Computer Communications 32 (2009) 1460–1484

Contents lists available at ScienceDirect

Computer Communications

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

A comprehensive TCP fairness analysis in high speed networks

Sándor Molnár, Balázs Sonkoly *, Tuan Anh Trinh

High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar tudósok körútja 2, H–1117 Budapest, Hungary

ARTICLE INFO

Article history:
Received 11 August 2008
Received in revised form 30 April 2009
Accepted 5 May 2009
Available online 13 May 2009

Keywords:
Congestion control
High speed networks
High speed TCP variants
Fairness analysis

ABSTRACT

The short-term dynamics of competing high speed TCP flows can have strong impacts on their long-term fairness. This leads to severe problems for both the co-existence and the deployment feasibility of different proposals for the next generation networks. However, to our best knowledge, no root-cause analysis of this observation is available. This is the major motivation of our work.

The contribution of the paper is twofold. First, we present our comprehensive performance evaluation results of both inter- and intra-protocol fairness behavior of different TCP versions to get an overall view of these protocols. The analysis has revealed not only the equilibrium behavior but also the transient characteristics besides the dynamic behavior. Second, we show the results of a root-cause analysis to get a deeper understanding in the case of some promising TCP versions. This study does not only fill the “black holes”, i.e. answers the questions which remained unanswered in some cases, but rather goes deeper and investigates questions which have never been asked yet. The work includes flow-level, packet-level, queueing and also spectral analyses. Three loss-based (HighSpeed TCP, Scalable TCP and BIC TCP) proposals and the delay-based FAST TCP are investigated in details with both “dumb-bell” and “parking-lot” topologies.

1. Introduction

TCP congestion control had managed to maintain the stability of the Internet in the past decades but it has reached its limitations of the challenges of present networks. However, the congestion control of TCP has been continuously developed from the first version which dates back to 1981 (RFC 793). The basic mechanism was incrementally developed and tuned introducing new additional algorithms, e.g., slow start, congestion avoidance, RTO calculation and delayed ACK in 1989 (RFC 1122), SACK in 1996 (RFC 2018) and NewReno in 2004 (RFC 3782) just to mention a few. The challenges of next generation networks (e.g., high speed communication or the communication over different media) generated an urgent need to further develop the congestion control of the Internet.

The research community has responded with many proposals developed in several excellent research groups all over the world. This significant research work, especially in the previous years, resulted in a huge number of new ideas. These ideas have been developed and implemented in new TCP version proposals. These proposals can efficiently address different aspects of the challenging future network environments and applications but it seems that the process towards finding the transport protocol of the future Internet has been slowed down. One of the main problems is that – in spite of the fact that many proposals have been published – finding the best one is difficult since these works are mostly independent research studies and do not compare their results to each others'. Or even if they do comparisons, usually there is no common performance evaluation framework (same set of metrics and topologies, etc.) where a comprehensive performance evaluation of the new versions can be made. Such a comprehensive study is very rear, however, without such studies finding the next step in the jungle of the proposed TCP versions is impossible. This is the need which motivated our work.

The goal of the paper is twofold. First, we have decided to make a comprehensive performance evaluation of the most promising proposals. Some of our results are already known, since when these proposals were suggested they were investigated too. In these cases our results can confirm or question already published results. However, we also aim at answering many questions which remained unanswered in the literature and we think that they are important to understand the mechanism and impacts of these new algorithms. We call this study as *horizontal research* because the results of this work give an overall picture about all the investigated TCP proposals where the still remained “black holes” are filled. The main question we have in the focus of the analysis is the fairness characteristics of the different TCP versions including both inter- and intra-protocol behavior with different topologies

* Corresponding author. Tel.: +36 14633884.
E-mail addresses: molnar@tmit.bme.hu (S. Molnár), sonkoly@tmit.bme.hu (B. Sonkoly), trinh@tmit.bme.hu (T.A. Trinh).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.05.003

and parameter settings. More specifically, we investigate the behavior in simple "dumb-bell" topology with one congested link and in more realistic topologies with multiple congested links and different round-trip times ("parking-lot" topologies). The interaction of flows using the same or different protocols is also analyzed (intra- and inter-protocol behavior). In order to get general statements, we investigate the interoperation of different congestion control approaches instead of carrying out the analysis for each combination of the protocols. We define the following groups of congestion control principles. The first group contains the loss-based protocols. This group can be further divided into the group of MIMD (Multiplicative Increase Multiplicative Decrease) mechanisms and the group of AIMD-like (Additive Increase Multiplicative Decrease) algorithms [1]. The second group corresponds to the delay-based protocols. We examine not only the interaction of single flows, but the analysis is extended to the competition of later entering individual flow against different traffic aggregates. In spite of the practical importance of these scenarios, we hardly find similar evaluations in the literature. On the one hand, we analyze the long-term behavior of interacting flows regarding fairness (fair/unfair) and stability (long-term oscillation/stable behavior) properties. In certain cases, the type of unfairness can also be important. More exactly, a TCP flow can force other flows to deviate from their normal operation and, in extreme cases this yields TCP Reno-like operation mode and starving of certain flows. On the other hand, we put emphasis not only on the equilibrium behavior but also on the transient characteristics with the dynamic behavior and investigate how the transient state can influence on the long-term fairness performance. We have recognized that this problem has been neglected by the research community but we show that the transient parameters can have significant impact on the equilibrium fairness results. As part of this transient analysis, we deeply analyze the impacts of starting time on the performance and fairness (delay-sensitivity, responsiveness, etc.). We also introduce a new fairness metric to describe the transient state behavior. The main message of this research shows that the dynamic characteristics should not be neglected in a comprehensive fairness analysis. This overall view is vital when making decisions regarding these proposals.

Second, we have decided to make a root-cause analysis to get a deeper understanding in the case of some promising TCP versions. This study not only fills the "black holes", the questions which remained unanswered in some cases, but rather goes deeper and investigates questions which have never been asked yet. We call this study *vertical research* because in this part we suggest novel methods – such as spectral analysis and identification of main operating frequencies – and get new results which help to provide an in-depth understanding and explanations of the investigated algorithms. For this purpose we have introduced a spectral analysis for the first time as far as we know to analyze and understand fairness performance. This tool can provide deeper understanding of interaction of different congestion control mechanisms, especially in scenarios with multiple bottleneck links where the macroscopic behavior is composed of several operating frequencies that can be propagated along the network routes (see Section 7.2.4). With this spectral analysis together with flow-level, packet-level and queueing analysis we have got a good understanding of the investigated phenomena.

In this paper, we summarize research results from a comprehensive performance evaluation study addressing the above listed research goals. This work has resulted in a huge amount of results which are impossible to include in a paper. Therefore, we have collected all of the results in a technical report [2] available from http://www.hsnlab.tmit.bme.hu/projects/tcp/. This paper is the extensive summary of that report. Partial results from our previous research can be found in [3,4].

It is certainly desirable to evaluate high speed TCP variants by measurements in real networks and there are already substantial works in this respect [5–10]. However, network parameters, especially in changing conditions, can be hardly controlled. In an end-to-end path, multiple network elements combined make the estimation of these parameters even harder. Furthermore, the uncertainty and/or inaccuracy of network parameters can have a significant impact on the outcome of the results. In our analysis, we need full control of network parameters, especially the buffer size, the bottleneck links to understand the fairness of competing flows. This justifies our choice of using simulations for this particular fairness analysis.

1.1. Related work: TCP versions

The new challenges of TCP have been addressed by several research groups in the last decade and, as a result, a number of new TCP versions have been developed. The research community has been working intensively to analyze these versions, see e.g., [11]. An overview of the most important proposals is given here, and some technical details of the TCP variants analyzed in the paper are summarized in Table 1. HighSpeed TCP (HSTCP) [12] is a modification to TCP's congestion control mechanism for use with TCP connections with large congestion windows. It changes the TCP response function to achieve better performance on high capacity links. HSTCP is based on an AIMD mechanism where the increase and decrease parameters ($a(W)$ and $b(W)$) are functions of the current value of the congestion window (see the corresponding row of Table 1) yielding an adaptive and more or less scalable algorithm. Ideas to introduce MIMD mechanisms for TCP have also been considered. Scalable TCP (STCP) [13] is a good example which has been suggested as an efficient transport protocol for high speed networks. Here, the multiplicative increase and multiplicative decrease algorithm guarantees the scalability of the protocol. The congestion window is increased by a constant parameter (a) as a response to a received acknowledgement, while it is reduced in a multiplicative manner (by bW) in case of packet losses (see Table 1). In order to solve the TCP's severe RTT (round-trip time) unfairness problems, BIC TCP has been developed [14]. BIC TCP combines two schemes called additive increase and binary search. When the BIC TCP source gets a packet loss event, the congestion window is reduced by a multiplicative factor (β); and the maximum window parameter (W_{max}) is set to the value of the congestion window just before the reduction while the minimum window parameter (W_{min}) is set to the current value. Then the protocol performs a binary search between these parameters by jumping to the "midpoint" between the bounds. (More exactly, this jump is based on the B parameter of the protocol.) If packet loss does not occur at the updated window size, that window size becomes the new minimum; if packet loss occurs, that window size becomes the new maximum. An important restriction is also introduced, the growing cannot be more aggressive than a linear one with a constant parameter (S_{max}). This process continues until the window increment is less than a small constant (S_{min}), when the window is settles down around W_{max} (increasing slowly on a "plateau"). This mechanism yields an "AIMD-like" behavior where the growing function is most likely composed of a linear phase (additive increase) and a logarithmic one (binary search). When the updated window size exceeds the current maximum, then a new equilibrium state has to be found and BIC TCP enters into the max probing state. During this phase, the growing function is the inverse of the previous ones, more exactly, the window is increased exponentially first (which is very slow at the beginning) and then linearly. This complex mechanism is also summarized in the corresponding row of Table 1. The good performance of the protocol, including good utilization, linear RTT fairness (RTT unfairness is proportional

Table 1
Details of TCP variants analyzed in the paper.

Protocol	Window adjustment	(When)	Reaction to loss
HSTCP	$\mathtt{w} \leftarrow \mathtt{w} + \frac{a(\mathtt{w})}{\mathtt{w}}$	(per-ACK)	$\mathtt{w} \leftarrow \mathtt{w} - b(\mathtt{w})\mathtt{w}$
STCP	$\mathtt{w} \leftarrow \mathtt{w} + a$	(per-ACK)	$\mathtt{w} \leftarrow \mathtt{w} - b\mathtt{w}$
BIC TCP	$\mathtt{w} \leftarrow \mathtt{w} + \frac{a}{\mathtt{w}},\ a \in \{S_{min}, \frac{W_{max}-\mathtt{w}}{B}, \frac{\mathtt{w}-W_{max}}{B-1}, S_{max}\}$	(per-ACK)	$\mathtt{w} \leftarrow \beta\mathtt{w}$
FAST TCP	$\mathtt{w} \leftarrow \min\{2\mathtt{w},\ (1-\gamma)\mathtt{w} + \gamma(\frac{\mathtt{baseRTT}}{\mathtt{RTT}}\mathtt{w} + \alpha)\}$	(periodically)	$\mathtt{w} \leftarrow 0.5\mathtt{w}$

to the RTT ratio as in AIMD), good scalability, and TCP-friendliness, comes from the slow increase around W_{max} and the aggressive linear increase of additive increase and max probing phases. Further research with BIC TCP has been resulted in CUBIC. CUBIC [15] is an enhanced version of BIC TCP. It simplifies the BIC window control and improves its TCP-friendliness and RTT-fairness. TCP Libra [16] is another solution to provide RTT-fairness while maintaining a good friendliness with TCP NewReno. The research to make further improvements yielded to the development of H-TCP [17]. The authors of H-TCP promise that the asymmetry due to the modification in HSTCP and also in STCP can be eliminated with their method. The idea of incorporating accurate bandwidth estimations into the TCP congestion control has also opened a new path in TCP research. TCP-Westwood [18] is a prominent example where eligible rate estimation methods to intelligently set the congestion window and slow-start threshold have been introduced. An interesting solution has been developed in LTCP [19]. LTCP has a two dimensional congestion control. The macroscopic control uses the concept of layering to quickly and efficiently make use of the available bandwidth, whereas microscopic control extends the existing AIMD algorithms of TCP to determine the per-ACK behavior.

The research on the delay-based ideas has resulted in FAST TCP [8,9]. FAST TCP has the same equilibrium properties as TCP Vegas [20] but it can also achieve weighted proportional fairness. FAST TCP seeks to restrict the number of its packets queued through the network path between an upper (β) and a lower (α) bound, however, the behavior is usually controlled by a single parameter (α) that can be considered as the targeted backlog (packets in the buffers) along the flow's path [8,9,21]. Under normal network conditions, FAST TCP periodically updates its congestion window based on the comparison between the measured average RTT and the estimated round-trip propagation delay (when there is no queueing). More exactly, the window is adjusted according to the formula presented in Table 1, where γ is the step size affecting the responsiveness of the protocol, and `baseRTT` is the minimum RTT observed so far which is an estimation of the round-trip propagation delay. The parameter α controls the equilibrium behavior, therefore the appropriate setting of this parameter is crucial (see [8,9,21] and Section 7.2.5). FAST TCP also reacts to packet losses halving its congestion window. The delay-based control also appears in other proposals like TCP-Africa [22]. TCP-Africa is a hybrid protocol that uses a delay metric to determine whether the bottleneck link is congested or not. In the absence of congestion it uses an aggressive, scalable congestion avoidance rule but in the presence of congestion it switches to the more conservative Reno congestion avoidance rule. A slightly different combined delay/loss-based mechanism is proposed in YeAH-TCP [23]. Two different operation modes are defined: during the "Fast" mode the operation is based on the Scalable TCP's algorithm while in the "Slow" mode, the protocol acts as TCP Reno. The combination of the delay-based and the loss-based approaches also appears in TCP-Illinois [24] and TCP-Adaptive Reno [25]. TCP-Illinois uses loss as a primary congestion signal and delay as a secondary one. The protocol uses an AIMD mechanism but adjusts the increase and decrease parameters based on experienced queueing delay. TCP-Adaptive Reno also dynamically adjusts the TCP response function based on the Buffer and Bandwidth Estimation (BBE) method. Compound TCP [26] is another example where a synergy of delay-based and loss-based approach has been implemented. It uses a scalable delay-based component in the standard TCP Reno congestion avoidance algorithm. The authors of LT-TCP [27] propose an approach that provisions proactive FEC on an end-to-end basis for TCP and also introduce a reactive FEC component to minimize the effect of erasures during the retransmission phase. The improvement significantly increases the performance over networks comprising lossy wireless links.

Another important group of congestion control protocols is based on explicit congestion notification instead of the implicit congestion signals such as packet loss or delay. These congestion control schemes require the assistance of network routers by this means the modification of the routers is also necessary. This is a serious disadvantage from the aspect of deployment feasibility. One of the main representatives of this group is the eXplicit Control Protocol (XCP) [28] which generalizes the Explicit Congestion Notification (ECN) proposal [29]. Instead of the one bit congestion indication used by ECN, XCP capable routers inform the senders about the degree of the congestion at the bottleneck. In addition, XCP decouples the utilization control from fairness control. Another good example for router assisted congestion control protocols is Rate Control Protocol (RCP) [30]. RCP approximates ideal processor sharing by explicitly emulating PS at each router. It is shown in [31] that RCP shows improvement in flow-completion time for realistic traffic mixes (containing not only long-lived flows).

1.2. Related work: performance analysis and its tools

The performance analysis of recently proposed mechanisms and TCP modifications is included in many papers. These works mainly deal with the performance of a new proposal or the interaction of standard TCP (Reno) and the new mechanism. In [32,33], a simulation-based performance analysis of HighSpeed TCP is presented and the fairness to regular TCP is analyzed. In [13], the author deals with the performance of Scalable TCP and analyzes the aggregate throughput of the standard TCP and Scalable TCP based on an experimental comparison. In [8], the performance of different TCP versions, such as HighSpeed TCP, Scalable TCP, Linux TCP and FAST TCP, are compared in an experimental test-bed environment. In all cases, the performance among connections of the same protocol sharing a bottleneck link is analyzed and different metrics are presented (throughput, fairness, responsiveness, stability). In [14], the authors compare the performance of BIC TCP using simulation with that of HighSpeed TCP, Scalable TCP and an AIMD mechanism. Bandwidth utilization, TCP friendliness, RTT unfairness, and convergence to fairness metrics are evaluated. Initial experimental results for combined delay/loss-based congestion control algorithms, namely, TCP-Illionis and Compound TCP are shown in [34].

The analysis of competing flows using different TCP versions has received less attention. In [35], the fairness of MIMD algorithms is evaluated and the interaction of AIMD mechanisms with static parameters (e.g., TCP NewReno) are analyzed. In [5], an experimental evaluation of different high speed protocols, such as HighSpeed TCP, Scalable TCP, BIC TCP, FAST TCP and H-TCP, is presented. In a series of benchmark tests, the intra-protocol behavior of these TCP variants are analyzed considering the effect of starting time of the flows, as well. In [36], experimental evaluation of different high speed TCP proposals is carried out mainly focusing on the relevant impacts of background traffic. In [37], the intra-, and inter-protocol fairness of HighSpeed TCP, Scalable TCP, FAST TCP, H-TCP, BIC TCP and CUBIC is analyzed focusing on the impacts of starting time of the flows. The evaluation is based on simulations conducting in a simple dumb-bell topology with two competing flows. The difficult problem of the parameter settings of delay-based TCP versions (Vegas, FAST TCP) has also been addressed by the research community. As an example, our previous research has resulted in solutions from a game-theoretic analysis [38,21]. We investigated fairness of delay-based TCP variants, FAST TCP in particular, from a game-theoretic point of view, [21]. We have shown that if the FAST TCP's users are assumed to be selfish in terms of setting their desired number of backlogged packets in the buffers along their paths, then the network as a whole, in certain circumstances, would operate *very inefficiently*. As a result, it is very vulnerable to selfish actions of the users. This poses a serious threat to the possible deployment of FAST TCP in the future Internet.

The need for creating a common performance evaluation framework for TCP versions has been identified and addressed by the IETF and IRTF working groups [39]. The goal of our paper is exactly in line with these goals of the community. In the internet draft [40] the metrics to be considered in an evaluation of new or modified congestion control mechanisms for the Internet has been collected. The most important candidate metrics like throughput, delay, packet loss rates, response to sudden changes or to transient events, oscillations in throughput or in delay, fairness and convergence times, robustness for challenging environments, robustness to failures and to misbehaving users, deployability, metrics for specific types of transport and user-based metrics are summarized and discussed in this document. The performance evaluation framework can be implemented in simulators using e.g., the network simulator Ns-2 [41], test-bed analysis and real world experiments. Of course, test-beds and real world experiments can produce much more realistic results than simulations but implementing a large and complex experimental scenario is challenging. This is the reason that most of the proposals so far are restricted to Ns-2 simulation frameworks.

A benchmark tool has been presented in [7]. This benchmark consists of a set of network configurations (i.e., topologies, routing matrix, etc.), a set of workloads (i.e., traffic generation rules), and a set of metrics. The authors propose that the benchmark should be implemented in both Ns-2 simulation mode and hardware experiment mode, and they present some results from their on-going research. Another TCP evaluation suite has been suggested in the internet draft [42]. This consists of an extendable tool that automates the Ns-2 TCP simulation process as much as possible. One can also define a set of commonly used network topologies, traffic models and performance evaluation metrics in the tool. A similar tool [43] has been developed based on an experiment scenario generator, consisting of a topology generator, a flow generator, and a workload generator, which are implemented in a set of tcl scripts for Ns-2 simulator. Based on this tool, some numerical results are presented in [44]. Different high speed TCP proposals are analyzed in complex network topologies with multiple bottlenecks, large number of short-lived and long-lived flows, and variety of RTTs. Here, the statistical behavior of large number of flows is examined in three groups of scenarios: (1) all flows use Reno, (2) half of the flows use Reno and the others use a high speed version, and (3) all flows use a high speed protocol.

Our previous research has also confirmed from several point of views that the dynamic characteristics of TCP cannot be neglected but play a significant role in the complex mechanisms observed in the Internet. For example, we have shown that TCP can propagate self-similarity between distant areas of the Internet due to its dynamic characteristics [45].

The rest of the paper is organized as follows. In Section 3, our main results and our methodology are summarized briefly. In Section 2, the motivation of our work is presented and the insufficiency of Jain's index is discussed. In Section 5 the transient and equilibrium behaviors of different TCP versions are analyzed and analytical results are presented. Novel metrics, namely the saturation time and the main operating frequencies, are also introduced and derived for different TCP variants. Sections 6 and 7 present the comprehensive fairness analysis of competing high speed TCP flows according to our methodology for intra-, and inter-protocol cases, respectively. The impacts of starting delay are also examined and the explanations of the experienced phenomena are given, as well. Conclusions are drawn in Section 9.

2. Why is Jain's index insufficient?

One of the most popular and widely accepted fairness indices is Jain's index [46]. It is used widespread because of its main benefits [47]. Jain's index has a very important role in measuring fairness among large number of flows. It is a normalized metric being bounded between 0 and 1, and can be defined as follows: $JI = (\sum x_i)^2/(n\sum x_i^2)$, where x_i is the normalized (e.g., average) throughput of the i-th flow and n is the number of flows. In contrast to other metrics, such as variance or standard deviation of throughput, it is independent of scale. Furthermore, it can be applied to any number of flows. Contrary to min–max ratio, it is continuous. And finally, this index has an intuitive relationship with user perception. Jain's index is particularly capable to describe the long-term behavior of large number of flows. In other words, the static characteristics of the flow competition is captured especially.

Emerging networks bring new challenges. First, the new architectures, heterogeneous networks, mobile and wireless environments exhibit different network characteristics requiring more attention on the dynamical aspects of the operation. For example, in mobile environment, during an inter-system handover, the delay and the bandwidth can suddenly change. As for a TCP connection, these sudden changes in delay or the high value of jitter can cause multiple back-offs and, in extreme case, disconnection. Second, the network traffic is mainly determined by the popular applications. For example, web applications – generating a lot of short-time connections (dragonflies, mice traffic) – have a great importance today [48]. This type of traffic cannot be treated considering only the long-term properties.

As a consequence of the new network environments and properties, the dynamic behavior of the TCP flows must be taken into consideration. On the one hand, it is obvious that the dynamic effects have significant impact on the performance and throughput of the TCP flows (see e.g., [49]). On the other hand, we argue that the fairness also needs to be reconsidered from the aspects of dynamic behavior. Jain's fairness metric is proposed assuming a simple control system model of n sources sharing the same bottleneck link and receiving the same feedback signal [46]. It can well describe the static properties of competing flows. However, the characteristics of the new network architectures and environments with new routing algorithms can not be well captured in all aspects by that model.

We show two simple examples – competition of two flows in the dumb-bell topology – to illustrate the deficiency of long-term analysis and available fairness metrics. In case of two competing flows, we expect that the equilibrium properties and thus, the fairness in the sense of Jain's index, do not depend on the starting time of the flows. For example, a few seconds of starting difference between the flows can be omitted when the experiment lasts for a very long time (e.g., more than 1 h). In Fig. 1, the performance of two competing Scalable TCP flows is presented for three different starting delays. In the first case, the delay is less than 5 s and the bandwidth shares are close to each other. This fact is also confirmed by Jain's index (0.960) approximating 1. The second scenario corresponds to a starting delay of 15 s, and the bandwidth is shared less fairly which is reflected by a smaller Jain's index (0.698). The most surprising result can be observed in the last scenario. Here, the delay is increased to 50 s and the second flow only achieves very low throughput. This unfairness is confirmed by Jain's index near to 1/2.

The importance of dynamic or transient analysis is illustrated by another example when the interaction of Scalable TCP and HSTCP can be observed. The long-term average behavior of both flows can be approximated analytically and the average throughput can be expressed in terms of packet drop probability [9]:

$$x_{stcp} = \frac{1}{T}\frac{\alpha_{stcp}}{p} \quad \text{and} \quad x_{hstcp} = \frac{1}{T}\frac{\alpha_{hstcp}}{p^{0.84}},$$

where x_{stcp}, x_{hstcp} denotes the average throughput of a single Scalable TCP and HSTCP flow, respectively. T is an approximation of round-trip time at the equilibrium state and $\alpha_{stcp}, \alpha_{hstcp}$ are constant parameters of the protocols. Thus, the equilibrium bandwidth share – assuming that the flows meet the same drop probability – can be expressed as follows:

$$\frac{x_{stcp}}{x_{hstcp}} = \frac{\alpha_{stcp}}{\alpha_{stcp}}\frac{1}{p^{0.16}} \approx 0.625\,\frac{1}{p^{0.16}}. \qquad (1)$$

For small values of p, it can be expected that the Scalable TCP flow gets a slightly more bandwidth than HSTCP. However, the experiences show significantly different behavior and HSTCP is starved. *Dynamic analysis is needed in order to understand the interaction*. In Fig. 2, the congestion window processes and the bottleneck queue are shown. (The simulation corresponds to the dumb-bell topology.) To our best knowledge, this is the first time to explain the poor performance of HSTCP when it co-exists with Scalable TCP. We have noticed that at the equilibrium, HSTCP source operates in Reno mode with very low values of congestion window. HSTCP – similarly to other high speed proposals – defines a lower bound for congestion window (*Low_W* parameter) and when the current value of congestion window does not exceed that, the HSTCP sender acts as TCP Reno applying the standard AIMD mechanism. We use the term *Reno mode* (or *Reno operation mode*) for this phenomenon in the rest of the paper. The root-cause of the poor performance of HSTCP is the Reno mode behavior. But why? We can answer this question by invoking one of the best tools for analysis in the frequency-domain: Fourier transform and FFT (Fast Fourier Transform). In Fig. 2, beside the time-domain plots, the spectrum plots are depicted, as well. (We used the `periodogram` and `psdplot` functions of Signal Processing Toolbox of MATLAB to estimate and plot the power spectral density of the data series. The `periodogram` function is based on the built-in `fft` function implementing Discrete Fourier transform.) It can be observed that the dynamic properties of Scalable TCP are enforced on HSTCP – as the losses occur according to the main frequency spike of Scalable TCP. A HSTCP flow operating at this frequency cannot leave Reno mode.

We found that spectral analysis can provide deeper understanding of the intrinsic behavior of congestion control mechanisms, especially when the interaction of different protocols is analyzed. For the very simple cases, when there exists a single dominant frequency, that can be inferred from the time-domain plots, as well. However, in case of multiple bottleneck links, the macroscopic behavior can be better understood and explained by this tool. Nevertheless, the time-domain and frequency-domain analysis need to be applied together because the time information of these non-stationary processes cannot be inferred from the spectrum plots.

It has been recently debated that in case of competing flows, starting delay can have a relevant impact on the performance and fairness [37]. However, the explanations of the phenomena have not been given, yet.

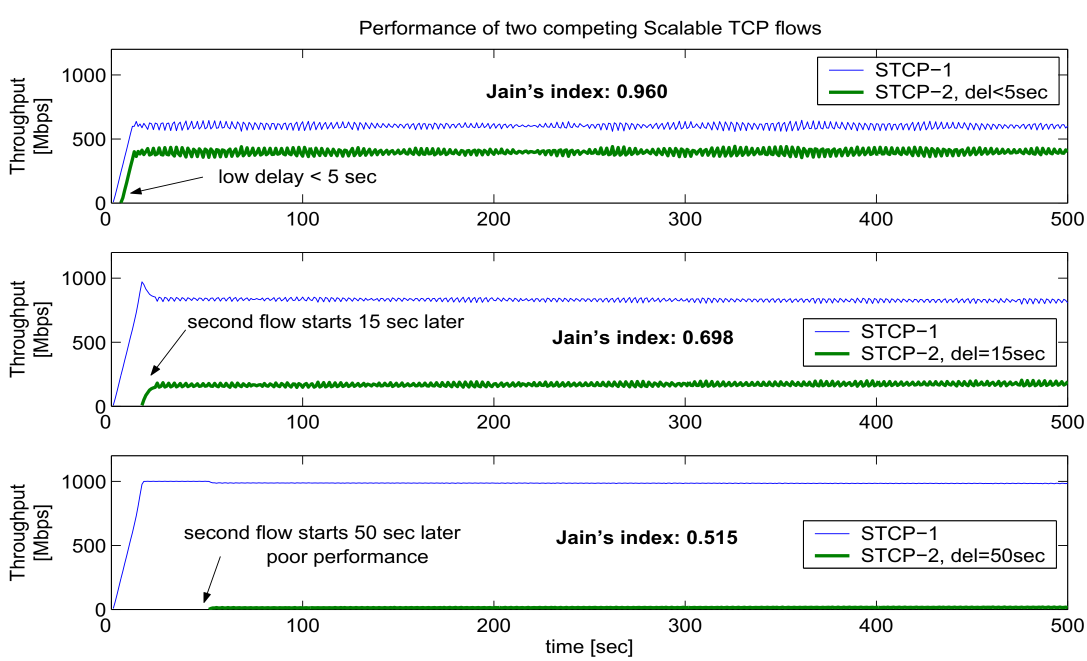


Fig. 1. Performance of two competing Scalable TCP flows for different starting delays.

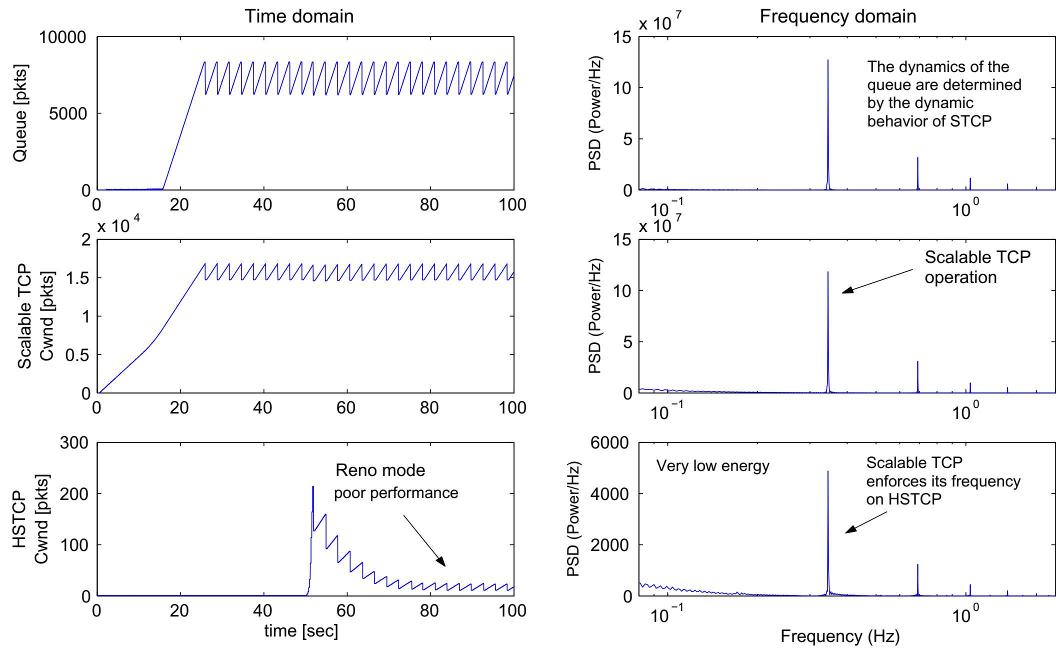


Fig. 2. Competition of Scalable TCP and later entering HSTCP.

3. Main results

In this section, our main results are summarized briefly while the details of the comprehensive fairness analysis are presented next. Since our results involve a wide range of scenarios and investigated properties, for the sake of clarity, we feel that a summary of the main findings is needed. This section is dedicated to that objective. (For the details of the investigated topologies and scenarios, see Section 4.) An overview of the analyzed scenarios and the observed/investigated properties is given in a self-explanatory table (see Table 2). Here, the rows correspond to different scenarios (including intra-, and inter-protocol behavior, interaction of various congestion control principles in different topologies, and different types of competition), while the columns give basic statements on the scenarios regarding fairness and responsiveness. The cells containing simple indicators show whether a property can or cannot be observed in the given scenario. The presented properties characterize the long-term and also the short-term behavior of the interactions.

Our results can be divided into two main classes: confirmatory results and new results. Our confirmatory observations are obviously contains already known results, as well, however, the focus is slightly different concerning the dynamical aspects and transient characteristics. On the one hand, by extensive analysis of a diverse set of scenarios we have found a set of new results. We have also shown that starting time indeed can play an important role on the fairness of competing high speed TCP flows.

- Scalable TCP is revealed by our analysis to be one of the most aggressive examined high speed variants. In particular, Scalable TCP dominates all other loss-based variants, including itself, leading to unfair share of bandwidth with other flows. In parking-lot topology, Scalable TCP also shows aggressive behavior, regardless of the length of the connection. In addition, our spectral analysis shows that this property is rooted in the MIMD design principle (MI – Multiplicative Increase – in particular) of Scalable TCP.
- FAST TCP with appropriate parameter setting shows good performance, in terms of fairness, with all other variants in a wide range of network environments.
- Other examined high speed variants perform differently in different situations and scenarios.

On the other hand, our analysis also reveals, to our best knowledge, new understanding to the subject:

- We showed that the impact of starting time on long-term fairness of competing high speed TCP flows is conditioned on the duration/interval of the starting itself. To illustrate this point, we introduced a novel metric, the "*saturation time*" of a connection. This is the time when the start up phase ends and the connection enters a periodic phase. In contrast to the somewhat rule-of-thumb choice of the starting in the literature, our analysis, taking the saturation time into account, quantifies the impact of the starting time on long-term fairness.
- In-depth spectral analysis of competing high speed TCP flows is also provided in the paper. On the one hand, we introduced a new fairness related metric, namely the *main operating frequencies* of congestion control mechanisms, and determined that for different protocols. On the other hand, the *spectral analysis tool* was intensively applied in order to characterize the interaction of congestion control algorithms. It was suitable for revealing interesting phenomena, especially in scenarios with multiple bottleneck links where the macroscopic behavior is composed of several operating frequencies that can be propagated along the network routes (see Section 7.2.4).

We found that Scalable TCP (and in general, MIMD mechanism) shows unfair behavior beside other loss-based flows in a wide range of network scenarios (see the corresponding cells of Table 2). The AIMD-like flows, even the traffic aggregates, are always starved, regardless of the starting times (long-term unfairness). Moreover, in certain cases, the Scalable TCP flow can force other flows to deviate from their normal operation by falling back to TCP Reno operation mode. For example, this type of unfairness can always be observed beside Scalable TCP aggregate (see Table 2). Furthermore, the aggressive behavior of Scalable TCP is also exhibited in the parking-lot topology with longer paths. The other flows, including other Scalable TCP flow, are always starved (see the parking-lot scenario of MIMD intra-protocol behavior and the parking-lot

Table 2
Main results.

Explanation of the marks: ✓ : YES; × : NO; ? : indefinite; Reno mode : starved flow operates as TCP Reno

Analyzed protocols		
loss-based	MIMD	Scalable TCP
loss-based	AIMD-like	HSTCP
loss-based	AIMD-like	BIC TCP
delay-based		FAST TCP

Scenarios				Properties: fairness – fair (long-term)	fairness – unfairness type: Reno mode	fairness – delay-sensitive	responsiveness – long-term (oscillation)	responsiveness – short-term (fast convergence)
INTRA-PROTOCOL	MIMD	dumb-bell top.	single flows	?	×	✓	✓	✓
			aggreg. – single flow	×	✓	✓	✓	✓
		parking-lot top., single flows		×	✓	×	✓	×
	AIMD-like	dumb-bell top.	single flows	✓	×	×	✓	×
			aggreg. – single flow	✓	×	×	✓	×
		parking-lot top., single flows		×	×	×	✓	×
	FAST TCP	dumb-bell top.	single flows	✓	×	×	?	✓
			aggreg. – single flow	parameter sensitive				
		parking-lot top., single flows		✓	×	×	?	✓
		complex parking-lot top.		×	×	✓	?	✓
INTER-PROTOCOL	MIMD – AIMD	dumb-bell top.	single flows	×	?	×	✓	✓
			MIMD agg. – AIMD	×	✓	×	✓	✓
			AIMD agg. – MIMD	×	?	×	✓	✓
		parking-lot top., single flows		×	✓	×	✓	✓
	FAST – loss-based	dumb-bell top.	single flows	✓	×	×	×	?
			loss-based agg. – FAST	✓	×	×	?	✓
		parking-lot top., single flows		✓	×	×	✓	✓
		inhomogeneous parking-lot top.		✓	×	×	✓	✓
		complex parking-lot top.		×	×	×	✓	✓
	HSTCP – BIC TCP	dumb-bell top.	single flows	?	×	✓	✓	?
			aggreg. – single flow	×	×	×	✓	?
		parking-lot top., single flows		×	×	×	✓	×

scenario of MIMD–AIMD interaction in Table 2). In addition, the intra-protocol behavior and the long-term fairness of the Scalable TCP flows are determined by the transient dynamics, i.e., the starting time of the flows.

We illustrated and showed some surprising benefits of FAST TCP, a delay-based TCP version proposed for next generation networks, in terms of fairness. In this respect, our main findings are summarized in the following. *In contrast to loss-based protocols, FAST TCP with appropriate parameters can always show fair (or almost fair) behavior beside loss-based TCP flows* (*HSTCP, Scalable TCP, and BIC TCP*) *in a network with a single congested link.* (We use the term "*almost fair*" when the long-term behavior is only close to a fair bandwidth share. This bias in the equilibrium state can be caused by the estimation error of `baseRTT`.) Concerning the dynamics of TCP starting times the fair or almost fair state is achieved by different ways:

- If the FAST TCP flow starts first then a fair and quasi stable equilibrium state can always be directly achieved.
- In case of a later entering of FAST TCP flow the equilibrium state is reached through an oscillating transient phase with a length depending on the starting time and other parameters.

We have also found that this fair behavior of FAST TCP seems to be a robust property of the protocol which still holds in an aggregated traffic mix or in different topologies (see Table 2). *More specifically, we have found that FAST TCP can achieve good utilization against aggregates of loss-based variants (HSTCP, Scalable TCP, and BIC TCP), and it can also share the bottleneck bandwidth fairly with single loss-based flows in simple parking-lot topology (with one congested link). Using appropriate parameters, the beneficial fairness properties still holds for network with multiple congested links, however, in general, the increasing number of bottleneck links may result in performance degradation.* We should also note that this property holds for a certain range of the parameter α depending on the actual network topology, flow parameters, etc. To find a method which can continuously change this parameter according to the network and flow environments to keep this property broadly general is a good point of future research.

4. Details of network environment and TCP parameters

Our comprehensive fairness analysis of competing high speed TCP protocols and the validation of the analytical results are carried out in the Ns-2 [41] simulation environment. Our simulation scripts regarding different network scenarios can be found in [2]. The different high speed transport protocols are integrated in the environment. Ns-2 version 2.27 includes the algorithm of High-Speed TCP, while the Scalable TCP control mechanism can easily be implemented. The Ns-2 source code of BIC TCP is used from [50], and the FAST TCP implementation is borrowed from [51].

The examined dumb-bell topology containing one bottleneck link is shown in Fig. 3(a). The quite large link delay simulates transatlantic link characteristics. However, the experienced phenomena are similar for networks with smaller propagation delays. (Illustrative examples are given in Section 8.) The queueing mechanism corresponding to the bottleneck link is drop-tail. We do not consider the impacts of the buffer size in our analysis and the buffers are set according to the bandwidth-delay product. We found that the quantitative properties of competing flows are affected by the size of the buffers in the network; however, the basic phenomena and the qualitative characteristics do not depend on this parameter. We also investigate a simple parking-lot topology (Fig. 3(b)), where the impacts of different round-trip times (RTT) can be revealed. Here, only the second link is congested. In case of these scenarios, a simulation contains two competing flows starting at different time instants and performing an infinite FTP download. Investigating the impacts of the starting time, different values are chosen. More exactly, on the one hand, we analyze scenarios when the second flow enters later than the saturation time of the first flow (e.g., with 50 s delay), and on the other hand, scenarios with smaller delay (e.g., with 15 s delay) are also examined. In the dumb-bell topology, the competition of a later entering flow against a traffic aggregate containing 10 flows using the same protocol is also analyzed.

Topologies with multiple congested links are also investigated. The first scenario (Fig. 3(c)), referred as inhomogeneous parking-lot topology, contains two congested links with different propagation delays. The link delays are significantly smaller in this environment than the transatlantic example. A complex parking-lot topology with five nodes (Fig. 3(d)) is also considered, where all links are congested. In these scenarios, one FAST TCP flow traverses across the backbone containing two or four congested links, respectively, and flows with loss-based protocols use single links.

During the evaluation, the default parameter set of the protocols is used (see [12,13]). HSTCP and Scalable TCP apply the Limited Slow-Start (LSS) mechanism [52], as well. In case of FAST TCP, the *appropriate* parameters (α and β) of the protocol are chosen to get fair operation as FAST TCP flow occupies the half of the bottleneck buffer (or buffers). The parameters of the simulations are summarized in Table 3.

5. Transient and equilibrium analysis of TCP versions

In order to understand the behavior of large number of TCP flows in realistic network environments, the first step has to be the investigation of individual flows. Different congestion control principles result in different traffic characteristics. In this section, the behavior of individual flows is summarized in order to gain a basic knowledge of the behavior of different congestion control principles. Here, the investigation is carried out considering the simple dumb-bell topology.

The performance of a single flow can be analyzed in two separate operating regimes. The first phase is a transient phase while the second one corresponds to an equilibrium behavior. Regarding the two operating regimes, we introduce novel metrics – capturing short-term and long-term dynamic properties – that can play important roles in the characterization of interacting flows and fairness performance. We present our methodology through the

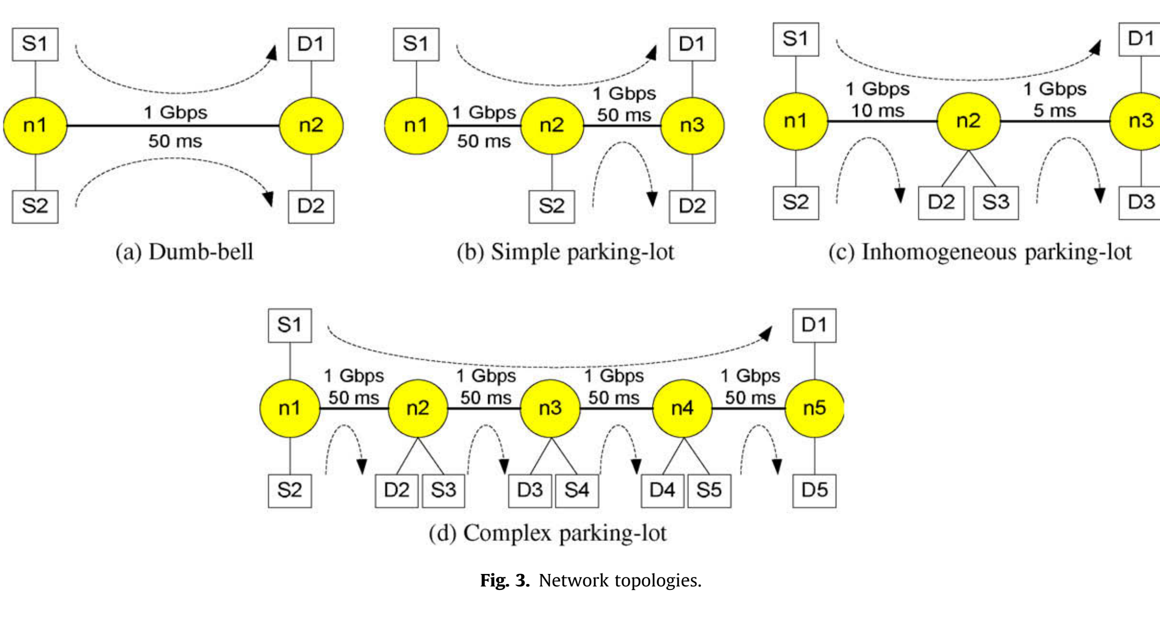


Fig. 3. Network topologies.

Table 3
Parameters.

Parameters	Topologies			
	dumb-bell	simple parking-lot	inhomogeneous parking-lot	complex parking-lot
capacity	1 Gbps	1 Gbps	1 Gbps	1 Gbps
delay	50 ms	50 ms	10 ms / 5 ms	50 ms
buffer size	8,333 pkts	25,000 pkts / 25,000 pkts	1,666 pkts / 833 pkts	8,333 pkts
packet size	1,500 bytes	1,500 bytes	1,500 bytes	1,500 bytes
cwnd / queue sampling	0.01 sec	0.01 sec	0.001 sec	0.01 sec
throughput sampling	1 sec	1 sec	0.1 sec	1 sec
FAST TCP $\alpha = \beta$	4,166 pkts	12,500 pkts	1,250 pkts	4,166 pkts

HSTCP

Low_W	38
$High\_W$	83,000
$High\_P$	10^{-7}
$High\_Dec$	0.1

STCP

a	0.01
b	0.125

BIC TCP

β	0.8
S_{max}	32
S_{min}	0.01
B	4

FAST TCP

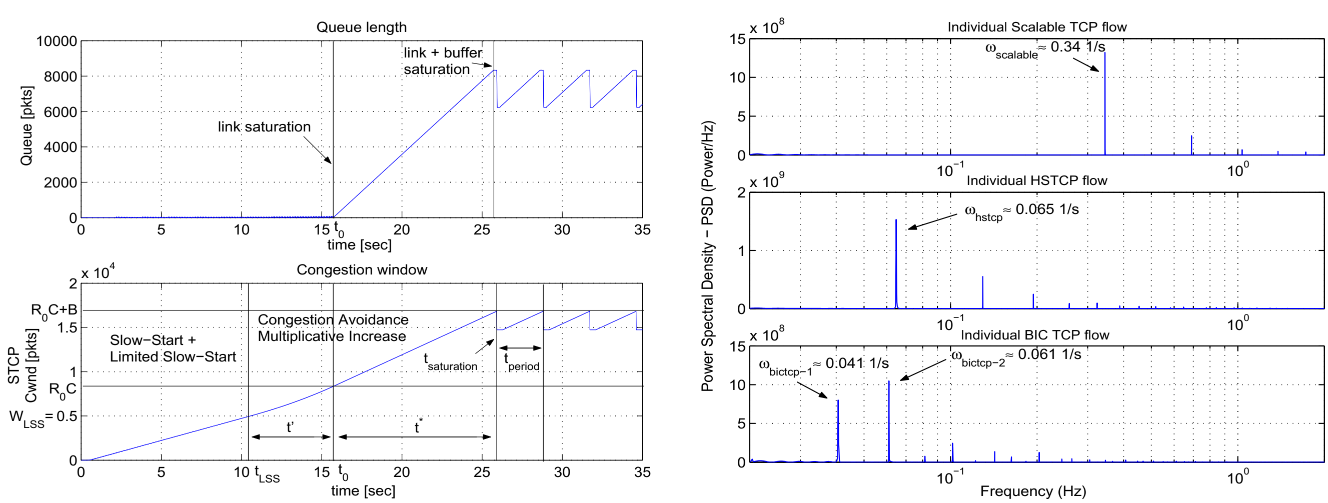
$\alpha = \beta$	see above
γ	0.5
cwnd_update_period	0.01
mi_threshold	0.0015

example of Scalable TCP protocol. By this approach, other TCP variants can easily be treated and the important parameters can be derived analytically.

5.1. Initial dynamics – saturation time

In this section, we focus on the initial phase which plays a significant role of the performance of an entering flow. We introduce a new performance metric, namely, the *saturation time*, as the length of this transient phase. This metric can be defined for a loss-based protocol as the time from the starting till the first packet drop. In Fig. 4, the saturation time and different phases of an individual Scalable TCP flow are presented as an illustration. Increasing the congestion window (and sending rate) of the source, the bottleneck link will be saturated after a while (link saturation). After this event, the buffer is filled by the new arriving packets. The time instant when the buffer is full at the first time is the saturation time. For a delay-based protocol, depending on the network environment (buffer size, parameters of the protocol), packet losses can be avoided during the operation. In these cases, the network is said to be saturated when the congestion window has settled down around the equilibrium state or the source has entered the delay-based ("AIAD-like") operating regime.

Various TCP versions apply different mechanisms during the initial phase. A source generally starts sending according to a Slow-Start-like manner using a multiplicative increase algorithm with a protocol-dependent parameter. This means that the congestion window is increased by a constant value for each acknowledgement received. In our particular cases, the protocols use the following mechanisms. The behavior of HSTCP and Scalable TCP is determined by the Slow-Start and Limited Slow-Start algorithms. With certain network parameters, the Limited Slow-Start phase can be left for the additive increase (HSTCP) or the multiplicative increase (Scalable TCP) phase, before the first packet drop. BIC TCP applies a multiplicative increase algorithm beside the standard Slow-Start mechanism. FAST TCP also increases its congestion



(a) Saturation time of Scalable TCP

(b) Spectrum of cwnd process of individual flows: STCP (top), HSTCP (center), BIC TCP (bottom)

Fig. 4. Transient and equilibrium behavior.

window according to a multiplicative increase algorithm if it is far from the equilibrium state. As a result, to understand the saturation behavior of different protocols, we have to understand the operation of basic algorithms used in the initial phases.

As an illustration, the saturation time of a Scalable TCP flow is derived. Our simulation results corresponding to this scenario are shown in Fig. 4 with the main parameters. During the consecutive phases of initial operation, the Slow-Start, Limited Slow-Start (LSS) and Scalable TCP's multiplicative increase mechanisms are applied. In [2], we summarize the main characteristics of these analytically tractable control algorithms and we derive the relevant parameters, as well. The following variables are used throughout this section: C, R_0, B denote the capacity, the round-trip propagation delay and the buffer size corresponding to the bottleneck link, while a and b are the parameters of Scalable TCP. In our scenarios, the Slow-Start phase is left when the initial threshold ($\texttt{ssthresh} =$ 100 pkts) is exceeded. This time instant can easily be expressed as $t_{SS} = R_0 \log_2 \texttt{ssthresh} \approx 0.664$ s. After t_{SS}, the source operates according to the LSS mechanism using the default parameter ($\texttt{max\_ssthresh} = \texttt{100}$ pkts). LSS operates in congestion avoidance mode in the Ns-2 implementation till the first packet drop. It affects the increase mechanism of `cwnd` comparing the increment of the congestion control mechanism (e.g., Scalable TCP, HSTCP) with its own increment and the maximum of these values is used. With this algorithm, a faster convergence can be achieved when the source sending rate is far from the equilibrium value. In Limited Slow-Start phase, `cwnd` is increased by at most $\texttt{max\_ssthresh}/2$ per round-trip time. The end of the LSS phase, actually, can be caused by a packet drop or the fact that the protocol's increase mechanism "suggests" more aggressive increment than the LSS algorithm. In our simulations, the end of this phase depends on the protocols and other network parameters, as well. In case of Scalable TCP, the end of LSS phase can be expressed as follows (see [2] for details):

$$t_{LSS} = R_0 \frac{\lg \texttt{max\_ssthresh}}{\lg 2} + R_0 \frac{W_{LSS} - \texttt{max\_ssthresh}}{\texttt{max\_ssthresh}/2} \approx 10.46 \text{ s}, \quad (2)$$

where W_{LSS} is the value of congestion window triggering the end of LSS (when the increment of the LSS algorithm and the MI mechanism are equal), and it can be expressed by the multiplicative increase parameter (a) of Scalable TCP and the LSS parameter ($\texttt{max\_ssthresh}$) as follows:

$$W_{LSS} = \frac{\texttt{max\_ssthresh}/2}{a} = 5000 \text{ pkts}. \quad (3)$$

After Limited Slow-Start, the multiplicative increase mechanism of the protocol operates. Till the first packet drop, two qualitatively different phases can be identified. During the first period (denoted by t'), the buffer is empty and the queuing delay is zero. The congestion window is growing from W_{LSS} to the BDP (R_0C) according to the MI mechanism, and by the end of this phase (denoted by t_0) the link has been saturated. Thus, the link saturation time can easily be determined:

$$t_0 = t_{LSS} + t' = t_{LSS} + R_0 \frac{\lg \frac{R_0 C}{W_{LSS}}}{\lg(1+a)} \approx 15.6 \text{ s}. \quad (4)$$

The second phase of the MI mechanism (denoted by t^*) lasts from the link saturation time till the saturation time (when the buffer is also saturated). During this interval, the bottleneck queue is growing, and the queueing delay has a significant impact on the congestion window process. However, the length of this period can also be determined by solving differential equations describing the dynamics of congestion window and the behavior of the queue. Instead of solving analytically not tractable differential equations (with varying delays and recursive arguments), a simple approximation can be applied. In this phase, the congestion window is increased from $W_0 = R_0C$ to $R_0C + B$ according to the multiplicative increase mechanism. Approximating the increase of the queueing delay by a linear function, the round-trip time can be treated as a constant with a mean value corresponding to a state when the half of the buffer capacity is used: $\widetilde{R} = R_0 + B/(2C)$. Thus, the saturation time can be expressed as follows:

$$\hat{t}_{saturation} = t_0 + t^* = t_0 + \widetilde{R} \frac{\lg \frac{R_0 C + B}{R_0 C}}{\lg(1+a)} \approx 26.05 \text{ s}. \quad (5)$$

The analytically derived parameters and the approximation of saturation time meet well the simulation results presented in Fig. 4. In Table 4, we summarize our results on the transient behavior of different protocols (for details, see [2]).

5.2. Equilibrium behavior

On the one hand, after the saturation time, an individual flow using a loss-based protocol shows periodic equilibrium behavior with periodic packet losses. The relevant parameters characterizing this state can also be derived analytically for the protocols, respectively. We introduce the *main operating frequencies* gained from the spectral characteristics of the congestion window processes as another novel metric that can be used beside the saturation time to get an enhanced characterization of congestion control mechanisms. On the other hand, a single delay-based protocol can realize stable equilibrium state without oscillation or packet drops in the network. To understand the long-term behavior of individual flows is crucial in order to understand the interaction of different flows later. In this section, we summarize the long-term characteristics of the examined protocols. Further details and the analytical derivations can be found in [2].

As a simple illustration, we present the long-term behavior of an individual Scalable TCP flow. The MIMD mechanism of the protocol – operating during the equilibrium state – yields a time period $k = -\log(1-b)/\log(1+a)$ (expressed in RTT). After the first packet drop, the Scalable TCP source operates around an operating point when the bottleneck queue is approximately full. Thus, the round-trip time ($R(t)$) can be approximated at that operating point by $R(t) \approx \widetilde{R} = R_0 + B/C$, where R_0 is the round-trip propagation delay, C is the bottleneck capacity, and B is the buffer length. In our example $\widetilde{R} \approx 0.2$ s. According to these results, the time of a period (t_{period}) can be approximated by $k\widetilde{R} \approx 2.6 \ldots 2.8$ s. The analytical result well captures the periodic behavior experienced in the simulations. In Fig. 4, the length of a period $t_{period} \approx 2.9$ s. This period also contains the time which is needed for retransmit and recovery. The spectrum of the congestion window process can also be derived. The relevant part of the spectrum is shown in Fig. 4(b) (top). The main spike corresponding to the dominant frequency can be seen at approximately $\omega = 0.34\,1/\text{s}$ which meets well the equilibrium time period derived analytically.

For the other loss-based protocols, the periodic characteristics can be determined in a similar way. The details of analytical derivations can be found in [2]. In Fig. 4(b), we show the spectrum

Table 4
Approximation of saturation time of different protocols.

Scalable TCP	$\hat{t}_{sat} = t_{LSS} + R_0 \frac{\lg \frac{R_0 C}{W_{LSS}}}{\lg(1+a)} + \widetilde{R} \frac{\lg \frac{R_0 C + B}{R_0 C}}{\lg(1+a)}$	≈ 26 s
HSTCP	$\hat{t}_{sat} = t_{SS} + R_0 \frac{R_0 C - \texttt{max\_ssthresh}}{\texttt{max\_ssthresh}/2} + \widetilde{R} \frac{B}{\texttt{max\_ssthresh}/2}$	≈ 42 s
BIC TCP	$\hat{t}_{sat} = R_0 \frac{\lg \texttt{max\_ssthresh}}{\lg 2} + R_0 \frac{\lg \frac{R_0 C}{\texttt{max\_ssthresh}}}{\lg(1+a)} + \widetilde{R} \frac{\lg \frac{R_0 C + B}{R_0 C}}{\lg 1+a}$	≈ 12 s
FAST TCP	$\hat{t}_{sat} = R_0 \frac{\lg R_0 C}{2} + t_{AIAD}$	≈ 2 s

plots of the congestion window processes of individual flows. The main frequency spike of HSTCP can be seen at approximately $\omega = 0.065\,1/\mathrm{s}$ which is significantly smaller than the frequency of Scalable TCP. The control mechanism of BIC TCP yields two main spikes at 0.04 1/s and 0.06 1/s, respectively. The equilibrium state of an individual delay-based FAST TCP flow shows stable behavior significantly different form the oscillating nature of loss-based protocols. During the connection, the size of the bottleneck queue is kept between the α and β parameters of the protocol, and losses do not occur.

6. Intra-protocol behavior

In the previous section, the behavior of individual flows and congestion control mechanisms were analyzed in a simple network environment. As an essential requirement, a TCP protocol should guarantee fair behavior among flows using that protocol. The next step of our investigation involves the analysis of the interaction of the same TCP flows in order to get a basic knowledge of the network behavior when all the sources use the same transport protocol. This topic has received more attention recently, and a lot of results can be found in the literature (see e.g., [8,5]). Therefore, the aim of this section is rather to get a basic knowledge and to confirm and explain certain results than to provide a comprehensive study.

The details of our analysis can be found in [2]. Here, we summarize only our main findings and the main properties of the intra-protocol behavior for different types of congestion control principles. More specifically, the interaction of similar Scalable TCP, HSTCP, BIC TCP and FAST TCP flows is examined in the dumb-bell and in the simple parking-lot topology. The impacts of the starting time and the competition of individual flow against traffic aggregate are also investigated.

6.1. Scalable TCP

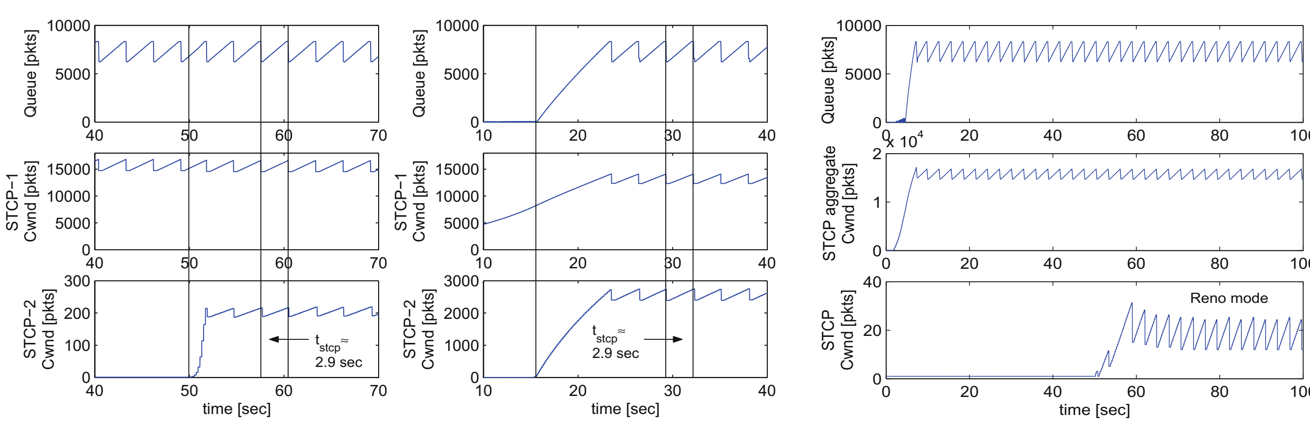
Our first statement is the following: *the MIMD mechanism cannot guarantee the fair behavior among flows using the same MIMD algorithm assuming synchronized losses even in very simple network environment.* More exactly, the performance of the competing Scalable TCP flows are mainly affected by the starting time of the flows.

As an illustration, the competition of two Scalable TCP flows in a very simple dumb-bell topology is presented. The congestion window processes and the dynamics of the bottleneck queue are shown in Fig. 5(a and b) for two different starting delays. In the first scenario, the second flow enters the network after the saturation time of the first one (50 s), while the second scenario corresponds to a delay (15 s) smaller than the saturation time. As a consequence of the properties of the MIMD algorithm, during the equilibrium phase, the two sources operate at the same frequency. The performance of the second flow and the fairness are mainly determined by the state of the first flow at the time instant of entering. Thus, the synchronized losses and synchronized periods of the two Scalable TCP flows can cause an unfair equilibrium state and unfair bandwidth share when the second flow is starved. Moreover, a later entering Scalable TCP flow shows very poor performance competing with a traffic aggregate of Scalable TCP flows. The queue process and the congestion window process of the traffic aggregate (as the sum of the individual processes) and the late entering flow are shown in Fig. 5(c). Here, the competition of a single flow against a traffic aggregate of 10 flows are presented. Starting times of the traffic aggregate flows are uniformly distributed within the first 5 s while the last flow enters 50 s later. Because of the basic properties of the MIMD algorithm, the dominant frequency of the traffic aggregate equals the one derived for a single Scalable TCP flow (see Fig. 4(b)). The main frequency spike can be observed at approximately 0.34 1/s in both cases. The later entering flow shows very poor performance operating in Reno mode which is a *serious disadvantage* of the protocol from practical aspects.

6.2. AIMD-like mechanisms

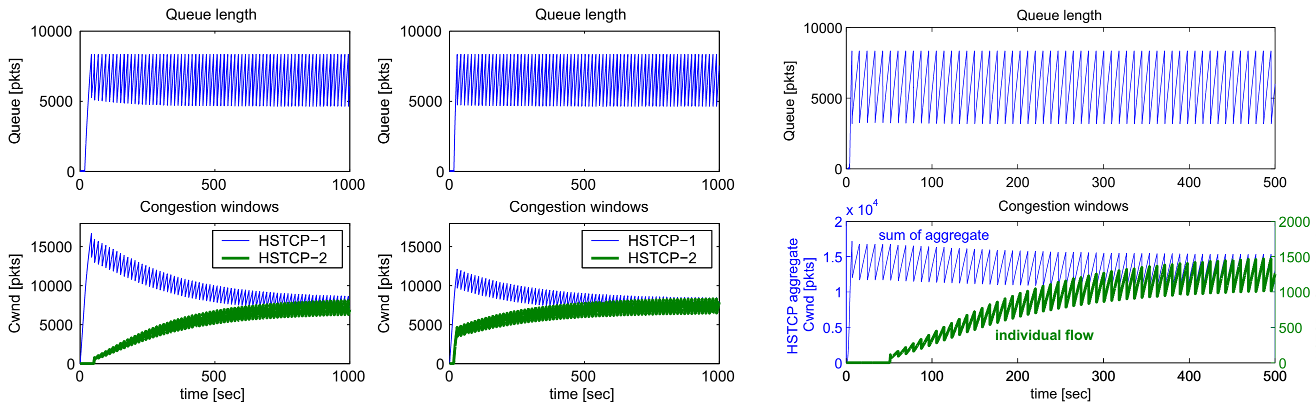
The less aggressive congestion control schemes using additive or slower (logarithmic) increase mechanisms are able to realize fair equilibrium states for similar flows; however, the transient phases can last unacceptable long time. In case of HSTCP and BIC TCP, the starting time has an impact on this transient phase but the long-term behavior is not affected.

As an illustration, we present some results on the intra-protocol behavior of HSTCP. The adaptive nature of the protocol originates from considering the current value of the congestion window during the "rate" adjustment. In Fig. 6, the dynamics of congestion windows and the bottleneck queue are shown for three different scenarios using the dumb-bell topology. These scenarios are similar to the previously presented ones for Scalable TCP (two flows – 50 s delay, two flows – 15 s delay, traffic aggregate – individual flow). Our results show that *in case of HSTCP flows operating in simple dumb-bell network environment, the starting delay has an impact on the convergence time; however, the long-term fairness is not*



(a) 2 flows, delay: 50s (b) 2 flows, delay: 15s (c) individual flow vs. traffic aggregate

Fig. 5. Intra-protocol behavior: Scalable TCP.



(a) 2 flows, delay: 50s (b) 2 flows, delay: 15s (c) individual flow vs. traffic aggregate

Fig. 6. Intra-protocol behavior: HSTCP.

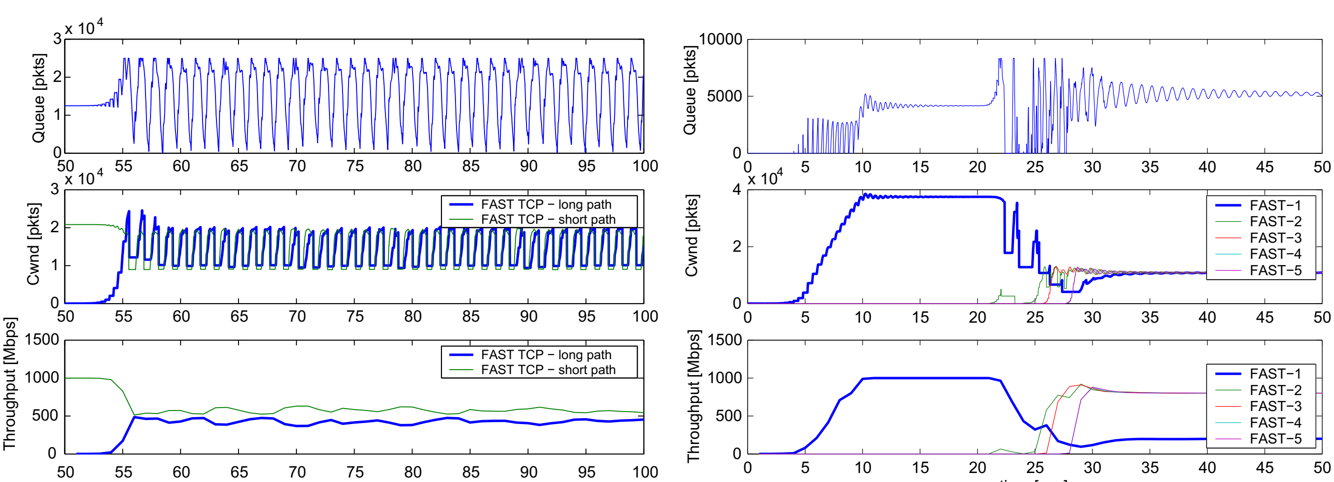
affected and a fair equilibrium state is realized. In Fig. 6(c), the vertical axes of the congestion window plot corresponding to the traffic aggregate (left axis) and the individual flow (right axis) are scaled differently. A fair equilibrium state is realized after a quite long transient period. (The ratio of the congestion window of the individual flow and the sum of the congestion windows of the aggregate is approximately 1:10.) HSTCP flows – similarly to TCP Reno – show RTT unfairness reaching unfair equilibrium states in the parking-lot topology. This property does not depend on the starting time of the flows.

6.3. FAST TCP

The delay-based FAST TCP with appropriate parameters can achieve fair or almost fair equilibrium states very quickly in simple network topologies. However, the good properties do not hold in more complex network environments. The equilibrium state in certain cases depends on the starting delay of the flows. (During our investigation, the parameters – α and β – of the protocol are set to similar values for intra-, and inter-protocol scenarios. These parameters yield an operation when the half of the bottleneck buffer is targeted by the FAST TCP flow.)

In Fig. 7(a), the behavior of two competing FAST TCP flows operating in the simple parking-lot topology (only one link is congested) is shown as an illustration. In the figure, the dynamics of the bottleneck queue, the congestion windows and the bandwidth shares are presented. An unstable equilibrium state with oscillation is achieved after a short transient phase. Both FAST TCP flows attempt to occupy the half of the buffer yielding buffer overflows and packet drops. This behavior can be explained by the "hybrid" congestion control mechanism of FAST TCP. When FAST TCP detects a packet loss, the congestion window is halved similarly to TCP Reno (multiplicative decrease). This adjustment causes the sudden decrease of bottleneck queue size and round-trip delay, as well. As the increment of FAST TCP is proportional to the distance from the targeted state, an aggressive growing of the congestion window can be observed after a loss-recovery period till the next packet drop that occurs around the same state as previously. These mechanisms yield an "MIMD-like" oscillation. Because of the loss synchronization effect, the two flows show the same long term operation. The main frequency of this periodic behavior – which is mainly determined by the round-trip time and loss-recovery time – is exhibited at the frequency of approximately 1.64 Hz. In spite of this unstable interaction, the average bandwidth share is near to a fair state. This fact is also confirmed by the Jain's index (0.861). We have further observed that the interaction of competing FAST TCP flows depends on the starting time of the sources, the estimation of the `baseRTT` and other parameters (e.g., threshold of multiplicative increase) of the protocol.

Surprisingly, the good fairness properties of FAST TCP do not hold in more complex network environments. In Fig. 7(b), the



(a) Simple parking-lot topology (b) Complex parking-lot topology

Fig. 7. Intra-protocol behavior of FAST TCP.

simulation results corresponding to a complex parking-lot topology consisting of four congested links are presented. In these scenarios, the first flow traverses four congested links (backbone) while the other flows use only single links. The sources use the same set of parameters and the flow on the backbone starts first. The later entering flows force packet drops and the congestion windows settle down at the same stable equilibrium state for all sources. Hence, the bandwidth share is not fair and the flow on the backbone is starved. (On the longer path, much higher `cwnd` would be needed for fair bandwidth share.) As a result, *FAST TCP flows with the same parameters in a complex network environment with multiple congested links cannot share the bandwidth fairly*. It is worth noting that increasing the α parameter of the flow on the backbone a better performance can be achieved.

7. Interaction of TCP versions

In the previous sections, the behavior of individual flows and the competition of flows applying the same congestion control mechanism have been discussed. Today, it is an important question that the recently proposed transport protocols how can *live* beside each other in a shared network environment. The current section is devoted to answer this question and investigate the interaction of the most important congestion control approaches. This comprehensive study is the main part of our work and contains the most exciting and sometimes surprising observations. On the one hand, we analyze known phenomena but also extend the investigation to a wider range of network environments and situations to get more general results (*horizontal research*). Beside this confirmatory type analysis, on the other hand, we explain the experienced phenomena (*vertical research*).

(In this paper, the interaction of AIMD-based mechanisms, more specifically, the competition of HSTCP and BIC TCP is not analyzed. The results in this case are not surprising and can be assessed according to our background knowledge. However, our results are summarized in [2].)

7.1. Scalable TCP – other loss-based protocol using additive increase mechanism

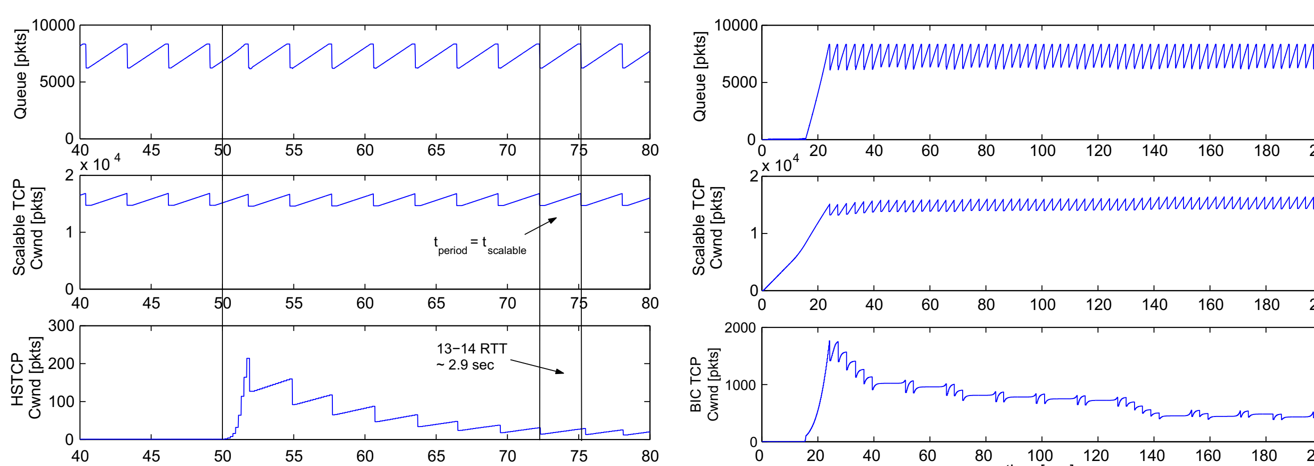
An essential part of a loss-based high speed transport protocol is the adequate increase mechanism. On the one hand, the multiplicative increase algorithm increases the congestion window by a constant value for each acknowledgement independently the current value of the window. On the other hand, adaptive additive increase mechanisms can change the increment according to the current value of the congestion window. (The exact parameters and the dynamic nature are different for these AIMD-like protocols.) It is an important question that how the two types of increase mechanism can work together. This section gives answers to this question based on the analysis of a diverse set of scenarios.

7.1.1. Dumb-bell topology: single flows

To reveal the basic properties of the interaction, first we investigate the competition of two single flows in a simple dumb-bell topology.

A surprising phenomenon can be observed when a HSTCP source enters the network after a Scalable TCP flow has achieved its maximal sending rate. Our first example is shown in Fig. 8. Here, the congestion window processes and the dynamics of the bottleneck queue are presented. In this scenario, the HSTCP source starts sending after the saturation time of the Scalable TCP flow (the delay is exactly 50 s). HSTCP starts sending according to the Slow-Start algorithm and the first packet drop occurs synchronized with the other flow and triggers the congestion avoidance phase. Losses (caused by buffer saturations) occur synchronized between the two flows during the connection. The periodic behavior of HSTCP is exactly determined by Scalable TCP and a common time period is exhibited. In our simulation example, the time period of Scalable TCP is approximately 13–14 RTT. During this time interval, HSTCP can achieve a maximum increment of $13a(W)$ of `cwnd`, while the decrement $(b(W))$ is greater yielding a decreasing trend. In the equilibrium state, `cwnd` will be smaller than *Low_Window* parameter of HSTCP which results in *TCP Reno operating mode* $(a(W) = 1, b(W) = 0.5)$ and poor performance. During the equilibrium state, Scalable TCP forces the HSTCP flow to deviate from its normal operation. The starting time of HSTCP affects only the length of the transient phase. Moreover, when HSTCP starts before the saturation time of Scalable TCP or the sources start at the same time, the equilibrium states are the same, only the length of the transient time differs.

The interaction between Scalable TCP and BIC TCP shows similar characteristics when the Scalable TCP flow starts first. The later entering BIC TCP flow cannot achieve significant rate; however, the performance is not as poor as in the previous case ("non-Reno mode"). As an illustration, the `cwnd` processes and the queue dynamics are shown for 15 s starting delay in Fig. 8(b). The equilibrium state does not depend on the starting time of the BIC TCP flow.



(a) Scalable TCP – HSTCP, delay: 50 sec (b) Scalable TCP – BIC TCP, delay: 15 sec

Fig. 8. Inter-protocol behavior: Scalable TCP – other loss-based protocol using AI mechanism.

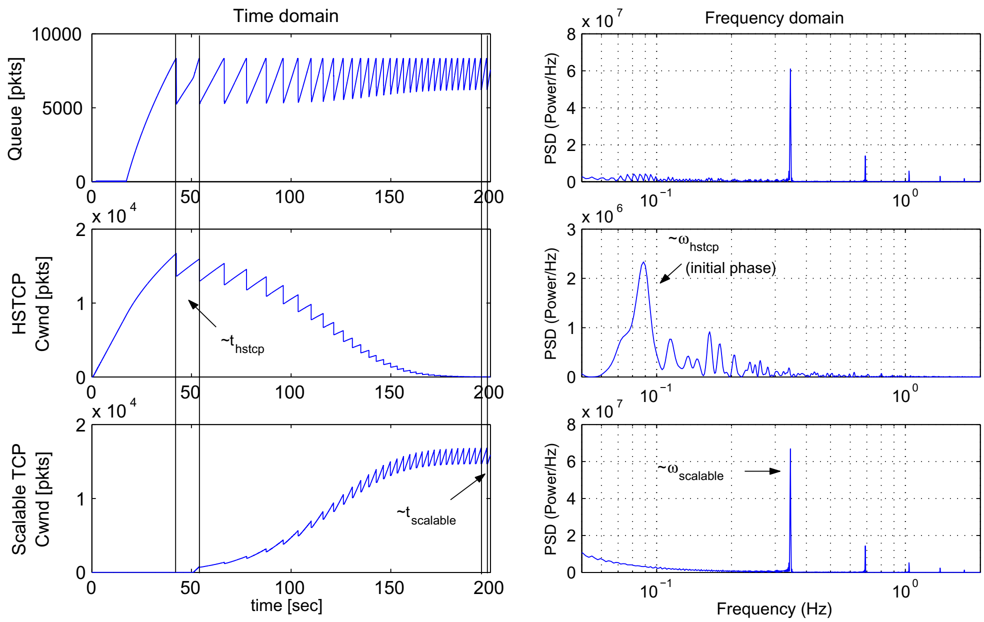


Fig. 9. HSTCP–Scalable TCP, delay: 50 s.

A better performance is expected when the Scalable TCP enters later the network. Surprisingly, the results do not meet the expectations and the starvation of AIMD-like flows can be observed in these cases, too. As an example, we examine the interaction of a HSTCP flow and a 50 s later entering Scalable TCP flow. The simulation results are shown in Fig. 9. At the time of starting Scalable TCP, HSTCP has achieved its equilibrium state with a time period of t_{hstcp}. Scalable TCP starts with Slow-Start/Limited Slow-Start and the bottleneck queue is fed by the traffic aggregate of the two flows. The extra traffic of Scalable TCP in the queue results in a decreasing time period, i.e., the synchronized losses occur more frequently. During shorter time periods, cwnd of HSTCP cannot achieve the value that was just before the reduction. In contrast with HSTCP, cwnd of Scalable TCP – adjusted according to the multiplicative increase algorithm – exceeds the value that has been reached before the reduction at the end of a period. Thus, the length of a period converges to the time period of Scalable TCP ($t_{scalable}$) and cwnd of HSTCP shows a decreasing trend while cwnd of Scalable TCP is increasing. The equilibrium state is the same as it was experienced previously. The same phenomena can also be observed in the frequency-domain. In Fig. 9, the spectrum plots are shown beside the time-domain plots. The diagrams confirm that the long-term network behavior is mainly determined by the Scalable TCP flow, since the bottleneck queue shows the same dominant frequency as the MIMD mechanism.

In case of BIC TCP, the phenomena are similar. The long-term performance and the equilibrium behavior are the same when the BIC TCP flow starts first or later. It is worth noting, that BIC TCP achieves better utilization than HSTCP because it can leave Reno mode.

We have found that single AIMD-like flows are always starved by an individual Scalable TCP flow and the starting delay of the flows has an impact only on the transient characteristics.

7.1.2. Dumb-bell topology: traffic aggregate – single flow

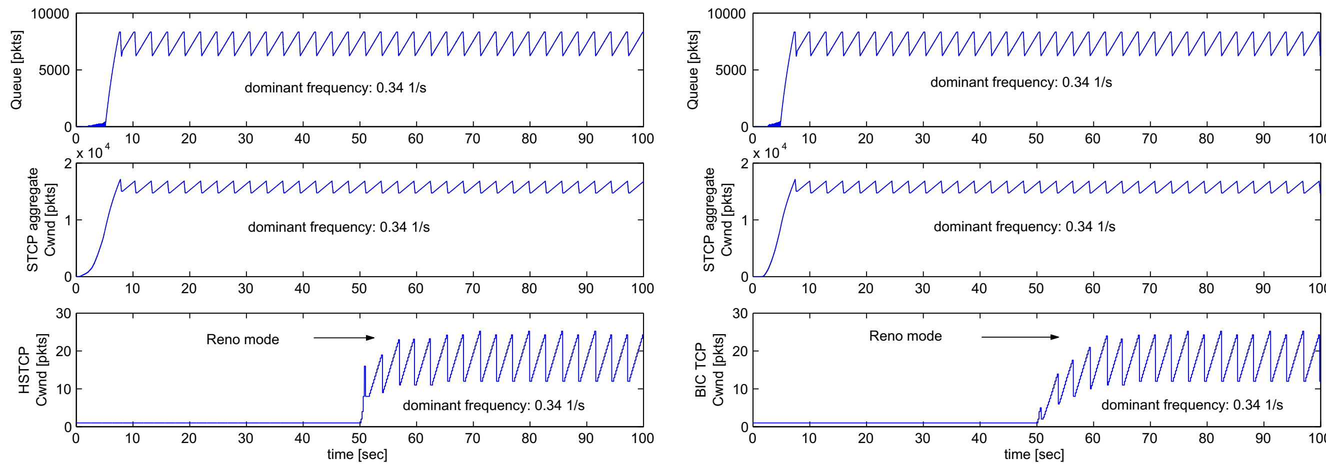
After the investigation of two interacting flows, this section deals with more realistic scenarios when one single flow competes with a traffic aggregate using the other type of congestion control mechanism. The simulation results are presented when the late entering flow competes with 10 other flows operating in their equilibrium state. (The starting times of the traffic aggregate flows are uniformly distributed within the first 5 s.)

Our first observation is that the later entering AIMD-like flow cannot achieve significant bandwidth share against the Scalable TCP aggregate and HSTCP and BIC TCP operate as TCP Reno ($cwnd < 30$). The bottleneck queue and the cwnd processes corresponding to the two scenarios are shown in Fig. 10. The upper parts of the figures relate to the queuing processes while the lower parts correspond to the traffic aggregates and the individual flows. In case of the aggregate, the data is the sum of the data series of the individual congestion windows. The periodic behavior of the individual AIMD-like flow is exactly determined by the MIMD mechanism of Scalable TCP, since the spectrum of HSTCP and BIC TCP shows the same dominant frequency spikes. (Here, we only present the dominant frequencies and the spectrum plots are omitted. Further analysis can be found in [2].) As a consequence of the MIMD algorithm, the spectral behavior of the Scalable TCP aggregate is similar to the behavior of an individual flow.

The next disadvantage of Scalable TCP is shown in Fig. 11. Surprisingly, the individual Scalable TCP flow can starve the HSTCP and BIC TCP aggregates, as well. On the one hand, HSTCP flows operate in Reno mode achieving very low utilization at the equilibrium state and their periodic behavior is determined by the single Scalable TCP flow. On the other hand, the performance of the BIC TCP aggregate is better ("non-Reno mode"); however, the dominance of the Scalable TCP flow is significant.

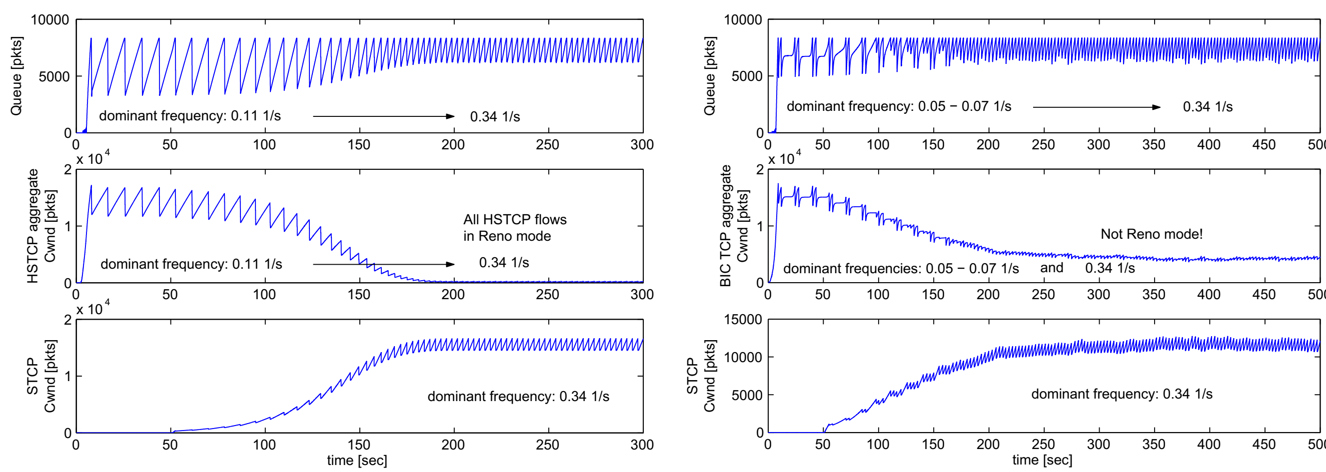
7.1.3. Simple parking-lot topology

Finally, the interaction of the MIMD and AIMD-like protocols is analyzed in a simple parking-lot topology (with one congested link). In these scenarios, the impacts of the different round-trip times can be revealed. The simulation results of the scenarios when the Scalable TCP flow possesses the shorter RTT are presented in Fig. 12(a and b) for HSTCP and BIC TCP, respectively. As it can be expected, Scalable TCP with the shorter RTT always starves the other flow. Starting time has an impact on the transient characteristics, while the equilibrium behavior is similar in different cases, i.e., HSTCP and BIC TCP operate in Reno mode achieving very low rate. In these scenarios, the network behavior is determined by the Scalable TCP flow.



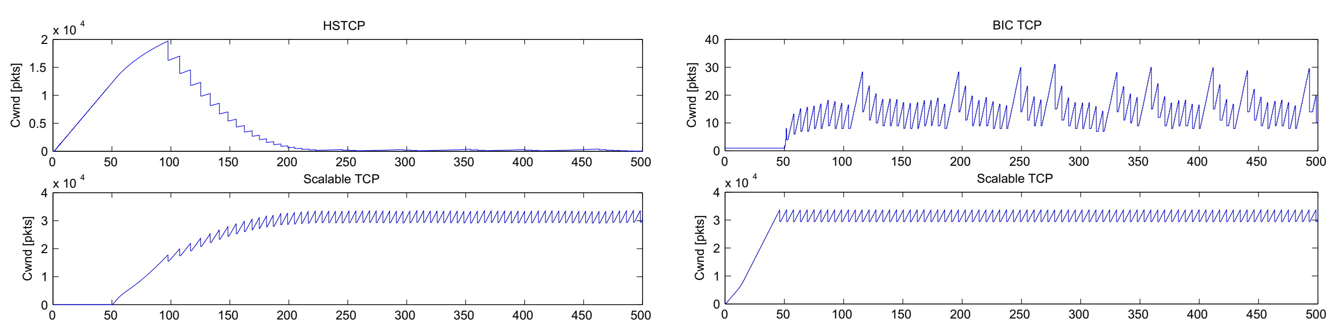
(a) STCP aggregate – HSTCP (b) STCP aggregate – BIC TCP

Fig. 10. Performance of individual flow vs. STCP traffic aggregate.



(a) HSTCP aggregate – STCP (b) BIC TCP aggregate – STCP

Fig. 11. Performance of individual STCP flow vs. traffic aggregate.



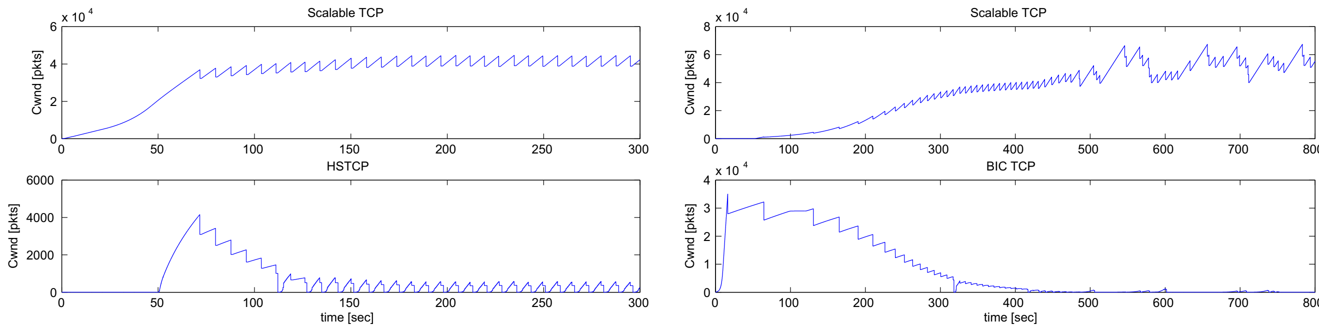
(a) HSTCP on the longer path starts first (b) BIC TCP on the longer path starts later

Fig. 12. Simple parking-lot topology: Scalable TCP on the shorter path.

An important property of Scalable TCP arises when it traverses the longer path. Illustrative results are shown in Fig. 13. The HSTCP and BIC TCP flows are also starved as a result of the aggressive nature of the MIMD algorithm; however, the performance is slightly better than in Reno mode.

7.1.4. Results

Our main findings are the following. The MIMD mechanism cannot guarantee the fair behavior with AIMD-like protocols using adaptive additive increase algorithm. Because of the aggressive and static characteristics of the multiplicative increase control of



(a) HSTCP on the shorter path starts later

(b) BIC TCP on the shorter path starts first

Fig. 13. Simple parking-lot topology: Scalable TCP on the longer path.

Scalable TCP, it starves other flows with AIMD protocols in a wide range of network environments. The aggressive nature of the MIMD mechanism is exhibited in the parking-lot topology and in the competition against traffic aggregates, as well. We found that the long-term interaction of these protocols is not affected by the starting delay. Starting delay has an impact on the transient phase and the convergence time.

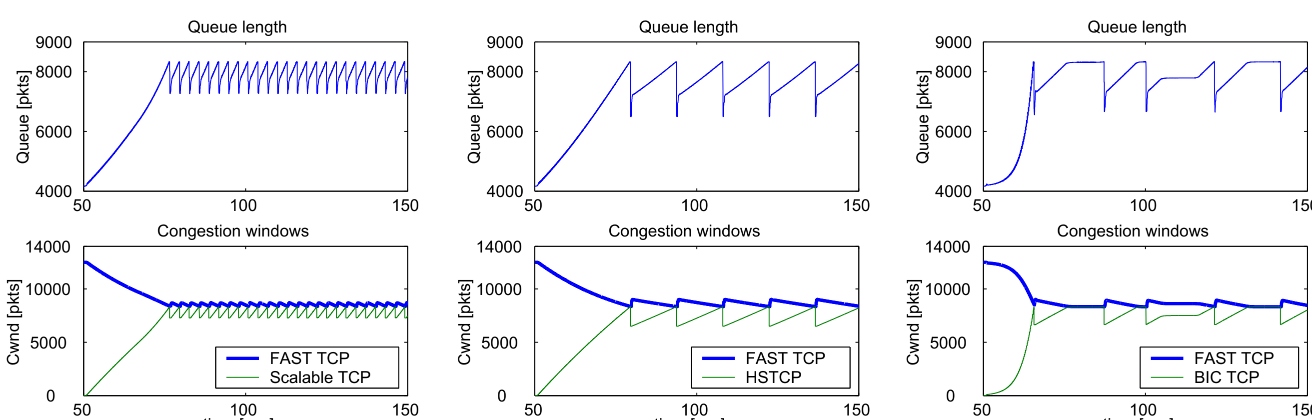
7.2. FAST TCP – loss-based protocols

In this section, we illustrate and show some surprising benefits of a promising delay-based protocol in terms of fairness. The interaction of FAST TCP and loss-based congestion control mechanisms is investigated in a diverse set of network scenarios.

7.2.1. Dumb-bell topology: single flows

First, we focus on the competition of single flows in the dumb-bell topology. In our first scenarios, FAST TCP source starts the transmission and the other flow – using loss-based protocol – enters the network when the first one has achieved its maximal sending rate. The simulation results corresponding to a starting delay of 50 s are presented in Fig. 14 for Scalable TCP (a), HSTCP (b) and BIC TCP (c). During the transient phase, the delay-based decrease algorithm of FAST TCP (roughly an additive decrease mechanism) interacts with the increasing control mechanism of the other protocol (Limited Slow-Start and multiplicative increase in case of Scalable TCP, Limited Slow-Start in case of HSTCP, and multiplicative increase in case of BIC TCP). As a consequence of the adequate parameter setting of FAST TCP, the congestion window processes converge to a fair equilibrium state. The convergence time is determined by the loss-based protocol. After the transient period, the two flows show periodic behavior. *The spectral behavior of FAST TCP follows the periodic operation of the loss-based protocols.* For example, when Scalable TCP reduces its congestion window, FAST TCP can increase the number of packets in the bottleneck queue performing a delay-based increase that is approximately equivalent to an additive increase mechanism. During the second part of a period, the multiplicative increase of Scalable TCP interacts with the roughly additive decrease of FAST TCP. Thus, the periodic behavior is affected by the interaction of an "AIAD-like" and an MIMD algorithm. The common time period and the dynamics of the bottleneck queue are determined by the time period of Scalable TCP. It is worth noting that losses do not occur during the FAST TCP connection and the equilibrium state is quasi stable. The equilibrium behavior of FAST TCP and HSTCP is very similar. Here, the interaction of the "AIAD-like" and the AIMD mechanisms results in a longer time period. BIC TCP also exhibits longer time period during the equilibrium phase.

In the next scenarios, the FAST TCP source enters later into the network and try to catch the half of the capacity of the bottleneck link. The simulation results corresponding to 15 and 9 s delays are presented in Fig. 15(a–c). In these scenarios, after a very short transient period, the congestion windows settle down again around an equilibrium state. An important benefit of FAST TCP can be observed in the plots. The protocol with our (adequate) parameters can achieve significant bandwidth share against Scalable TCP, HSTCP and BIC TCP, too. Moreover, the equilibrium state is fair if the `baseRTT` estimation of the FAST mechanism is accurate. This

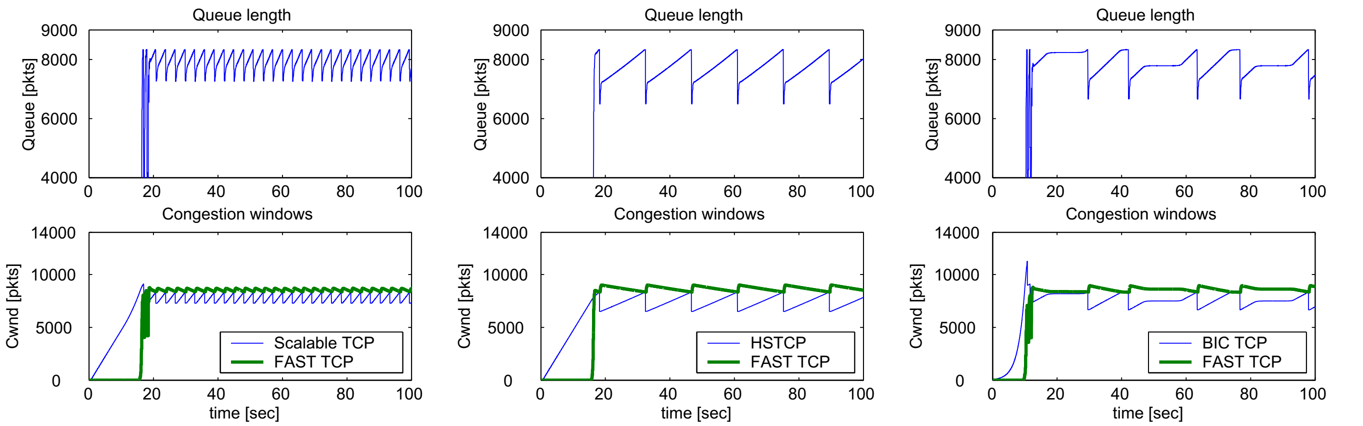


(a) FAST TCP – STCP

(b) FAST TCP – HSTCP

(c) FAST TCP – BIC TCP

Fig. 14. FAST TCP starts first, delay: 50 s.



(a) STCP – FAST TCP (b) HSTCP – FAST TCP (c) BIC TCP – FAST TCP (del: 9s)

Fig. 15. FAST TCP starts later, delay: 15, 15, 9 s.

estimation depends on the current state of the bottleneck buffer at the time of entering, i.e., exact estimation can be computed before the link saturation time (if the buffer is empty).

A significantly different behavior can be experienced by increasing the starting delay of the FAST TCP flow (to be greater than the saturation time of the other protocol). As an illustration, first, the simulation results of the competition of a Scalable TCP flow (MIMD mechanism) and a FAST TCP flow corresponding to 50 s delay are shown in Fig. 16. This behavior can be examined in the frequency-domain, too. The power spectral density (PSD) functions of the `cwnd` process and the bottleneck queue process are also shown in Fig. 16. The good performance of FAST TCP can be explained by the special control algorithm used by the protocol. When FAST TCP is far from the equilibrium sending rate, the window is increased aggressively in a roughly multiplicative manner. As the bottleneck queue operates around its full state, during the transient period, FAST TCP also suffers from losses and halves the `cwnd`. After a recovery period, the "MI-like" increase is performed until the next loss. After a long and oscillating transient phase, the common periodic equilibrium behavior previously seen is exhibited when FAST TCP does not suffer from losses. The dominant frequency of a single Scalable TCP flow ($\omega \approx 0.341/\mathrm{s}$) occurs in the PSD of FAST TCP (with lower energy value) as well, while the presence of a higher frequency component can also be observed corresponding to the oscillation of the transient phase. These two frequency spikes mainly determine the dynamics of the bottleneck queue. It is worth noting, that the equilibrium state is only near to the fair behavior and rather an *almost fair* bandwidth share is realized. This long-term bias is caused by the overestimation of `baseRTT` which originates from the fact that FAST TCP does not meet empty buffer. However, this slight bias in fairness can be compensated by the tuning of the α parameter of FAST TCP (see Section 7.2.5).

The interaction with AIMD-like protocols possesses similar characteristics. As an illustration, the simulation results of BIC TCP are presented in Fig. 17. Here, the transient period is shorter and FAST TCP follows the spectral behavior of BIC TCP at the equilibrium state. We observed that the length of the transient period

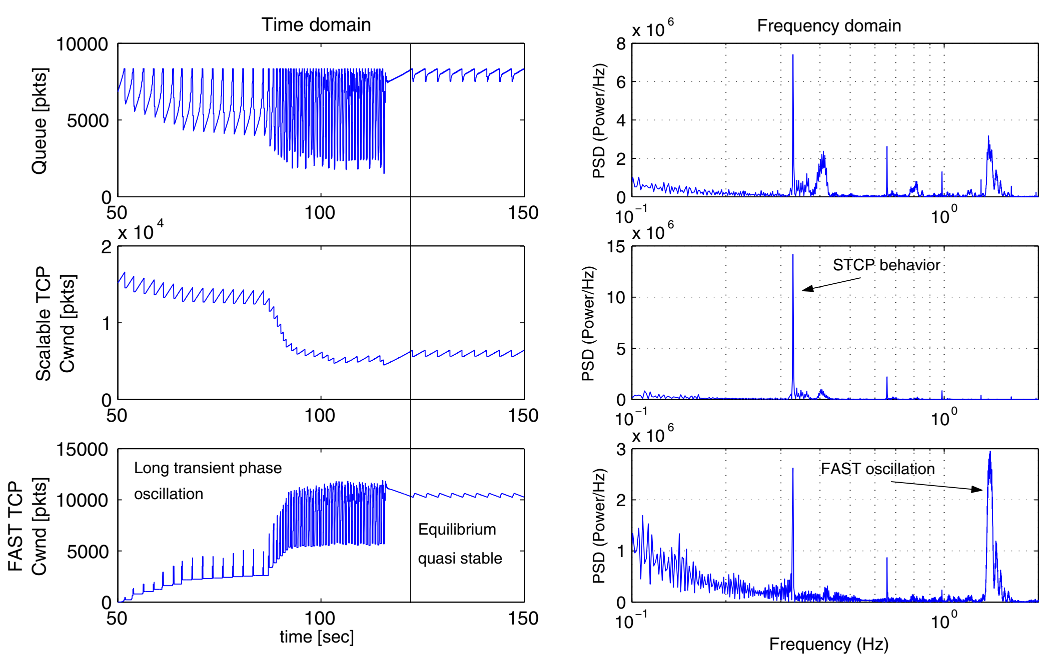


Fig. 16. STCP–FAST TCP, delay: 50 s.

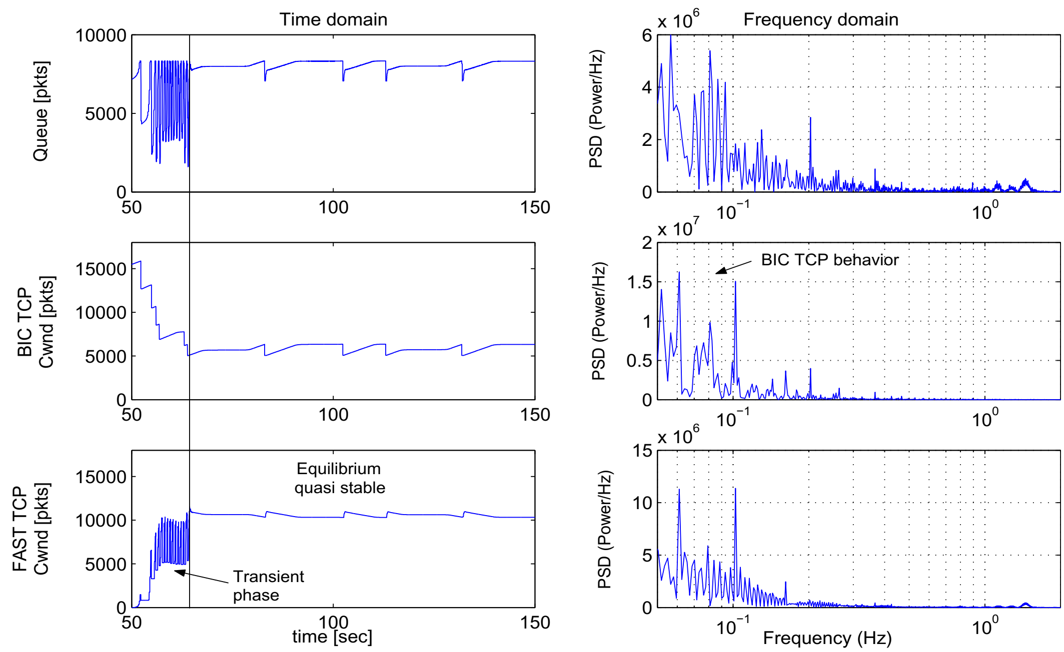


Fig. 17. BIC TCP–FAST TCP, delay: 50 s.

depends on the starting time and other parameters of FAST TCP, as well. In all cases, the almost fair behavior is exhibited.

7.2.2. Dumb-bell topology: traffic aggregate – single flow

In this section, a more realistic situation is analyzed. More specifically, a FAST TCP flow enters the network (simple dumb-bell topology) where a traffic aggregate of a loss-based protocol has fully utilized the bottleneck link.

First, the interoperation with AIMD-like flows is presented. As a demonstrative example, the simulation results of competing HSTCP aggregate and a single FAST TCP flow are shown in Fig. 18. In this scenario, the aggregate contains 10 HSTCP flows and the FAST TCP source enters with a delay of 50 s. The FAST TCP flow is able to occupy half of the bottleneck bandwidth beside HSTCP flows (if the parameters are well chosen), and the behavior is similar to the behavior of two competing flows: FAST TCP realizes a quasi-stable equilibrium state without losses. The competition with BIC TCP aggregate is also similar to the interoperation of single flows.

Second, we focus on the performance of FAST TCP beside a traffic aggregate of MIMD flows. In contrast to AIMD-like protocols, the interaction of FAST TCP with Scalable TCP aggregate shows long-term oscillation. An example is given in Fig. 19. (The scenario is similar to the previous one.) Here, the characteristics of the

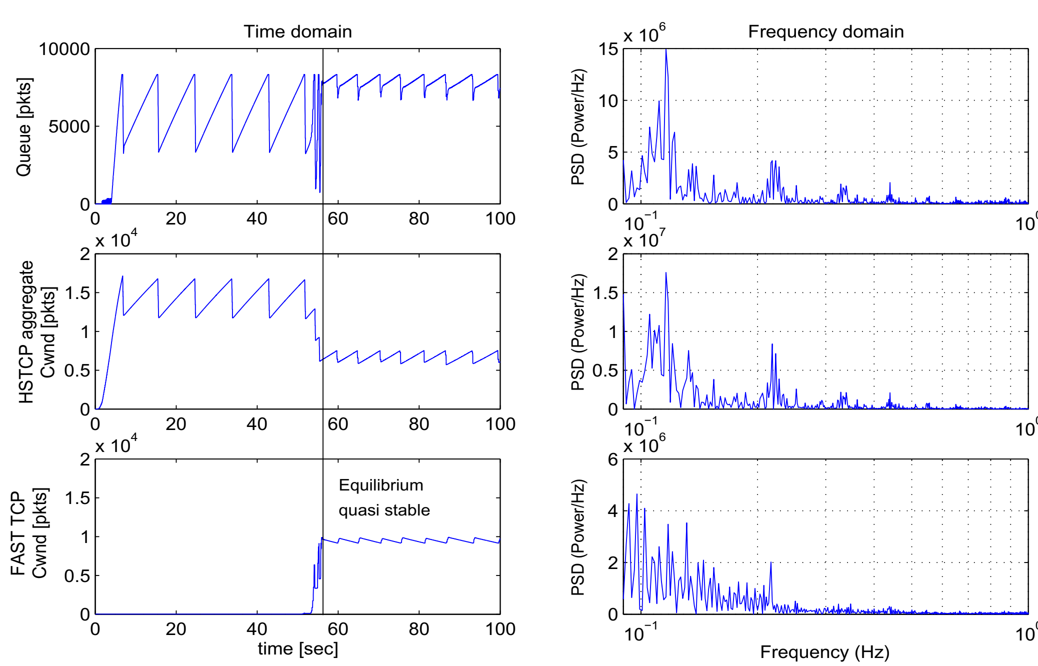


Fig. 18. HSTCP aggregate–FAST TCP.

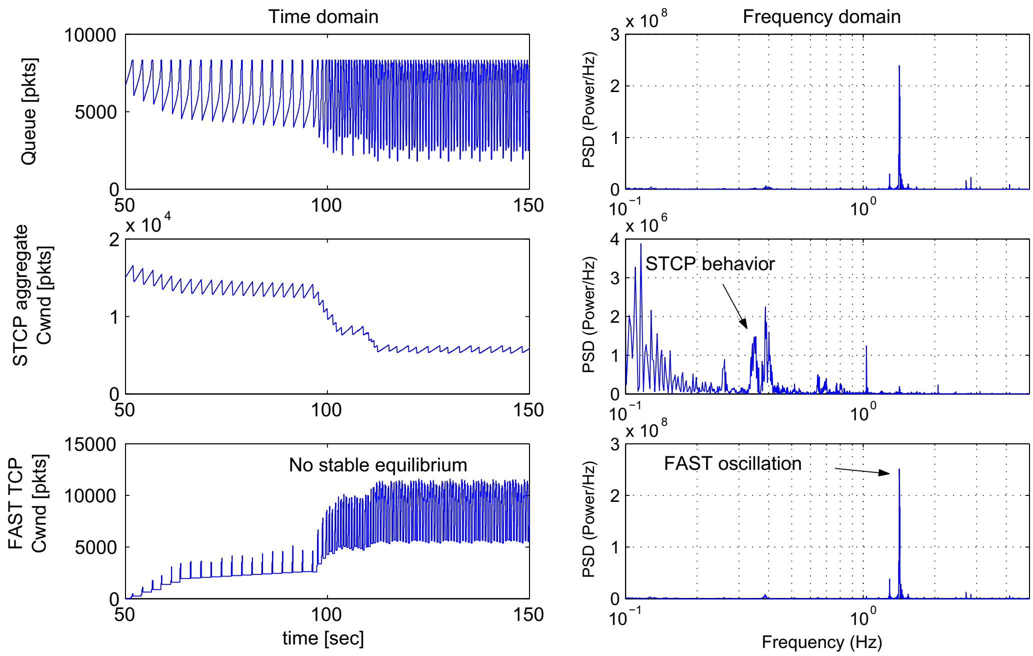


Fig. 19. STCP aggregate–FAST TCP.

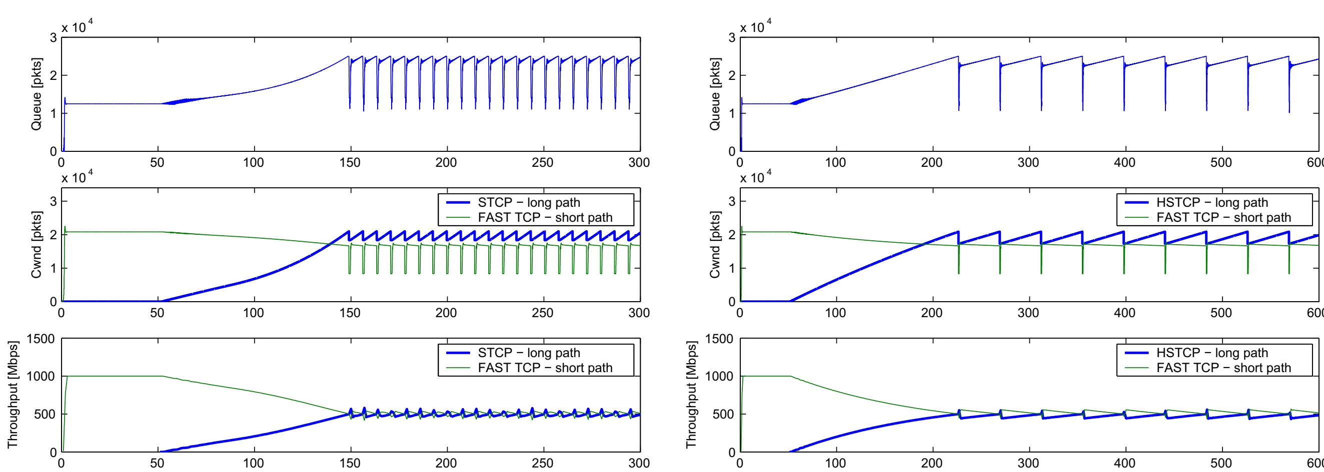
equilibrium behavior is determined by the interaction of the MIMD mechanism of Scalable TCP and the "MIMD-like", oscillating operation of FAST TCP. In this network environment, FAST TCP is not able to reach a stable (or quasi stable) state and suffers from losses. The spectral characteristics of the bottleneck queue process is mainly determined by this "FAST oscillation" as it can be inferred from the spectrum plots. Despite the unstable operation, the achieved throughput of the FAST TCP flow approximates the half of the bottleneck capacity (as it is targeted by the parameters) resulting in good performance. Nevertheless, this oscillation can be alleviated by increasing the congestion window update period of FAST TCP resulting in a less responsive behavior.

We found that FAST TCP can always achieve good performance and fair (or almost fair) behavior against traffic aggregate of loss-based protocols; however, the long-term behavior depends on the other protocol, as well.

7.2.3. Simple parking-lot topology

Our next step is the examination of the impact of different round-trip times on fairness. Here, we show some demonstrative results.

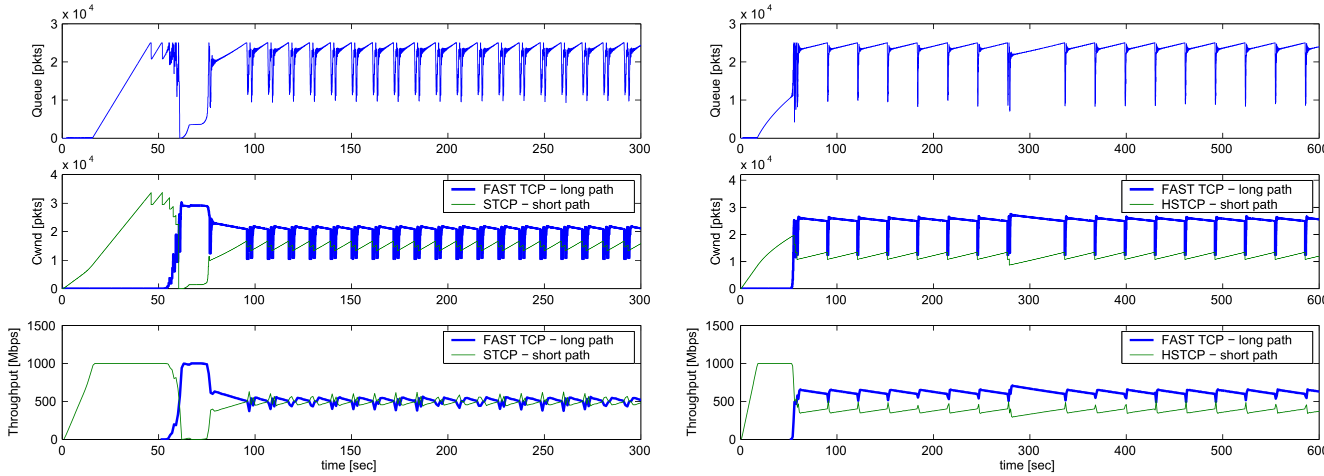
In the first scenarios, the FAST TCP flow traverses the shorter path. As an illustration, the simulation results are shown in Fig. 20 for later entering Scalable TCP and HSTCP flow. Beside the dynamics of the bottleneck queue and the congestion windows, the figure includes the throughput diagrams, too. Similarly to the competition in the dumb-bell topology, fair or almost fair behavior is always exhibited. The FAST TCP flow with shorter RTT does not starve the loss-based protocols with longer paths. FAST TCP reduces its sending rate till a fair equilibrium bandwidth share is realized. The only difference in the long-term behavior is the slight oscillation of the `cwnd` process of FAST TCP indicating packet losses.



(a) Scalable TCP starts later

(b) HSTCP starts later

Fig. 20. Simple parking-lot topology: loss-based protocol with longer RTT - FAST TCP.



(a) Scalable TCP starts first

(b) HSTCP starts first

Fig. 21. Simple parking-lot topology: FAST TCP with longer RTT – loss-based protocol.

In the next scenarios, FAST TCP flow operates with longer round-trip times. The illustrative results are presented in Fig. 21. Here, FAST TCP enters into the network 50 s later than Scalable TCP and HSTCP. It can be observed that the FAST TCP flow can always achieve fair or almost fair equilibrium state. In certain cases, FAST TCP outperforms the loss-based protocol which can be explained by the estimation error of `baseRTT` (FAST TCP flow does not experience empty buffer). If the starting phase of the FAST TCP causes timeout to the loss-based flow (see Fig. 21(a)), then the `baseRTT` estimation is accurate (because of the empty buffer).

We found that the fair behavior of FAST TCP still holds in simple parking-lot topology with one congested link.

7.2.4. Multiple congested links

In realistic network environments, more than one link can be congested across the route of a connection. In this section, more exciting scenarios are investigated where the topologies contain multiple bottlenecks. More exactly, the performance in the inhomogeneous parking-lot and the complex parking-lot topologies is discussed.

FAST TCP shows promising fairness characteristics beside loss-based flows in the inhomogeneous parking-lot topology, however, the interaction is qualitatively different for various loss-based mechanisms. Our default parameter setting for FAST TCP aims at occupying the half buffer capacity along the network route. First, the fairness performance against MIMD flows is presented. In Fig. 22, the congestion window process of a FAST TCP flow traversing across two congested links and the STCP flows operating on separate links are shown both in the time and the frequency domains (for the details of the topology, see Fig. 3(c)). The round-trip propagation time of the first STCP flow is two times greater than the other one. In these scenarios, the spectral analysis provides a good tool to get a deeper understanding of the performance. The Scalable TCP flows show a significant frequency spike as it is

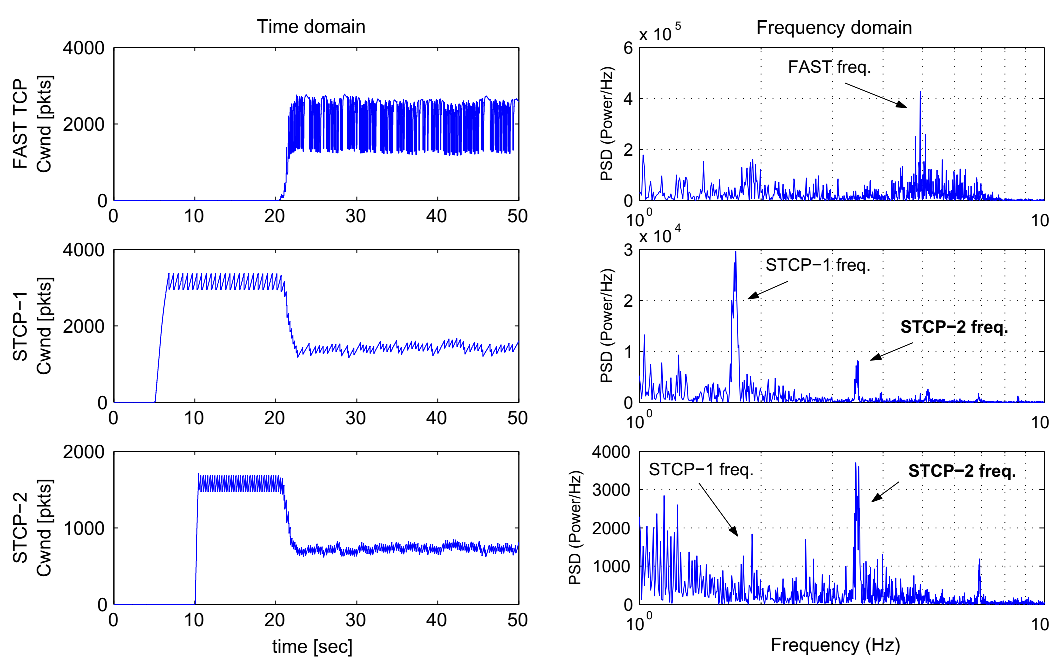


Fig. 22. Inhomogeneous parking-lot topology: FAST TCP–STCP.

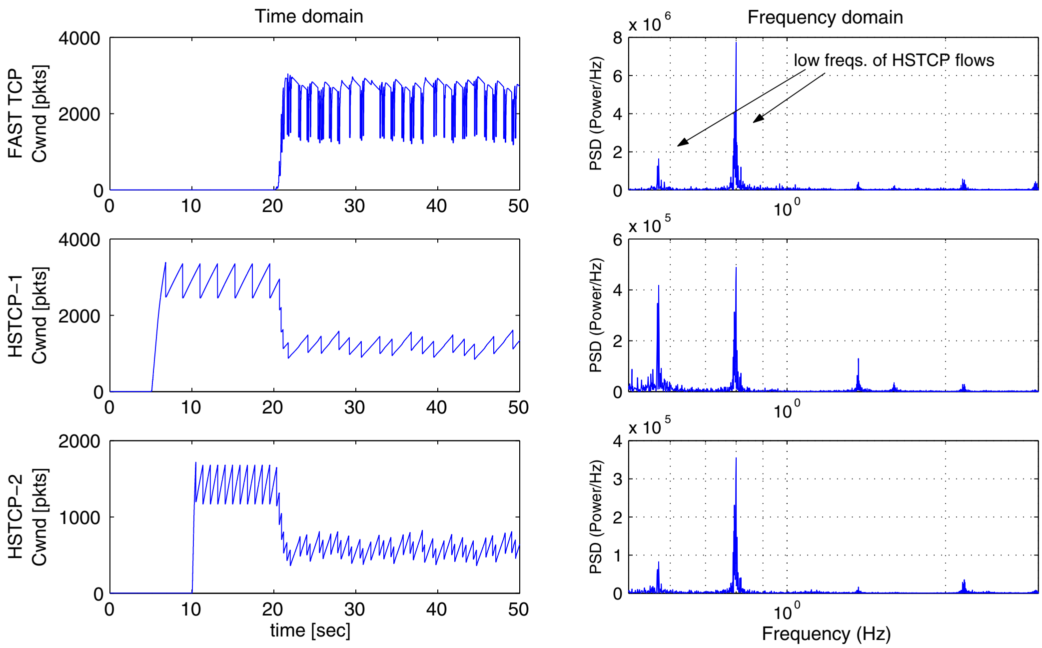


Fig. 23. Inhomogeneous parking-lot topology: FAST TCP–HSTCP.

shown in the figure. But surprisingly, the main frequency of the other flow is also appearing with lower energy values (see the marked spikes, STCP-1 freq. and STCP-2 freq., in the plots). This synchronization between the flows on *separate links* is propagated by the FAST TCP flow that experiences the overflow of both buffers along its route (as packet losses). This surprising phenomenon means that the dynamic characteristics of different links can be propagated along the network routes. Here, the FAST TCP behavior is determined by its own oscillation frequency which is significantly higher than the frequency of the Scalable TCP flow on the shorter path. This oscillation also means high loss rate for the FAST flow. Despite the unstable equilibrium operation of FAST TCP, the long-term bandwidth share is fair as it is shown in the bottom part (default parameters) of Fig. 26(b).

Second, the fairness performance against AIMD-like flows is illustrated by the example of HSTCP. In Fig. 23, the interaction of FAST TCP and two HSTCP flows is shown in a similar scenario. The main difference can be observed in the operation of FAST TCP. Here, the long-term behavior of the delay-based protocol is mainly determined by the periodic behavior of the two HSTCP flows, and the FAST oscillation is not considerable. Thus, the loss rate experienced by the FAST flow is much lower than previously. Moreover, the equilibrium bandwidth share is also fair.

The frequency propagation effect can be presented well in a similar scenario when the FAST TCP flow competes with a HSTCP and a Scalable TCP flow, respectively. Here, the HSTCP flow operates on the first link with larger propagation delay, and the STCP flow is used on the other one. In Fig. 24, a typical section of the

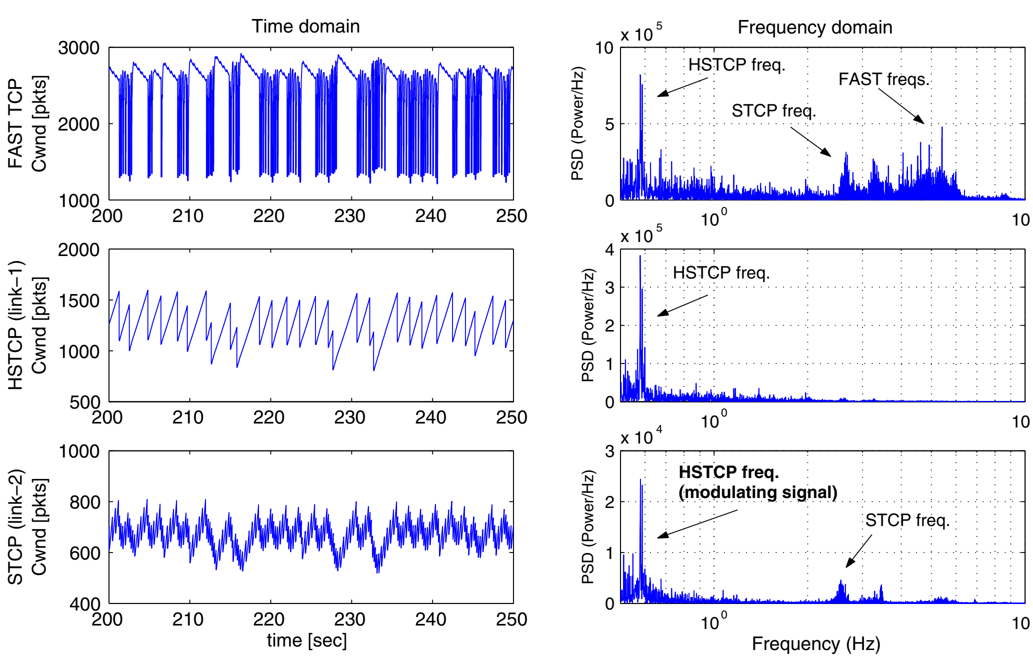


Fig. 24. Inhomogeneous parking-lot topology: FAST TCP–HSTCP, STCP.

equilibrium operation is shown in the time-domain and the corresponding spectrum plots are also given. HSTCP operating at a lower frequency modulates the congestion window process of Scalable TCP which exhibits higher frequency. This phenomenon is similar to the amplitude modulation: the first flow corresponds to the modulating signal while the second one acts as the carrier. In this scenario, the periodic behavior of FAST TCP is affected by the loss-based flows and its own oscillation frequency also appears; while the fair average bandwidth share is achieved, too.

Surprisingly, the performance of FAST TCP shows degradation in the complex parking-lot topology with more congested links. Here, the α and β parameters of FAST TCP are set to the same value as it was used in the dumb-bell scenarios.

First, the interaction of a single FAST TCP traversing the backbone (four congested links) and four AIMD-like (here HSTCP) flows using separate links is shown in Fig. 25(a). The congestion window of FAST TCP can settle down around the same equilibrium state where the other flows operate. Therefore, the bandwidth share of FAST TCP is significantly below the fair state. The equilibrium behavior slightly depends on the starting time of the flows, as well. The characteristics of the competition are similar for BIC TCP flows (for further details, see [2]).

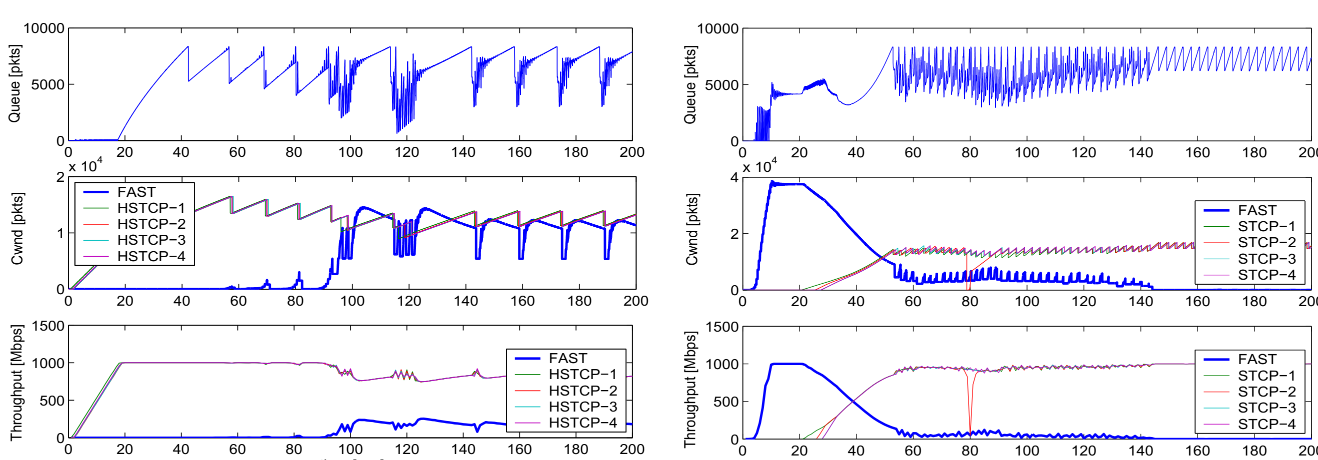
Second, the interaction with MIMD flows is analyzed, and an illustration is given in Fig. 25(b). In contrast to AIMD-like flows, Scalable TCP flows always starve the FAST TCP flow traversing the long path. When the Scalable TCP sources start the transmission, then the later entering FAST TCP cannot achieve significant throughput. In the other case – which is shown in the figure, when the Scalable TCP flows enter the network, an unfair state is realized during a transient phase (with low bandwidth share of FAST TCP) and later at the equilibrium state the starvation of FAST TCP can be observed. Obviously, the performance of FAST TCP can be enhanced increasing the α and β parameters of the protocol. However, this can yield unstable network behavior with degraded link utilization.

7.2.5. Impacts of FAST parameters on fairness performance

The fairness performance of FAST TCP is affected by several parameters that should be chosen well in order to get the presented beneficial properties. Thus, to find the adequate parameter settings for a wide range of network conditions or give an adaptive algorithm to adjust the parameters is crucial. The research on parameter settings of FAST TCP is addressed in several papers (see e.g., [8,53,21]). In this section, the impacts of some parameters of FAST TCP is discussed.

From the perspective of fairness, the most important parameter is α (and β) determining the targeted share of buffer capacity and queueing delay. The α parameter has a direct relation to the equilibrium bandwidth share and fairness performance in most cases. Moreover, the characteristics of the long-term behavior (stability or oscillation) can also be affected by this parameter. On the one hand, in case of co-existence of FAST TCP and AIMD-like flows, tuning α parameter can control the fairness performance in a wide range of network scenarios. For example, the slight bias that was observed in certain situations (which is caused by the `baseRTT` estimation error) can be compensated by reducing α. On the other hand, the incompatibility of FAST TCP and Scalable TCP cannot always be alleviated by α tuning. Especially in case of multiple bottlenecks or multiple MIMD flows, the performance degrades due to the high frequency oscillation of FAST TCP yielding very high loss rate. The unstable long-term behavior is the result of the interaction of two aggressive control mechanisms that cannot be compensated by reducing α. However, the average throughput (but not goodput) can be tuned by this parameter. In Fig. 26(a), a simple example is given when the α parameter tuning cannot alleviate the starvation of FAST TCP. Here, the topology is the complex parking-lot where the FAST TCP flow competes with four STCP flows. The first scenario is similar to the previously presented case (see Fig. 25(b)), but now the FAST TCP enters the network when the others have achieved their maximal sending rates. The serious starvation of FAST TCP can be observed, and the increase of the α parameter cannot solve the problem here. (α = 16,000 corresponds to the half of the entire buffer capacity along the network path.)

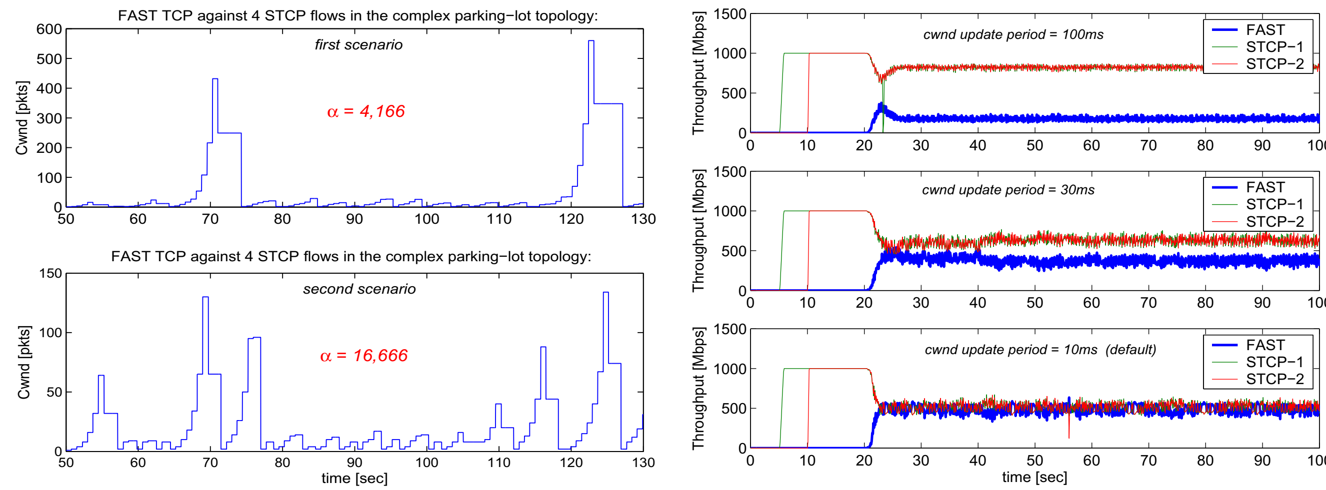
Another important FAST parameter is the congestion window update period determining the time of window adjustments. This parameter can significantly affect the long-term dynamic characteristics, and in some scenarios, the unstable behavior is rooted in the inadequate setting of that. Furthermore, the update period can have impacts on the equilibrium bandwidth share, as well. As an example, results of the STCP–FAST TCP interaction in the inhomogeneous parking-lot topology are presented belonging to three different congestion window update periods. Namely, 100 ms (above the RTT), 30 ms (around the RTT), and 10 ms (below the RTT) time constants are investigated and shown in Fig. 26. The default value in the Ns-2 implementation is 10 ms (in [8], 20 ms is suggested). The throughput plots well indicate that the long-term fairness is considerably affected by the update period. Moreover, we found that the oscillation frequency of FAST TCP increases when this time constant is reduced. A reasonable choice for the



(a) FAST TCP – HSTCP, FAST starts later

(b) FAST TCP – Scalable TCP, FAST starts first

Fig. 25. Complex parking-lot topology: 5 nodes, FAST TCP – loss-based protocols.



(a) FAST TCP against STCP flows in the complex parking-lot topology (b) Impacts of cwnd update period: FAST TCP – STCP

Fig. 26. Impacts of FAST parameters on fairness performance.

update period is below the round-trip time of the FAST flow, while the too small values result in unnecessary oscillation.

Other parameters – such as γ, *mi_threshold* or `baseRTT` estimation – are also able to affect the fairness performance. For example, γ and *mi_threshold* concerns the responsiveness of the protocol while the accuracy of the `baseRTT` estimation affects the equilibrium state.

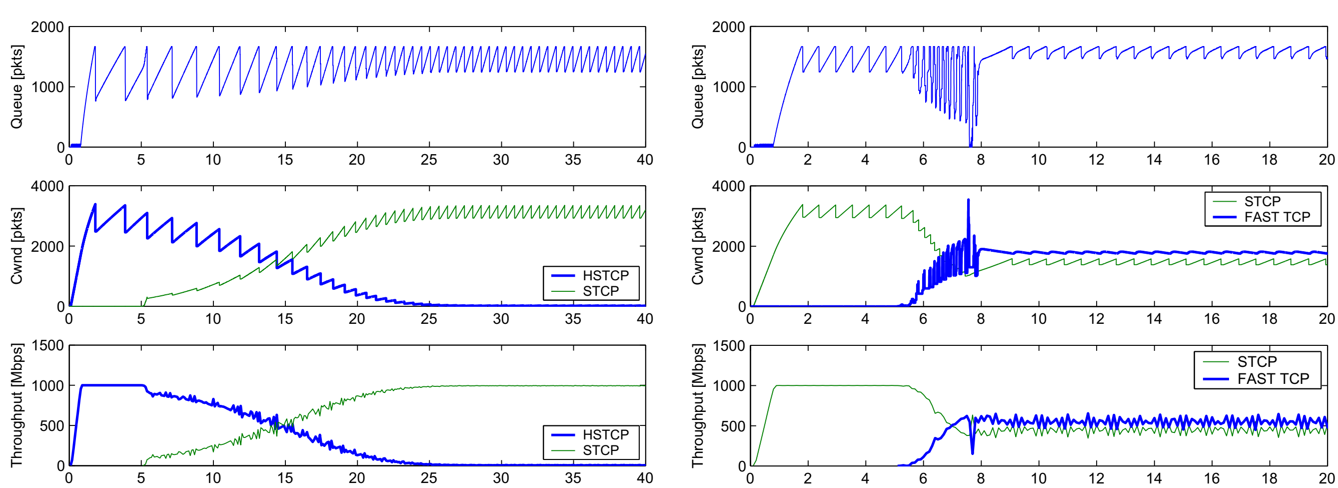
7.2.6. Results

Our main findings are as follows. *In contrast to loss-based protocols, FAST TCP with appropriate parameters can always show fair or almost fair behavior beside loss-based protocols in simple network environments with single congested link.* On the one hand, the long-term behavior of FAST TCP is similar in different scenarios with quasi stable equilibrium state. On the other hand, the transient characteristics are affected by the starting delay of the flows and the other protocol, as well. *We have also found that this fair behavior of FAST TCP seems to be a robust property of the protocol and FAST TCP can achieve good utilization against traffic aggregate of loss-based protocols. The beneficial fairness properties of FAST TCP still holds for networks containing more than one congested links (appropriate parameters can be chosen), however, in general, the increasing number of bottleneck links may result in performance degradation.*

We should also note that the performance and the fairness properties are significantly affected by FAST parameters. This parameter-sensitive property of the protocol yields the importance of choosing adequate parameters. In this paper, we always set α parameter to get an operation when the FAST TCP flow occupies the half of the buffer. With other parameters, FAST TCP can achieve unfair states or can be starved even in simple scenarios. The beneficial properties of the protocol holds for a certain range of parameter α depending on the actual network topology, flow parameters, etc. To find a method which can continuously change this parameter according to the network and flow environments to keep this property broadly general is a good point of future research.

8. Discussion

As FAST TCP and other high speed protocols are proposed for networks characterized by high bandwidth-delay product, we focused on mainly high BDP networks during our investigation. However, the results are more general and hold in network environment with smaller BDP, too. As an illustration, the competition



(a) HSTCP – Scalable TCP (b) Scalable TCP – FAST TCP

Fig. 27. Impacts of lower link delay.

of Scalable TCP and HSTCP, and the interaction of Scalable TCP and FAST TCP in a simple dumb-bell topology with small link propagation delay are presented. Here, the link delay is 10 ms. In Fig. 27, it can be observed that the qualitative characteristics are not changed; only the quantitative properties are different, and the whole network operates at a higher frequency than it was experienced previously (see Figs. 9 and 15).

9. Conclusion

In this paper, we have presented a comprehensive fairness performance evaluation analysis of different high speed TCP versions. The study includes an overall analysis including flow-level, packet-level, queueing and spectral analysis with both intra- and inter-protocol characteristics in different topologies and parameter settings. The analysis also includes a root-cause analysis of the starting time impact on competing high speed TCP flows. We have proved that the short-term dynamic characteristics of the protocols can have a major impact on long-term fairness.

Our study has emphasized the important need for finding a dynamic sensitive fairness metric for performance evaluation of transport protocols for next generation high bandwidth-delay product networks. As a step to this direction we have proposed a new metric called the saturation time. By the help of this metric we can quantify the impact of the starting time on long-term fairness.

We have derived analytical results on relevant parameters for HighSpeed TCP and Scalable TCP in different scenarios, including both inter-protocol and intra-protocol settings. We have analyzed and explained the "starving" effect of competing high speed TCP flow, when a flow force other flows from their proper operation. We demonstrate that FAST TCP with proper parameter settings can achieve fair behavior with HighSpeed TCP, Scalable TCP and BIC TCP. We have also shown that this behavior is rather robust property of the protocol concerning different traffic mix and wide range of network topologies. However, we have shown that in case of a later entering of FAST TCP flow the equilibrium state is reached through an oscillating transient phase with a length depending on the starting time and other parameters.

Our future plan addresses the continuation of this research in line with the related IETF activity with creating a general performance evaluation framework to analyze and compare promising TCP proposals.

Acknowledgements

This work was supported by the Inter-University Center for Telecommunications and Informatics (ETIK). The first author was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1] M. Welzl, Network Congestion Control: Managing Internet Traffic, John Wiley & Sons, Ltd., 2005.

[2] B. Sonkoly, T.A. Trinh, S. Molnár, Benchmarking high speed TCP fairness, Tech. rep., Budapest University of Technology and Economics, BME (Oct. 2007). Available from: URL <http://hsnlab.tmit.bme.hu/projects/tcp/>.

[3] T.A. Trinh, B. Sonkoly, S. Molnár, Revisiting FAST TCP fairness, in: Proceedings of 18th ITC Specialist Seminar on Quality of Experience, Karlskrona, Sweden, May 29–30, 2008.

[4] B. Sonkoly, T.A. Trinh, S. Molnár, Understanding the impacts of short-term dynamics on long-term fairness of competing high speed TCP flows: a root-cause analysis, in: Proceedings of NETWORKS 2008, Budapest, Hungary, Sep. 28–Oct. 2, 2008.

[5] Y. Li, D. Leith, R.N. Shorten, Experimental evaluation of TCP protocols for high-speed networks, IEEE/ACM Transactions on Networking (ToN) 15 (5) (2007) 1109–1122.

[6] G.S. Lee, L.L.H. Andrew, A. Tang, S.H. Low, WAN-in-Lab: Motivation, deployment and experiments, in: Proceedings of Fifth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2007), Marina Del Rey, CA, USA, Feb. 7–9, 2007.

[7] D.X. Wei, P. Cao, S.H. Low, Time for a TCP benchmark suite? Online (Aug. 2005). Available from: URL <http://www.cs.caltech.edu/weixl/research/technicals/benchmark/summary.ps>.

[8] C. Jin, D.X. Wei, S.H. Low, FAST TCP: motivation, architecture, algorithms, performance, in: Proceedings of IEEE Infocom 2004, vol. 4, Hong Kong, China, Mar. 7–11, 2004, pp. 2490–2501.

[9] D.X. Wei, C. Jin, S.H. Low, S. Hegde, FAST TCP: motivation, architecture, algorithms, performance, IEEE/ACM Transactions on Networking (ToN) 14 (6) (2006) 1246–1259.

[10] H. Bullot, R.L. Cottrell, R. Hughes-Jones, Evaluation of advanced TCP stacks on fast long-distance production networks, Journal of Grid Computing 1 (4) (2003) 345–359.

[11] S. Molnár, E. Altman (Guest Eds.), Telecommunication Systems, special issue on High Speed Transport Protocols, vol. 30 (4).

[12] S. Floyd, Highspeed TCP for large congestion window, IETF RFC 3649 (Dec. 2003).

[13] T. Kelly, Scalable TCP: improving performance in highspeed wide area networks, ACM SIGCOMM Computer Communication Review 33 (2) (2003) 83–91.

[14] L. Xu, K. Harfoush, I. Rhee, Binary increase congestion control (BIC) for fast long-distance networks, in: Proceedings of IEEE Infocom 2004, vol. 4, Hong Kong, China, Mar. 7–11, 2004, pp. 2514–2524.

[15] I. Rhee, L. Xu, CUBIC: a new TCP-friendly high-speed TCP variant, in: Proceedings of Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2005), Lyon, France, Feb. 3–4, 2005.

[16] G. Marfia, C.E. Palazzi, G. Pau, M. Gerla, M.Y. Sanadidi, M. Roccetti, TCP-Libra: exploring RTT fairness for TCP, Tech. Rep. UCLA-CSD TR-050037, UCLA Computer Science Department (2005). Available from: URL <http://www.cs.ucla.edu/cpalazzi/papers/TCP-Libra-UCLA-CSD-TR050037.pdf>.

[17] D. Leith, R. Shorten, H-TCP protocol for high-speed long distance networks, in: Proceedings of Second International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2004), Argonne, IL, USA, Feb. 16–17, 2004.

[18] R. Wang, K. Yamada, M.Y. Sanadidi, M. Gerla, TCP with sender-side intelligence to handle dynamic, large, leaky pipes, IEEE Journal on Selected Areas in Communications 23 (2) (2005) 235–248.

[19] S. Bhandarkar, S. Jain, A.L.N. Reddy, Improving TCP performance in high bandwidth high RTT links using layered congestion control, in: Proceedings of Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2005), Lyon, France, Feb. 3–4, 2005.

[20] S.H. Low, L. Peterson, L. Wang, Understanding vegas: a duality model, Journal of ACM 49 (2) (2002) 207–235.

[21] S. Molnár, S. Kardos, T.A. Trinh, Parameter setting games in TCP Vegas and FAST TCP, in: Proceedings of 19th International Teletraffic Congress, ITC19, Beijing, China, Aug. 29–Sep. 2, 2005.

[22] R. King, R. Riedi, R. Baraniuk, Evaluating and improving TCP-Africa: an adaptive and fair rapid increase rule for scalable TCP, in: Proceedings of Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2005), Lyon, France, Feb. 3–4, 2005.

[23] F. Vacirca, A. Baiocchi, A. Castellani, YeAH-TCP: yet another highspeed TCP, in: Proceedings of Fifth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2007), Marina Del Rey, CA, USA, Feb. 7–9, 2007.

[24] S. Liu, T. Başar, R. Srikant, TCP-Illinois: a loss and delay-based congestion control algorithm for high-speed networks, in: Proceedings of First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS), Pisa, Italy, Oct. 11–13, 2006.

[25] H. Shimonishi, T. Hama, T. Murase, TCP-Adaptive Reno: improving efficiency-friendliness tradeoffs of TCP congestion control algorithm, in: Proceedings of Fourth International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2006), Nara, Japan, Feb. 2–3, 2006.

[26] K. Tan, J. Song, Q. Zhang, M. Sridharan, A compound TCP approach for high-speed and long distance networks, in: Proceedings of IEEE Infocom 2006, Barcelona, Spain, Apr. 23–29, 2006.

[27] O. Tickoo, V. Subramanian, S. Kalyanaraman, K.K. Ramakrishnan, LT-TCP: end-to-end framework to improve TCP performance over networks with lossy channels, in: Proceedings of 13th International Workshop on Quality of Service (IWQoS 2005), Passau, Germany, Jun. 21–23, 2005.

[28] D. Katabi, M. Handley, C. Rohrs, Congestion control for high bandwidth-delay product networks, in: Proceedings of ACM SIGCOMM 2002, Aug. 19–23, Pittsburgh, PA, USA, 2002.

[29] K. Ramakrishnan, S. Floyd, D. Black, The addition of explicit congestion notification (ECN) to IP, IETF RFC 3168 (Sep. 2001).

[30] N. Dukkipati, M. Kobayashi, R. Zhang-shen, N. McKeown, Processor sharing flows in the internet, in: Proceedings of 13th International Workshop on Quality of Service (IWQoS), Passau, Germany, Jun. 21–23, 2005.

[31] N. Dukkipati, N. McKeown, Why flow-completion time is the right metric for congestion control, ACM SIGCOMM Computer Communication Review 36 (1) (2006) 59–62.

[32] E. Souza, D. Agarwal, A highspeed TCP study: characteristics and deployment issues, Tech. Rep. LBNL-53215, Lawrence Berkeley National Lab (2003). Available from: URL <http://www-itg.lbl.gov/evandro/hstcp/hstcp-lbnl-53215.pdf>.

[33] T.A. Trinh, B. Sonkoly, S. Molnár, A HighSpeed TCP study: observations and re-evaluation, in: Proceedings of 10th Eunice Summer School and IFIP Workshop on Advances in Fixed and Mobile Networks, Tampere, Finland, Jun. 14–16, 2004.
[34] D.J. Leith, L.L.H. Andrew, T. Quetchenbach, R.N. Shorten, K. Lavi, Experimental evaluation of delay/loss-based TCP congestion control algorithms, in: Proceedings of Sixth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2008), Manchester, UK, Mar. 5–7, 2008.
[35] E. Altman, K.E. Avrachenkov, B.J. Prabhu, Fairness in MIMD congestion control algorithms, in: Proceedings of IEEE Infocom 2005, vol. 2, Miami, USA, Mar. 13–17, 2005, pp. 1350–1361.
[36] S. Ha, Y. Kim, L. Le, I. Rhee, L. Xu, A step toward realistic performance evaluation of high-speed TCP variants, in: Proceedings of Fourth International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet 2006), Nara, Japan, Feb. 2–3, 2006.
[37] M.C. Weigle, P. Sharma, J. Freeman, Performance of competing high-speed TCP flows, in: Proceedings of IFIP Networking 2006, Coimbra, Portugal, May 15–19, 2006.
[38] T.A. Trinh, S. Molnár, Quality of Service in the Emerging Networking Panorama, vol. 3266 of Lecture Notes in Computer Science, Ch. A Game-Theoretic Analysis of TCP Vegas, Springer-Verlag, 2004, pp. 338–347.
[39] L. Andrew, C. Marcondes, S. Floyd, L. Dunn, R. Guillier, W. Gang, L. Eggert, S. Ha, I. Rhee, Towards a common TCP evaluation suite, in: Proceedings of Sixth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2008), Manchester, UK, Mar. 5–7, 2008.
[40] S. Floyd, Metrics for the evaluation of congestion control mechanisms, Internet Draft, draft-irtf-tmrg-metrics-09.txt (Mar. 2007).
[41] Ns-2 Network Simulator. Available from: URL <http://www.isi.edu./nsnam/ns>.
[42] G. Wang, Y. Xia, D. Harrison, An NS2 TCP evaluation tool suite, Internet Draft, draft-irtf-tmrg-ns2-tcp-tool-00 (Apr. 2007).
[43] TCP Evaluation Suite. Available from: URL <http://netlab.cs.ucla.edu/tcpsuite/>.
[44] H. Shimonishi, M. Sanadidi, T. Murase, Assessing interactions among legacy and high-speed TCP protocols, in: Proceedings of Fifth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2007), Marina Del Rey, CA, USA, Feb. 7–9, 2007.
[45] A. Veres, Z. Kenesi, S. Molnár, G. Vattay, On the propagation of long-range dependence in the internet, in: Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, Aug. 28–Sep. 1, 2000, pp. 243–254.
[46] D.M. Chiu, R. Jain, Analysis of the increase and decrease algorithms for congestion avoidance in computer networks, Computer Networks and ISDN Systems 17 (1) (1989) 1–14.
[47] R. Jain, A. Durresi, G. Babic, Throughput fairness index: an explanation, ATM Forum/99-0045 (Feb. 1999). Available from: URL <http://www.cse.wustl.edu/jain/atmf/a99-0045.htm>.
[48] N. Brownlee, K. Claffy, Understanding internet traffic streams: dragonflies and tortoises, IEEE Communications Magazine 40 (10) (2002) 110–117.
[49] A. Gurtov, Effect of delays on TCP performance, in: Proceedings of IFIP Personal Wireless Communications 2001 (PWC'2001), Aug 2001, Lappeenranta, Finland, pp. 8–10.
[50] BI-TCP implementation for NS 2. Available from: URL <http://www4.ncsu.edu/rhee/export/bitcp/bitcp-ns/bitcp-ns.htm>.
[51] FAST TCP simulator module for ns-2. Available from: URL <http://www.cubinlab.ee.unimelb.edu.au/ns2fasttcp/>.
[52] S. Floyd, Limited slow-start for TCP with large congestion windows, IETF RFC 3742 (2004).
[53] L. Tan, W. Zhang, C. Yuan, On parameter tuning of fast TCP, IEEE Communications Letters 11 (5) (2007) 458–460.