^{TO}] = \frac{G(p)T_0}{1-p}, \tag{22}$$

where T_0 is the average duration of the first timeout in a sequence of one or more successive timeouts, and $G(p)$ is given by

$$G(p) = 1 + p + 2p^2 + 4p^3 + 8p^4 + 16p^5 + 32p^6. \tag{23}$$

Third, $E[T_{ca}]$ is given by

$$E[T_{ca}] = \frac{E[d_{ca}]}{R(p, RTT, T_0, W_{max})}, \tag{24}$$

where

$$E[d_{ca}] = d - E[d_{ss}] \tag{25}$$

is the mean number of data segments left after slow start and any following loss recovery, and

$$R = \begin{cases} \dfrac{\frac{1-p}{p} + \frac{w}{2} + Q(p,w)}{RTT\left(\frac{b}{2}w + 1\right) + \frac{Q(p,w)G(p)T_0}{1-p}} & \text{if } w < W_{max} \\ \dfrac{\frac{1-p}{p} + \frac{W_{max}}{2} + Q(p,W_{max})}{RTT\left(\frac{b}{8}W_{max} + \frac{1-p}{pW_{max}} + 2\right) + \frac{Q(p,W_{max})G(p)T_0}{1-p}} & \text{otherwise,} \end{cases} \tag{26}$$

and where w is given by Eq. (11). Finally, $E[T_{delay}]$, the expected delay between the reception of a single segment and the delayed ACK for that segment is Operating System specific, and typically take a value in the range of 100–150 ms.

References

[1] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, IEEE/ACM Transactions on Networking 8 (2000) 133–145.
[2] N. Cardwell, S. Savage, T. Anderson, Modeling TCP latency, in: Proceedings of INFOCOM 2000, Tel Aviv, Israel, 2000.
[3] M. Mathis, J. Semke, J. Mahdavi, The macroscopic behavior of the TCP congestion avoidance algorithm, Computer Communication Review 27 (1997).
[4] T. Ott, J. Kemperman, M. Mathis, The stationary behavior of ideal TCP congestion avoidance. Preprint.
[5] Network Simulator 2 (software). Available from: <http://www.isi.edu/nsnam/ns>.
[6] ITU-T Rec. G.1030. Estimating end-to-end performance in IP networks for data applications, International Telecommunication Union, Geneva, Switzerland, 2005.
[7] V. Paxson, M. Allman, Computing TCP's retransmission timer, in: Internet Archives RFC 2988, November 2000.
[8] C. Boutremans, J.Y. Le Boudec, Adaptive joint playout buffer and FEC adjustment for internet telephony, in: Proceedings of IEEE Infocom 2003, 2003.
[9] E. Altman, K.E. Avrachenkov, C. Barakat, TCP in presence of bursty losses, in: Proceedings ACM Sigmetrics, Santa Clara, CA, 2000.
[10] E. Altman, K.E. Avrachenkov, C. Barakat, P. Dube, TCP over a multi-state Markovian path, in: K. Goto, T. Hasegawa, H. Takagi, Y. Takahashi (Eds.), Performance and QoS in Next Generation Networking, Springer, Berlin, 2000, pp. 103–122.
[11] A. Hafsaoui, D. Collange, G. Urvoy-Keller, Revisiting the performance of short TCP transfers, in: L. Fratta et al. (Eds), NETWORKING 2009, LNCS vol. 5550, 2009, pp. 260–273.
[12] D. Damjanovic, P. Gschwandtner, M. Welzl. Why is this web page coming up so slow? Investigating the loss of SYN packets, in: L. Fratta et al. (Eds), NETWORKING 2009, LNCS, vol. 5550, 2009, pp. 895–906.
[13] ITU-T Rec. P.800. Methods for subjective determination of transmission quality, International Telecommunication Union, Geneva, Switzerland, 1996.
[14] J. Nielsen, Response Times: The Three Important Limits, 1994. Available from: <http://www.useit.com/papers/responsetime.html>.