A new methodology for TCP evaluation in a multiuser web environment

E. González-Parada*, J.M. Cano-García, A. Díaz-Estrella

Dpto. Tecnología Electrónica, ETSI Telecomunicación, University of Málaga, Campus de Teatinos, 29071 Malaga, Spain

Received 16 February 2004; revised 3 August 2004; accepted 19 August 2004
Available online 13 September 2004

Abstract

This paper presents a new methodology to evaluate and graphically represent TCP performance in a web environment. The main novelty of this work is that it focuses on web environments involving a large number of connections, where the traffic model is extracted from real traces. In these cases, conventional tools are not efficient to handle the complexity of the analysis. The proposed framework includes: (i) a set of representative web browsing scenarios affected by different types of losses; (ii) a new analysis methodology to cope with the huge data volume related to the simulated connections and; (iii) a new graphical representation to allow an easy visualisation of the simulation results. To test the proposed methodology, an evaluation of the impact of two representative TCP configuration parameters for web traffic has been included.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Web traffic; TCP; Network protocols modelling; Simulation and analysis

1. Introduction

TCP is the most widely used Internet transport protocol, as most popular Internet application (FTP, e-mail and web browsing) rely on it. TCP achieves correct data transmission by means of retransmission mechanisms and also includes congestion control mechanisms to avoid network overloading. Because of its popularity, TCP is also highly dynamic. Since it is under continuous research to adapt itself to emerging requirements, it keeps evolving into more complete and efficient versions, which depend on different configuration parameters. Many researchers have focused on evaluating the performance of different TCP versions and parameters [1–6]. However, most of them deal with a few connections because of the enormous complexity of web traffic, which involves a large number of coexisting connections. It has been stated [7] that it would be interesting to study TCP in more complex scenarios, but these scenarios should be controlled so that: (i) the TCP version and parameters of the users are known; and (ii) background traffic can be controlled. Thus, it is necessary to work with simulators. Simulations involving so many coexisting connections are not simple and have not been feasible until recently [8], because they involve: (i) accurate models capturing the main features of real web traffic; (ii) a reliable Internet model; and (iii) a model of all protocols involved in the study. It is also difficult to graphically represent the results of such simulations in a simple, intuitive way.

This paper presents a new methodology to cope with the increasing complexity and data volume of TCP simulations involving web traffic. Its subgoals include: (i) a traffic model to extract the main features of Internet traffic which is computationally feasible for efficient simulations; (ii) a new analysis and graphic representation method to present results. Section 2 presents a fairly realistic simulation scenario for web traffic and the proposed analysis framework, while Section 3 presents the new analysis method. In order to test the proposed methodology, it has been used to evaluate how TCP is affected by changes in some important configuration parameters (Initial Window and MSS) in a complex scenario. Even though it could be extended to any other version, this example study focus on NewReno TCP because it is one of the most widely deployed ones [9] and includes all the basic congestion control and loss recovery TCP mechanisms. These results are presented in Section 4.3. Finally, conclusions are given in Section 5.
2. An overview of the evaluation scenario

The main novelty of the proposed methodology is that it has been developed to handle web traffic involving a large number of coexisting connections. Thus, it is necessary to create a simulated scenario capable of reproducing the main features of Web traffic. As aforementioned, simulation is required to keep a controlled scenario [10]. Specifically, we have chosen a simulated scenario, which is both representative and fairly realistic (Fig. 1). It consists of several web clients connected to Internet through an access network. These clients extract information from different servers during web browsing. Such a scenario is fairly realistic because web applications have been reported to generate most TCP transactions in the Internet for years [11] and, consequently, the performance of web browsing has a strong impact on the client’s perception of the service. Obviously, to achieve a totally realistic scenario, the topology of the network should be more complex, but this would very significantly increase the computational load of the simulations and also make them sensitive to many extra configuration parameters not really related with TCP.

We work with a discrete event simulation kernel, where we have included a new traffic model extracted from real web traffic, a TCP model, which has been validated against the Network Simulator 2 (NS2) TCP model, and the network topology in Fig. 1. In fact, we could have used NS2, but its TCP versions are unidirectional except TCP Reno. In web traffic, even though connections are significantly asymmetric, traffic is bidirectional. Hence, we developed a new TCP simulation library where all TCP versions are bidirectional. All simulations have been conducted on a Ultra10 Sun Spare Station, where each run took around 12 h to simulate 500,000 s for the scenarios further described in this paper. The following subsections explain the most significant features in our simulations. It is important to point out, though, that the proposed methodology could be applied to any other simulator.

2.1. Access network

Our access network is modelled as a full-duplex 10 Mbps link with drop-tail policy queues. Different TCP connections belonging to the same or different users may be simultaneously established and, thus, contend for the access network bandwidth. The access network represents the bottleneck link and captures the correlation between the load generated by traffic sources and packet drops in the network.

2.2. The internet model

To model Internet, we have taken into account two main factors: (i) network delays and losses affect TCP performance [12]; and (ii) network conditions may also affect traffic because of feedback caused by TCP congestion control [13]. Even though it is a very difficult task [8], several authors have addressed the characterisation of Internet end-to-end delay and packet drops [14–16].

The Packet flying time through Internet is not constant because it depends on the number of hops between the endpoints and the links bandwidth, latency and load. In some works delay in packets flowing between two given endpoints has been modelled as a stochastic process using a probability distribution function. The statistical properties of this delay can be found in [14–16], but no simulation models are available. Kalden [17] proposes a Gaussian distribution to model packet delay, but does not take into account the delay correlation in packets belonging to the same connection. However, close packets are expected to suffer a similar delay because they follow the same path under similar conditions [18]. Unfortunately, both the models in [18] and [17] may present a spurious behaviour because they frequently produce packet disorder that provokes unnecessary retransmission and congestion control activation, heavily affecting TCP performance. Although packet reordering is possible for certain network configurations [19], the reordering degree induced by the previous models is not frequent in Internet. Since no suitable delay model is available, we use a fixed delay for all the packets in each given connection.

In our scenario, losses are modelled by capturing the packet drop process in a given path. Our framework includes two microscopic level models: a simple Bernoulli model and a Loss Burst model. In the Bernoulli model, a loss probability $p$ is independently assigned to each packet in a connection. $p$ determines if the packet is lost or successfully transmitted through Internet. Besides, Internet losses are not generally due to errors, but due to the congestion of some routers in the path of the transmitted packets. Thus, losses usually appear in bursts, as proven in several studies [15,16], so we also include burst losses in our framework. We use a two-state Markov chain to model burst losses as proposed in [15], where each state presents an exponentially distributed duration. It is necessary to note, though, that the burst length is often bimodal rather than exponentially distributed [16,20]. Some works focus on this behaviour [20,21], but we have not included this effect to keep our scenario well controlled.
2.3. Traffic model

World Wide Web (WWW), now representing more than 50% of the total traffic on the Internet [11], is nowadays the most important application in the Internet community. The traffic pattern generated by WWW browsing is rather complex and, hence, most studies work with simple, non-realistic traffic models. However, a great effort has been dedicated recently to develop more realistic traffic models for Internet applications [22–28]. Accurate WWW traffic modelling is quite hard [8] because of (i) the coexistence of different protocol versions and configurations, and (ii) the changing nature of the web environment. Besides, web traffic is tightly coupled with the human user behaviour, the browsing pattern of different users may be complex and diverse. Finally, web traffic complexity is also increased by non-user triggered information retrieval, typically caused by pages designed with frames, redirections or popup banners.

There are basically two different traffic modelling strategies. The *behaviourist* modelling approach (also known as *black-box* modelling) relies on statistical traffic characterisation. Its main drawback is that it cannot predict the behaviour of the network when new applications are introduced or when current ones evolve into new, non-statistically characterised traffic patterns. *Structural* modelling strategies are more computationally expensive but they offer a more comprehensive and meaningful way to fully characterise the complex nature of traffic. Structural modelling is specially suitable to resemble the Internet changing nature traffic because a proper design of the model structure allows to cope with the application evolution and to track changes in traffic patterns. Consequently, the model proposed in our framework is a structural one. In structural models the activity of a WWW user session is reproduced by means of a hierarchical layered structure, whose parameters are statistically characterised to capture the most relevant real traffic features [22,25–27]. Although all these models are different, they present a similar structure presented in Fig. 2. Each one of these layers is modelled as described below:

- **Session Level**: there are two different approaches to model the session level when reproducing the aggregated traffic of several users: (i) a birth–death model may be used to simulate a large population of users or (ii) many simultaneous ON/OFF sources can also be used. This level is typically defined by the following parameters:
  - The *number of pages* visited in each session.
  - The *session interarrival time* for the birth-death model or the *time between session* for the ON/OFF model.

- **Page Level**: this layer is modelled according to an ON/OFF type process corresponding to the activity after receiving a web request (ON) plus a silent period for page viewing after all objects are retrieved (OFF). After expiration of the viewing period, it is assumed that a new web page is immediately generated. Each page usually consists of a HTML file and a set of inline objects. A similar concept is defined in [26] as web-request. The parameters in this layer are:
  - The user *reading time* (usually named as page viewing time or HTTP OFF) is the period of time between two successive page downloads.
  - The *number of objects* in a web page. This parameter comprises the main HTML as well as the inline objects, and thus, provides a model for the web document structure.

- **Connection Level**: the main task at this level is to model the TCP connection activation and deactivation process. Most studies model this level by means of a birth–death
process and assume that each object is downloaded in a separate connection. The parameters of this level are:
- The connection interarrival time is the time between the beginning of two consecutive connections.
- The connection size is related to the downloaded amount of data, and reflects the objects size. Several models characterise the main and the inline objects size separately whereas others consider a single parameter for both sizes.
- The object request size represents the size of the HTTP request header sent from the browser to the server to download each object.

- Packet Level: the packet level is meant to reproduce traffic generated by a single connection, which is related to TCP dynamics. The behaviour of the TCP connections in the third level can be either statistically characterised to be incorporated to the model or simulated using a detailed model of TCP. In the first option, traffic bursts generated by TCP dynamics during the data transfer are characterised by means of several parameters like the packet burst size and packet burst interarrival time. However, it has been stated that TCP imposes a feedback between the network and the traffic sources that cannot be represented by the packet level models. Consequently, if a better accuracy is required or the interaction between the network and the traffic sources has to be considered, it is necessary to simulate the TCP behaviour. This is the approach selected in this paper.

It is important to note that this paper does not focus on proposing a new traffic model but rather on using a fairly realistic model in simulations to evaluate the impact of different parameters configurations for TCP under conditions as close to reality as possible. Thus, rather than proposing a completely new model, we use one similar to that proposed in [26]. In our case, all parameters have been extracted from traces captured both in an academic and a corporative network in 1999, assuming the HTTP version to be 1.0 without keep-alive. Table 1 shows the values of the parameters defined by the model for any given session.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Distribution</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session interarrival time</td>
<td>Exponential</td>
<td>Sets the traffic load. The value considered in our simulations is defined in Table 2.</td>
</tr>
<tr>
<td>Number of pages per session</td>
<td>Lognormal</td>
<td>μ = 14, σ = 36</td>
</tr>
<tr>
<td>Page viewing time</td>
<td>Gamma</td>
<td>μ = 46.8s, σ = 168.6s</td>
</tr>
<tr>
<td>Number of object per page</td>
<td>Lognormal</td>
<td>μ = 5.3s, σ = 12s</td>
</tr>
<tr>
<td>Connection interarrival time</td>
<td>Gamma</td>
<td>μ = 2.3s, σ = 4.5s</td>
</tr>
<tr>
<td>Connection size (downlink)</td>
<td>Pareto</td>
<td>μ = 5616 bytes, σ = 1.77</td>
</tr>
<tr>
<td>Object request size (uplink)</td>
<td>Lognormal</td>
<td>μ = 364 bytes, σ = 101 bytes</td>
</tr>
</tbody>
</table>

It has been considered that the session interarrival time follows an exponential distribution, as stated in [29]. Also, it can be noted that the main and in-line objects sizes have been characterised together. The first- and second-order statistical moments of the analysed parameters are very similar to those reported by other characterisations. Detected variations are not significant: they appear because both cached and not cached objects are treated equally in this model, whereas other models characterise them separately. As previously mentioned, it is important to note that packet level parameters have not been considered in our simulations. Instead, we have thoroughly simulated the TCP behaviour as proposed in [12].

2.4. TCP/IP model

In order to simulate the TCP connections behaviour, a complete TCP model has been included in our simulator. The TCP simulation model resembles the per packet behaviour of TCP full-duplex connections, and implements all the TCP basic mechanisms (including connection establishment and tear down, slow-start, congestion avoidance, fast retransmit and fast recovery [30]). Also, some of the most popular improvements [2,3,31,32] are included in the TCP model, so that different TCP flavours can be simulated. The operation of the different TCP mechanisms included in our simulation model have been validated against the one-way TCP models of NS2 by using time vs. sequence number diagrams [33].

Finally, it is necessary to note that we have not included a detailed model of IP because no routing aspects have been considered in our simulations. However, the IP header overhead is included in every transmitted packet.

3. A new methodology and graphical representation for TCP evaluation in a multiuser web environment

A key issue to evaluate TCP performance is to define a set of criteria so that tests can be performed in a clearly defined framework. These criteria must be clearly defined so that all experiments can be reproduced and compared under the same conditions. Popular criteria include the throughput, the efficiency and the goodput. The throughput is the ratio between the total amount of bytes generated by a connection-including TCP/IP protocol overhead and also retransmitted packets-and the total transmission time. It provides information about the amount of network resources used per time unit and can be either measured for a single connection or for all users. The efficiency is related to the effectiveness and usefulness of the resources used by the connection. It is equal to the ratio between the total amount of useful application data transferred through a connection and the total amount of bytes generated by that connection, including protocol overhead and retransmitted packets.
The goodput is equal to the effective application data transferred per time unit for a given connection and, thus, represents the performance of that connection from the user’s point of view. We have consequently chosen the goodput in our study to evaluate TCP on the user side. However, even though the goodput is immediate to use in a study when a single connection is involved, it is not obvious how to use this ratio in a web environment involving a very large number of them. Similarly, it is not either easy to provide an efficient graphic representation of this ratio in such conditions. The following subsections provide a new methodology to evaluate and graphically represent the performance of TCP in a web environment, respectively.

3.1. Goodput vs. connection size

When evaluating the performance of TCP, it is important to note that the goodput depends on the connection size because of the TCP dynamics and, particularly, because of the slow-start mechanism included in all TCP versions. Small sized connections are heavily affected by the slow-start mechanism because they are usually finished before further control can be applied, while larger connections are more likely affected by other control mechanisms (Section 2.4). Consequently, it is better to characterise a connection by means of a vector including both the goodput and the connection size. At this point, the obvious choice to represent goodput-connection size vectors would be a scatter plot, where goodput is represented on one axis, connection size is represented on the other, and thus the vector becomes a point in the \( \mathbb{R}^2 \) plane. However, since the size of a given connection may range from a few bytes to several megabytes, to achieve a better visualisation the goodput is represented against the common logarithm of the connection size rather than against connection size itself.

In each simulation run, a finite set of connections is generated and, consequently, a finite sets of vectors is returned. The traffic model in our work has been extracted from real traces and resembles the Long Range Dependence exhibited by the Internet traffic. Hence, a long period is simulated to take full advantage from the model statistical properties, and several millions of connections are generated in each simulation run. When so many connections are represented in a scatter plot, these plots become blurred and present no clear pattern. Thus, it would be preferable to represent the joint probability density function (joint pdf) of both magnitudes, which would represent the probability of occurrence of a given vector goodput-size \((g, s)\) in a simulation. In order to estimate the joint pdf using a finite series of vectors, the bidimensional histogram \( h_{GS} \) is used.

In a simulation, not all connection sizes and goodputs can be achieved. Thus, in order to compute \( h_{GS} \) in an efficient way, an interest window where most vectors are clustered is set. Our goodput range is defined by a minimum and maximum value \((G_{\text{min}} \text{ and } G_{\text{max}})\) and our connection sizes range from \( S_{\text{min}} \text{ to } S_{\text{max}} \). In all our experiments, \( G_{\text{min}} \) is equal to 0 and \( G_{\text{max}} \) is equal to 200 Kbps. Similarly, we have also set \( S_{\text{min}} \) and an \( S_{\text{max}} \) to 3.16 log(bytes) and 6.6 log(bytes), corresponding approximately to 1460 bytes and 4 Mb, respectively. Our interest window is consequently defined by \((0, 3.16), (2 \times 10^5, 3.16), (0, 6.6) \) and \( (2 \times 10^5, 6.6) \). The histogram is neither computed nor represented for values outside the window, but they are considered in order to properly compute the histogram and other functions derived from it.

After defining an interest window, \( h_{GS} \) is calculated by defining \( N_g \) and \( N_s \) equally spaced intervals in axes \( G \) and \( S \), respectively, and computing how many connections are inside each square interval, which we call rectangular bins. It must be noted that \( N_g \) is not necessarily equal to \( N_s \), because both magnitudes present different ranges. \( N_g \) and \( N_s \) must be carefully chosen: if they are too low, the resulting figures tend to be as noisy as a scattered plot but if they are too large, resolution is poor. In our work, \( N_g \) and \( N_s \) have been heuristically chosen after a large number of simulations.

After \( h_{GS} \) is obtained, it can be observed that since in our traffic model connection size follows a Pareto distribution, most connections present small or medium size and, consequently, are cluttered on the left side of the histogram. In order to improve the visibility of the representation, instead of the joint pdf, we use the conditional probability density function of the goodput given a connection size (conditional pdf). The conditional pdf is defined as the ratio between the joint pdf and \( f_S(s) \), the marginal probability density function of \( S \). Again, this function is estimated by means of a conditional histogram \( h_{GS} \) derived from the bidimensional histogram \( h_{GS} \). As shown in Eq. (1), \( h_{GS} \) can be computed by dividing the number of occurrences in a rectangular bin \((i, j)\), namely \( h_{GS}(i, j) \), by the total number of occurrences for a certain connection size range \([\beta_{j-1}, \beta_j]\), given by \( h_S(j) \) (Eq. (2)). It is important to note that \( h_S(j) \) must also include all occurrences out of the interest window to be complete. Occurrences out of the interest window are given by Eqs. (3) and (5), where \( P(G[k], S[k]) \) and \( P(G[k], S[k]) \) are defined in Eqs. (4) and (6).

\[
h_{GS}(i,j) = \frac{h_{GS}(i,j)}{h_S(j)} \quad (1)
\]

\[
h_S(j) = h_S^0(j) + h_S^1(j) + \sum_{i=1}^{N_G} h_{GS}(i,j) \quad (2)
\]

\[
h_S^0(j) + \sum_{k=1}^{K} \Pi_u^0(G[k], S[k]) \quad (3)
\]

\[
h_S^1(j) + \sum_{k=1}^{K} \Pi_u^1(G[k], S[k]) \quad (4)
\]

\[
h_S^0(j) = \left\{ \begin{array}{ll} 1 & \text{if } \beta_{j-1} \leq S[k] < \beta_j \text{ and } G_{\text{max}} \leq G[k] \\ 0 & \text{otherwise} \end{array} \right. \]

\[
h_S^1(j) = \left\{ \begin{array}{ll} 1 & \text{if } S[k] \leq \beta_{j-1}, G_{\text{max}} \leq G[k] \text{ and } \beta_{j-1} < S[k] \text{ or } G_{\text{max}} \leq G[k] \\ 0 & \text{otherwise} \end{array} \right. \]

\[
P_u^0(G[k], S[k]) = \left\{ \begin{array}{ll} 1 & \text{if } \beta_{j-1} \leq S[k] < \beta_j \text{ and } G_{\text{max}} \leq G[k] \\ 0 & \text{otherwise} \end{array} \right. \]

\[
P_u^1(G[k], S[k]) = \left\{ \begin{array}{ll} 1 & \text{if } S[k] \leq \beta_{j-1}, G_{\text{max}} \leq G[k] \text{ and } \beta_{j-1} < S[k] \text{ or } G_{\text{max}} \leq G[k] \\ 0 & \text{otherwise} \end{array} \right. \]
In this subsection, we address the graphical representation of $h_{GS}$ Traditionally, functions of two variables like $h_{GS}$ are either represented by 3D surfaces or bidimensional grayscale plots. 3D surfaces are quite useful in a computer, where the point of view can be changed at will. However, printed data is bound to rely on a planar view of a tridimensional graph and hence peaks can be overlapped and the representation may become confusing. In bidimensional grayscale plots, occurrence is given by a gray intensity: higher peaks are represented by dark colors and valleys are represented by bright ones. The main problem of such plots is that slight differences in grays cannot be perceived by the human eye. Thus, we propose a new representation for $h_{GS}$ that we call isopercentile plots. These plots represent several goodput percentiles for each connection size range: the $n$th goodput percentile conditioned to connection size is given by Eq. (7), where $F_{GS}(q, s)$ is the goodput probability distribution function conditioned to connection size, as defined in Eq. (8).

$$G_{s}^{\text{th}}(s) = \left\{ q|F_{GS}(q, s) = \frac{n}{100} \right\}$$ (7)

$$F_{GS}(q, s) = \int_{-\infty}^{q} f_{GS}(g, s)dg$$ (8)

The conditional probability distribution function is estimated by estimating the conditional histogram $h_{GS}$ using Eq. (9). Thus, the $n$th goodput percentile conditioned to connection size can be estimated according to Eqs. (10) and (11).

$$F_{GS}^{\ast}(r, j) = \frac{h_{GS}(j)}{h_{GS}(j)} + \sum_{i=1}^{\frac{r}{100}} h_{GS}(i, j)$$ (9)

$$G_{s}^{\ast}(j) = G_{\min} + (r^{\ast}(j) - 1)L_{G}$$ (10)

$$r^{\ast}(j) = \min \left( \left\{ r = 1...N_{G}|F_{GS}^{\ast}(r, j) \geq \frac{n}{100} \right\} \right)$$ (11)

Isopercentile plots are the level curves of $F_{GS}^{\ast}$ so not only they represent the probability distribution function but, also, the curves closeness provide information about the conditional probability density function: the closer the curves are for a given connection size, the more connections achieving a similar goodput for that size range and, consequently, the higher the conditional probability density function is. It is also useful to represent the mean goodput conditioned to connection size, which is defined by Eq. (12) and estimated according to Eqs. (13) and (14).

$$\bar{G}_{s}(g) = \int_{-\infty}^{+\infty} g f_{GS}(g, s)dg$$ (12)

$$\hat{G}_{s}(j) = \frac{\sum_{k=0}^{K} G(K)I_{[\beta_{j-1}, \beta_{j}]}(S[k])}{\sum_{k=0}^{K} I_{[\beta_{j-1}, \beta_{j}]}(S[k])}$$ (13)

$$I_{[\beta_{j-1}, \beta_{j}]}(S[k]) = \begin{cases} 1 \quad \text{if } \beta_{j-1} \leq S[k] < \beta_{j} \\ 0 \quad \text{Otherwise} \end{cases}$$ (14)

In order to represent the aforementioned goodput percentiles, we propose two different plots (Fig. 3). In our first plot (Fig. 3a), we present a set of goodput percentiles against connection size for a given TCP configuration. The number of represented percentiles affects the resolution of the plot, but should be limited so that the plot does not become blurred. Thus, a percentile-sampling interval of 10% has been heuristically chosen, and 10, 20, 30, 40, 50, 60, 70, 80, 90% percentiles are represented in the plot. The 5 and 95% percentile are also represented to improve the plot, but should be limited so that the plot does not become blurred. Thus, a percentile-sampling interval of 10% has been heuristically chosen, and 10, 20, 30, 40, 50, 60, 70, 80, 90% percentiles are represented in the plot.
the resolution at the edges of the plot. Finally, the goodput conditional mean against connection size for a given TCP version is also represented in this plot as a bold line. The statistical information in this plot is easy to read. For example, let \((g, s)\) be a point in Fig. 3a which belongs to the \(i\)th percentile line. This means that, for a connection size equal to \(s\), a \(n\%\) of the connections achieves a goodput equal or lower than \(g\), so that the percentile acts as a goodput frontier. Besides, we also know that, given a connection size \(s\), a \(n\%\) of the connections present a goodput lower than \(g\) and larger than \(g'\), where \(g\) and \(g'\) are given by percentiles \(i\)th and \((i+n)\)th for \(s\). Thus, when percentiles are close around a given goodput \(g\), it means that the goodput probability density function for \(g\) is high and, hence, that most connections approximately achieve \(g\).

The proposed isopercentile plot in Fig. 3a is useful to present the behaviour of a given TCP configuration for different connection sizes. However, if the goal is to compare different TCP versions or configurations in a simple way, there would be too many lines for correct visualisation if a single plot is used. Consequently, we use a second type of plot to compare different TCP versions and configurations. In this second representation, we use a plot like the one in Fig. 3b for each connection size range. Each line in these plots represents the goodput percentiles for different TCP configurations and versions using a polar diagram. The goodput percentiles for each TCP configuration or version are represented in the radii of the figures whereas different configurations or versions are represented in different angles. The closer a line in a plot is to the external circumference, the better the goodput achieved for that connection size. This representation allows an easy and intuitive performance analysis for each connection range. It must be noted that angle does not have a physical meaning in this type of representation. However, it has been observed that global shapes resulting in this type of representation allow a simple understanding of a simulation at a glance, as will be proven in the results section.

In brief, this subsection has proposed a new kind of representation to provide easy and intuitive visualisation of the goodput conditional probability density function that we propose to use in Section 4.3 for evaluating TCP performance in a web environment when a large number of connections is involved. Our representation includes three different kinds of plots that work together to provide all information required about the condition pdf in a simple way.

4. Evaluation of the impact of TCP configuration parameters

In order to test the validity of the proposed analysis and representation methodology, we have chosen a TCP evaluation scenario consisting of a World Wide Web application involving a huge number of connections. As an example, this section focuses on evaluating the impact of some relevant TCP configuration parameters on its performance in such a scenario. Particularly, the study is focused on two relevant parameters: the Initial Window Size and the Maximum Segment Size (MSS). Even though the main contribution of this paper is the aforementioned methodology, several interesting conclusions have been extracted from this example analysis in this section. It is necessary to note, though, that even though simulations have been designed to be representative, they are not totally exhaustive because the main goal of the paper is not this example analysis. A more thorough analysis is proposed as future work.

The evaluated TCP version is NewReno because it is known to be one of the most deployed TCP version at the present [9]. It is necessary to note, though, that the same study can be performed for any other TCP version using the proposed methodology. A similar study, for less realistic traffic model and a relatively short simulation run time (360 s), is presented in [5,34] studies the impact of the initial windows by means of test conducted in the real Internet, where the effect of the aggressiveness increase in other connections is difficult to evaluate. Also, an analytical study showing the effect of the initial congestion window on TCP performance for different connection sizes for a single TCP connection appears in [35]. Finally, [36] considers a simulation framework based on the SURGE web server traffic model that is somewhat similar to ours. However, it does not deal with a data volume as large as ours and their statistical analysis does not deal with the effect of parameters in different size connections. In our case, the used traffic model provides a correct proportion of long- and short-lived flows, similar to the one existing in the Web traffic. Thus, the impact of configuration parameters on the performance of different size flows and also on the interaction between them when they share the same link can be evaluated and results can be compared with previous work.

4.1. Evaluated parameters

An important feature of the Internet traffic is the heavy tailed distribution of the connection size. This means that there is a huge number of small size connections but also that a significant data volume is carried by a small number of large size connections, usually named as mice and elephants, respectively [37]. Short lived TCP flows end before reaching the congestion avoidance stage and, therefore, they are mainly affected by the slow-start mechanism. Since slow-start determines that the connection bandwidth is low at the beginning of the connection and is progressively increased to roughly fit all available bandwidth, short connections achieve a lower bandwidth than long lived flows. Thus, much effort has focused on boosting the slow-start stage. A first common approach relies on increasing the size of the initial congestion window (henceforth IW) in order to accelerate this stage. Also,
increasing IW may reduce the transmission time, because timeouts due to lack of ACKs at the beginning of the transfer are avoided and delayed ACK timeout are removed. However, it has been proven that a big initial congestion window may lead to a significant burstiness increase [5,38]. The upper bound for IW is usually considered to be roughly equal to 4 KB [38]. Although it has been suggested that this size should not be larger than one or two segments [30], a recent revision [39] removed the limits in [30] and set them back to those in [38]. It can be observed that the initial window (IW) is a key configuration parameter and, hence, its impact is going to be evaluated in this section.

A second important TCP configuration parameter, which affects both short and long lived flows, is the MSS. It has been stated that the most commonly used MSS in Internet is 1460 bytes, which corresponds to the 1500 bytes size limit of the Ethernet frames [11]. However, the default MSS specified by earliest TCP recommendation for connections flowing through the Internet is 536 bytes [30,40]. Naturally, when there is enough bandwidth available, a larger MSS allows a faster growth of the congestion window not limited to the initial slow-start phase, thus improving the performance of TCP connections. However, a bigger MSS increases the aggressiveness of the TCP flow, and hence, under congestion conditions a smaller MSS could be a better choice.

Given the importance of both IW and MSS and the different discussions about their limits, in this section the most common configurations will be compared for WWW traffic using the methodology proposed in Section 3.

4.2. Scenario description

In the example study in this section, two types of scenarios have been defined: homogeneous and heterogeneous ones. In homogeneous scenarios, all connections share the same IW and MSS, whereas in heterogeneous scenarios connections presenting different configuration parameters coexist in the same proportion. This evaluation is performed under different congestion conditions and, therefore, different loss patterns have been regarded in our simulations.

As commented in Section 2, our scenarios basically consist of a set of users accessing HTTP servers through an access network and Internet (Fig. 1). All our simulations in this section lasted 500,000 s. In each simulation run approximately 60,000,000 TCP connections were generated according to the Web traffic model described in Section 2.3. During a whole simulation, several TCP connections coexist simultaneously in the access link and compete for its bandwidth. The average number of TCP connection simultaneously sharing the link along the simulation run is about 100.

There are three sets of parameters in the presented simulations: TCP parameters, network parameters and traffic parameters. In order to evaluate different IW and MSS configuration under the same conditions, the remaining TCP parameters are kept constant. These parameters, which are presented in Table 2, have been chosen because they either correspond to typical existing configurations or recommendations [41]. It is important to note that the TCP version used is NewReno including the careful version of the Avoiding Multiple Fast Retransmit mechanism, as recommended in [31]. As aforementioned, NewReno is selected because it is one of the most deployed TCP version at present [9]. Our NewReno TCP implementation have been validated by means of conformance tests like the ones in [33]. Regarding the network parameters, the server rate has been fixed to 1.25 Mbps and the Internet latency has been fixed to 50 ms. Finally, the session interarrival time in the traffic parameters has been set to 0.5 s. Thus, the global traffic load is approximately equal to 1 Mbps.

4.3. Simulation and results

In the analysis in this subsection, three different scenarios, described in Table 3, have been used. In all three scenarios, simulations have been conducted for two different loss patterns: access network queues over flow losses and Internet independent random losses. In the first case, losses occur when the access network queues over flow and, therefore, these losses depend on the traffic in the network. All simulations using this loss pattern have been performed for a queue buffer size equal to 10 KB. This queue buffer size is quite extreme and it is intended to resemble a congestion situation in the access network. It must be noted that congestion could also occur for a larger buffer size if the average traffic generated by the sources practically fills all available bandwidth. The loss pattern experienced in this case could be very different from the loss pattern simulated in this scenario, and should be studied in a paper completely focused on evaluating the impact of configuration parameters on TCP.
performance rather than on a methodology of evaluation. This is meant to be done in future work.

In the second case, losses are random and do not depend on the traffic load, because they are supposed to be caused by a huge number of traffic sources coexisting in a very high bandwidth network (the Internet backbone). The Internet loss rate is equal to 1%, which is a reasonable value for an Internet non-excessively congested path.

### 4.3.1. Homogeneous scenario (scenario 1)

This scenario includes the simulations where all coexisting connections in the access link share the same MSS and IW values. Six simulations have been performed in this scenario for each type of losses (Table 3). The results of these simulations can be observed in Figs. 4 and 5, corresponding to queue overflow losses and independent losses, respectively. Each plot in these Figures presents the TCP performance for a given connection size range. Dashed lines in these plots represent several goodput percentiles (5, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 95%) for different TCP configuration parameters using the proposed polar diagrams. The goodput percentiles for the different simulated MSS and IW are represented in each one of the six radii of the figures. The goodput conditional mean is represented as a thicker line. This representation allows an easy and intuitive analysis for each connection range. The closer a line is to the external circumference in a figure, the better the goodput achieved by that connection size. It must be noted, though, that these plots present different scales, so they must be compared with care. Also, plots yielding close lines correspond to configurations where all connections receive a similar treatment, whereas in plots with separate lines some connections are well over the mean while others are clearly below.

It can be observed in Fig. 4 that the best results for connections smaller than 39 KB (Fig. 4a–e) are achieved for an MSS equal to 1460 bytes and IW equal to 4 MSS. However, for larger connection sizes (Fig. 4i and j) the best results are obtained for MSS=536 bytes and IW equal to 1 or 2 MSS. This happens because all connections share the same medium and present the same MSS and IW and, hence, connections presenting an aggressive slow-start with a large MSS and IW are bound to generate losses in the network at medium term. These losses appear because an aggressive slow-start cannot fine tune the connection throughput to the available bandwidth and they may provoke a reduction of the transmission window and, consequently, a bandwidth loss for all connections coexisting in the link. Small connections are either finished before they are affected by these losses or, anyway, less affected by them. In addition, they take advantage of the faster slow-start stage. On the other hand, the benefits of a faster initial slow-start stage for long-lived flows are not significant, while large connections are very affected by losses during their life span. Consequently, even though it is better to have an aggressive slow-start for an isolated connection, large connections behave better for more conservative MSS and IW when they share the medium with many other connections. This behaviour can also be appreciated in the percentage of connections affected by losses due to queue overflow (Table 4), which is larger when the configuration parameters are more aggressive. The same effect can be also noted in the percentage of lost packets, also shown in Table 4. As observed in (Fig. 4f–h), the best for medium size connections is an intermediate configuration, either.

### Table 3

Simulation scenarios

<table>
<thead>
<tr>
<th>Scenario 1 (Losses: queue (10 KB)/independent)</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW = 1</td>
<td>IW = 1</td>
<td>IW = 4</td>
<td>IW = 1</td>
<td>IW = 2</td>
<td>IW = 4</td>
<td></td>
</tr>
<tr>
<td>MSS = 536</td>
<td>MSS = 536</td>
<td>MSS = 536</td>
<td>MSS = 1460</td>
<td>MSS = 1460</td>
<td>MSS = 1460</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario 2 (Losses: queue (10 KB)/independent)</th>
<th>S1</th>
<th>S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>33.3% IW = 1</td>
<td>33.3% IW = 1</td>
<td>33.3% IW = 2</td>
</tr>
<tr>
<td>33.3% IW = 4</td>
<td>33.3% IW = 4</td>
<td>33.3% IW = 4</td>
</tr>
<tr>
<td>MSS = 536</td>
<td>MSS = 1460</td>
<td>MSS = 1460</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scenario 3 (Losses: queue (10 KB)/independent)</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW = 1</td>
<td>IW = 2</td>
<td>IW = 4</td>
<td></td>
</tr>
<tr>
<td>50% MSS = 536</td>
<td>50% MSS = 536</td>
<td>50% MSS = 536</td>
<td></td>
</tr>
<tr>
<td>50% MSS = 1460</td>
<td>50% MSS = 1460</td>
<td>50% MSS = 1460</td>
<td></td>
</tr>
</tbody>
</table>

Regarding the relative positions of the percentiles in the plots, several conclusions can be extracted. First, it can be noted that lower percentiles are in general worse for more aggressive configurations even when the connection size is small, meaning that in this cases, the goodput achieved by the slowest connections is the worst. Also, the variability of the goodput is higher for these configurations: while most short-lived flows benefit from a more aggressive behaviour, some of them are badly affected. It can easily be observed (Fig. 4 j–l) that the connections presenting less variability are large ones with non-aggressive parameters. It is also interesting to note how connections behave with respect to the goodput conditional mean for each configuration (thick black line). It can be observed that in medium and large connections, approximately half the percentiles are above the mean and half are below in most cases. However, aggressive configurations in very small connections (Fig. 4a) make most connections go near or over the mean while only a 35% are below that mean. If the connection size is a bit larger, though (Fig. 4b), the situation is different: less than a 40% of the connections are above the mean. Also, in this second case the fastest connections receive a far better service that the rest. Despite all these facts, goodput is better for most small connections using aggressive...
configurations, but the safest configurations to achieve a fairer bandwidth sharing are clearly the non-aggressive ones, though even in these cases small connections present some variability. These conclusions can be further supported by handling two different averages of the connection goodput. The mean goodput $G$ is the arithmetic mean of the goodput of all the simulated connections (Eq. (15)). Since there are a huge number of small size connections, this metric hardly reflects the performance of big size connections. However, since large connections represent a significant amount of traffic, the weighted

<table>
<thead>
<tr>
<th>MSS (bytes)</th>
<th>1460</th>
<th>536</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW (MSSs)</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>% of affected connections</td>
<td>16.79</td>
<td>19.5</td>
</tr>
<tr>
<td>% of lost packets</td>
<td>4.03</td>
<td>4.65</td>
</tr>
</tbody>
</table>
average goodput $A_G$ is also computed (Eq. (16)). This metric is defined in Eq. (16) where $T(k)$ is the transmission time of connection $k$. Table 5 shows the values of both metrics for queue overflow losses in this scenario. According to these metrics, a moderated configuration would be the best choice in this scenario in average.

$$G_k = k$$

$$A_G = \frac{\sum_{k=1}^{K} S(k) \cdot T(k)}{\sum_{k=1}^{K} T(k)}$$  \hspace{1cm} (16)

Fig. 5 shows the same simulations for independent losses. It can be observed at a glance that results for small connections are very similar to those in Fig. 4. Basically, small connections are less affected by losses than large ones, especially for aggressive configuration parameters and hence results tend to be the same. It is remarkable, though, that in the worst case the goodput achieved by small connection for random losses is better than for queue overflow losses. This happens because aggressive configurations provoke congestion and, consequently, queue overflow losses that affect not only medium and large connections but also some small ones. These affected small connections present a very low goodput. Random losses, on the other hand, do not depend on the chosen configuration parameters. Thus, all connections are more or less affected depending on their size and, consequently, no particularly affected small connection appears in these plots. It can also be easily observed that medium size connections behave similarly but in this case an aggressive configuration is usually better: since losses are not related to congestion in this case, it is better for a connection to have a life span as short as possible to achieve the best goodput. The most significant change can be appreciated for large connections. First, they achieve a far better goodput than in the previous case because they are not so damaged by small connections when an aggressive configuration is set. It can also be noted that for large connections the best goodput is related to a large packet size, basically because slow-start is faster for large packets and, thus, they achieve a better loss recovery.

In general, it can be observed that results are better for MSS equal to 1460 and IW equal to 4 despite the connection size. This happens because, as aforementioned, independent

| Table 5 | Connection goodput mean and weighted average for different TCP configuration simulations for scenario 1 and queue overflow losses |
|--------------------------------|-----------------|-----------------|
| MSS (bytes) | 1460 | 536 |
| IW (MSSs) | 1 | 2 | 4 | 1 | 2 | 4 |
| Goodput mean (kBps) | 4.3052 | 6.1262 | 5.9446 | 3.9436 | 5.5162 | 6.2137 |
| Weighted average goodput (kBps) | 5.3756 | 6.1467 | 5.1714 | 5.6203 | 7.6628 | 7.7286 |

| Table 6 | Connection goodput mean and weighted average for different TCP configuration simulations for scenario 1 and Internet random losses |
|--------------------------------|-----------------|-----------------|
| MSS (bytes) | 1460 | 536 |
| IW (MSSs) | 1 | 2 | 4 | 1 | 2 | 4 |
| Goodput mean (kBps) | 5.0487 | 7.2548 | 8.4607 | 3.9519 | 5.4345 | 6.6544 |
| Weighted average goodput (kBps) | 6.1426 | 7.7530 | 8.4265 | 5.0694 | 6.6861 | 7.7693 |

losses are not related to the connection size and, consequently, they equally affect large and small connections. Thus, aggressive configurations are better for any connection because they can occupy as much bandwidth as possible despite the behaviour of the rest of the connections in the medium. This same effect is reflected in Table 6, where the previously defined mean goodput and weighted average goodput are shown.

4.3.2. Heterogeneous scenario in IW (scenario 2)

Thus far, we have analysed scenarios where all connections present the same configuration parameters, but this is not the usual case. Thus, in this scenario all coexisting connections in the access link have the same MSS, but their IW may be equal to 1, 2 or 4 MSS. In this subsection, connections presenting IW equal to 1, 2 or 4 appear in the same proportion. The results of simulations corresponding to queue overflow losses can be observed in Fig. 6. Whereas in the previous scenario each plot corresponded to six simulations, in this case it corresponds to only two (Table 3): MSS equal to 1460 and 536 bytes. The goal is to observe how connections sharing the access link are affected by the different values of IW.

The plots in this new scenario are divided into two halves: the left one corresponds to MSS equal to 1460, whereas the right one correspond to MSS equal to 536. Thus, it can be considered for a first fast evaluation that the right side of the plots correspond to more aggressive configurations. Thus, globally regarding MSS, plots shifted to the left correspond to connection sizes, which take advantage of more aggressive configurations: in this scenario, the smaller ones. However, this scenario is designed to evaluate the impact of IW and this is done within each side of the plot. A first conclusion when evaluating this impact is that small connections are quite affected by variations of IW but large connections are practically unaffected by these variations. This was expected because IW only affects the initial slow-start stage for all connections, which is not very significant in the life span of a large connection.

It can be noted in Fig. 6 that, as in the previous scenario, the most aggressive configurations benefit small connections with respect to the larger ones. Also, it can be appreciated that the best results for all connection sizes are obtained for IW equal to 4, both for MSS equal to 1460 and
536 bytes. This happens because in the previous scenario all connections had the same IW and, hence, it was better for large connections that all connections had a small IW. In this case, even if a connection presents a small IW, it cannot force the rest to use the same parameters. Hence, it is better for all connections to be aggressive. This can be also observed in Table 7, where the percentage of connections affected by losses for both MSS is lower for IW equal to 4 and larger for IW equal to 1 with respect to percentages calculated at scenario 1 (Table 4). It is interesting to note that the percentage of affected connections for MSS equal to 536 does not depend as much on IW as for MSS equal to 1460.

Table 7
Percentage of connections affected by queue losses and percentage of lost packets in scenario 2

<table>
<thead>
<tr>
<th>MSS (bytes)</th>
<th>1460</th>
<th>536</th>
</tr>
</thead>
<tbody>
<tr>
<td>IW (MSSs)</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>% of affected connections</td>
<td>19.22</td>
<td>20.69</td>
</tr>
<tr>
<td>% of lost packets</td>
<td>5.9</td>
<td>2.39</td>
</tr>
</tbody>
</table>

Fig. 6. Goodput percentiles for different connection size and TCP configuration. Results correspond to scenario 2 in presence of queue overflow losses.
1460. This fact basically indicates that this percentage grows with the aggressiveness of the configuration in a non-linear way. Table 7 also shows the percentage of packets dropped in the network queues for each simulation performed in this scenario.

Results for independent losses are not shown for this scenario because they only present slight differences with the results included in the homogeneous scenario. When no congestion exists in the shared link, connections with a cautious configuration are not damaged by the presence of connections with a more aggressive configuration. As in the previous case, the most aggressive configuration achieves the best performance: IW equal to 4 MSS, for MSS equal to 1460 and 536 bytes.

4.3.3. Heterogeneous scenario in MSS (scenario 3)

This last scenario is used to evaluate cases where all coexisting connections in the access link have the same IW. However, their MSS may be equal to 1460 or 536 bytes. In this scenario, the same proportion of connections presenting different MSS has been used. The results of these simulations for queue loss pattern can be observed in Fig. 7. In this case, each figure corresponds to three different simulations (Table 3): IW equal to 1, 2 and 4 MSS. The goal
of the simulations is to observe how connections sharing the access link are affected by the different values of MSS.

The simulations here are arranged over the plot in the same order than in the previous scenarios for the sake of comparison among all simulations. This means that in this case a given simulation is printed over a single axis of the plot where connections presenting MSS of 1460 and 536 are confronted. Thus, if there are symmetries along a given configuration axis in a plot, it means that MSS does not affect much the goodput of the connections for the size in such a plot. It can be clearly appreciated that the less symmetric plots are related to small connection sizes, which in this case also achieve a better goodput for more aggressive configuration parameters. Even though this is the case for all connection sizes in this scenario, it can be noted that the effect is less remarkable for larger connections. It can be globally observed that, while in the previous scenario IW had not much influence on the performance of large connection, in this scenario MSS has an important impact on such a performance. Also, in the previous scenario a moderated configuration was the best choice for large connections. In this scenario, though, it is better to use MSS configurations as aggressive as possible even for large connections because, since there are connections yielding an MSS equal to 1460 in the access link anyway, any given connection benefits from a large MSS.

In Fig. 7, presenting simulations with queue overflow losses, the best results are achieved for MSS equal to 1460 in all cases. With regard to IW, small connections prefer a large IW, whereas large connections, which are not significantly affected by IW, are severely affected by small ones if IW is large. This behaviour is also reflected in Table 8, where the percentage of connections with losses grows for MSS equal to 536 and decreases for MSS equal to 1460 with respect to results in our homogeneous scenario. The percentage of packets dropped in the network queues for each simulation performed in this scenario is also presented in this table.

Simulations with independent losses present practically the same results that in previous scenarios and, therefore, they are not included in the paper. Again in this case, the more aggressive the configuration parameters, the better the achieved goodput.

Table 8  
Percentage of connections affected by queue losses and percentage of lost packets in scenario 3

<table>
<thead>
<tr>
<th>IW (MSSs)</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSS (bytes)</td>
<td>1460</td>
<td>536</td>
<td>1460</td>
</tr>
<tr>
<td>% of affected connections</td>
<td>15.33</td>
<td>13.81</td>
<td>17.15</td>
</tr>
<tr>
<td>% of lost packets</td>
<td>2.79</td>
<td>3.05</td>
<td>5.61</td>
</tr>
</tbody>
</table>

5. Conclusions and future work

In this paper we have presented a new methodology and representation technique to analyse the performance of TCP in scenarios involving a huge number of connections. We propose to statistically evaluate TCP in terms of the goodput achieved by the different sized connections in the access link because it is one of the factors having a major impact on how users perceive a service. Regarding representation, we propose two different types of plots for a simple and intuitive visual evaluation of TCP versions and configurations under different circumstances. These plots are adequate to evaluate a single TCP version or configuration and to compare the performance of several ones absolutely or with respect to a baseline version.

To test the suitability of our proposal, a fairly realistic framework has been developed to simulate a world wide web scenario. This framework includes a traffic model extracted from real traces where a large number of simultaneous connections are allowed. Even though more realistic frameworks can be developed, this one is computationally feasible and, consequently, useful for simulations. This framework supports most TCP versions and enhancements. However, to test our analysis and representation methodology, we chose to evaluate the behaviour of NewReno TCP because it is, most likely, the most deployed TCP version in the web. Specifically, we tested its performance with respect to two major configuration parameters, IW and MSS. It is important to note, though, that this has been simply done to test the methodology and representation technique that we propose and that any other analysis could have been performed using the same tools.

All simulations were performed in three different scenarios where: (i) all connections shared the same IW and MSS; (ii) connections shared only the same IW; and (iii) connections shared only the same MSS. In these scenarios, both simulations with independent random Internet losses and queue overflow losses were performed. In the first case, results were practically the same in all scenarios.

In homogeneous scenarios, it has been observed that aggressive configurations yielding large MSS and IW values in congested links benefit small connections. Under these conditions, small connections steal bandwidth from large connections. If it is taken into account that: (i) small connections are more frequent in web traffic; (ii) large connections move a heavier data load and, hence, use the link a longer time; and (iii) large connections, because of the nature of TCP, achieve a larger goodput, it seems natural to use aggressive configuration parameters to benefit small connections. However, it is important to note that even though there are less large connections, they represent a huge data volume of the link. Consequently, since an aggressive policy damages large connections making them last longer, an user is bound to perceive them as slow and, consequently, its opinion about the quality of the link may
be harmed. In non-congested scenarios, though, the most aggressive configurations are always the best option. In heterogeneous scenarios, it has been observed that aggressive configurations are once more enhanced with respect to those yielding more conservative parameters. Thus, in these cases, a greedy behaviour is always the best option for any connection despite its size.

If both homogeneous and heterogeneous scenarios are regarded and it is taken into account that the most usual MSS in Internet is 1460, it seems to be recommendable to use this MSS and not 536 for all connections. However, it would be advisable to use a moderate value for IW. IW equal to 2 is a good value because it enhances small connections but does not harm much large connections. Thus, the global performance of the link is good from the user point of view. This combination of parameters (MSS = 1460, IW = 2) is reasonable both for congested and non-congested links. All in all, it can be stated that this analysis corroborates previous assumptions about the behaviour of TCP configurations for web traffic that were predicted from studies performed for only a few connections.

In brief, we have tested that our methodology and representation technique is a valuable tool to evaluate the behaviour of TCP configurations and versions for web traffic. Therefore, future work will focus on extending the present study to others configuration parameters and to test recently proposed TCP improvements such as DSACK [42] and EIFEL [43]. Also, even though closer to reality than previously existing evaluation scenarios, our framework can be further enhanced. Particularly, more complex and accurate models of the Internet behaviour (i.e. Internet delay and loss models) can be included, and also different network topologies may be considered.

Acknowledgements

This paper have been partially supported by the Spanish Ministry of Science and Technology under project TIC2003-07953-C02-01 (AIRES).

References


