Budapest University of Technology and Economics
Department of Telecommunications and Media Informatics

Fairness and Stability Analysis of High Speed Transport Protocols

Ph.D. Dissertation

Balázs Sonkoly

Scientific supervisor:
Sándor Molnár, Ph.D.
High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Budapest, Hungary
2010

Balázs Sonkoly

High Speed Networks Laboratory

Department of Telecommunications and Media Informatics

Budapest University of Technology and Economics [1]

[1]The reviews and the minutes of the Ph.D. Defense are available from the Dean's Office.

Abstract

TCP congestion control had managed successfully the stability of the Internet in the past decades but it has reached its limitations in "challenging" network environments. The new 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 current Internet. In recent years, several new proposals and modifications of the standard congestion control mechanism have been developed by different research groups all over the world. These new mechanisms and TCP versions address different aspects of future networks and applications and improve the performance of regular TCP. One of the important network environments where the serious drawbacks of standard TCP Reno can be experienced is high speed wide are networks. These networks can be characterized by high bandwidth-delay product (BDP) and TCP cannot efficiently utilize them due to its conservative congestion control scheme. As a response to this problem, the research community has proposed several new transport protocols recently referred as high speed TCPs.

On the one hand, this dissertation provides a comprehensive fairness analysis of promising TCP proposals and congestion control mechanisms designed for high bandwidth-delay product networks. We put emphasis not only on the equilibrium behavior but also on the transient characteristics with the dynamic behavior to gain an in-depth insight into the interaction and co-existence of different congestion control schemes. We reveal the analytic properties of the transient and dynamic behavior of individual flows of different TCP protocols and define novel metrics in order to characterize them. The dynamic characteristics of competing flows are also investigated in a wide range of network scenarios. We investigate how the transient state can influence the long-term fairness performance. As part of this transient analysis, we deeply analyze the impacts of starting time on the performance and fairness. Here, the main focus is on inter-protocol properties when different TCP versions are interacting. The explanations of the experienced phenomena are also given based on the novel metrics and methodology.

On the other hand, a comprehensive control-theoretic analysis of HSTCP is also carried out to estimate the performance of the protocol in a very high bandwidth-delay product network environment where the queues are regulated by the RED active queue management. The motivation behind our approach is to gain analytical insight into the performance of HSTCP. To

achieve this goal, firstly, we establish a fluid-flow model for HSTCP/RED networks. Secondly, we describe a systematic implementation methodology in detail in order to make the non-linear model tractable. The results of the model are validated by packet-level simulations. Thirdly, a stability condition for HSTCP/RED networks is derived which can be applied in RED design and parameter setting.

Kivonat

A mai Internet egyik legfontosabb szállítási protokollja a megbízható adatátvitelt biztosító TCP protokoll (Transmission Control Protocol, szállításvezérlő protokoll) és annak különböző verziói. Számos funkciója közül az egyik legfontosabb a torlódásszabályozás, a hálózat megóvása a túlterheléstől és az összeomlástól. A kapcsolatorientált TCP protokoll zárthurkú szabályozást végez, melynek során az adó entitás az adási sebességét a beérkező nyugták alapján állítja a hálózati körülményekhez azzal a céllal, hogy a hálózat jó kihasználtsággal működjön, de a túlterhelés ne okozzon összeomlást. Az Internet több évtizedes folyamatos fejlődése során jelentősen változtak a hálózati technikák és a jellemző alkalmazások, ami a TCP folyamatos módosítását és javítását tette szükségessé. A szállítási protokollok számára komoly kihívást jelent a nagysebességű (és nagykiterjedésű) hálózati környezetben való hatékony működés, amit a hagyományos TCP protokoll a konzervatív torlódásszabályozási mechanizmusa miatt nem képes biztosítani. A közelmúltban számos TCP módosítási javaslat született, melyek a nagysebességű hálózati környezetben igyekeznek hatékonyabb működést garantálni. Ezek a nagysebességű TCP variánsok a hagyományos, konzervatív torlódásszabályozási mechanizmust cserélik le és az új környezethez igazodó módszereket javasolnak.

Munkám első részében különböző nagysebességű TCP protokoll (HighSpeed TCP, Scalable TCP, BIC TCP, FAST TCP), illetve torlódásvezérlési mechanizmus (AIMD alapú, MIMD alapú, késleltetés alapú) széleskörű analitikus és szimulációs vizsgálatát végeztem el a "fairness" szempontjából, különös tekintettel a dinamikus tényezőkre és azok hosszú távú viselkedésre gyakorolt hatására. A különböző hálózati topológiákkal (dumb-bell, parking-lot) és versenyhelyzetekkel (egy–egy folyam, egy folyam – aggregátum, indulási késleltetés hatása) széleskörű intra-protokoll és inter-protokoll vizsgálatokat végeztem. A kutatásokat tovább folytattam a tapasztalt jelenségek magyarázatával, illetve az ehhez alkalmazott módszertan kidolgozásával és új teljesítménymetrikák bevezetésével.

Munkám második részében az egyik első nagysebességű TCP javaslat, a HighSpeed TCP vizsgálatával foglalkoztam. Felállítottam egy tisztán HighSpeed TCP-t használó folyamokat tartalmazó hálózat folyadékmodelljét, ahol a hálózat egy szűk sávszélességű linket tartalmaz és a

várakozási sor menedzselése a RED aktív sorkezelési mechanizmus szerint történik. A megalkotott hálózati modell vizsgálatához és a bonyolult differenciálegyenletekből felépülő rendszer numerikus approximációval történő megoldásához módszertant dolgoztam ki, illetve egy keretrendszert fejlesztettem. A kialakított MATLAB/Simulink környezetben a differenciálegyenlet-rendszer különböző kezdeti értékek mellett megoldható numerikus approximáció segítségével. A modell eredményeit Ns-2 környezetben végzett csomagszintű hálózati szimulációkkal validáltam. Ezenkívül stabilitási feltételt adtam meg HighSpeed TCP/RED hálózatokhoz, ami alapján a RED paraméterei úgy állíthatók be, hogy a hálózat stabil működést mutasson.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Sándor Molnár for his guidance and support in my research work. I am also grateful to Dr. Tuan Anh Trinh for his help and the fruitful discussions. A have done my work at the High Speed Networks Laboratory (HSNLab), and I wish to express my grateful thanks to them for the permanent support. And of course, I would like to thank my family for their continuous support and encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Transmission Control Protocol (TCP) has played an important role in the success of Internet. The original protocol providing a reliable, connection-oriented service on top of IP networks dates back to 1981 (RFC 793 [68]). In the mid 1980s, serious incidents were experienced in the Internet when the network performance fell down by several orders of magnitudes. This phenomenon, called *congestion collapse*, raised the urgent need of some more sophisticated control mechanism in the transport layer. The original solution for the congestion collapse was provided in [31] by Van Jacobson. An essential part was added to TCP including the congestion control mechanisms. The congestion management of TCP is composed of two important algorithms. The Slow-Start and Congestion Avoidance algorithms allow the protocol to increase the data sending rate of sources without overwhelming the network and help to avoid congestion collapse. The protocol updates a variable called *congestion window* that directly affects the sending rate by means of limiting the number of unacknowledged packets in the network based on a sliding window mechanism. This variable is adjusted according to various algorithms in different phases of the connection. The basic mechanism was incrementally developed and tuned introducing new additional algorithms, e.g., RTO calculation and delayed ACK in 1989 (RFC 1122 [8]), SACK in 1996 (RFC 2018 [56]) and NewReno in 2004 (RFC 3782 [22]) just to mention a few.

TCP congestion control had managed successfully the stability of the Internet in the past decades but it has reached its limitations in "challenging" network environments. The new 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 current Internet. In recent years, several new proposals and modifications of the standard congestion control mechanism have been developed by different research groups all over the world. These new mechanisms and TCP versions address different aspects of future networks and applications and improve the performance of regular TCP.

For example, standard TCP (Reno version) cannot provide acceptable performance in wireless or mobile environments where the propagation delay and the available bandwidth can suddenly change (e.g., during inter-system handover) which can result in multiple back-offs or in extreme cases in disconnection. In order to remedy this problem, new TCP versions have been dedicated to this environment. The drawbacks of standard TCP Reno can be experienced in high speed wide area networks, as well. These networks can be characterized by high bandwidth-delay product (BDP) and TCP cannot efficiently utilize them due to its conservative congestion control scheme. As a response to this problem, the research community has proposed several new transport protocols recently referred as *high speed TCPs* or *high speed transport protocols.*

1.1 Research Objectives

As a consequence of the existence of various transport protocol proposals, future networks will carry a mixed traffic generated or controlled by heterogeneous transport mechanisms, however, little is known about the co-operation of them. In this respect, the understanding of fairness performance is crucial. *Our first main research goal is to provide a deeper insight into the interaction of recent congestion control schemes and fairness characteristics.* The relevance of this research direction is supported by the fact that high speed TCP proposals are not only theoretical mechanisms but they have appeared in recent operating systems, as well. For example, from Linux kernel version 2.6.19, the default TCP protocol is CUBIC [70], while Microsoft Windows Vista and Server 2008 use Compound TCP [80] as default protocol.

Most of the papers on fairness analysis examine the long-term behavior of the TCP flows, however, we claim that dynamical aspects should be taken into consideration, as well, due to the characteristics of today's challenging network environments. *Thus, our next goal is to reveal the analytic properties of the transient and dynamic behavior of individual flows of different protocols and to define appropriate metrics in order to characterize them. The investigation of dynamic characteristics of competing flows in a wide range of network scenarios is also addressed by our research.* Revealing the impact of flow starting times on long-term fairness performance is also included in this research goal. Here, the main focus is on inter-protocol properties when different TCP versions interact. Another relevant part of our research aims at providing the explanations of the experienced phenomena based on the novel metrics and methodology.

TCP is an important element of the congestion control of the Internet, however, the dynamic behavior of the network is not solely affected by this mechanism but network routers, especially with Active Queue Management (AQM) schemes, also play important roles. In this respect, Random Early Detection (RED) [23] gateways are the most remarkable ones as RED is implemented in a wide range of commercial routers. In the second part of our research, we focus on the performance of HSTCP[18]/RED networks. *Our goal is to provide a control-theoretic model to estimate the performance of HSTCP in a very high bandwidth-delay product network environment when the queues are regulated by RED.* The motivation behind our approach is to gain analytical insight into the performance of HSTCP. *We also address the stability analysis of such a network in order to give guidelines for setting RED parameters appropriately yielding stable network operation.*

1.2 Methodology

To achieve the above mentioned research goals, several tools are applied from the field of mathematical modeling and network simulation.

In Chapter 3, the comprehensive fairness analysis of TCPs is carried out based on packet-level simulations conducted in Ns-2 [64]. Our approach includes the analytical characterization of protocols based on novel metrics, such as saturation time and identification of main operating frequencies, as well.

In Section 3.4, the evolution of congestion window controlled by different transport mechanisms is captured by simple formulae gained from an analytical analysis at the packet-level. In certain cases, approximations are also applied and the results are validated by packet-level simulations.

As a novel method in fairness characterization, spectral analysis is applied for individual flows and competing flows, as well. The main operating frequencies of protocols are identified based on this approach in order to explain the experienced phenomena. In Section 3.5 and 3.6, the dynamics of bandwidth shares, congestion windows and queue processes obtained from packet-level simulations are extensively analyzed both in the time- and frequency-domain.

In Chapter 4, a control-theoretic approach is applied to get an in-depth analytical insight into HSTCP/RED networks. We provide flow-level, fluid-flow models regarding HSTCP and other network elements. This non-linear model consists of coupled differential equations with complex dependences and varying time delays, therefore, analytically not tractable. Thus, we apply numerical approximations and describe a systematic implementation methodology. A MATLAB/Simulink [57] framework is designed and implemented for analyzing proposed fluid-models. The model is validated by packet-level simulations using Ns-2.

The stability of the HSTCP/RED network as a feedback system is also investigated. The intractable non-linear model is simplified by the technique of linearization around an operating point and a local stability condition is derived analytically applying tools from the field of classical linear feedback control systems, such as Laplace transform and Nyquist criterion. The local stability condition on the linear system is validated by non-linear flow-level results and packet-level simulations.

1.3 Structure of the dissertation

The rest of the dissertation is organized as follows. In Chapter 2, a detailed overview is given on the related works. First, the relevant TCP proposals and AQM schemes are summarized. Second, the results on TCP performance and stability analysis are overviewed.

Chapter 3 is devoted to a comprehensive TCP fairness analysis. In Section 3.2, the motivation of the work is presented through simple illustrations. Then the simulation environment with the investigated scenarios is shown in details in Section 3.3. The main results and statements can be found in the subsequent sections. In Section 3.4, the transient and equilibrium analysis of individual flows are carried out, and novel metrics, such as saturation time and main operating frequencies, are suggested. In Section 3.5, the intra-protocol fairness characteristics are summarized. The most important results regarding the inter-protocol fairness performance are presented in Section 3.6. At the end of the chapter a brief discussion and the summary of the main results are presented before the conclusion.

In Chapter 4, a control-theoretic analysis of HSTCP/RED networks is provided. A fluid-flow model of the HSTCP source and other network elements are given in Section 4.2 and a systematic implementation methodology is provided in a MATLAB/Simulink framework in Section 4.3. The model is validated by packet-level simulations. The corresponding results and examples are presented in Section 4.4. In Section 4.5, based on the established model, a stability condition for HSTCP/RED networks is derived with numerical examples. Finally, a discussion on the possible extension of the framework is given in Section 4.6 and the conclusions of this chapter are drawn.

In Chapter 5, the possible applications of the new results are outlined briefly.

The main conclusions of the dissertation are drawn in Chapter 6.

Chapter 2

Overview on Related Works

The congestion control mechanisms of TCP had managed to maintain the stability of the Internet in the past decades but today it has reached its limitations due to the new 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 [68]). The basic mechanism was incrementally developed and tuned as new challenges appeared introducing new additional algorithms. 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. As a response, the research community, including several excellent research groups all over the world, has developed many proposals. The huge number of new ideas has resulted in different new TCP versions implemented in several environments. These proposals can efficiently address different aspects of the challenging future network environments and applications. In order to select the "optimal" transport protocol, extensive performance analysis is necessary in a wide range of network environments and applications. In the recent years, many papers were published deepening our understanding of these new protocols regarding performance characteristics, co-existence issues, and other important properties affecting the deployability of them.

In this chapter, an overview is given on the recent TCP versions and related active queue management techniques. The related papers and research works on TCP performance and stability analysis are also presented here.

2.1 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., [60]. An overview of the most important proposals (see Table 2.1) is given here, and some technical details of the TCP variants analyzed in the dissertation are summarized in Table 2.2. HighSpeed TCP (HSTCP) [18] is a

Table 2.1: TCP versions

	protocol	main mechanisms	network environment	main properties
loss-based	HighSpeed TCP	adaptive AIMD	high BDP	minor modifications to Reno
	Scalable TCP	MIMD	high BDP	performance improvements, scalability
	BIC TCP	AI, binary search, max probing, MD	high BDP	scalability, stability, linear RTT-fairness
	CUBIC	enhanced BIC, cubic growth	high BDP	simplified window control, improved TCP-friendliness, linear RTT-fairness
	TCP Libra	new utility fn.	general	RTT-fairness, TCP-friendliness
	H-TCP	adaptive AIMD	high BDP	performance improvements
	TCP-Westwood	eligible rate est., agile probing, PNCD	high BDP, wireless	performance improvements, scalability, TCP-friendliness
	LTCP	macroscopic/microscopic control	high BDP	scalability, AIMD characteristics
delay/loss-based	TCP Vegas	"AIAD-like"	general	fairness problems
	FAST TCP	"fast Vegas"	high BDP	performance improvements, good fairness properties
	TCP-Africa	aggressive mode/ Reno mode	high BDP	performance improvements, scalability, TCP-friendliness
	YeAH-TCP	"Fast" mode/ "Slow" mod	high BDP	performance improvements, scalability, TCP-friendliness
	TCP-Illinois	adaptive AIMD based on delay	high BDP	scalability, performance improvements
	TCP-Adaptive Reno	adaptive AIMD based on BBE	high BDP	scalability, performance improvements
	Compound TCP	delay-based component + Reno	general	good utilization, TCP-friendliness
	LT-TCP	proactive FEC/ reactive FEC	wireless	performance improvements
explicit	XCP	router assistance, utilization/fairness control	general	modification of the routers is necessary
	RCP	approximates ideal PS	general	improved flow-completion times

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 2.2) yielding an adaptive and more or less scalable algorithm. HSTCP introduces a new relation between the average congestion window and the steady-state packet drop (or marking) rate. It is designed to have the standard TCP response in environments with mild to heavy congestion (packet loss rates of at most 10^{-3}) and to have a different, more aggressive response in environments of very low congestion event rate.

Ideas to introduce MIMD mechanisms for TCP have also been considered. Scalable TCP (STCP) [39] 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 2.2). A proposed setting for the constants are $a = 0.01$ and $b = 0.125$ [39].

In order to solve the TCP's severe RTT (round-trip time) unfairness problems, BIC TCP has been developed [100]. 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 2.2. The good performance of the protocol, including good utilization, linear RTT fairness (RTT unfairness is proportional to the RTT ratio as in AIMD), good scalability, and TCP-friendliness, comes from the slow

Table 2.2: Details of TCP variants analyzed in the dissertation

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	$\mathtt{w} \leftarrow \mathtt{w} + \frac{a}{\mathtt{w}}\ , \ \ a \in \left\{ S_{min}, \frac{W_{max} - \mathtt{w}}{B}, \frac{\mathtt{w} - W_{max}}{B-1}, S_{max} \right\}$	(per-ACK)	$\mathtt{w} \leftarrow \beta\mathtt{w}$
FAST TCP	$\mathtt{w} \leftarrow \min\left\{ 2\mathtt{w},\ (1-\gamma)\mathtt{w} + \gamma\left(\frac{\mathtt{baseRTT}}{\mathtt{RTT}}\mathtt{w} + \alpha\right)\right\}$	(periodically)	$\mathtt{w} \leftarrow 0.5\mathtt{w}$

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 [70] is an enhanced version of BIC TCP. It simplifies the BIC window control and improves its TCP-friendliness and RTT-fairness. TCP Libra [55] 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 [45]. 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 [95] 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 [4]. 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 [35, 97]. FAST TCP has the same equilibrium properties as TCP Vegas [54] 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 [35, 61, 97]. 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 2.2, 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 [35, 61, 97] and Section 3.6.2). FAST TCP also reacts to packet losses halving its congestion window. The delay-based control also appears in other proposals like TCP-Africa [41]. 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 [90]. 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 [49] and TCP-Adaptive Reno [71]. 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 [80] 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 [83] 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) [36] which generalizes the Explicit Congestion Notification (ECN) proposal [69]. 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) [14]. RCP approximates ideal processor sharing by explicitly emulating PS at each router. It is shown in [15] that RCP shows improvement in flow-completion time for realistic traffic mixes (containing not only long-lived flows).

2.2 Active Queue Management (AQM)

In order to get efficient network operation and to avoid congestion collapse, TCP's congestion control mechanism is an important component. However, the dynamic behavior of the network is not solely affected by these mechanisms but network routers also play important roles. A typical network router maintains a set of queues (buffers) belonging to different outputs. Traditional drop-tail routers hold the packets to be scheduled in such queues. When the queue is full, the incoming packets are dropped. By this scheme, the congestion cannot be controlled. Therefore, several Active Queue Management schemes were proposed in order to avoid congestion at the routers by means of efficient buffer management. Packet dropping policies are congestion management mechanisms implemented in the network routers that reactively or proactively drop packets in order to reduce congestion and free up buffer space [43]. An exhaustive review of possible AQM schemes is given in [43] and a detailed classification of them is proposed, as well. This classification is based on the network environment (ATM or IP), the type of congestion management mechanism (avoidance or control and recovery), the number of used thresholds (none or global or per-connection), the state information (global or per-connection), and the queue behavior (static or dynamic). This classification is extended in [42] for a wide variety of AQM schemes and many different characteristics of IP networks are also considered. The main goal of packet dropping policies is to avoid or control congestion, however, other performance metrics, such as application throughput, network utilization, fairness performance, and synchronization problems of TCPs, are also significantly affected. Therefore, several research papers deals with the performance analysis of AQM schemes in different network environments (see e.g., [17, 28, 42, 59]).

One of the most important AQM scheme is Random Early Detection (RED) [23] as it is implemented in a wide range of commercial routers, as well. However, other AQM mechanisms were proposed to enhance the capability of the basic RED algorithm, e.g., SRED [67], Adaptive RED [21], REM [3], Blue [16] just to mention a few. The deeper understanding of the behavior of TCP/RED networks by the means of exhaustive mathematical evaluation has also got significant attention. For example, (linear and non-linear) time series analysis of such networks proved to be a good tool to reveal the root-causes of complex behavior exhibited by RED-based networks [5, 6, 34].

Using RED, in contrast to traditional drop-tail algorithm, packets from the queue can be dropped or marked – if it is used together with the Explicit Congestion Notification (ECN) proposal [69] – according to a marking profile when the buffer is not full. When the average queue size is under a threshold then the incoming packets are accepted. As the average queue size increases, the dropping or marking probability of an arriving packet increases, too (*early drop*). Reaching an upper threshold (or the buffer size), the incoming packets are dropped or marked with a probability of 1 (*forced drop*). This mechanism helps to avoid global synchronization which is a known problem of drop-tail queues, and enhances the global power (defined as the ratio of throughput to delay) and fairness characteristics, as well [23].

The basic RED algorithm has three parameters (p_{max}, t_{min}, t_{max}) [23] characterizing the packet marking profile. This function is shown in Figure 2.1. The average queue size which is the input parameter of the marking profile is calculated using a low-pass filter with an exponential weighted moving average (EWMA). The time constant of the filter or the weighting factor (w_q) is also an important parameter of a RED gateway. The main parameters are summarized in Table 2.3.

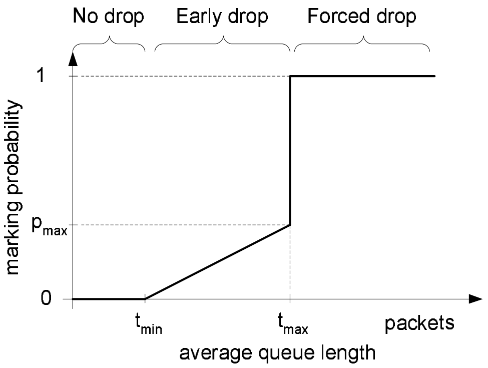


Figure 2.1: RED packet marking profile

Table 2.3: RED parameters

p_{max}	maximum probability of early drop
t_{min}	minimum threshold
t_{max}	maximum threshold
w_q or q_weight	weighting factor or queue weight

The main drawback of RED rooted in the careful parameter settings that is required in order to provide good performance. This parameter tuning and the performance evaluation of RED is addressed by several research papers (see e.g., [7, 13, 29, 48]), however, the optimal configuration for different network environments is still an open issue.

2.3 TCP performance analysis

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 [78, 87], a simulation-based performance analysis of HighSpeed TCP is presented and the fairness to regular TCP is analyzed. In [39], 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 [35], 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 [100], 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 [46].

The analysis of competing flows using different TCP versions has received less attention. In

[1], the fairness of MIMD algorithms is evaluated and the interaction of AIMD mechanisms with static parameters (e.g., TCP NewReno) are analyzed. In [47], 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 [26], experimental evaluation of different high speed TCP proposals is carried out mainly focusing on the relevant impacts of background traffic. In [98], 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 [61, 85]. We investigated fairness of delay-based TCP variants, FAST TCP in particular, from a game-theoretic point of view, [61]. 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 [2, 84, 89]. The goal of our research work is exactly in line with these goals of the community. In the internet draft [20] the metrics to be considered in an evaluation of new or modified congestion control mechanisms for the Internet has been collected. Here, we summarize briefly the most relevant ones.

- **Throughput:** it can be measured as a router-based metric of aggregate link utilization, as a flow-based metric of per-connection transfer rate, and as user-based metric based on utility functions or user waiting times. Sometimes it is distinguished from goodput that considers only useful traffic.

- **Delay:** similarly to throughput, delay can be measured as a router-based metric regarding the queueing delay over time, or as a flow-based metric in terms of per-packet transfer time. This transfer time can also include waiting time for the transport layer service or retransmissions.

- **Packet loss rate:** it can be considered as a network-based or a flow-based metric. Packets can be lost in the network due to congestion at the routers or corruption.

- **Response to changes, responsiveness:** this metric describes the ability of a congestion control mechanism to respond to sudden changes or transient events in the network (e.g., changes of routing or bandwidth, starting of a new flow, etc.).

- **Oscillations:** fluctuation in rates, throughput, and queueing delay can result in unstable network behavior with low utilization. Therefore, the minimizing of oscillation is another goal that should be considered in transport protocol design. This metric is related to the variance of the corresponding metrics in different time scales.

- **Fairness and convergence times:** fairness metrics describe very important characteristics of resource sharing regarding that how "fair" the current allocation is. It can easily be considered intuitively, however, a clear and widely accepted measure does not exist. The most important fairness metrics proposed so far will be discussed later in this section. The convergence time concerns the time for convergence to fairness between an existing and a new coming flow.

- **Robustness for challenging environments:** robustness needs to be explored for paths with reordering, corruption, variable bandwidth, asymmetric routing, router configuration changes, mobility, etc.

- **Robustness to failures and misbehaving users:** another goal of transport protocol design is to be as robust as possible to network failures and misbehaving users (e.g., receivers).

- **Deployability:** issues regarding the deployability of a new congestion control mechanism, such as "which network elements need to be changed", code complexity, etc.

In our respects, fairness is the most relevant performance metric. There are number of different fairness measures proposed by the research community so far regarding different aspects of the resource allocation [20, 79]. An early fairness measure is *max–min fairness* [32]. Under max–min fair resource allocation, the only way to increase a source's sending rate is to decrease the rate of some other source whose rate is less than or equal to the rate of the former one. Based on a resource allocation model [37], the seminal paper [38] presented the first mathematical model and analysis of congestion control algorithms for general topology networks. The resource allocation is considered as a global optimization problem where utility functions are assigned to network users. With this utility maximization problem, several fairness metrics can be described. In case of *proportional fairness* [38], an allocation is defined as proportionally fair if for any other feasible allocation, the aggregate of proportional changes is less than or equal to zero. Proportional fairness can be described by logarithmic utility functions. When we use the weighted sum of the proportional changes, we get *weighted proportional fairness*. Defining other utility functions, several *utility-based fairness* metrics can be achieved. In [51], the flow control algorithm is modeled based on a similar utility-based optimization approach. And finally, one of the most popular and widely used fairness metric is *Jain's index* [12, 33]. In contrast to the above metrics, Jain's index considers the flows with identical resource requirements and does not take into account different characteristics of flows, such as the number of links on the path or round-trip times or others. Jain's index is overviewed in details in Section 3.2.

The performance evaluation framework can be implemented in simulators using e.g., the network simulator Ns-2 [64], 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 [96]. 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 [93]. 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 [82] 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 [72]. 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.

2.4 TCP stability analysis

The current Internet – including network routers, links, data sources and sinks, congestion control mechanisms – is the largest artificial distributed feedback system. The modeling of such a system is a very challenging mathematical issue. In recent years, several mathematical tools and results have been adopted in order to analyze the dynamic behavior of this network or to provide designing guidelines regarding different network elements. Two types of studies are of fundamental interest [52]. On the one hand, for the analysis of equilibrium properties, optimization-theory and game-theory provide suitable tools. On the other hand, the dynamic characteristics and the stability of the feedback control system can be analyzed based on a control-theoretic approach.

In [38], the first mathematical model and analysis of congestion control algorithms for general topology networks is presented. The resource allocation is considered as a global optimization problem where utility functions are assigned to the sources. Decentralized primal and dual algorithms are given to implement solutions to relaxations of the optimization problems, respectively. In case of the primal algorithm, first-order source dynamics and static link laws are applied. On the other hand, the dual algorithm uses static source control with first-order link dynamics. The model is used to analyze the stability and the fairness of a class of rate control algorithms. A Lyapunov-based proof of global stability is provided in the absence of delays. A similar approach can be found in [51] where a gradient projection algorithm is given for solving the dual problem. In [92], the basic model of [38] is modified and a general rate-based model is given with dynamic sources and dynamic links, as well. The propagation delays (forward and backward direction) are also included in the model. The non-linear model is linearized around an operating point and stability conditions are derived with basic ideas in the proofs. A possible window-based extension of the model is also investigated. In [50], the equilibrium properties of different TCPs and AQM schemes are analyzed based on a duality model. The utility functions are derived for TCP Reno and TCP Vegas, while the RED and REM AQM mechanisms are modelled, as well.

In [58], the behavior of regular TCP (in congestion avoidance) is modeled by jump process driven Stochastic Differential Equations and a fluid model of the traffic is given. A special network centric loss model is used where the loss indications arrive at the sources from the network as a Poisson arrival process. The main deficiency of this model is the independence of the packet loss process and the data flow. In [59], this deficiency is remedied by modeling a complete system, where losses and TCP sending rates are closely coupled. Thus, the model corresponds to a closed loop control system giving rise to a set of coupled differential equations. Simulation results demonstrate that this model is able to capture the dynamics of TCP Reno accurately. In [29], the previous model given by coupled non-linear differential equations is converted to a linear system via the technique of linearization, and the well developed tools in classical linear feedback control theory are applied. With this technique, the stability margins of TCP Reno/RED networks with single bottleneck link can be analyzed and the parameter settings of RED can be examined. A stability condition for this network is given which is validated by packet-level simulations using Ns-2. In [30], improved controllers for AQM routers are provided. Namely, the P and PI controllers are investigated and shown some benefits of PI controller comparing to RED. In [28], a nonlinear stability analysis of a TCP/AQM network regulated by a Proportional controller is carried out and in the case of delay-free marking, the asymptotic stability is proven for all gains. For delayed feedback, a condition of local asymptotic stability is provided with a region of attraction. The basic single link model of [29] is generalized to network case in [53] and [27].

In order to get a deeper understanding of dynamic characteristics of recent congestion control mechanisms and to provide stability conditions for different network scenarios, the control-theoretic approach has proved to be a very useful tool. For example, in [11], the stability analysis of TCP Westwood is carried out, while in [94], a discrete-time model of FAST TCP is introduced and stability properties are derived analytically.

Chapter 3

TCP Fairness Analysis

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 our research work 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.

This chapter is based on [62, 75, 77, 86, 88].

3.1 Introduction

As a response to the limitations and performance problems of traditional TCP congestion control mechanism experienced in present networks, many proposals have been developed by the research community 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 our work 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 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 [99]. 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 3.6.2). With this spectral analysis together with flow-level, packet-level and queueing analysis we have a good understanding of the investigated phenomena.

In this chapter, 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 this dissertation. Therefore, we have collected all of the results in a technical report [75] available from `http://hsnlab.tmit.bme.hu/projects/tcp/`. This chapter is the extensive summary of that report. Partial results from our previous research can be found in [77, 88].

It is certainly desirable to evaluate high speed TCP variants by measurements in real networks and there are already substantial works in this respect, [10, 35, 44, 47, 96, 97]. 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.

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 [91].

The rest of this chapter is organized as follows. In Section 3.2, the motivation of our work is presented and the insufficiency of Jain's index is discussed. The simulation environment with the investigated topologies and scenarios are presented in Section 3.3. In Section 3.4 the transient and equilibrium behaviors of different TCP versions are analyzed and analytical results are derived. Novel metrics, namely the saturation time and the main operating frequencies, are also introduced and derived for different TCP variants. Section 3.5 and Section 3.6 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. Section 3.7 provides a brief discussion on the impact of lower link delays. In Section 3.8, our main results and contributions are summarized briefly. Conclusions are drawn in Section 3.9.

3.2 Why is Jain's index insufficient?

One of the most popular and widely accepted fairness indices is Jain's index [12]. It is used widespread because of its main benefits [33]. 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: $JI = \left(\sum x_i\right)^2 / \left(n \sum x_i^2\right)$, 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 [9]. 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., [25]). 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 [12]. It can well describe the static properties of competing flows. However, the characteristics of the new network architectures and environments with new routing algorithms cannot be well captured in all aspects by that model.

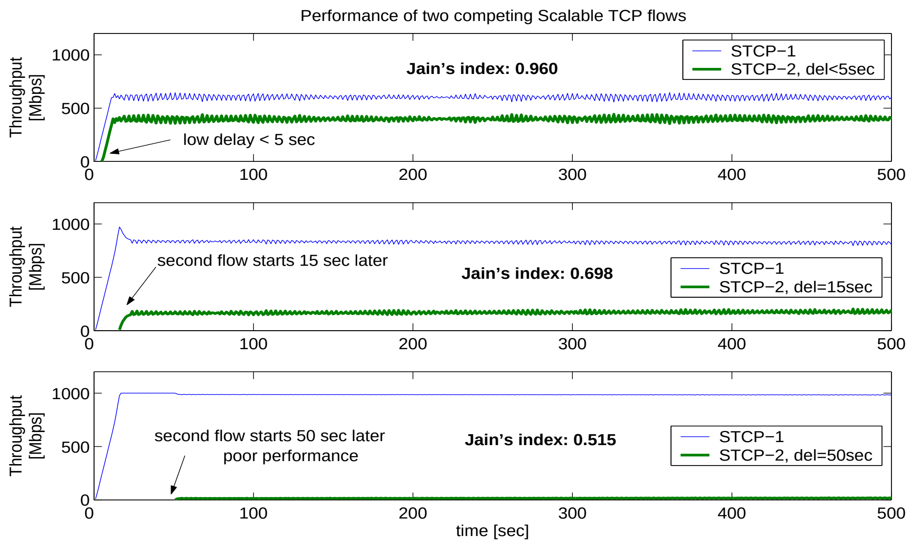


Figure 3.1: Performance of two competing Scalable TCP flows for different starting delays

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 one hour). It can be expected that the equilibrium bandwidth shares of the flows and the long-term fairness and thus, the Jain's indices are similar for scenarios with different starting times. In Figure 3.1, 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 sec 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 sec, 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 sec and the second flow only achieves very low throughput. This unfairness is confirmed by Jain's index near to 1/2. Surprisingly, we have experienced different long-term performance characterized by different Jain's indices which is mainly affected by the transient dynamics. However, Jain's index is not capable to explain the phenomenon and reveal the root-cause of this operation.

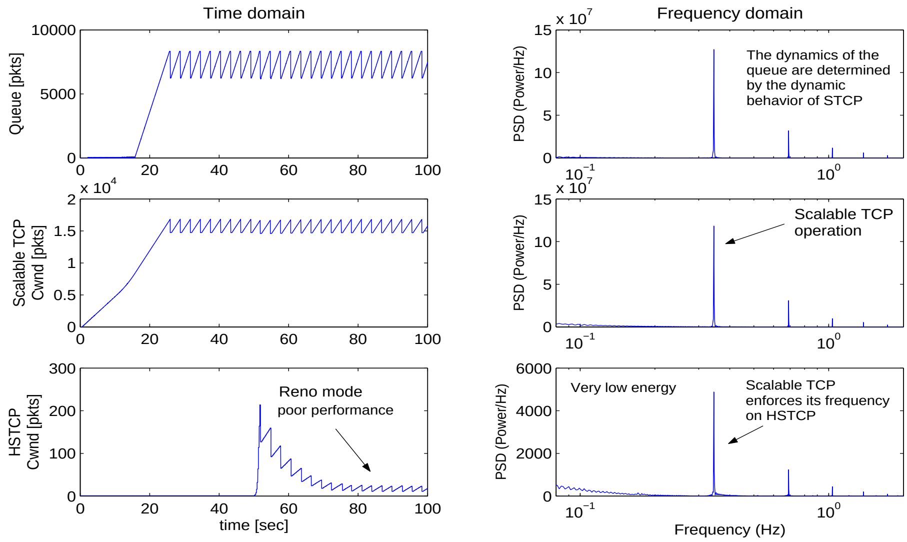


Figure 3.2: Competition of Scalable TCP and later entering HSTCP

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 [97]:

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

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 α_{stcp}, α_{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:

$$\frac{x_{stcp}}{x_{hstcp}} = \frac{\alpha_{stcp}}{\alpha_{stcp}}\frac{1}{p^{0.16}} \approx 0.625\,\frac{1}{p^{0.16}}. \tag{3.1}$$

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 Figure 3.2, 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 chapter. 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 Figure 3.2, 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 v6.5 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.

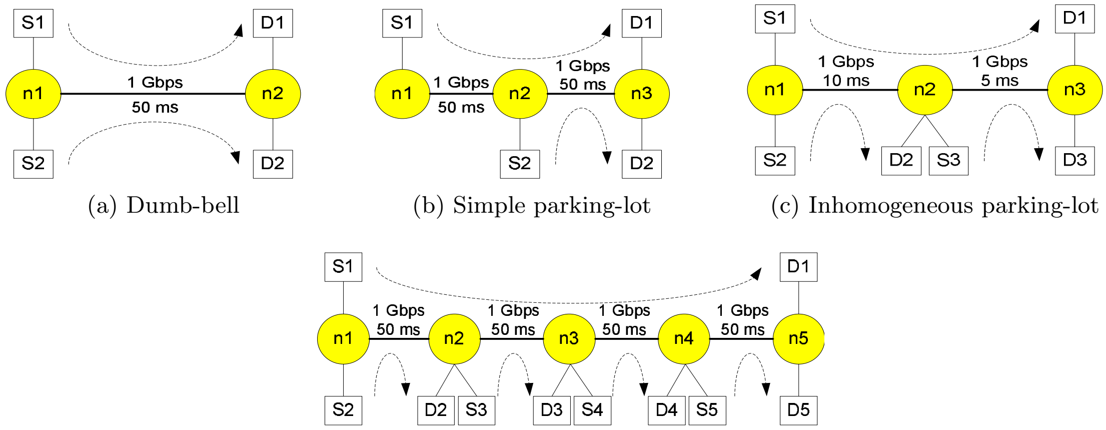
It has been recently debated that in case of competing flows, starting delay can have a relevant impact on the performance and fairness [98]. However, the explanations of the phenomena have not been given, yet.

As we have seen there is no widely accepted definition and metric for fairness, in the rest of the chapter, "fairness" or "fair behavior" is often used as a qualitative property regarding the fair usage of the shared links by the competing flows.

3.3 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 [64] simulation environment. Our simulation scripts regarding different network scenarios can be found in [75]. The different high speed transport protocols are integrated in the environment. Ns-2 version 2.27 includes the algorithm of HighSpeed TCP, while the Scalable TCP control mechanism can easily be implemented. The Ns-2 source code of BIC TCP is used from [65], and the FAST TCP implementation is borrowed from [66].

The examined dumb-bell topology containing one bottleneck link is shown in Figure 3.3a. 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 3.7.) The queueing mechanism corresponding to the bottleneck link is drop-tail.



(a) Dumb-bell (b) Simple parking-lot (c) Inhomogeneous parking-lot

(d) Complex parking-lot

Figure 3.3: Network topologies

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 (Figure 3.3b), 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 sec delay), and on the other hand, scenarios with smaller delay (e.g., with 15 sec 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 (Figure 3.3c), 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 (Figure 3.3d) 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 [18] and [39]). HSTCP and Scalable TCP apply the Limited Slow-Start (LSS) mechanism [19], 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 3.1.

Table 3.1: 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

3.4 Transient and equilibrium analysis of TCP versions

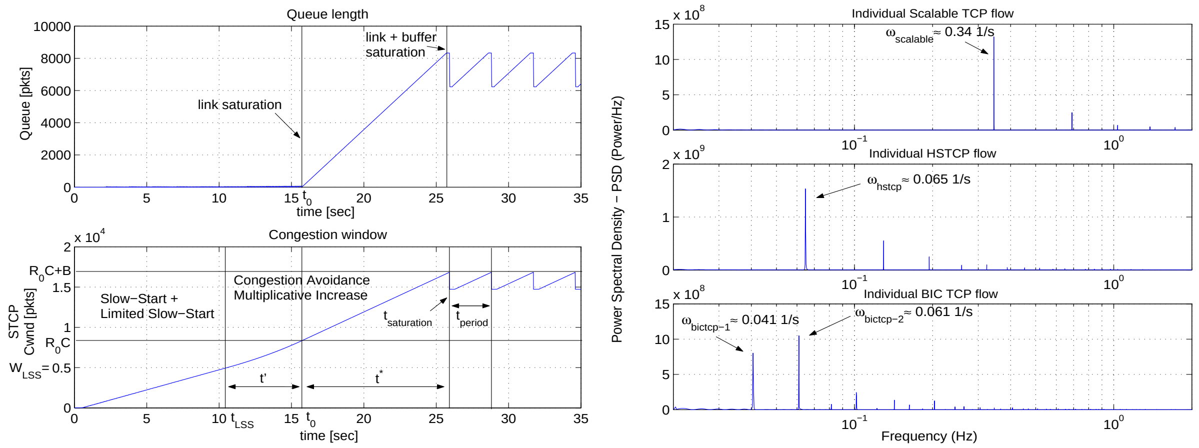
In order to understand the behavior of large number of TCP flows in realistic network environments, the first step has to be the investigation of individual flows. Different congestion control principles result in different traffic characteristics. In this section, the behavior of individual flows is 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 example of Scalable TCP protocol. By this approach, other TCP variants can easily be treated and the important parameters can be derived analytically.

The following variables are used throughout this section: C, R_0, R, B, W denote the capacity, the round-trip propagation delay, the round-trip time, the buffer size corresponding to the bottleneck link, and the congestion window.

3.4.1 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



(a) Saturation time of Scalable TCP

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

Figure 3.4: Transient and equilibrium behavior

the starting till the first packet drop. In Figure 3.4a, 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 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 Figure 3.4a 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 Appendix A and in [75], we summarize the main characteristics of these analytically tractable control algorithms and we derive the relevant parameters, as well. 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 $t_{SS} = R_0 \log_2 \texttt{ssthresh} \approx 0.664$ sec. After t_{SS}, the source operates according to the LSS mechanism using the default parameter ($\texttt{max\_ssthresh} = 100$ 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 Appendix A or [75] for details):

$$t_{LSS} = R_0 \frac{\lg \texttt{max\_ssthresh}}{\lg 2} + R_0 \frac{W_{LSS} - \texttt{max\_ssthresh}}{\texttt{max\_ssthresh}/2} \approx 10.46\,\text{sec}, \tag{3.2}$$

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:

$$W_{LSS} = \frac{\texttt{max\_ssthresh}/2}{a} = 5,000\,\text{pkts}. \tag{3.3}$$

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 ($R_0 C$) 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:

$$t_0 = t_{LSS} + t' = t_{LSS} + R_0 \frac{\lg \frac{R_0 C}{W_{LSS}}}{\lg(1+a)} \approx 15.6\,\text{sec}. \tag{3.4}$$

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

Table 3.2: Approximation of saturation time of different protocols

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

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 $W_0 = R_0 C$ to $R_0 C + B$ 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: $\tilde{R} = R_0 + B/(2C)$. Thus, the saturation time can be expressed as follows:

$$\hat{t}_{saturation} = t_0 + t^* = t_0 + \tilde{R} \frac{\lg \frac{R_0 C + B}{R_0 C}}{\lg(1+a)} \approx 26.05\,\text{sec}. \tag{3.5}$$

The analytically derived parameters and the approximation of saturation time meet well the simulation results presented in Figure 3.4a. In Table 3.2, we summarize our results on the transient behavior of different protocols (for details, see Appendix A and [75]).

3.4.2 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 loss-based protocols.

First, the long-term behavior of an individual Scalable TCP flow is analyzed. The MIMD mechanism of the protocol – operating during the equilibrium state – yields a time period

$$k = -\frac{\log(1-b)}{\log(1+a)} \tag{3.6}$$

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

$$R(t) \approx \tilde{R} = R_0 + B/C, \tag{3.7}$$

where R_0 is the round-trip propagation delay, C is the bottleneck capacity, and B is the buffer length. In our example $\tilde{R} \approx 0.2$ sec. According to these results, the time of a period (t_{period}) can be approximated by $k\tilde{R} \approx 2.6 \ldots 2.8$ sec. The analytical result well captures the periodic behavior experienced in the simulations. In Figure 3.4a, the length of a period $t_{period} \approx 2.9$ sec. 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 Figure 3.4b (top). The main spike corresponding to the dominant frequency can be seen at approximately $\omega = 0.34\,1/s$ which meets well the equilibrium time period derived analytically.

Second, the periodic characteristics of HSTCP can be determinded in a similar way. The equilibrium performance of a HSTCP source depends on the value of `cwnd`, since the increase and decrease factor changes with the congestion window. However, analytical results can be derived using approximations for $a(W)$, $b(W)$ and $R(t)$ at the neighborhood of the operating point (a_0, b_0, R_0). The maximum of the congestion window of an individual HSTCP flow can be determined based on the bandwidth-delay product and the buffer size:

$$W_{max} = R_0 C + B. \tag{3.8}$$

On the one hand, the decrease parameter at the operating point can be expressed easily ($b_0 = b(W_{max})$). On the other hand, the increase parameter is varying during one period but the value of that is bounded as follows:

$$a(W_{max}(1 - b_0)) < a_0 < a(W_{max}). \tag{3.9}$$

The length (k) of a period in RTT for HSTCP can be expressed in the following way (using approximations around the operating point):

$$\begin{aligned} (1 - b_0)W_{max} + k\, a_0 &= W_{max} \\ k &= \frac{b_0}{a_0} W_{max} \\ k &= \frac{b_0}{a_0} (R_0 C + B). \end{aligned} \tag{3.10}$$

In our example $k \approx 78 \ldots 96$ and $t_{period} \approx 15 \ldots 18$ sec. This is illustrated in Figure A.1a while the corresponding spectrum plot is shown in Figure 3.4b. The main frequency spike of HSTCP can be seen at approximately $\omega = 0.065\,1/s$

And finally, the equilibrium characteristics of BIC TCP is investigated, however, this derivation is slightly longer due to the special congestion control algorithm of the protocol. An individual BIC TCP flow also exhibits periodic long-term behavior. We recall the increase mechanism of BIC TCP from Table 2.2. When an acknowledgement is received, the congestion window is updated as follows:

$$W \leftarrow W + \frac{a}{W}, \text{ where } a \in \left\{ S_{min}, \frac{W_{max} - W}{B_{BIC}}, \frac{W - W_{max}}{B_{BIC} - 1}, S_{max} \right\}. \tag{3.11}$$

Thus, the increase method can be linear, logarithmic or exponential according to the state of the network and the protocol. These consecutive segments can be approximated by simple analytic formulae, respectively. As we use the BIC fast convergence option in Ns-2, the steady-state periodic behavior is realized by two different types of periods following each other alternately as it is shown in Figure A.1b. During the first period, mainly the additive increase mechanism of the congestion control algorithm operates with an increase parameter of S_{max}. The length of this period in RTT can easily be approximated as follows:

$$k_1 = \frac{(1-\beta)(R_0 C + B)}{S_{max}} \approx 104. \tag{3.12}$$

The round-trip time can be approximated at that operating point by $\tilde{R} = R_0 + B/C = 0.2$ sec which yields a time period with $t_{period1} \approx 20.8$ sec. At the end of this phase when the packet loss occurs, the congestion window is decreased by a multiplicative factor ($\beta = 0.8$) and the new target of the window is set as follows:

$$W_{max} = \frac{1+\beta}{2}\left(R_0 C + B\right) \tag{3.13}$$

because the BIC fast convergence option is used.

The second period consists of different segments. During this period, the congestion window is increased from $W_0 = \beta(R_0C+B)$ to $W_{end} = (R_0C+B)$. At the beginning, the additive increase mechanism operates with parameter $S_{max} = 32$. This linear phase lasts till the binary search increment goes below S_{max}. At that time instance, the congestion window can be expressed as

$$W_1 = \frac{1+\beta}{2} W_{end} - S_{max} B_{BIC}, \tag{3.14}$$

where $B_{BIC} = 4$ is the binary search parameter of BIC TCP. It is easy to express the length of this first phase in RTT (when W is increased from W_0 to W_1):

$$l_1 = \frac{W_1 - W_0}{S_{max}} = \frac{\frac{1-\beta}{2}\left(R_0 C + B\right) - S_{max} B_{BIC}}{S_{max}} \approx 48 \text{ RTT}. \tag{3.15}$$

The additive increase mechanism is followed by a logarithmic increase (binary search increase) phase when the congestion window is increased from W_1 until it roughly reaches W_{max}. More precisely, this mechanism ends when the difference between the current value of the congestion window and W_{max} is smaller than S_{min}. The length of this logarithmic phase can be expressed as follows:

$$l_2 = \frac{\log \frac{S_{min}}{W_{max}-W_1}}{\log \frac{B_{BIC}-1}{B_{BIC}}} = \frac{\log \frac{S_{min}}{S_{max}B_{BIC}}}{\log \frac{B_{BIC}-1}{B_{BIC}}} \approx 32.8 \text{ RTT.} \tag{3.16}$$

Now, the max probing phase is activated when the congestion window is increased exponentially from W_{max} to W_2. At the end of this section, the increment of max probing and the additive increase mechanism is equal. Hence, W_2 can easily be expressed by the following equation:

$$W_2 = \frac{1+\beta}{2} W_{end} + S_{max}(B_{BIC} - 1). \tag{3.17}$$

And now, the length of this phase is

$$l_3 = \frac{\log \frac{W_2 - W_{max}}{S_{min}}}{\log\left(1 + \frac{1}{B_{BIC}-1}\right)} = \frac{\log \frac{S_{max}(B_{BIC}-1)}{S_{min}}}{\log\left(1 + \frac{1}{B_{BIC}-1}\right)} \approx 31.8 \text{ RTT.} \tag{3.18}$$

And finally, the additive increase mechanism with parameter S_{max} is applied again when the window is increased from W_2 to W_{end}. Thus, the length can be derived easily as

$$l_4 = \frac{W_{end} - W_2}{S_{max}} = \frac{\frac{1-\beta}{2}(R_0 C + B) - S_{max}(B_{BIC} - 1)}{S_{max}} \approx 49 \text{ RTT.} \tag{3.19}$$

After some arithmetic arrangements, we get the length of the complete period as follows:

$$\begin{aligned} k_2 &= l_1 + l_2 + l_3 + l_4 \\ &= \frac{(1-\beta)(R_0 C + B)}{S_{max}} - (2B_{BIC} - 1) + \\ &+ \frac{2(\log S_{max} - \log S_{min})}{\log B_{BIC} - \log(B_{BIC} - 1)} + \frac{\log B_{BIC} + \log(B_{BIC} - 1)}{\log B_{BIC} - \log(B_{BIC} - 1)} \\ &\approx 162 \text{ RTT.} \end{aligned} \tag{3.20}$$

Applying our approximation for the round-trip time ($\tilde{R} = R_0 + B/C = 0.2$ sec), the time period of this second phase is $t_{period2} \approx 32$ sec.

It should be noted that due to the non-trivial trajectory of congestion window, the spectrum components cannot be directly calculated from these sub-periods but can be determined by a more detailed spectrum analysis. The spectrum of the periodic signal composed by these segments is shown in the bottom part of Figure 3.4b. The main frequency spikes can be observed at $0.04\,1/s$ and $0.06\,1/s$, respectively.

The relevant time periods of different loss-based TCPs are summarized in Table 3.3 while in Figure 3.4b, the spectrum plots of the congestion window processes of individual flows are shown. The main observation here is the significantly higher operation frequency of Scalable TCP compared to the analyzed AIMD-like flows.

For the sake of completeness, the equilibrium state of an individual delay-based FAST TCP flow shows stable behavior which is 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.

Table 3.3: Approximation of relevant time periods of different protocols

Scalable TCP:	$k = -\log(1-b)/\log(1+a)$	$t_{period} \approx 2.9\,\mathrm{sec}$
HSTCP:	$k \approx \frac{b_0}{a_0}\,(R_0C+B)$	$t_{period} \approx 15\,\mathrm{sec}$
BIC TCP:	$k_1 \approx \frac{1-\beta}{S_{max}}\,(R_0C+B)$	$t_{period1} \approx 21\,\mathrm{sec}$
	$k_2 \approx \frac{(1-\beta)(R_0C+B)}{S_{max}} - (2B_{BIC}-1) + \frac{2(\log S_{max} - \log S_{min})}{\log B_{BIC} - \log(B_{BIC}-1)} + \frac{\log B_{BIC} + \log(B_{BIC}-1)}{\log B_{BIC} - \log(B_{BIC}-1)}$	$t_{period2} \approx 32\,\mathrm{sec}$

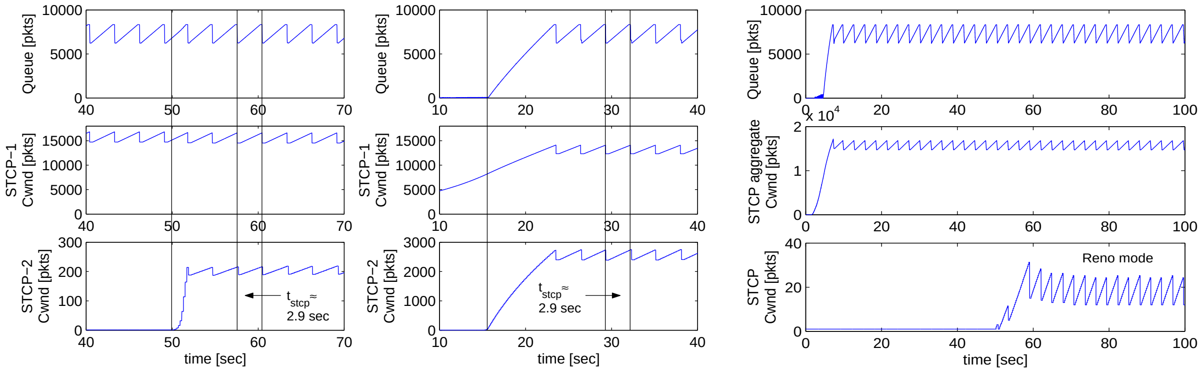
3.4.3 Contribution summary

To sum up, in this section, we have suggested *saturation time* as a novel fairness-related performance metric regarding dynamic aspects of the protocols that is able to characterize the initial transient behavior of congestion control mechanisms. Generally, saturation time gives the length of the transient phase. For a loss-based protocol, it is defined as the time from start till the first packet drop. For delay-based protocols, we have defined it as the length of time from start till the congestion window has settled down around the equilibrium state or the source has entered the delay-based operating regime. Analytical and approximative results have also been derived for the saturation time of Scalable TCP, HSTCP, BIC TCP, and FAST TCP. The approximations have been validated by packet-level simulations conducted in Ns-2.

Beside saturation time, we have introduced the *main operating frequencies* gained from the spectral characteristics of the congestion window processes as another novel metric that can be used to get an enhanced characterization of loss-based congestion control mechanisms. Main operating frequencies concern the dynamics of the long-term behavior. Moreover, identification of main operating frequencies plays an important role in the inter-protocol analysis, as well, in the subsequent sections. In addition, we have derived these frequencies analytically for individual flows of Scalable TCP, HSTCP, and BIC TCP protocols.

3.5 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



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

Figure 3.5: Intra-protocol behavior: Scalable TCP

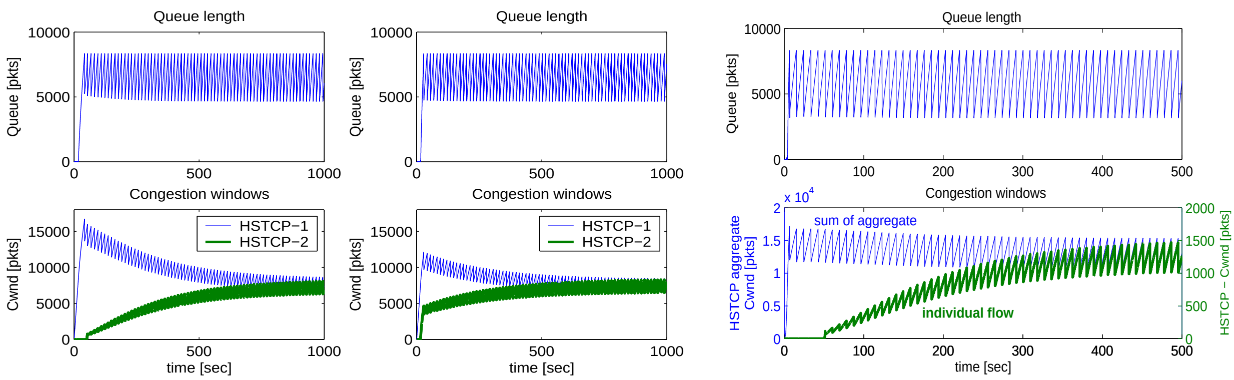
of the network behavior when all the sources use the same transport protocol. This topic has received more attention recently, and a lot of results can be found in the literature (see e.g., [35, 47]). Therefore, the aim of this section is rather to get a basic knowledge and to confirm and explain certain results than to provide a comprehensive study.

The details of our analysis can be found in [75]. 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.

3.5.1 Scalable TCP

Our first statement is the following: *the MIMD mechanism can not 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 is 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 Figure 3.5(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 sec), while the second scenario corresponds to a delay (15 sec) 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



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

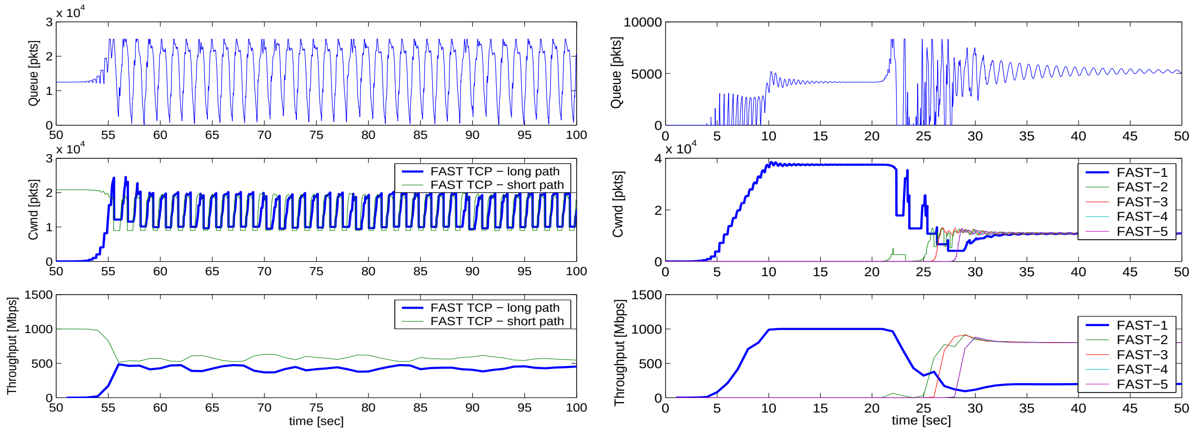
Figure 3.6: Intra-protocol behavior: HSTCP

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 Figure 3.5(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 sec while the last flow enters 50 sec 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 Figure 3.4b). 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.

3.5.2 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 Figure 3.6, 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 sec delay, two flows – 15 sec 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 affected and a fair*



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

Figure 3.7: Intra-protocol behavior of FAST TCP

equilibrium state is realized. In Figure 3.6(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.

3.5.3 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 Figure 3.7a, 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 Figure 3.7b, the 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 can not 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.

3.6 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 chapter, 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 [75].)

3.6.1 Scalable TCP and other loss-based protocol using AI mechanism

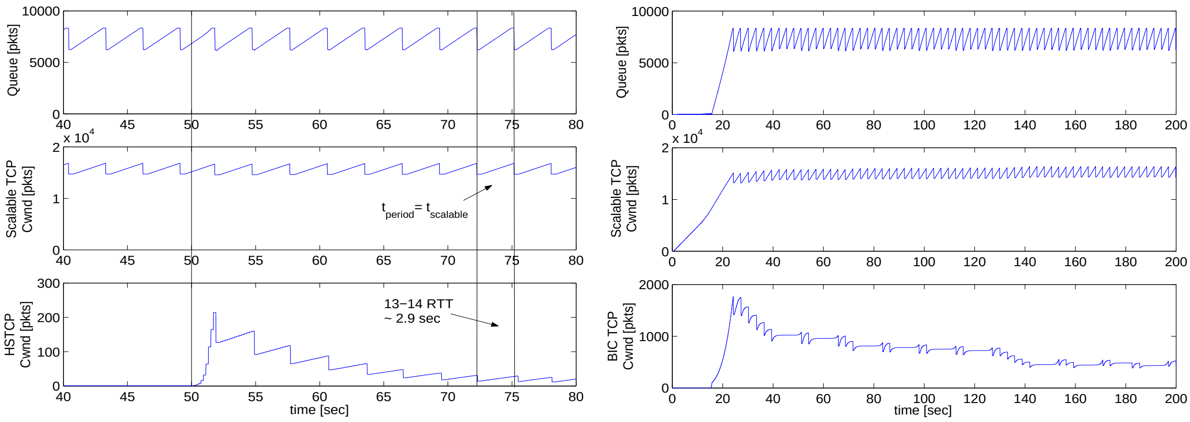
An essential part of a loss-based high speed transport protocol is the adequate increase mechanism. On the one hand, the multiplicative increase algorithm increases the congestion window by a constant value for each acknowledgement independently the current value of the window. On the other hand, adaptive 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.

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 Figure 3.8a. 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 sec). 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 $13\,a(W)$ 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;



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

Figure 3.8: Inter-protocol behavior: Scalable TCP – other loss-based protocol using AI mechanism

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 sec starting delay in Figure 3.8b. The equilibrium state does not depend on the starting time of the BIC TCP flow.

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

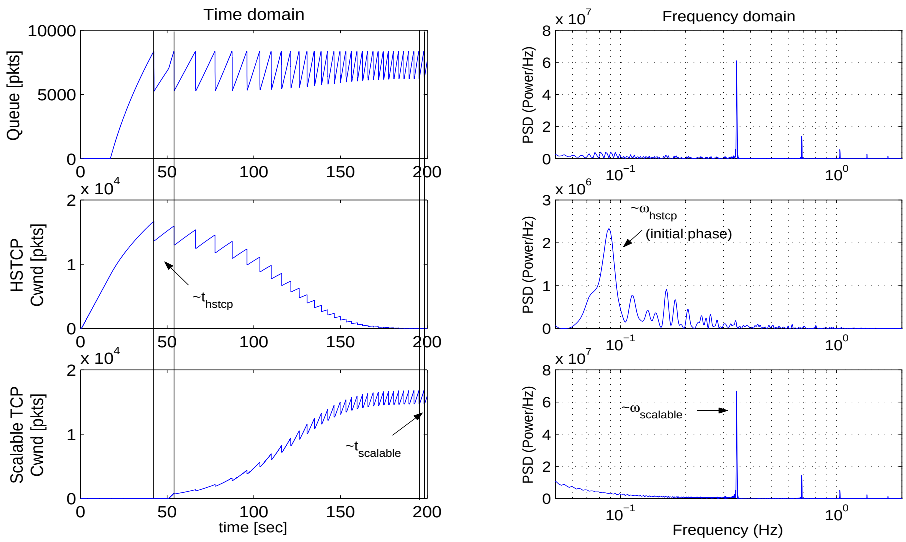


Figure 3.9: HSTCP – Scalable TCP, delay: 50s

50 sec later entering Scalable TCP flow. The simulation results are shown in Figure 3.9. 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

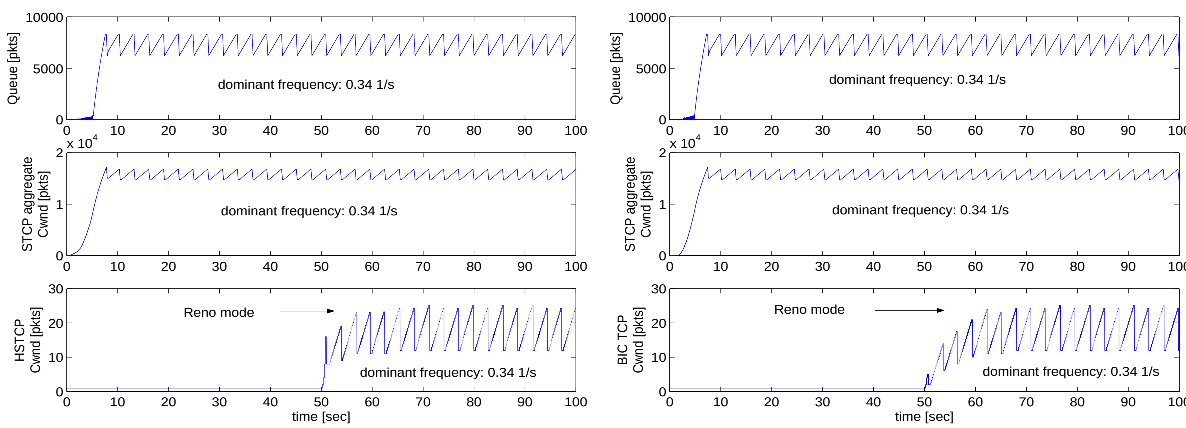
by the traffic aggregate of the two flows. The extra traffic of Scalable TCP in the queue results in a decreasing time period, i.e., the synchronized losses occur more frequently. During shorter time periods, `cwnd` of HSTCP cannot achieve the value that was just before the reduction. In contrast with 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 Figure 3.9, 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.

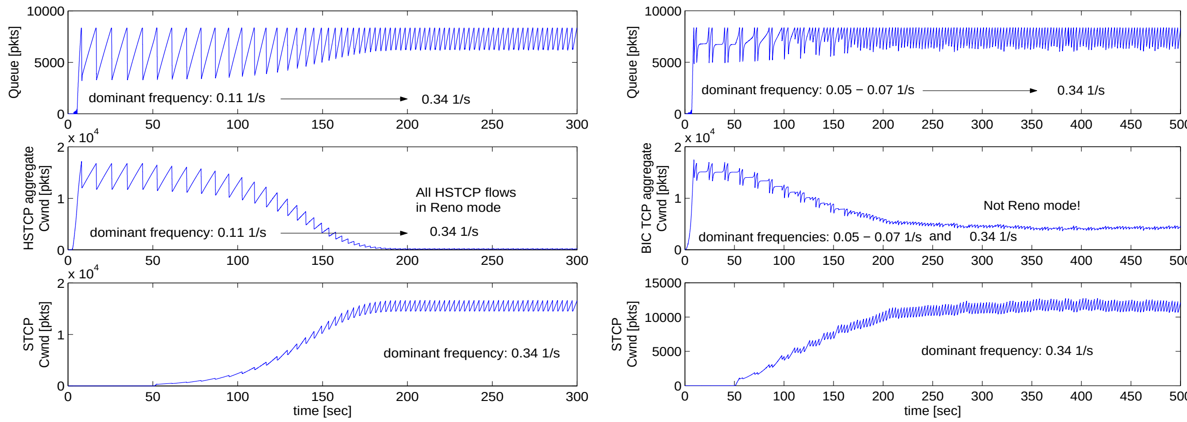
Dumb-bell topology: traffic aggregate – single flow

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 sec.)



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

Figure 3.10: Performance of individual flow vs. STCP traffic aggregate



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

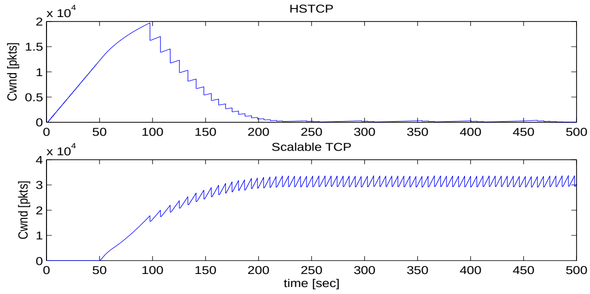
Figure 3.11: Performance of individual STCP flow vs. traffic aggregate

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 Figure 3.10. 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 [75].) As a consequence of the MIMD algorithm, the spectral behavior of the Scalable TCP aggregate is similar to the behavior of an individual flow.

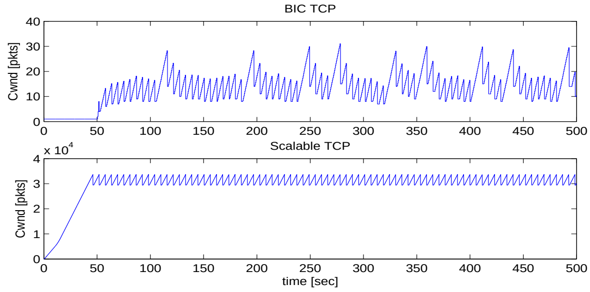
The next disadvantage of Scalable TCP is shown in Figure 3.11. Surprisingly, the individual Scalable TCP flow can starve the HSTCP and BIC TCP aggregates, as well. On the one hand, HSTCP flows operate in Reno mode achieving very low utilization at the equilibrium state and their periodic behavior is 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.

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 Figure 3.12a and Figure 3.12b for HSTCP and BIC

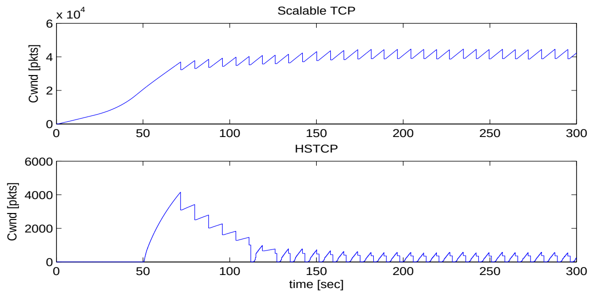


(a) HSTCP on the longer path starts first

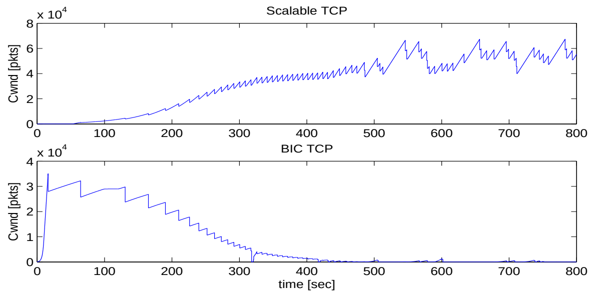


(b) BIC TCP on the longer path starts later

Figure 3.12: Simple parking-lot topology: Scalable TCP on the shorter path



(a) HSTCP on the shorter path starts later



(b) BIC TCP on the shorter path starts first

Figure 3.13: Simple parking-lot topology: Scalable TCP on the longer path

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.

An important property of Scalable TCP arises when it traverses the longer path. Illustrative results are shown in Figure 3.13. 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.

Results

Our main findings on the fairness characteristics of the MIMD mechanism, and specifically for Scalable TCP, are the following.

- The MIMD mechanism shows unfair behavior beside other loss-based flows in a wide range of network scenarios, and the AIMD-like flows using adaptive additive increase algorithms are always starved. 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. In addition, the spectral analysis showed that this property is rooted in the MIMD design principle (MI - Multiplicative Increase - in particular) of Scalable TCP.

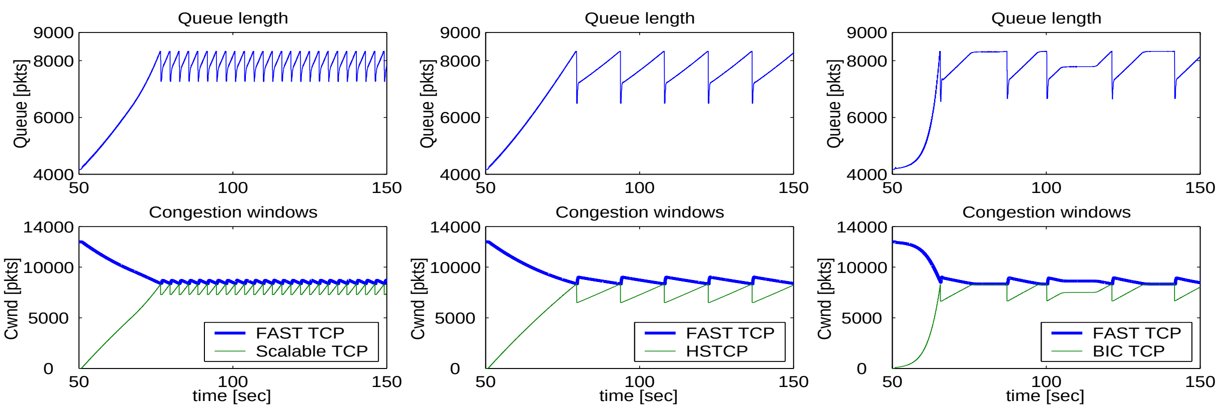
- The aggressive nature of the MIMD mechanism is exhibited in the parking-lot topology and in the competition against traffic aggregates, as well.
- On the one hand, the long-term interaction of MIMD and AIMD-like protocols is not affected by the starting delay. Starting delay can influence only the transient phase and the convergence time. On the other hand, the long-term performance of competing MIMD flows is directly affected by starting times and therefore, intra-protocol fairness cannot be achieved assuming synchronized losses (see Section 3.5).
- In addition, we have provided the explanations of the experienced phenomena based on the previously suggested metrics and methodology.

3.6.2 FAST TCP and 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.

Dumb-bell topology: single flows

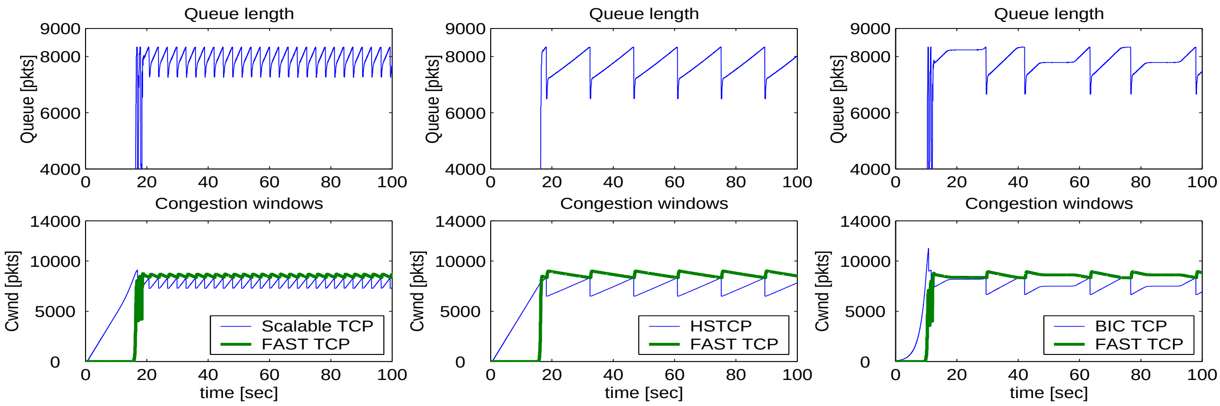
First, we focus on the competition of single flows in the dumb-bell topology. In our first scenarios, FAST TCP source starts the transmission and the other flow – using loss-based protocol – enters the network when the first one has achieved its maximal sending rate. The simulation results corresponding to a starting delay of 50 sec are presented in Figure 3.14 for Scalable TCP (a), HSTCP (b) and BIC TCP (c). During the transient phase, the delay-based decrease al-



(a) FAST TCP – STCP (b) FAST TCP – HSTCP (c) FAST TCP – BIC TCP

Figure 3.14: FAST TCP starts first, delay: 50s

gorithm of FAST TCP (roughly an additive decrease mechanism) interacts with the increasing



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

Figure 3.15: FAST TCP starts later, delay: 15s, 15s, 9s

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 sec delays are presented in Figure 3.15(a), (b) and (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 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,

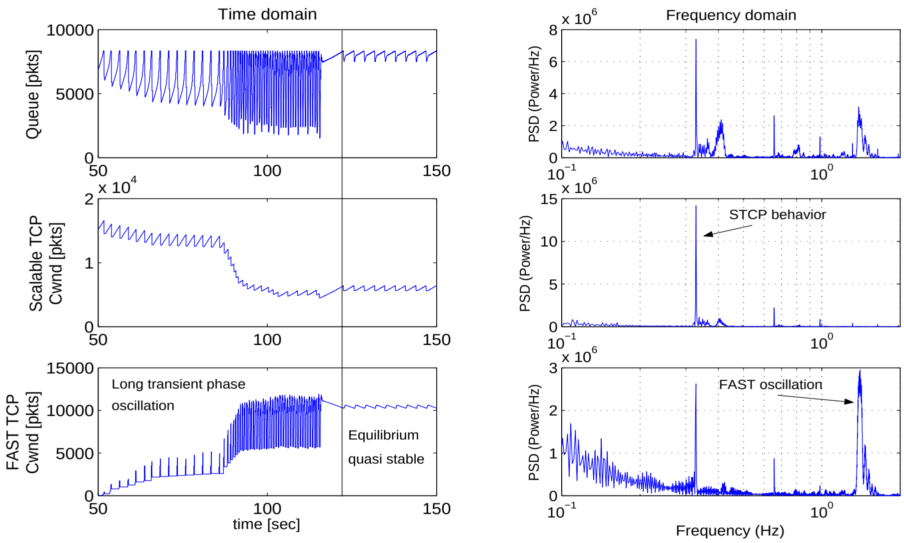


Figure 3.16: STCP – FAST TCP, delay: 50s

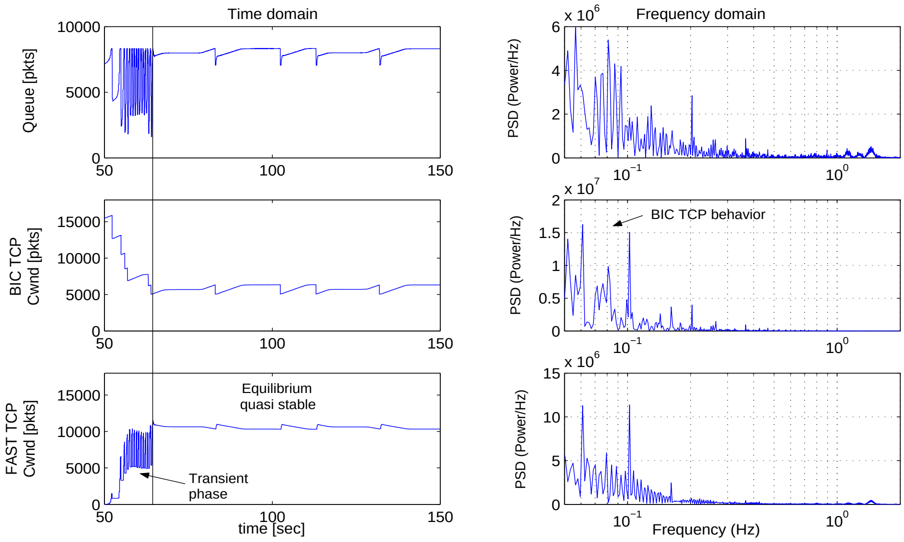


Figure 3.17: BIC TCP – FAST TCP, delay: 50s

first, the simulation results of the competition of a Scalable TCP flow (MIMD mechanism) and a FAST TCP flow corresponding to 50 sec delay are shown in Figure 3.16. 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 Figure 3.16. 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 ($\omega \approx 0.34\,1/s$) 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 3.6.2).

The interaction with AIMD-like protocols possesses similar characteristics. As an illustration, the simulation results of BIC TCP are presented in Figure 3.17. 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 depends on the starting time and other parameters of FAST TCP, as well. In all cases, the almost fair behavior is exhibited.

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.

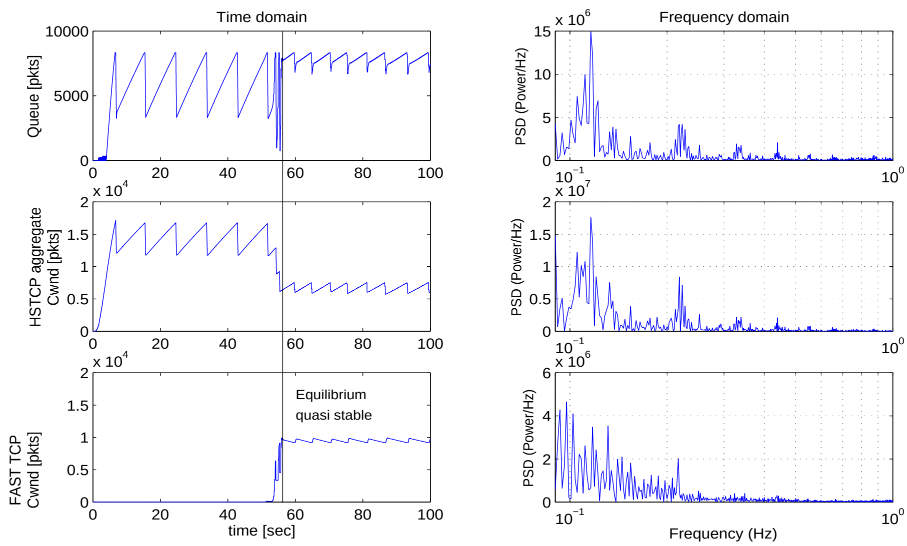


Figure 3.18: HSTCP aggregate – FAST TCP

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 Figure 3.18. In this scenario, the aggregate contains 10 HSTCP flows and the FAST TCP source enters with a delay of 50 sec. 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.

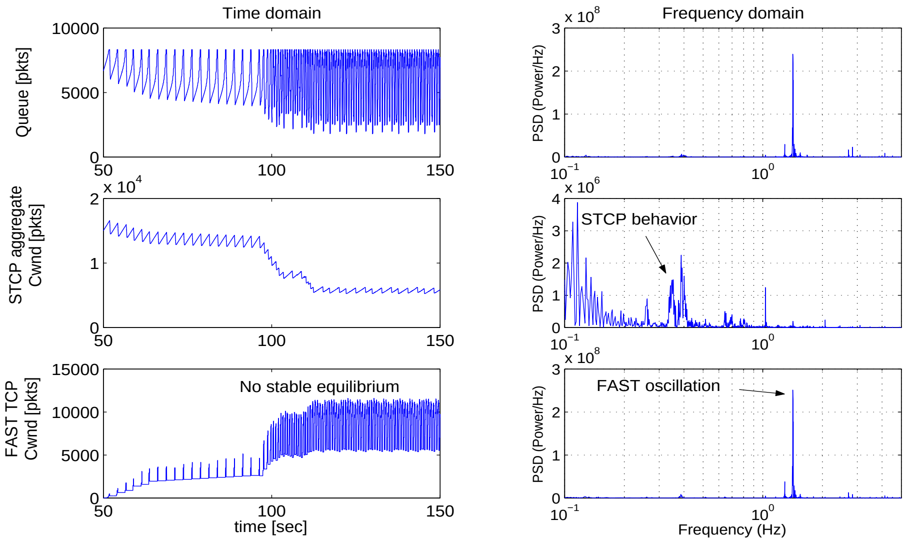


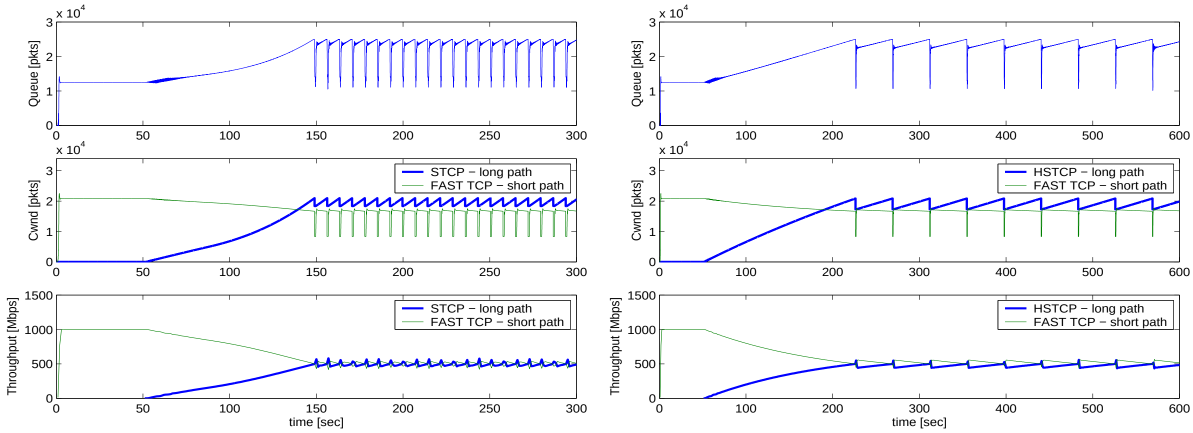
Figure 3.19: STCP aggregate – FAST TCP

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 Figure 3.19. (The scenario is similar to the previous one.) Here, the characteristics of the 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 are 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.

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.



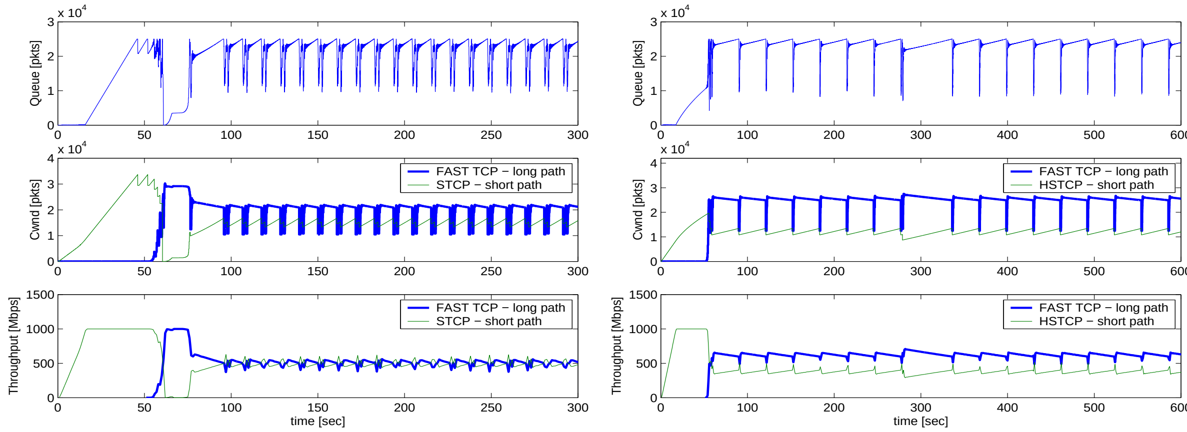
(a) Scalable TCP starts later (b) HSTCP starts later

Figure 3.20: Simple parking-lot topology: Loss-based protocol with longer RTT – FAST TCP

In the first scenarios, the FAST TCP flow traverses the shorter path. As an illustration, the simulation results are shown in Figure 3.20 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.

In the next scenarios, FAST TCP flow operates with longer round-trip times. The illustrative results are presented in Figure 3.21. Here, FAST TCP enters into the network 50 sec 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 Figure 3.21a), 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.



(a) Scalable TCP starts first (b) HSTCP starts first

Figure 3.21: Simple parking-lot topology: FAST TCP with longer RTT – loss-based protocol

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.

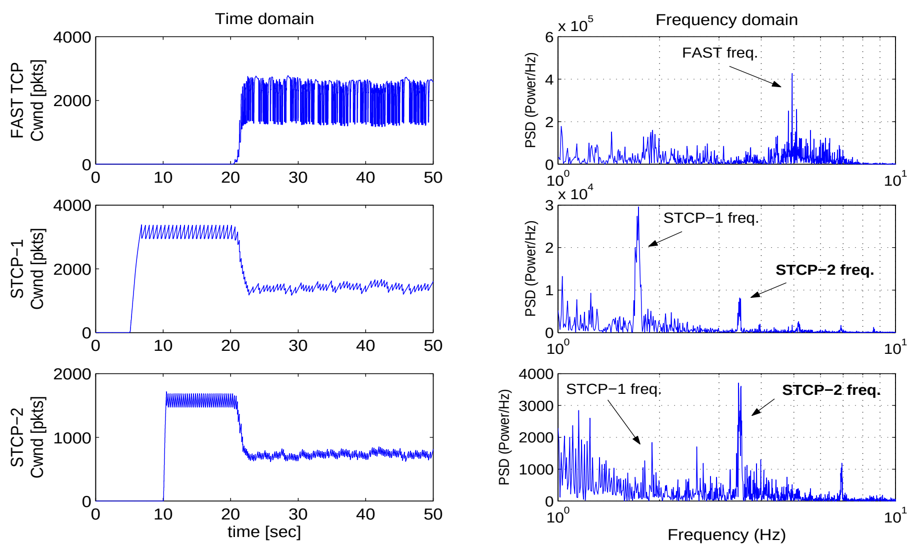


Figure 3.22: Inhomogeneous parking-lot topology: FAST TCP – STCP

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 Figure 3.22, 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 Figure 3.3c). 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 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 Figure 3.26b.

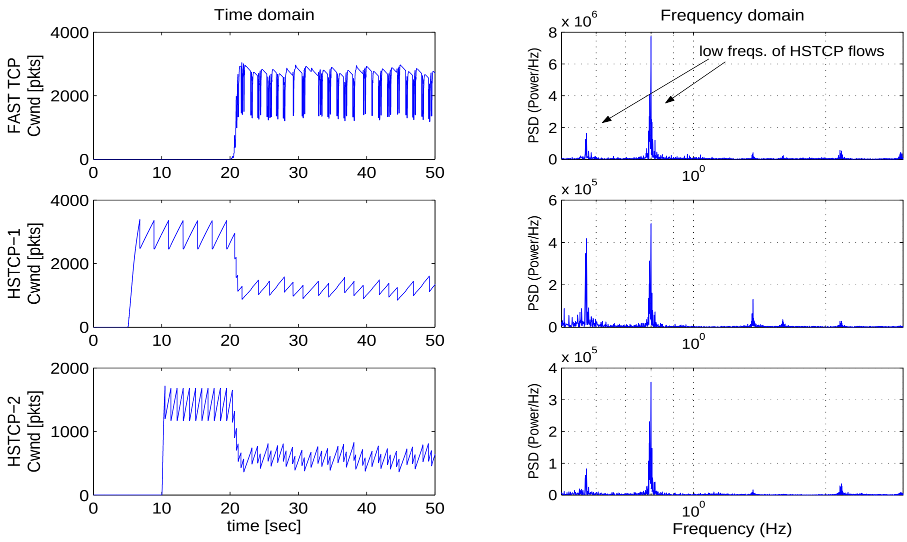


Figure 3.23: Inhomogeneous parking-lot topology: FAST TCP – HSTCP

Second, the fairness performance against AIMD-like flows is illustrated by the example of HSTCP. In Figure 3.23, 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.

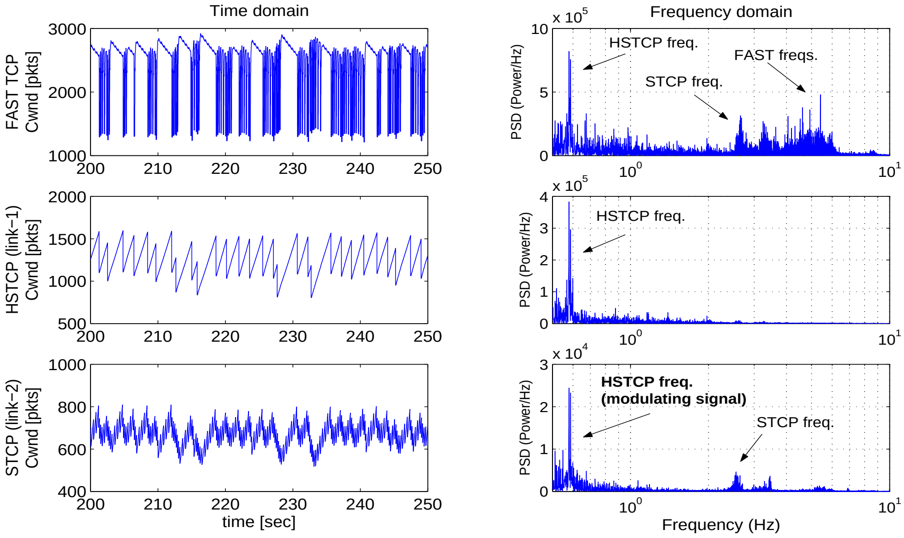
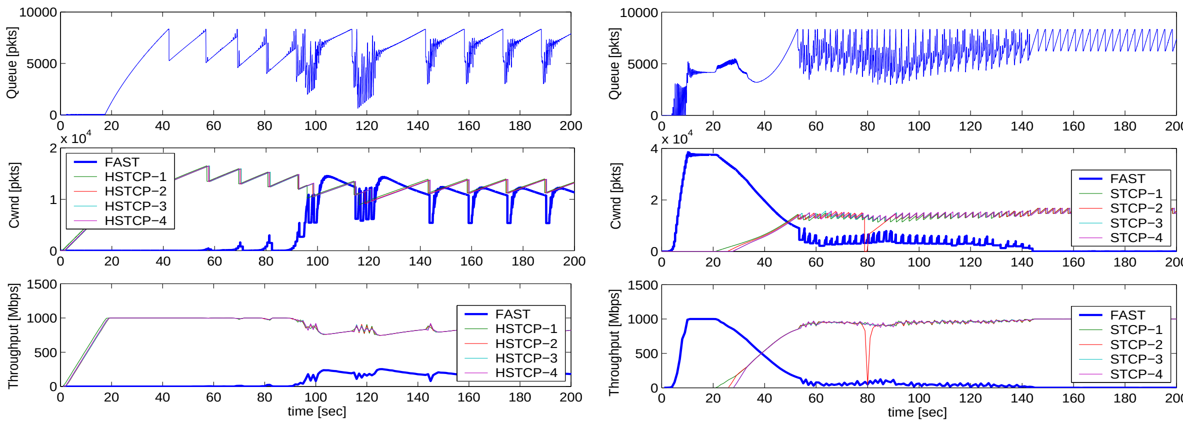


Figure 3.24: Inhomogeneous parking-lot topology: FAST TCP – HSTCP, STCP

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 Figure 3.24, a typical section of the 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.



(a) FAST TCP – HSTCP, FAST starts later (b) FAST TCP – Scalable TCP, FAST starts first

Figure 3.25: Complex parking-lot topology: 5 nodes, FAST TCP – loss-based protocols

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.

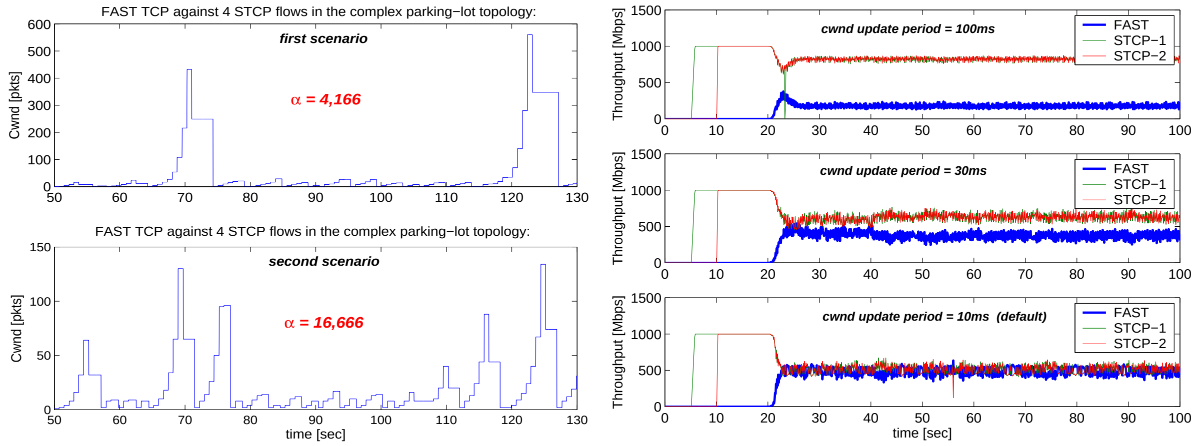
First, the interaction of a single FAST TCP traversing the backbone (four congested links) and four AIMD-like (here HSTCP) flows using separate links is shown in Figure 3.25a. The congestion window of FAST TCP can settle down around the same equilibrium state where the other flows operate. Therefore, 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 [75]).

Second, the interaction with MIMD flows is analyzed, and an illustration is given in Figure 3.25b. 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 in certain cases. However, this can yield unstable network behavior with degraded link utilization.

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 has been addressed in several papers (see e.g., [35, 61, 81]). 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 which yields very high loss rate. The unstable long-term



(a) FAST TCP against STCP flows in the complex parking-lot topology

(b) Impacts of cwnd update period: FAST TCP – STCP

Figure 3.26: Impacts of FAST parameters on fairness performance

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 Figure 3.26a, 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 Figure 3.25b), 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. ($\alpha = 16,000$ 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, 100ms (above the RTT), 30ms (around the RTT), and 10ms (below the RTT) time constants are investigated and shown in Figure 3.26. The default value in the Ns-2 implementation is 10ms (in [35], 20ms 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 update period is below the round-trip time of the FAST flow, while the too small values results 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.

Results

Our main findings on the fairness performance of the delay-based FAST TCP protocol in networks with a single bottleneck link 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 (such as HSTCP, Scalable TCP, and BIC TCP) in simple network environments with a single congested link.

- The long-term behavior of FAST TCP is similar in different scenarios showing a fair equilibrium state, however, concerning the dynamics of starting times this state is achieved by different ways. If the FAST TCP flow starts first then the equilibrium state can always be directly achieved, whereas, in case of a later entering FAST TCP flow, the equilibrium state is reached through an oscillating transient phase with a length depending on the starting time and the other protocol, as well. The spectral behavior of FAST TCP follows the periodic operation of the loss-based protocols.

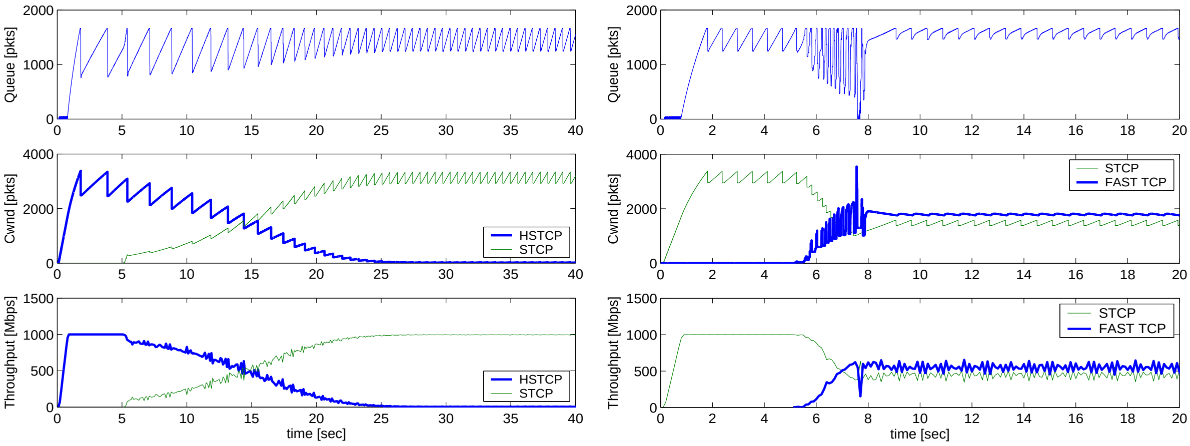
- In addition, this fair behavior of FAST TCP seems to be a robust property of the protocol and FAST TCP can achieve good utilization against aggregates of loss-based variants (HSTCP, Scalable TCP, and BIC TCP), and it can also share the bottleneck bandwidth fairly with single loss-based flows in simple parking-lot topology (with one congested link).

- With appropriate parameters, FAST TCP shows fair or almost fair intra-protocol fairness, however, characteristics of the interaction between FAST TCP flows depend on the starting time, the `baseRTT` estimation and other parameters of the protocol.

We have claimed the following statements on FAST TCP fairness properties when there are multiple congested links in the network:

- The beneficial fairness properties of FAST TCP still holds for certain network scenarios containing more than one congested links (appropriate parameters can be chosen), however, in general, the increasing number of bottleneck links results in performance degradation.

- The interaction is qualitatively different for MIMD and AIMD-like flows. When competing with AIMD-like flows then the spectral characteristic of FAST TCP at the equilibrium state is mainly determined by the AIMD-like flows. On the other hand, in case of MIMD flows, FAST TCP shows heavy oscillations or serious performance degradation depending on the network scenario. In general, we have concluded that the fairness performance of



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

Figure 3.27: Impacts of lower link delay

FAST TCP with AIMD-like flows is always better than with MIMD flows in case of multiple congested links.

- When FAST TCP traverses multiple links it is able to synchronize loss-based flows on separate links.

And finally, we have revealed the following impacts of FAST TCP parameters on fairness performance:

- 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. 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 which yields very high loss rate.

- The congestion window update period can also 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. A reasonable choice for the update period is below the round-trip time of the FAST TCP flow, while the too small values result in unnecessary oscillation.

3.7 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 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 Figure 3.27, 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 Figure 3.9 and 3.15).

3.8 Main results

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 3.3.) An overview of the analyzed scenarios and the observed/investigated properties is given in a self-explanatory table (see Table 3.4). 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.

Table 3.4: Main results

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

Analyzed protocols		
loss-based	MIMD	Scalable TCP
	AIMD-like	HSTCP
		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		×	×	×	✓	×

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. 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 3.6.2).

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 3.4). 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 3.4). 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 scenario of MIMD–AIMD interaction in Table 3.4). 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 3.4). *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.

3.9 Conclusion

In this chapter, 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. In addition, we introduced the *main operating frequencies* gained from the spectral characteristics of the congestion window processes as another novel metric that can be used to get an enhanced characterization of loss-based congestion control mechanisms.

We have derived analytical results on relevant parameters for Scalable TCP, HighSpeed TCP, BIC TCP and FAST 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 flows, when a flow force other flows to deviate 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 when there is a single bottleneck in the network. We have also shown that this behavior is rather robust property of the protocol concerning different traffic mix and wide range of network topologies. The beneficial fairness properties of FAST TCP still holds for certain network scenarios containing more than one congested links (appropriate parameters can be chosen), however, in general, the increasing number of bottleneck links results in performance degradation.

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.

Chapter 4

Stability Analysis of HSTCP/RED networks

HighSpeed TCP is a promising solution for transport protocol over very high bandwidth-delay product networks. However, little is known about its performance as well as its interaction with other elements of the network (such as the RED queue management). In this chapter, a comprehensive control-theoretic analysis of HighSpeed TCP is provided. We develop a fluid-flow model of the HighSpeed TCP/RED network and use it to study its performance. The main contributions of this research work are the following. Firstly, we provide a fluid-flow model for HighSpeed TCP/RED networks. Secondly, a comprehensive and *systematic* implementation methodology is described in detail. We also provide a Simulink-based framework for analyzing fluid-based models. Thirdly, we give stability conditions for HighSpeed TCP/RED networks. Finally, the results are validated by simulations using Ns-2.

This chapter is based on [73, 74, 76].

4.1 Introduction

Recent experience indicates that the congestion control of current TCP prevents it from fully utilize high-speed wide-area connections. Thus, network applications demanding high bandwidth are rarely able to take full advantage of high-speed networks and they are often not utilizing the available bandwidth. The major reason for under utilization is that the additive increase of traditional TCP is too slow and the multiplicative decrease is too harsh. In order to improve the performance of standard TCP Reno, several new TCP variants have been proposed recently as it is presented in Section 2.1. HighSpeed TCP (HSTCP) [18] is a promising solution due to its incremental improvement that makes the AIMD mechanism an adaptive and more or less scalable algorithm. It is specifically designed for use in networks with high bandwidth-delay product. However, there exist very few studies on the performance implication of HighSpeed

TCP so far (see e.g., [78, 87]). Neither its interaction with other elements of the network (such as the RED queue management) is well understood. These issues have motivated our work.

This research work provides a control-theoretic model to estimate the performance of HighSpeed TCP in a very high bandwidth-delay product network environment. The motivation behind our approach is to gain analytical insight into the performance of HighSpeed TCP. As it is presented in Section 2.4, control-theoretic approach proves to be a very promising tool to model the dynamics of traditional TCP/AQM networks (TCP/RED network in particular) [11, 29, 53, 59]. The main contributions of this research work are the following. First, we provide a fluid-flow model for HighSpeed TCP/RED networks. Second, a comprehensive and *systematic* implementation methodology is described in detail and a Simulink-based framework for analyzing fluid-based models is also designed and implemented. Third, we give stability conditions for HighSpeed TCP/RED networks that can be used to find adequate parameter settings for RED gateways. Finally, the analytical results are validated by packet-level simulations using Ns-2.

The rest of the chapter is organized as follows. In Section 4.2, the fluid-flow model of HighSpeed TCP/RED networks is given. Section 4.3 and 4.4 describes the Simulink implementation of the model and validates the results based on packet-level simulations, respectively. In Section 4.5, a control-theoretic stability analysis for HighSpeed TCP/RED networks is provided. Finally, a discussion on the possible extension of the framework is given in Section 4.6 and the conclusions of this chapter are drawn in Section 4.7.

4.2 Fluid-flow model

This section is devoted to summarize our fluid-flow model that captures the expected (average) transient behavior of a HSTCP/RED network. The model includes a single bottleneck link fed by identical HSTCP sources. Our model is based on the model presented in [29, 59] for TCP Reno/RED networks. We generalize this model for networks carrying HSTCP traffic.

The following variables are used in the rest of this chapter:

$$\begin{aligned}
W &\doteq \text{expected TCP window size (packets)},\\
q &\doteq \text{expected queue length (packets)},\\
x &\doteq \text{expected queue length estimation (packets)},\\
R &\doteq \text{round-trip time (RTT)} = \tfrac{q}{C} + T_p \text{ (secs)},\\
C &\doteq \text{link capacity (packets/sec)},\\
T_p &\doteq \text{fix round-trip propagation delay (secs)},\\
N &\doteq \text{load factor (number of TCP sessions)},\\
p &\doteq \text{probability of packet mark/drop}.
\end{aligned}$$

4.2.1 HighSpeed TCP source model

In [58], Poisson Counter Driven Stochastic Differential Equations are applied to model the dynamic behavior of regular TCP. This model is further developed in [59] in order to closely couple TCP sending rates and loss arrivals. Taking the expectation of the variables and the equations, and applying certain approximations, the expected transient behavior of TCP Reno/RED networks (including queue dynamics and the dynamics of the congestion windows) can be accurately captured in a wide range of network scenarios. A similar methodology can be applied for HSTCP and as a result, the expected (average) dynamics of HSTCP's congestion window can be established. In the analyzed network environment, including RED routers, this behavior can be described by the following differential equation:

$$\dot{W}(t) = \frac{a(W(t))}{R(t)} - b(W(t))\,W(t)\frac{W(t-R(t))}{R(t-R(t))}\,p(t-R(t)) \tag{4.1}$$

where the first term corresponds to the additive increase part of the algorithm using the increase parameter $a(W(t))$. This term expresses that the congestion window is increased by $a(W(t))$ packets per one round-trip time. The second term corresponds to the multiplicative decrease part depending on the decrease parameter $b(W(t))$. RED realizes a proportional marking scheme marking packets of flows according to the flows' bandwidth share. Thus, the decrease of the congestion window is weighted by the delayed rate $\frac{W(t-R(t))}{R(t-R(t))}$ and the marking probability $p(t-R(t))$. We emphasize that this differential equation gives an approximation for the expected dynamics of the congestion window.

The main difference between our HSTCP source model and the published TCP Reno model [59] originates from the fact that the increase and decrease parameters of the HSTCP protocol depend on the current value of congestion window ($W(t)$) which yields a more complicated differential equation than the one for TCP Reno.

4.2.2 Network model

Considering a network with one single bottleneck link fed by identical TCP flows, the expected transient behavior of the queue can be captured by the following differential equation [59]:

$$\dot{q}(t) = N(t)\frac{W(t)}{R(t)} - 1_{q(t)}C \tag{4.2}$$

where the first term reflects the increase of the queue length according to the arrival rate and the second term reflects the decrease part depending on the service rate. $1_{q(t)} = 1$ if $q(t) > 0$, and zero otherwise. There is an AQM (Active Queue Management) policy associated with this router. We focus on the classical RED (Random Early Detection) [23] policy characterized by a packet discard function $p(x)$ taking the average queue length estimation $x(t)$ as its argument. At packet level simulation, e.g., in Ns-2 [64], the average queue length is computed after every packet arrival applied an exponentially weighted moving average. This can be described by a *difference* equation that can be approximated by the following *differential* equation [59]:

$$\dot{x}(t) = -Kx(t) + Kq(t) \tag{4.3}$$

where $K = -C \ln(1-\alpha)$ and α is the forgetting factor. This is a first order low pass filter with a cutoff frequency of K. The approximation of the difference equation is based on the assumption that in the steady-state (if the queue is stable) the packet arrival rate is equal to the service rate. Thus, the sampling period can be estimated by the reciprocal of the capacity.

The AQM module acting as a controller (using control-theoretic terms) feeds back the congestion measure (marking or dropping probability) to the TCP senders according to its parameters. The TCP sources as parts of the controlled plant adjust their sending rates based on the experienced marking probability.

The equations (4.1), (4.2) and (4.3) describing the HighSpeed TCP sources and the network dynamics, respectively, form coupled differential equations modeling the HSTCP/RED network. This system of differential equations due to the complex dependence between the variables and containing variable delay in some arguments, is analytically not tractable. Thus, we apply numerical approximation to solve these complex equations.

4.3 Implementation of the model

As far as we know there is no detailed explanations about the numerical approximation methods used to solve related differential equations modeling TCP/RED networks. In this section, we illustrate our new, *systematic* approach to implement the previously presented system of differential equations.

The system described by non-linear differential equations has been implemented in the Simulink environment of MATLAB. It includes blocks describing the behavior of each part and we get a clear and tractable framework. The basic elements of the model is presented in Figure 4.1.

The HSTCP source is modeled by the block called "Cwnd dynamics" which captures the behavior of the HSTCP's congestion window control algorithm. The size of the congestion window W is set according to the packet-marking probability p and the round-trip time R. The marking probability is derived from the instantaneous queue length q by the AQM module, while the current round-trip time is originated from the queue length. The dynamics of the queue length is affected by the current window size, round-trip time and the system load N (number of sources). The input of the system is the load (N_input). The key network variables are saved to MATLAB workspace by the corresponding elements.

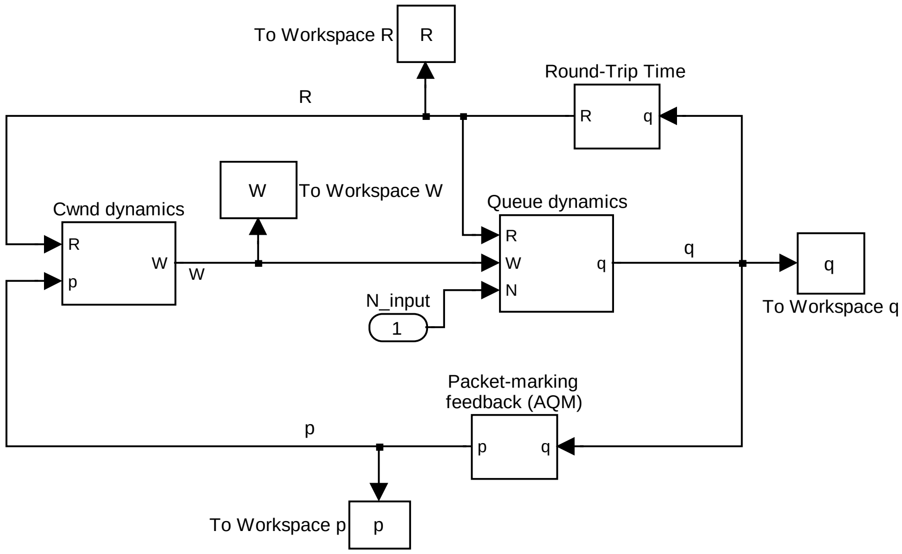


Figure 4.1: Basic elements of the model (top-level)

4.3.1 Implementation of HSTCP source model

The building blocks of the subsystem representing the dynamics of HSTCP's congestion window are presented in Figure 4.2. The increase and decrease parameters are derived by an implemented MATLAB function and affects the additive increase and multiplicative decrease algorithms according to the specification of HighSpeed TCP [18].

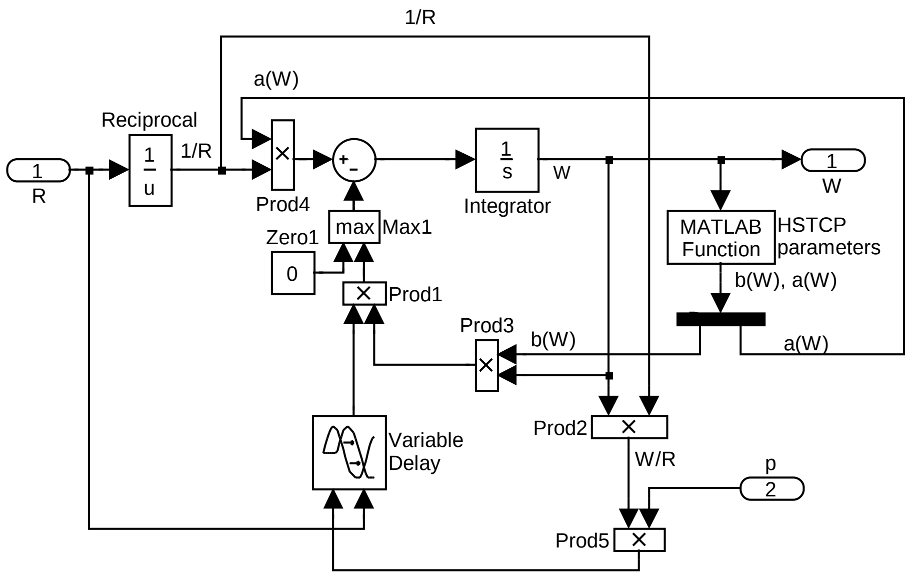
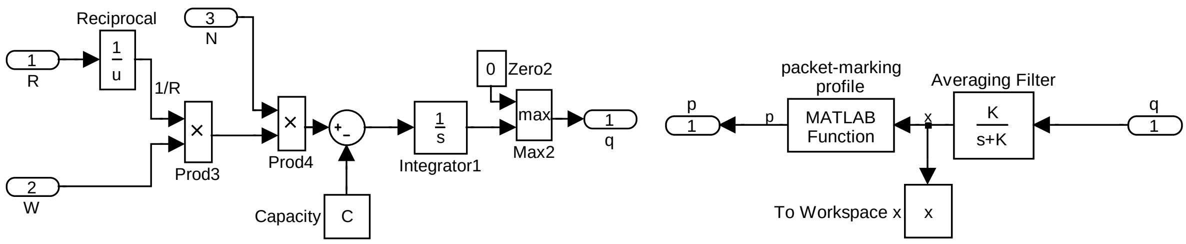


Figure 4.2: Dynamics of Cwnd - HighSpeed TCP

4.3.2 Implementation of network model

The block belonging to the single bottleneck queue dynamics is shown in detail in Figure 4.3a. It realizes the corresponding differential equation (4.2).



(a) Queue dynamics (b) AQM module

Figure 4.3: Elements of the network model

The module realizing the AQM policy is shown in Figure 4.3b. It consists of a low-pass filter averaging the instantaneous queue length and a marking function associated with the RED marking profile (see in Figure 2.1). The output of this subsystem is the packet marking probability p.

The round-trip time module implements a simple correspondence between the queue length and the RTT according to the following equation:

$$R = T_p + \frac{q}{C}. \tag{4.4}$$

4.4 Validation of the model

The previously introduced model has been analyzed using Simulink's imbedded differential equation solver tools with different parameters and it has been validated by simulations, as well, conducted in the Ns-2 [64] simulation environment. In this section, the main results of this analysis are presented.

During our investigation, the "ode45" differential equation solver is used which implements the Domand-Prince algorithm (numeric approximation). The investigated network scenario contains identical TCP flows feeding a single queue belonging to the bottleneck link. The parameters of the network scenario used for the validation of the model, namely the network parameters, the parameters of HSTCP and RED (packet-marking profile), are summarized in Table 4.1. In this work, the default parameter set of HSTCP is used [18], and a single link with high bandwidth-delay product is examined. (We note that in [76], we used another parameter set. Here, the same network parameters are chosen as it is used in Chapter 3.)

As a simple example, the results of a single flow scenario are presented first. We compare the congestion window and the instantaneous queue length of Simulink flow-level simulation (model)

Table 4.1: Parameters of the network scenario used for validation

Network parameters

capacity	1 Gbps
delay	50 ms
packet size	1,500 bytes

HSTCP parameters

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

RED parameters

p_{max}	0.01
t_{min}	4,000
t_{max}	8,000
$w_q = \alpha$	0.002

with Ns-2 packet-level simulation in Figure 4.4. Since we focus on the Congestion Avoidance phase, only the steady-state results are shown. (In Ns-2, the Slow Start mechanism is active from the beginning of a connection until a threshold is reached. This behavior is modeled by an initial condition of the congestion window that is related to that threshold.) It can be observed that the analytic model is capable of accurately capturing the oscillating dynamics of the single HSTCP flow and the bottleneck queue. It is worth noting that an individual TCP-like flow shows oscillation not only at the packet-level but at the flow-level, as well.

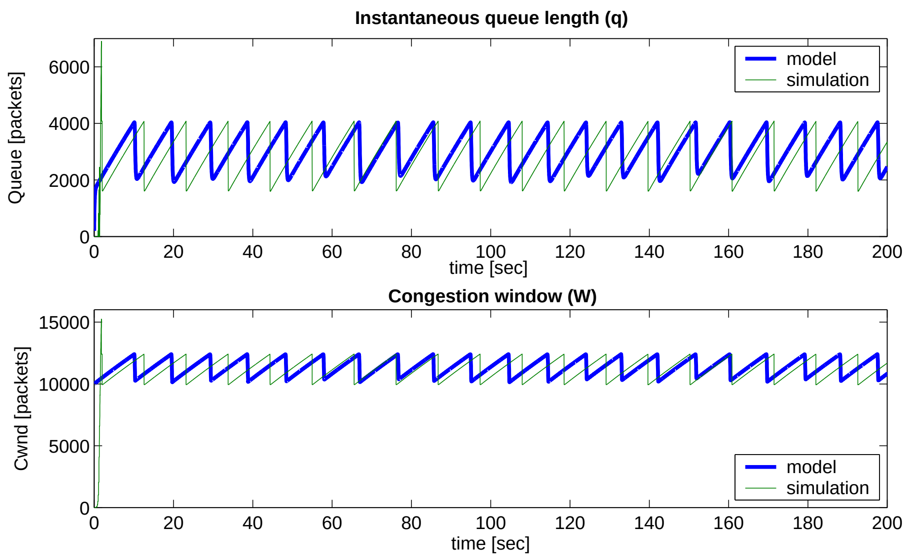


Figure 4.4: Validation of the model - single HSTCP flow

The model is also analyzed in a more realistic scenario, more specifically, when the bottleneck link is shared among 100 HSTCP flows. Two different settings of queue weight or forgetting factor (w_q or q_{weight} in the packet-level simulation and α in the model) are shown here as an illustration. Figure 4.5 presents the outcome of the model and the simulation regarding the instantaneous queue length at the bottleneck. The results are able to validate the accuracy of the model and also emphasize the importance of the appropriate setting of RED parameters. For $\alpha = 0.002$, the bottleneck queue exhibits a (quasi) stable behavior (see the top part of Figure 4.5), whereas $\alpha = 0.00001$ yields an oscillating system (see the bottom part of Figure 4.5). In the rest of the chapter, we carry out a stability analysis in order to provide an analytical stability condition for

HSTCP/RED networks that can be used in RED design.

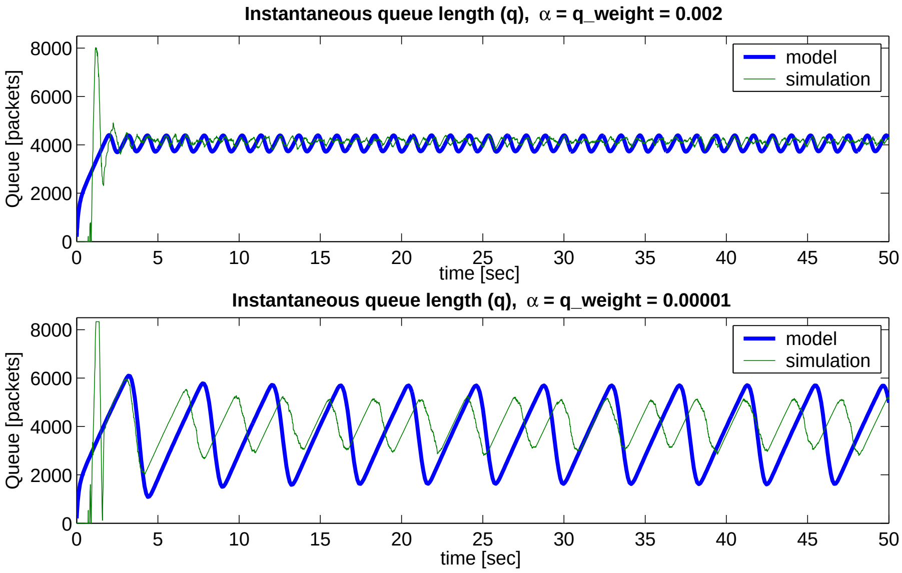


Figure 4.5: Validation of the model – 100 HSTCP flows: $\alpha = 0.002$ (top) and $\alpha = 1e{-}5$ (bottom)

4.5 Stability analysis

In control-system language, the RED module can be referred as the controller or compensator and the rest of the system as the plant. The objective of the controller design is to provide a stable and robust closed-loop system with acceptable transient response. In order to give a stability condition on HSTCP/RED networks, we apply the following methodology. The stability of the system described by our non-linear fluid-flow model (see Section 4.2) is difficult to analyze directly. Thus, we linearize the model about an equilibrium point first, and the *global asymptotic stability* of the linear system is analyzed. Then we give a condition for RED parameters to stabilize the linear feedback control system for a given range of parameters. Based on the condition, the *local asymptotic stability* of the original system (and the physical system) with reasonable region of attraction can be inferred. Finally, the results are validated and illustrated by packet-level simulations (corresponding to another non-linear system).

A similar stability analysis is carried out for TCP Reno/RED networks in [29].

4.5.1 Characterizing stability

First, we summarize various notions of stability and presents some useful results on Lyapunov stability theory. For further details, see [40, 63, 79].

Intuitively and crudely speaking, we say an equilibrium point is locally stable if all solutions which start near the equilibrium point x^* remain near x^* for all time. Precisely, the mathematical definitions are the following:

Definition 1 (Stability in the sense of Lyapunov) *The equilibrium point $x^* = 0$ of a dynamical system satisfying*

$$\dot{x} = f(x,t) \quad x(t_0) = x_0 \quad x \in \mathbb{R}^n \tag{4.5}$$

is stable in the sense of Lyapunov at $t = t_0$, if for any $\epsilon > 0$ there exists a $\delta(t_0, \epsilon) > 0$ such that

$$\|x(t_0)\| < \delta(t_0, \epsilon) \quad \Rightarrow \quad \|x(t)\| < \epsilon, \quad \forall t \geq t_0. \tag{4.6}$$

Uniform stability is a concept which guarantees that the equilibrium point is not losing stability. We insist that for a uniformly stable equilibrium point x^*, δ not be a function of t_0.

The definition of asymptotic stability states that the trajectory eventually reaches the equilibrium point. It can be defined precisely as follows:

Definition 2 (Asymptotic stability) *An equilibrium point $x^* = 0$ is asymptotically stable at $t = t_0$ if*

1. *$x^* = 0$ is stable , and*

2. *$x^* = 0$ is locally attractive; i.e., there exists $\delta(t_0)$ such that*

$$\|x(t_0)\| < \delta(t_0) \quad \Rightarrow \quad \lim_{t \to \infty} x(t) = 0. \tag{4.7}$$

Similarly, *uniform asymptotic stability* requires that $x^* = 0$ is uniformly stable, and uniformly locally attractive. Further, it is required that the convergence in Equation 4.7 is uniform. For time-invariant systems, stability implies uniform stability and asymptotic stability implies uniform asymptotic stability.

Global asymptotic stability strengthens this further by requiring asymptotic stability starting from any initial condition:

Definition 3 (Global asymptotic stability) *We say an equilibrium point x^* is globally stable if it is stable for all initial conditions $x_0 \in \mathbb{R}^n$.*

In general, for a physical system, the global asymptotic stability is to be satisfied. The goal of the controller design is to make the feedback system stable in the sense of global asymptotic stability. When we use stability, usually this notion of stability is referred.

There is a strong form of stability which demands an exponential rate of convergence:

Definition 4 (Exponential stability, rate of convergence) *The equilibrium point $x^* = 0$ is an exponentially stable equilibrium point if there exist constants m, $\alpha > 0$ and $\epsilon > 0$ such that*

$$\|x(t)\| \leq m e^{-\alpha(t-t_0)} \|x(t_0)\|$$

for all $\|x(t_0)\| \leq \epsilon$ and $t \geq t_0$. The largest constant α which may be utilized is called the rate of convergence.

Exponential stability is a strong form of stability implying uniform asymptotic stability.

Lyapunov theorem gives *sufficient* conditions on different notions of stability. If an appropriate scalar function, referred as Lyapunov candidate or Lyapunov function, of the system can be found then the stability of the system can be inferred from some properties of this function. More exactly, the Lyapunov theorem states the following.

Theorem 1 (Lyapunov theorem) *Consider a continuously differentiable function $V(x,t)$ such that*

$$V(x,t) > 0, \qquad \forall x \neq 0$$

and $V(0,t) = 0$. Now we have the following conditions on various notions of stability.

1. *If $\dot{V}(x,t) \leq 0\ \forall x$, then the equilibrium point is (Lyapunov) stable.*
2. *In addition, if $\dot{V}(x,t) < 0\ \forall x \neq 0$, then the equilibrium point is asymptotically stable.*
3. *In addition to the above conditions, if V is radially unbounded, i.e., $V(x,t) \to \infty$ when $||x|| \to \infty$, then the equilibrium point is globally asymptotically stable.*

Lyapunov theorem only gives sufficient conditions, and it could be difficult to find the appropriate candidate function V for general non-linear systems.

On the other hand, the stability theory of linear time-invariant systems is well known in the literature, and both necessary and sufficient conditions are provided. To check if a linear system is stable, it is enough to verify that the poles of the transfer function $L(s)$ lie in the complex left half-plane. However, to find the roots of the transfer function in general, could also be difficult. Nyquist criterion provides and alternate condition that could be verified in an easier way in most cases, and it is given in the following theorem [24, 79].

Theorem 2 (Nyquist criterion) *Let C be a closed contour in the complex plane such that no zeros or poles of $L(s)$ lie on C. Let Z and P denote the number of zeroes and poles of $L(s)$, respectively, that lie inside C. Then, the number of times that $L(j\omega)$ encircles the origin in the clockwise direction, as ω is varied from $-\infty$ to ∞, is equal to $Z - P$.*

Nyquist criterion can be applied for the stability analysis of systems with negative feedback, as well. In this case, the stability of a closed-loop system can be inferred from the open-loop transfer function. More exactly, let $P(s)$ and $C(s)$ denote the transfer function of the plant and the controller, respectively. Then, the closed-loop transfer function $L(s)$ is in the form

$$L(s) = \frac{P(s)}{1 + P(s)C(s)} = \frac{P(s)}{1 + H_0(s)}, \tag{4.8}$$

where the denumerator is referred as the the characteristic equation of the system, and $H_0(s)$ is the open-loop transfer function. To check the stability of the system, the roots (zeroes) of the characteristic equation should be analyzed instead of checking the poles of $L(s)$. Let Z and P denote the number of zeroes and poles, respectively, of $1+H_0(s)$ in the complex right half-plane. In terms of Theorem 2, we want $Z=0$. Applying the Nyquist criterion with the contour being the imaginary axis and fictitious half-circle connecting $+j\infty$ and $-j\infty$ that lies entirely on the right half-plane, we get the following conditions on the asymptotic stability of the linear feedback system. (We suppose that $H_0(s)$ has no poles on the imaginary axis; for further details, see [24]).

1. If $P=0$, then the feedback system is asymptotically stable $\Leftrightarrow$ there are no encirclements of the point $-1+j0$ by the plot of $H_0(j\omega)$ as ω is varied from $-\infty$ to ∞

2. If $P>0$, then the feedback system is asymptotically stable $\Leftrightarrow$ the point $-1+j0$ is encircled P times in the counterclockwise direction by the plot of $H_0(j\omega)$ as ω is varied from $-\infty$ to ∞

These criteria can be checked in the Bode diagram, as well. In simple cases (when $P=0$), this means that the closed-loop system is stable when the Bode diagram of the open-loop system shows positive phase margin at the cut-off frequency. Cut-off frequency or crossover frequency (ω_c) is the frequency when the magnitude Bode plot reaches the unit gain, while the phase margin is $\phi_m = 180° + \phi_c$, where ϕ_c denotes the phase angle at the cut-off frequency.

4.5.2 Linearization of the model

The plant – including HSTCP sources and bottleneck queue – is modeled by differential equations (4.1), (4.2). Taking W and q as state variables and p as input of the plant, the operating point (W_0, q_0, p_0) is given by $\dot{W}=0$ and $\dot{q}=0$. Using small-signal linearization, we assume that load factor and round-trip delay are constants:

$$N(t) \equiv N \quad \text{and} \quad R(t) \equiv R_0 = \frac{q_0}{C} + T_p.$$

Moreover, the increase and decrease parameters of HSTCP can be considered similarly:

$$a(W(t)) \equiv a(W_0) = a_0 \quad \text{and} \quad b(W(t)) \equiv b(W_0) = b_0.$$

a_0 and b_0 can be derived directly from the basic parameters (Low_Window, $High\_Window$, $High\_P$ and $High\_Decrease$) of HSTCP (see [18] for more details):

$$b(W_0) = \frac{(High\_Dec - 0.5) \times (\log W_0 - \log Low\_W)}{\log High\_W - \log Low\_W} + 0.5 \tag{4.9}$$

$$a(W_0) = \frac{W_0^2 \times p(W_0) \times 2 \times b(W_0)}{2 - b(W_0)} \tag{4.10}$$

where

$$\begin{aligned} p(W_0) &= \exp\left\{\frac{\log W_0 - \log Low\_W}{\log High\_W - \log Low\_W} \cdot \right. \\ & \qquad \left. (\log High\_P - \log Low\_P) + \log Low\_P \right\} \\ Low\_P &= \frac{1.5}{Low\_W^2}. \end{aligned}$$

Substituting zero for $\dot{W}$ and $\dot{q}$, the following equations can be derived for the operating point:

$$W_0^2 p_0 = \frac{a_0}{b_0} \quad \text{and} \quad W_0 = \frac{R_0 C}{N}. \tag{4.11}$$

Taking the partial derivatives of the right hand side of equation (4.1) and (4.2), we linearize the system about the operating point:

$$\begin{aligned} \delta\dot{W}(t) &= -\frac{a_0 N}{R_0^2 C}\left(\delta W(t) + \delta W(t - R_0)\right) \\ & \quad -\frac{b_0 R_0 C^2}{N^2}\delta p(t - R_0) \\ \dot{\delta q}(t) &= \frac{N}{R_0}\delta W(t) - \frac{1}{R_0}\delta q(t) \end{aligned} \tag{4.12}$$

where the variables denote perturbations ($\delta W \doteq W - W_0$, $\delta q \doteq q - q_0$ and $\delta p \doteq p - p_0$).

This linear system can be transformed into the Laplace transform domain and the transfer function can be derived:

$$\begin{aligned} P(s) &= -e^{-sR_0}\, P_{hstcp}(s)\ P_{queue}(s) = \\ &= -e^{-sR_0}\, \frac{\frac{b_0 R_0 C^2}{N^2}}{s + \frac{a_0 N}{R_0^2 C}\left(1 + e^{-sR_0}\right)}\ \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}} \end{aligned} \tag{4.13}$$

where the first term e^{-sR_0} corresponds to a delay of R_0 and $P_{hstcp}(s)$ and $P_{queue}(s)$ describe the behavior of HSTCP and the queue, respectively. The delay term (e^{-sR_0}) that appears in the denominator of the transfer function $P_{hstcp}(s)$ can be eliminated by a similar approximation as it was used in [29]. The subtle difference to be taken into consideration is the changed condition that makes the approximation acceptable. For regular TCP, this condition requires that $W_0 \gg 1$. In our case, the new requirement

$$W_0 \gg a_0 \tag{4.14}$$

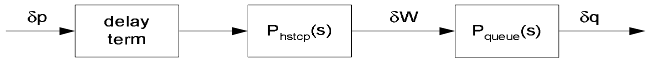
is also a reasonable assumption for typical network conditions.

Thus, we get a simplified linear model as it is shown by the block diagram in Figure 4.6.

Remarks:

1. The negative eigenvalues of HSTCP and queue dynamics ($-\frac{2a_0 N}{R_0^2 C}$, $-\frac{1}{R_0}$) indicate that the equilibrium state of the non-linear dynamics is locally asymptotically stable.

2. If we substitute the increase and decrease parameters of HSTCP $a_0 = 1$ and $b_0 = 0.5$ we get the linear model of the network with regular TCP Reno sources.



$$P_{hstcp}(s) = \frac{\frac{b_0 R_0 C^2}{N^2}}{s + \frac{2a_0 N}{R_0^2 C}} \quad ; \quad P_{queue}(s) = \frac{\frac{N}{R_0}}{s + \frac{1}{R_0}}$$

Figure 4.6: The simplified linear model

4.5.3 Designing RED for HighSpeed TCP

The previously presented plant – including the HSTCP sources and the bottleneck queue – and the RED controller form a feedback control system. The transfer function of the RED controller can be modeled in a range of queue length as follows [29]:

$$C(s) = C_{red}(s) = \frac{L}{\frac{s}{K} + 1} \tag{4.15}$$

where

$$L = \frac{p_{max}}{t_{max} - t_{min}} \quad \text{and} \quad K = -\frac{\ln(1-\alpha)}{\delta} \approx -C\ln(1-\alpha),$$

α is the queue averaging parameter (or forgetting factor) and δ is the sample time which is approximated by $1/C$ in steady-state [59]. L is the slope of the curve characterizing the RED marking profile, whereas K is the cutoff frequency of the RED controller.

The objective of our RED design for a network with HSTCP sources is to select RED parameters L_{hstcp} ($= \frac{p_{max}}{t_{max}-t_{min}}$) and K_{hstcp} to stabilize the feedback control system for a given range of N and R_0. That range can be defined as follows:

$$N \geq N^- \quad \text{and} \quad R_0 \leq R^+.$$

For the sake of clarity, we summarize the used parameters:

L_{hstcp}, K_{hstcp}	:	RED-based control system parameters (explained above)
a_0, b_0	:	HSTCP increase/decrease parameters at the operating point
C	:	Capacity of the link
N^-	:	Minimum number of flows
R^+	:	Maximum value of the RTT

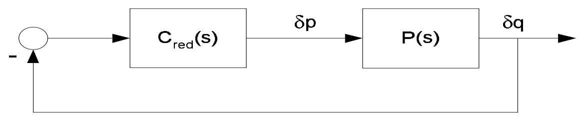
Proposition 1 (Stability condition) *If L_{hstcp} and K_{hstcp} satisfy*

$$\frac{L_{hstcp}\, b_0\, (R^+ C)^3}{2a_0\, (N^-)^2} \leq \sqrt{\frac{\omega_g^2}{K_{hstcp}^2} + 1} \tag{4.16}$$

where

$$\omega_g = 0.1 \min\left\{\frac{2a_0 N^-}{(R^+)^2 C}, \frac{1}{R^+}\right\} \tag{4.17}$$

then, the linear feedback control system showed in Figure 4.7 is stable for all $N \geq N^-$ *and* $R_0 \leq R^+$.



$$P(s) = \frac{\frac{b_0 C^2}{N} e^{-sR_0}}{\left(s + \frac{2a_0 N}{R_0^2 C}\right)\left(s + \frac{1}{R_0}\right)} \quad ; \quad C_{red}(s) = \frac{L_{hstcp}}{\frac{s}{K_{hstcp}} + 1}$$

Figure 4.7: Block diagram of the linearized feedback control system

This condition gives a stability region for the HSTCP/RED network, hence, a set of parameters within that region gives stable system. In the condition, ω_g is an upper bound of the crossover frequency and the constant factor 0.1 is a design parameter that provides the stability margins and the dominance of the RED controller's time constant in the closed-loop behavior.

Proof: The frequency response function of the loop transfer function is

$$L(j\omega) = \frac{L_{hstcp}\frac{b_0 (R_0 C)^3}{2a_0 N^2} e^{-j\omega R_0}}{\left(\frac{j\omega}{K_{hstcp}} + 1\right)\left(\frac{j\omega}{\frac{2a_0 N}{R_0^2 C}} + 1\right)\left(\frac{j\omega}{\frac{1}{R_0}} + 1\right)}. \tag{4.18}$$

The goal of this controller design is to force the RED module ($C_{red}(s)$) to dominate closed-loop behavior which is achieved by making the closed-loop time constant ($\approx 1/\omega_g$) greater than the maximum of the other two time-constants $\left(\frac{(R^+)^2 C}{2a_0 N^-},\ R^+\right)$ of the transfer function. Thus, the following approximation can be applied

$$L(j\omega) \approx \frac{L_{hstcp}\frac{b_0 (R_0 C)^3}{2a_0 N^2} e^{-j\omega R_0}}{\frac{j\omega}{K_{hstcp}} + 1} \quad \text{for} \quad \forall \omega \in [0, \omega_g]. \tag{4.19}$$

For a given range of N and R_0 we get an upper bound for the gain at ω_g:

$$|L(j\omega)| \leq \frac{L_{hstcp}\,\frac{b_0 \left(R^+ C\right)^3}{2a_0 (N^-)^2}}{\sqrt{\frac{\omega_g^2}{K_{hstcp}^2} + 1}}. \tag{4.20}$$

From this and (4.16) it follows that $|L(j\omega_g)| \leq 1$ for all $N \geq N^-$, $R_0 \leq R^+$. Thus, the crossover frequency ω_c (where $L(j\omega_c) = 1$) is bounded above by ω_g which yields

$$\angle L(j\omega_c) \geq \angle L(j\omega_g) = \angle \frac{L_{hstcp}\frac{b_0 (R_0 C)^3}{2a_0 N^2}}{\frac{j\omega_g}{K_{hstcp}} + 1} - \omega_g R_0 \geq -90^\circ - 0.1\frac{180^\circ}{\pi} > -180^\circ \tag{4.21}$$

where we used the condition (4.17). We get that $\angle L(j\omega_c) \geq -180°$ indicating the stability based on Nyquist stability criterion (see Theorem 2). ■

Remarks:

1. It is important to note that if we substitute the increase and decrease parameters of HSTCP $a_0 = 1$ and $b_0 = 0.5$ in (4.16) and (4.17) we get the conditions for the network with regular TCP Reno sources [29].

2. Stability margins of the feedback control system can be derived in a similar way as it was given for networks with regular TCP [29]. Thus, if the RED parameters satisfy conditions (4.16) and (4.17) in Proposition 1, then a lower bound can be derived for the gain margin (GM) and for the phase margin (PM) of the linear control system, respectively, as follows:

$$GM \geq 5\pi \quad ; \quad PM \geq 85° \tag{4.22}$$

   and the proof from [29] can be applied with the same considerations.

3. In the stability condition, the increase and decrease parameters of HSTCP at the operating point (a_0 and b_0) can be approximated using the steady-state value of congestion window that can be estimated based on the bandwidth-delay product.

4.5.4 Numerical examples

In this section, some numerical results are given in order to illustrate the above presented analytical stability condition. Based on the stability condition (Proposition 1) for HSTCP/RED networks, a stability region can be defined. Taking Equations (4.16) and (4.17), the RED parameter L_{hstcp} can be expressed in the terms of N^- and R^+ assuming that K_{hstcp} is fixed as follows

$$L_{hstcp} \leq \frac{2a_0\,(N^-)^2}{b_0\,(R^+C)^3}\sqrt{\frac{\omega_g^2}{K_{hstcp}^2}+1}. \tag{4.23}$$

The parameters of the network and HSTCP are the same as it is shown in Table 4.1 while a_0 and b_0 can be approximated based on the bandwidth-delay product and ω_g is given by Equation 4.17. Therefore, the stability region can be plotted in 3D. In Figure 4.8, the maximum values of RED parameter L_{hstcp} yielding stable behavior according to the stability condition is plotted in terms of N^- and R^+. Thus, the values of L_{hstcp} below the plotted surface give stable system according to Proposition 1.

As an illustration, three different scenarios with single bottleneck link are examined and the results of the fluid-flow model and Ns-2 simulations are compared. Here, the RED parameters are fixed ($K_{hstcp} = -C\ln(1-\alpha) = 166.83$, $L_{hstcp} = 2.5 \cdot 10^{-6}$), and only the number of flows is varied.

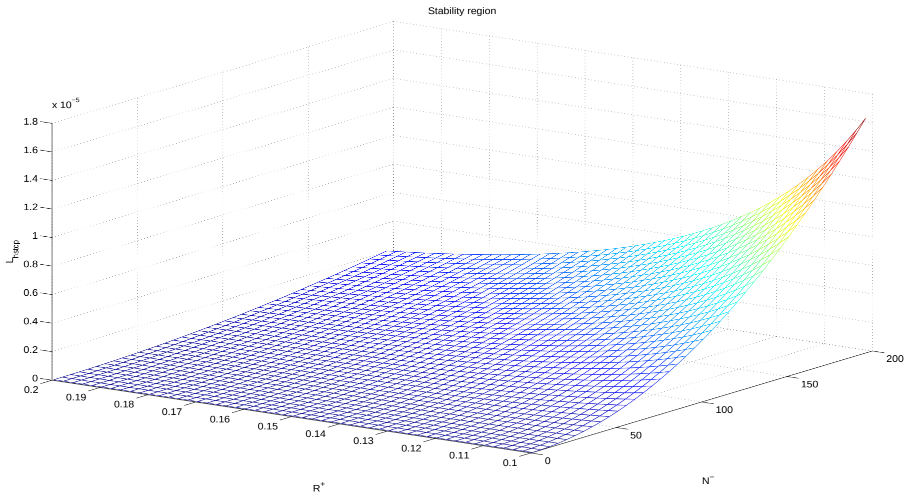


Figure 4.8: Stability region of HSTCP/RED network

Firstly, a parameter set is chosen that yields an unstable system. The number of HSTCP flows feeding the same queue belonging to the single bottleneck link is $N = 10$. In this case, the stability condition predicts instability. The instantaneous queue length derived from the model and the simulation are shown in the upper part of Figure 4.9. The steady-state characteristics of the queue dynamics and the *severe oscillation* are well captured by the model, however, the transient behavior shows difference due to the Slow Start mechanism that is not considered by our model. The permanent oscillation exhibited by the non-linear fluid-flow model regards the expected value of the queue length process. The amplitude of the oscillation does not degrade in time which shows that the asymptotic stability is not satisfied.

Increasing the flow number, the feedback system moves toward the stable operation. In the second scenario, $N = 50$ is chosen which results in a *moderate oscillation* of queue length as it is shown in the center part of Figure 4.9. However, the asymptotic stability is not satisfied by the model and the amplitude of the oscillation remains approximately constant in time. More exactly, the oscillation can be observed at a higher frequency and a lower amplitude than previously. We note that the vertical axis of this plot (and the next one) is scaled from 3000 to 5000 in order to provide a better view on the steady-state behavior. According to the stability condition, the limit of the stable behavior (with safe margins) can be observed at around $N = 160$. The bottom part of the figure, shows the queue behavior from the model and the simulation when it is fed by 165 HSTCP flows. The *asymptotic stability* of the steady-state of the model is well demonstrated by the plot as the amplitude of the oscillation is decreasing continually in time until it reaches zero. In the equilibrium state, the expected queue length settles down at an exact queue length. On the other hand, the queue process from the packet-level simulation does not settle down exactly but shows perturbation around the same equilibrium state. The difference

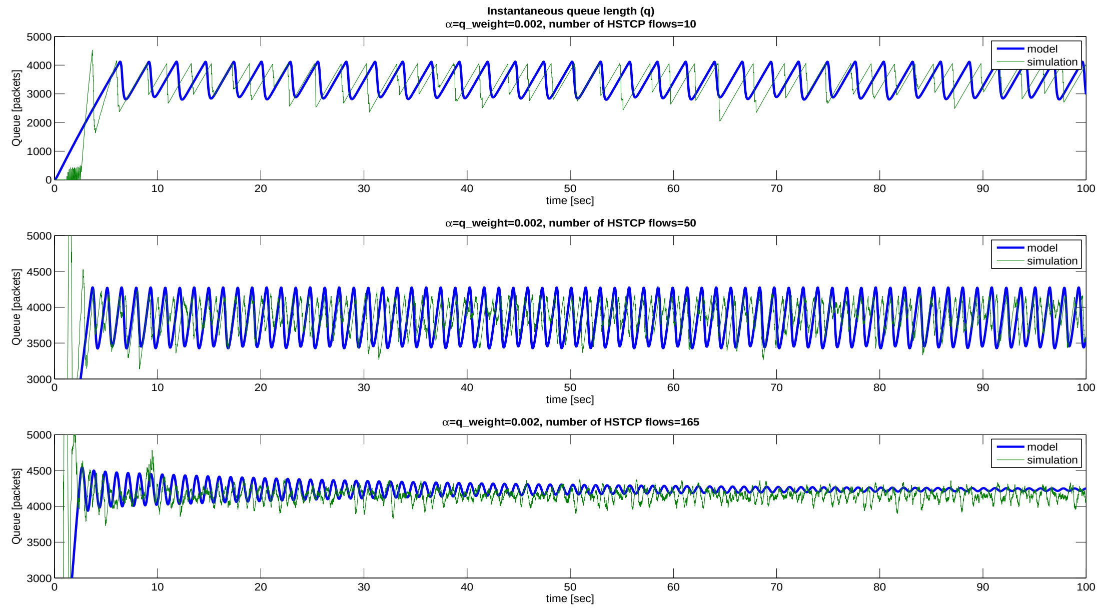


Figure 4.9: Queue dynamics from model and simulation: 10 flows (top), 50 flows (center), and 165 flows (bottom)

in the transient characteristics can be observed in this case, as well.

The stability condition can be used in the design of RED parameters for given network conditions, as well. For example, in the previous scenario, when there exist $N = 50$ HSTCP connections in the network, the closed-loop system shows oscillation. This phenomenon can be compensated by better choice of RED parameters. For this network scenario, the stability condition suggests much lower values of L_{hstcp} than the current one ($L_{hstcp} = 2.5 \cdot 10^{-6}$). Of course, decreasing the gain parameter of RED, we get a slower controller. This gain parameter can be decreased by different ways. One solution is reducing the p_{max} parameter which yields a lighter slope in the marking profile (see Figure 2.1). Another possibility is expanding the interval of the early drop ($t_{max} - t_{min}$). In the bottom part of Figure 4.10, $L_{hstcp} = 5 \cdot 10^{-7}$ is achieved by setting p_{max} to 0.002. As it is expected from the stability condition, the queue is stabilized by the RED controller. Here, the parameters are chosen to get an asymptotically stable behavior of the model which results in significantly reduced oscillation at the packet-level. In the center part of the figure, the original behavior is presented again while the upper part shows the impact of a non adequate parameter setting ($p_{max} = 0.05$ and $L_{hstcp} = 1.25 \cdot 10^{-5}$) which results in severe oscillation.

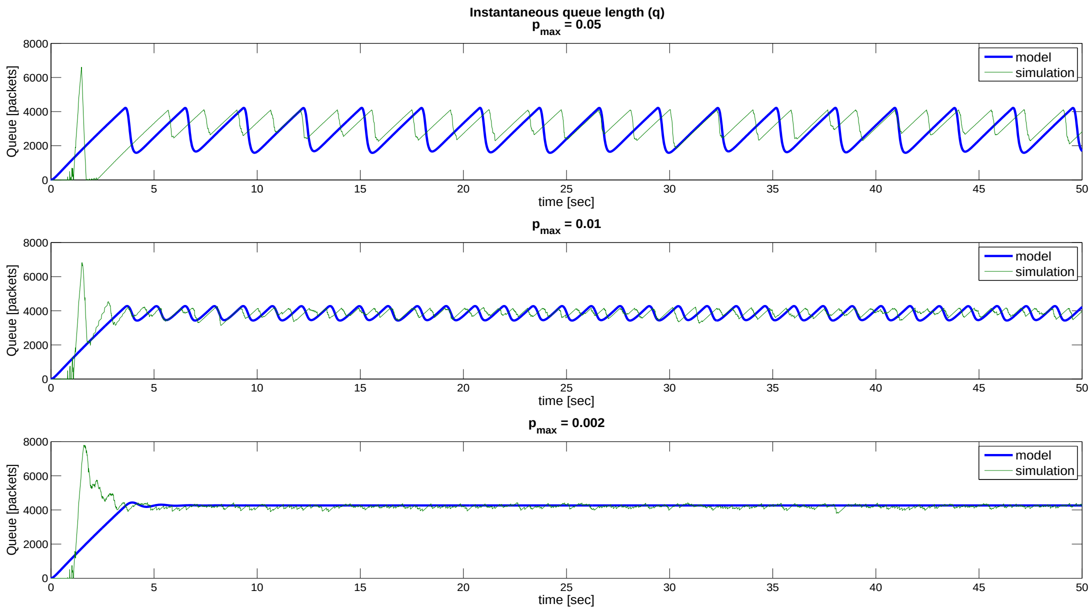


Figure 4.10: Queue dynamics from model and simulation: $p_{max} = 0.05$ (top), $p_{max} = 0.01$ (center), and $p_{max} = 0.002$ (bottom)

4.6 Extension of the framework

The presented framework can easily be extended with other TCP protocols, as well. For example, the MIMD behavior of an individual Scalable TCP sender can be modeled by a similar way as it has been shown for HSTCP in Section 4.2.1:

$$\dot{W}(t) = \frac{aW(t)}{R(t)} - bW(t)\,\frac{W(t-R(t))}{R(t-R(t))}\,p(t-R(t)). \tag{4.24}$$

The first term of the differential equation describes the multiplicative increase part of the control mechanism while the second term corresponds to the multiplicative decrease mechanism. Here, the increase (a) and decrease (b) parameters are constant values, and the first term (capturing the window increase) explicitly depends on the current value of the congestion window. In [73], we have implemented this model in the Simulink framework and analyzed, as well.

Furthermore, the model is capable of examining the interaction of different congestion control mechanisms at the flow-level. In the top-level model, different TCP algorithms feed the bottleneck queue and the marking probability is fed back to both sources as it is shown in Figure 4.11. In [73], we presents also some experimental results on the co-operation of HighSpeed TCP (AIMD mechanism) and Scalable TCP (MIMD mechanism).

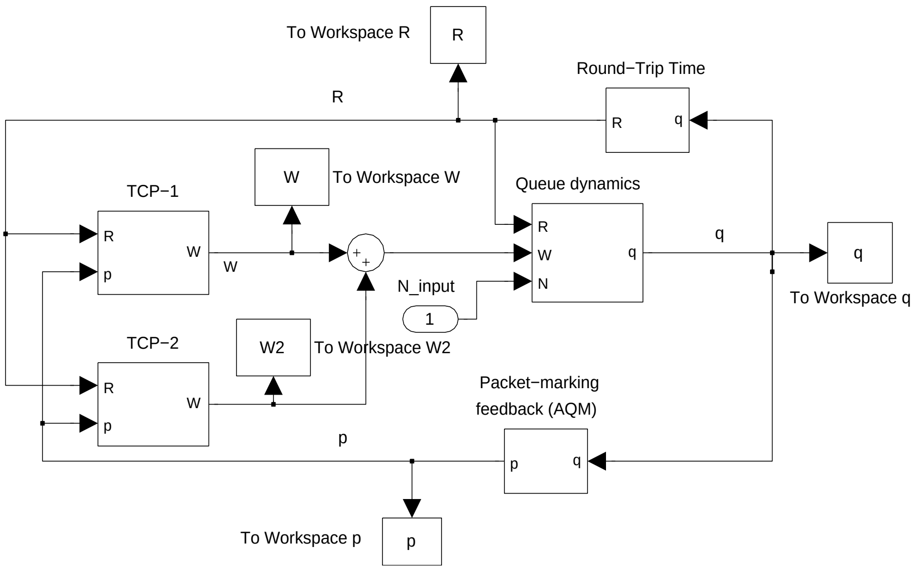


Figure 4.11: Basic elements of the model (top-level)

4.7 Conclusion

In this chapter, a control-theoretic analysis of HighSpeed TCP/RED network has been carried out. We have proposed a fluid-flow model for HighSpeed TCP and investigated its performance in a very high bandwidth-delay product network with RED active queue management at the router. We have described a comprehensive and systematic implementation methodology in detail to get the solutions of the system of non-linear differential equations with given initial conditions. We have also designed and implemented a MATLAB/Simulink-based framework for analyzing fluid-based models applying numerical approximation based differential equation solvers. Based on the model we have derived the stability conditions for HighSpeed TCP network regulated by RED queueing mechanism at the router. The model as well as results are validated through packet-level simulations by using Ns-2.

Our analysis raises the issue but it leaves many questions unaddressed. In the future, we plan to extend the current model to study other high speed variants of TCP, such as FAST TCP and XCP. We also plan to use the model to study fairness issues of these protocols in a very high bandwidth-delay product network environment.

Chapter 5

Application of New Results

The comprehensive fairness analysis presented in Chapter 3 provides a deeper understanding of recently proposed protocols regarding fairness characteristics. As several proposals have been implemented under various operating systems, the question of co-existence of these protocols is raised accordingly. The performance of the applications and user experience are also significantly affected by these mechanisms. In this work, advantages and drawbacks of recent proposals are revealed in a wide range of network scenarios. Based on the results and the presented methodology, the new TCPs can be characterized from different aspects, and eventually, the deployability of these mechanisms can be evaluated. These results could be applied in future transport protocol design or in methods choosing adequate parameters of certain protocols (e.g., FAST TCP) for given network environments.

The flow-level model established in Chapter 4 is able to estimate the dynamic performance of the network elements, such as HSTCP senders and RED routers, and to understand the behavior of the network as a whole. The provided stability condition could be used in network design and dimensioning. More specifically, based on the criterion, the RED parameters could be configured in order to get stable behavior when the network carries HSTCP traffic. Furthermore, the presented framework makes it possible to include other transport protocol models and evaluate the individual operation and the inter-operability of them at the flow-level.

Chapter 6

Conclusion

TCP congestion control had managed successfully the stability of the Internet in the past decades but it has reached its limitations in "challenging" network environments. Serious drawbacks of standard TCP Reno can be experienced in high speed wide are networks. These networks can be characterized by high bandwidth-delay product (BDP) and TCP cannot efficiently utilize them due to its conservative congestion control scheme. As a response to this problem, the research community has proposed several new transport protocols recently referred as high speed TCPs. The first part of the dissertation (see Chapter 2) was devoted to give a detailed overview on the most important TCP versions, especially on high speed variants, and related active queue management techniques. The related papers and research works on TCP performance and stability analysis were also presented here.

In Chapter 3, a comprehensive fairness analysis of promising TCP proposals and congestion control mechanisms designed for high bandwidth-delay product networks was carried out. We put emphasis not only on the equilibrium behavior but also on the transient characteristics with the dynamic behavior to gain an in-depth insight into the interaction and co-existence of different congestion control schemes. We revealed the analytic properties of the transient and dynamic behavior of individual flows of different TCP protocols and defined novel metrics in order to characterize them. The dynamic characteristics of competing flows were also investigated in a wide range of network scenarios. We investigated how the transient state can influence the long-term fairness performance. As part of this transient analysis, we deeply analyzed the impacts of starting time on the performance and fairness. Here, the main focus was on inter-protocol properties when different TCP versions are interacting. The explanations of the experienced phenomena were also given based on the novel metrics and methodology.

Chapter 4 was devoted to a comprehensive control-theoretic analysis of HSTCP to estimate the performance of the protocol in a very high bandwidth-delay product network environment where the queues are regulated by the RED active queue management. The motivation behind our approach was to gain analytical insight into the performance of HSTCP. To achieve this goal, firstly, we established a fluid-flow model for HSTCP/RED networks. Secondly, we described a systematic implementation methodology in detail in order to make the non-linear model tractable. The results of the model were validated by packet-level simulations. Thirdly, a stability condition was derived for HSTCP/RED networks that can be applied in RED design and parameter setting.

Finally, in Chapter 5 the possible application of the new results in protocol design and network dimensioning was outlined briefly.

Appendix A

Initial Dynamics

The notation of the variables and the parameters of the packet-level simulations used in Appendix A are summarized in Table A.1.

Table A.1: Used parameters and variables

Notations	Parameters	Values
C	capacity	$83,333$ pkts/sec
B	buffer size	$8,333$ pkts
R_0	round-trip propagation delay	0.1 sec
R or $R(t)$	round-trip time	variables
W or $W(t)$	congestion window	
q or $q(t)$	queue length	

A.1 Dynamics of Slow-Start

A discrete model of Slow-Start algorithm on a time scale of one RTT can be given as follows:

$$W[k] = 2^k.$$

A continuous approximation of the congestion window can be expressed as follows:

$$W(t) = 2^{t/R},$$

where W is the congestion window (`cwnd`), while R is the round-trip time. Actually, RTT is a function of queueing delay (RTT = propagation time + queueing delay). As a first and simple approach, RTT can be approximated by the propagation delay ($R(t) \approx R_0$).

The Slow-Start phase ends when `cwnd` reaches a threshold (`ssthresh`) or a packet loss can be detected. (`ssthresh` has an initial value in the beginning of a connection.) The time (t_{th})

needed to reach the threshold can be expressed as follows:

$$\begin{aligned} W(t_{th}) &= \texttt{ssthresh} \\ t_{th} &= R_0 \log_2 \texttt{ssthresh} = R_0 \frac{\lg \texttt{ssthresh}}{\lg 2} \end{aligned}$$

The current sending rate can be approximated by

$$x(t) \approx \frac{W(t)}{R(t)}$$

The link saturation time (when the sending rate reaches the bottleneck capacity and the filling of the buffer starts) t_0 can be expressed by the following equations:

$$\begin{aligned} x(t_0) &= C \\ \frac{W(t_0)}{R_0} &= C \\ t_0 &= R_0 \frac{\lg R_0 C}{\lg 2}, \end{aligned}$$

where C denotes the bottleneck capacity. At this time the congestion window has reached the bandwidth-delay product (BDP) which is equal to R_0C. After t_0 the bottleneck queue increases. The first drop occurs at time instance t_{drop} when the buffer size B is reached. The queue dynamics ($q(t)$) in the increasing phase can be expressed in the terms of `cwnd`:

$$\begin{aligned} q(t) &= \int_{t_0}^{t} (x(\tau) - C)\, d\tau \\ &= \int_{t_0}^{t} \frac{2^{\tau/R_0}}{R_0} d\tau - \int_{t_0}^{t} C d\tau \\ &= \frac{2^{t/R_0}}{\lg 2} - \frac{2^{t_0/R_0}}{\lg 2} - Ct + Ct_0. \end{aligned}$$

t_{drop} can be determined by solving the following equation for t:

$$\begin{aligned} q(t) &= B \\ \frac{2^{t/R_0}}{\lg 2} - Ct + \left(Ct_0 - \frac{2^{t_0/R_0}}{\lg 2} - B \right) &= 0. \end{aligned}$$

In this case, the source detects the loss with a delay t_{delay} (queueing delay and one-way propagation delay), thus, the Slow-Start phase ends slightly later than t_{drop}.

The time instance t_{drop} can be approximated by the following simple formula assuming that the congestion window is equal to the sum of the bandwidth-delay product and the buffer size at the saturation time:

$$t_{drop} \approx R_0 \log_2 (R_0 C + B) = R_0 \frac{\lg(R_0 C + B)}{\lg 2}. \tag{A.1}$$

Thus, the end of the Slow-Start phase t_{ss} can be expressed by the minimum of the above derived values:

$$t_{ss} = \min\left\{t_{th}, t_{drop} + t_{delay}\right\}.$$

Configuring initial Slow-Start threshold to be 100 packets, the end of Slow-Start phase is triggered by the event that `cwnd` exceeds `ssthresh`. During our analysis, we always assume this initial value of `ssthresh`.

A.2 Dynamics of Limited Slow-Start

In Limited Slow-Start phase, `cwnd` is increased by at most `max_ssthresh`/2 per round-trip time. In the simulations, the parameter is set to the proposed value (100 packets). It can easily be derived analytically [19] that

$$\log_2(\texttt{max\_ssthresh}) + \frac{\texttt{cwnd} - \texttt{max\_ssthresh}}{\texttt{max\_ssthresh}/2}$$

RTT is needed to reach `cwnd` (that is greater than the threshold parameter) where the first term corresponds to the standard Slow-Start phase.

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 protocol version. As `cwnd` increases, there will be a state (W_{LSS}) when the increment of LSS (denoted by $1/K$) and the increment of Scalable TCP's or HSTCP's algorithm will be equal triggering the end of this phase.

On the one hand, a Scalable TCP source exits from LSS when $1/K = a$. The value of `cwnd` (W_{LSS}) corresponding to this state can be easily expressed as follows

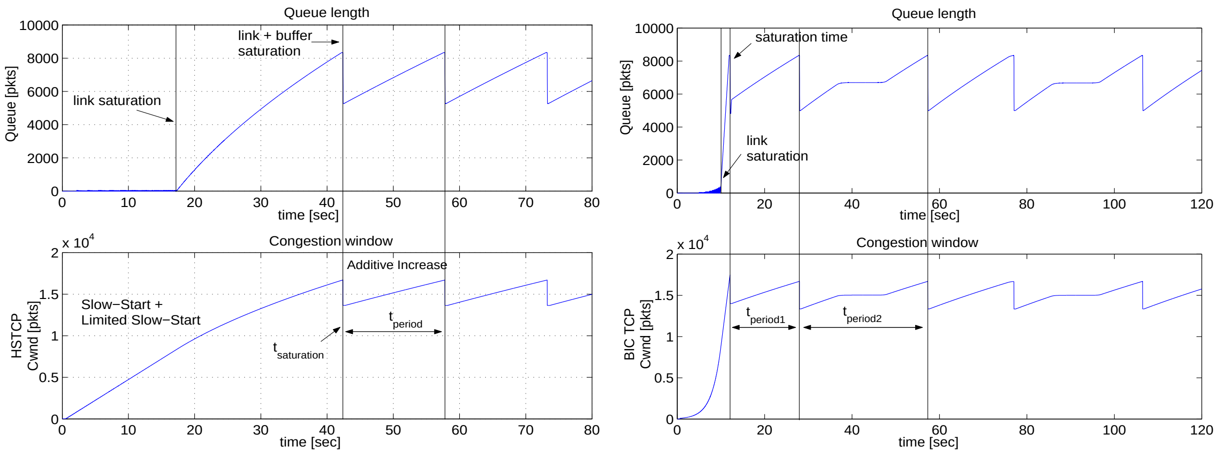
$$\begin{aligned}
\frac{1}{K} &= a \\
\mathrm{int}\left(\frac{W}{\texttt{max_ssthresh}/2}\right) &= \frac{1}{a} \\
W_{LSS} &= \frac{\texttt{max_ssthresh}}{2}\frac{1}{a},
\end{aligned}$$

where a is the increase parameter of Scalable TCP. Our parameters give that the end of LSS phase is expected to be around $W_{LSS} = 5,000$ at $t \approx 10.46$ sec.

On the other hand, a HSTCP source exits from LSS when the increment of HSTCP's additive increase algorithm ($a(W)/W$) will be greater than the increment of LSS ($1/K$). Hence, W_{LSS} can be derived as follows:

$$\begin{aligned}
\frac{1}{K} &= \frac{a(W)}{R} \\
\frac{\texttt{max_ssthresh}/2}{W} &= \frac{a(W)}{W} \\
a(W_{LSS}) &= \texttt{max_ssthresh}/2,
\end{aligned}$$

where $a(W_LSS)$ is the `cwnd` dependent increase parameter of HSTCP. In our scenario, $W_{LSS} \approx 29,000$. This high value of `cwnd` can not be reached with the computed simulation parameters resulting in HSTCP source operating in LSS till the first packet drop. Thus, the initial behavior is determined by Slow-Start and Limited Slow-Start algorithms.



(a) HSTCP (b) BIC TCP

Figure A.1: Saturation time and equilibrium behavior of HSTCP and BIC TCP

A.3 HSTCP – saturation time

In case of individual HSTCP flow, the time of the first packet drop (saturation time) can similarly be determined as it was outlined for Scalable TCP in Section 3.4.1. The link saturation time can be expressed as follows:

$$t_0 = R_0 \frac{\lg \texttt{max\_ssthresh}}{\lg 2} + R_0 \frac{R_0 C - \texttt{max\_ssthresh}}{\texttt{max\_ssthresh}/2} \approx 17.13\,\text{sec}. \tag{A.2}$$

Determining the saturation time, a similar approximation can be used as it was applied for Scalable TCP:

$$\hat{t}_{saturation} = t_0 + t^*,$$

where

$$t^* = \tilde{R} \frac{R_0 C + B - R_0 C}{\texttt{max\_ssthresh}/2} = \tilde{R} \frac{B}{\texttt{max\_ssthresh}/2}. \tag{A.3}$$

Our parameters yield that $\hat{t}_{saturation} = 42.13$ sec which meets well the simulation results as it is shown in Figure A.1a.

A.4 BIC TCP – saturation time

The initial behavior of an individual BIC TCP can be treated in a similar way. The source starts sending according to the Slow-Start mechanism. Exceeding the threshold, BIC TCP performs a multiplicative increase algorithm till the first packet drop. During this phase, `cwnd` is increased by $a = 0.05$ for each acknowledgement with our simulation settings.

The link saturation time in this case can also be determined easily according to the following equation:

$$t_0 = R_0 \frac{\lg \texttt{max\_ssthresh}}{\lg 2} + R_0 \frac{\lg \frac{R_0 C}{\texttt{max\_ssthresh}}}{\lg(1+a)} \approx 9.73\,\text{sec}. \tag{A.4}$$

Using a similar approximation method as it was outlined previously, the saturation time can be derived, as well.

$$\hat{t}_{saturation} = t_0 + t^*, \tag{A.5}$$

where

$$t^* = \tilde{R} \frac{\lg \frac{R_0 C + B}{R_0 C}}{\lg 1 + a}, \tag{A.6}$$

where

$$\tilde{R} = R_0 + B/(2C). \tag{A.7}$$

Using this approximation, we get that $\hat{t}_{saturation} = 11.86$ sec which approximates well the simulation results (see Figure A.1b).

A.5 FAST TCP – saturation time

Far from the equilibrium state of `cwnd`, FAST TCP converges exponentially to that equilibrium performing a Slow-Start-like multiplicative increase algorithm. If queueing delay exceeds a threshold (which is a constant parameter of the protocol in the Ns-2 implementation), the AIAD-like control algorithm is used instead of multiplicative increase.

$$\hat{t}_{sat} = R_0 \frac{\lg R_0 C}{2} + t_{AIAD} \approx 2\,\text{sec}. \tag{A.8}$$

In our simulation environment, FAST TCP (with $\alpha = 4166$) reaches the equilibrium approximately after 2 seconds.

Bibliography

[1] E. Altman, K. E. Avrachenkov, and 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.

[2] L. Andrew, C. Marcondes, S. Floyd, L. Dunn, R. Guillier, W. Gang, L. Eggert, S. Ha, and I. Rhee, “Towards a common TCP evauation suite,” in *Proceedings of Sixth International Workshop on Protocols for FAST Long-Distance Networks (PFLDnet 2008)*, Manchester, UK, Mar. 5-7, 2008.

[3] S. Athuraliya, V. H. Li, S. H. Low, and Q. Yin, “REM: Active queue management,” *IEEE Network*, vol. 15, pp. 48–53, 2001.

[4] S. Bhandarkar, S. Jain, and 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.

[5] N. Bigdeli and M. Haeri, “Time-series analysis of TCP/RED computer networks, an empirical study,” *Chaos, Solitons & Fractals*, vol. 39, no. 2, pp. 784–800, 2009.

[6] N. Bigdeli, M. Haeri, S. Choobkar, and F. Jannesari, “Characterization of complex behaviors of TCP/RED computer networks based on nonlinear time series analysis methods,” *Physica D: Nonlinear Phenomena*, vol. 233, no. 2, pp. 138–150, 2007.

[7] T. Bonald, M. May, and J.-C. Bolot, “Analytic evaluation of RED performance,” in *Proceedings of IEEE Infocom 2000*, vol. 3, Tel Aviv, Israel, Mar. 26-30, 2000, pp. 1415–1424.

[8] R. Braden, “Requirements for internet hosts – communication layers,” IETF RFC 1122, Oct. 1989.

[9] N. Brownlee and K. Claffy, “Understanding internet traffic streams: Dragonflies and tortoises,” *IEEE Communications Magazine*, vol. 40, no. 10, pp. 110–117, Oct. 2002.

[10] H. Bullot, R. L. Cottrell, and R. Hughes-Jones, "Evaluation of advanced TCP stacks on fast long-distance production networks," *Journal of Grid Computing*, vol. 1, no. 4, pp. 345–359, 2003.

[11] J. Chen, F. P. amd R. Wang, M. Y. Sanadidi, and M. Gerla, "Fluid-flow analysis of TCP Westwood with RED," in *Proceedings of IEEE GLOBECOM 2003*, San Francisco, USA, Dec. 1-5, 2003.

[12] D. M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1–14, 1989.

[13] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith, "Tuning RED for Web traffic," *IEEE/ACM Transactions on Networking (ToN)*, vol. 9, no. 3, pp. 249–264, 2001.

[14] N. Dukkipati, M. Kobayashi, R. Zhang-shen, and N. McKeown, "Processor sharing flows in the internet," in *Proceedings of Thirteenth International Workshop on Quality of Service (IWQoS)*, Passau, Germany, Jun. 21-23, 2005.

[15] N. Dukkipati and N. McKeown, "Why flow-completion time is the right metric for congestion control," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 59–62, Jan. 2006.

[16] W. Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "BLUE: A new class of active queue management algorithms," U. Michigan, Tech. Rep. CSE-TR-387-99, Apr. 1999.

[17] V. Firoiu and M. Borden, "A study of active queue management for congestion control," in *Proceedings of IEEE Infocom 2000*, Tel Aviv, Israel, Mar. 26-30, 2000, pp. 1435–1444.

[18] S. Floyd, "HighSpeed TCP for large congestion window," IETF RFC 3649, Dec. 2003.

[19] ——, "Limited slow-start for TCP with large congestion windows," IETF RFC 3742, Mar. 2004.

[20] ——, "Metrics for the evaluation of congestion control mechanisms," Internet Draft, draft-irtf-tmrg-metrics-09.txt, Mar. 2007.

[21] S. Floyd, R. Gummadi, and S. Shenker, "Adaptive RED: An algorithm for increasing the robustness of RED's active queue management," ACIRI, Tech. Rep., 2001.

[22] S. Floyd, T. Henderson, and A. Gurtov, "The NewReno modification to TCP's fast recovery algorithm," IETF RFC 3782, Apr. 2004.

[23] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking (ToN)*, vol. 1, no. 4, pp. 397–413, Aug. 1993.

[24] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*. Prentice Hall, 2002.

[25] A. Gurtov, "Effect of delays on TCP performance," in *Proceedings of IFIP Personal Wireless Communications 2001 (PWC'2001)*, Lappeenranta, Finland, Aug. 8-10, 2001.

[26] S. Ha, Y. Kim, L. Le, I. Rhee, and 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.

[27] H. Han, C. V. Hollot, Y. Chait, and V. Misra, "TCP networks stabilized by buffer-based AQMs," in *Proceedings of IEEE Infocom 2004*, vol. 2, Hong Kong, China, Mar. 7-11, 2004, pp. 964–974.

[28] C. V. Hollot and Y. Chait, "Nonlinear stability analysis for a class of TCP/AQM network," in *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida, USA, Dec. 4-7, 2001.

[29] C. V. Hollot, V. Misra, D. Towsley, and W. B. Gong, "A control theoretic analysis of RED," in *Proceedings of IEEE Infocom 2001*, Anchorage, USA, Apr. 22-26, 2001.

[30] ——, "On designing improved controllers for AQM routers supporting TCP flows," in *Proceedings of IEEE Infocom 2001*, Anchorage, USA, Apr. 22-26, 2001.

[31] V. Jacobson, "Congestion avoidance and control," in *Proceedings of ACM SIGCOMM 1988*, Stanford, CA, USA, Aug. 16-18, 1988, pp. 314–329.

[32] J. M. Jaffe, "A decentralized "optimal" multiple-user flow control algorithm," *IEEE Transactions on Communications*, vol. 29, pp. 954–962, 1981.

[33] R. Jain, A. Durresi, and G. Babic, "Throughput fairness index: An explanation," ATM Forum/99-0045, Feb. 1999. [Online]. Available: http://www.cse.wustl.edu/~jain/atmf/a99-0045.htm

[34] K. Jiang, X. Wang, and Y. Xi, "Nonlinear analysis of RED – a comparative study," *Chaos, Solitons & Fractals*, vol. 21, no. 5, pp. 1153–1162, 2004.

[35] C. Jin, D. X. Wei, and 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.

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

[37] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, pp. 33–37, 1997.

[38] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," in *Journal of the Operational Research Society*, vol. 49, no. 3, 1998, pp. 237–252.

[39] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 2, pp. 83–91, Apr. 2003.

[40] K. Khalil, *Nonlinear Systems*. Prentice Hall, 1996.

[41] R. King, R. Riedi, and 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.

[42] J. Koo, S. Ahn, and J. Chung, *Computational Science - ICCS 2004*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2004, vol. 3036, ch. Performance Analysis of Active Queue Management Schemes for IP Network, pp. 349–356.

[43] M. Labrador and S. Banerjee, "Packet dropping policies for ATM and IP networks," *IEEE Communications Surveys*, vol. 2, no. 3, pp. 2–14, Third Quarter 1999.

[44] G. S. Lee, L. L. H. Andrew, A. Tang, and 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.

[45] D. Leith and 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.

[46] D. J. Leith, L. L. H. Andrew, T. Quetchenbach, R. N. Shorten, and 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.

[47] Y. Li, D. Leith, and R. N. Shorten, "Experimental evaluation of TCP protocols for high-speed networks," *IEEE/ACM Transactions on Networking (ToN)*, vol. 15, no. 5, pp. 1109–1122, Oct. 2007.

[48] D. Lin and R. Morris, "Dynamics of random early detection," *SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 4, pp. 127–137, 1997.

[49] S. Liu, T. Başar, and 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.

[50] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Transactions on Networking (ToN)*, vol. 11, no. 4, pp. 525–536, Aug. 2003.

[51] S. H. Low and D. E. Lapsley, "Optimization flow control, I: Basic algorithm and convergence," *IEEE/ACM Transactions on Networking (ToN)*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[52] S. H. Low, F. Paganini, and J. Doyle, "Internet congestion control: an analytical perspective," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 28–43, Feb. 2002.

[53] S. H. Low, F. Paganini, J. Wang, S. Adlakha, and J. C. Doyle, "Dynamics of TCP/RED and a scalable control," in *Proceedings of IEEE Infocom 2002*, New York, USA, Jun. 23-27, 2002.

[54] S. H. Low, L. Peterson, and L. Wang, "Understanding Vegas: a duality model," *Journal of the ACM*, vol. 49, no. 2, pp. 207–235, Mar. 2002.

[55] G. Marfia, C. E. Palazzi, G. Pau, M. Gerla, M. Y. Sanadidi, and M. Roccetti, "TCP-Libra: Exploring RTT fairness for TCP," UCLA Computer Science Department, Tech. Rep. UCLA-CSD TR-050037, 2005. [Online]. Available: http://www.cs.ucla.edu/~cpalazzi/papers/TCP-Libra-UCLA-CSD-TR050037.pdf

[56] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgment options," IETF RFC 2018, Oct. 1996.

[57] "The MathWorks: MATLAB." [Online]. Available: http://www.mathworks.com/products/matlab/

[58] V. Misra, W. B. Gong, and D. Towsley, "Stochastic differential equation modeling and analysis of TCP windowsize behavior," Department of Electrical and Computer

Engineering, University of Massachusetts, Tech. Rep. ECE-TR-CCS-99-10-01, 1999. [Online]. Available: ftp://gaia.cs.umass.edu/pub/Misra00_AQM.ps.gz

[59] ——, "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED," in *Proceedings of ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 28 - Sep. 1, 2000, pp. 151–160.

[60] S. Molnár and E. Altman, "Guest eds." *Telecommunication Systems, special issue on High Speed Transport Protocols*, vol. 30, no. 4, Dec. 2005.

[61] S. Molnár, S. Kardos, and 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.

[62] S. Molnár, B. Sonkoly, and T. A. Trinh, "A comprehensive TCP fairness analysis in high speed networks," *Computer Communications*, vol. 32, no. 13-14, pp. 1460–1484, Aug. 2009.

[63] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc., 1994.

[64] "Ns-2 Network Simulator." [Online]. Available: http://www.isi.edu./nsnam/ns

[65] "BI-TCP implementation for NS 2." [Online]. Available: http://www4.ncsu.edu/~rhee/export/bitcp/bitcp-ns/bitcp-ns.htm

[66] "FAST TCP simulator module for ns-2." [Online]. Available: http://www.cubinlab.ee.unimelb.edu.au/ns2fasttcp/

[67] T. J. Ott, T. V. Lakshman, and L. Wong, "SRED: Stabilized RED," in *Proceedings of IEEE Infocom 1999*, 1999, pp. 1346–1355.

[68] J. Postel, "Transmission control protocol," IETF RFC 793, Sep. 1981.

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

[70] I. Rhee and 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.

[71] H. Shimonishi, T. Hama, and 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.

[72] H. Shimonishi, M. Sanadidi, and 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.

[73] B. Simon, B. Sonkoly, and S. Molnár, "Modelling the inter-operation of high speed TCP protocols," in *Proceedings of 7th European Conference of Young Research and Science Workers in Transport and Telecommunications (TRANSCOM 2007)*, Žilina, Slovak Republic, Jun. 25-27, 2007.

[74] B. Sonkoly, B. Simon, T. A. Trinh, and S. Molnár, "A research framework for analyzing high speed transport protocols based on control-theory," *International Journal of Network Protocols and Algorithms*, vol. 1, no. 2, pp. 1–26, Dec. 2009.

[75] B. Sonkoly, T. A. Trinh, and S. Molnár, "Benchmarking high speed TCP fairness," Budapest University of Technology and Economics, BME, Tech. Rep., Oct. 2007. [Online]. Available: http://hsnlab.tmit.bme.hu/projects/tcp/

[76] ——, "Understanding HighSpeed TCP: A control-theoretic perspective," in *Proceedings of The Third IASTED International Conference on Communications and Computer Networks (CCN 2005)*, Marina del Rey, CA, USA, Oct. 24-26, 2005.

[77] ——, "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.

[78] E. Souza and D. Agarwal, "A HighSpeed TCP study: Characteristics and deployment issues," Lawrence Berkeley National Lab, Tech. Rep. LBNL-53215, 2003. [Online]. Available: http://www-itg.lbl.gov/~evandro/hstcp/hstcp-lbnl-53215.pdf

[79] R. Srikant, *The Mathematics of Internet Congestion Control.* Birkhauser, 2004.

[80] K. Tan, J. Song, Q. Zhang, and 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.

[81] L. Tan, L. Dong, C. Yuan, and M. Zukerman, "Fairness comparison of FAST TCP and TCP Reno," *Computer Communications*, vol. 30, no. 6, pp. 1375–1382, Apr. 2007.

[82] "TCP Evaluation Suite." [Online]. Available: http://netlab.cs.ucla.edu/tcpsuite/

[83] O. Tickoo, V. Subramanian, S. Kalyanaraman, and K. K. Ramakrishnan, "LT-TCP: End-to-end framework to improve TCP performance over networks with lossy channels," in

Proceedings of Thirteenth International Workshop on Quality of Service (IWQoS 2005), Passau, Germany, Jun. 21-23, 2005.

[84] "Transport Modeling Research Group (TMRG), IRTF," http://wiki.tools.ietf.org/group/irtf/trac/wiki/tmrg.

[85] T. A. Trinh and S. Molnár, *Quality of Service in the Emerging Networking Panorama*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2004, vol. 3266, ch. A Game-Theoretic Analysis of TCP Vegas, pp. 338–347.

[86] T. A. Trinh, B. Sonkoly, and S. Molnár, "On the fairness characteristics of FAST TCP," *Annals of Telecommunications*, vol. 65, no. 1, pp. 73–85, Feb. 2010.

[87] ——, "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.

[88] ——, "Revisiting FAST TCP fairness," in *Proceedings of 18th ITC Specialist Seminar on Quality of Experience*, Karlskrona, Sweden, May 29-30, 2008.

[89] "Transport Area Working Group (tsvwg), IETF," http://www.ietf.org/dyn/wg/charter/tsvwg-charter.html.

[90] F. Vacirca, A. Baiocchi, and 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.

[91] A. Veres, Z. Kenesi, S. Molnár, and 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.

[92] G. Vinnicombe, "On the stability of networks operating TCP-like congestion control," in *Proceedings of the IFAC World Congress on Automatic Control*, Barcelona, Spain, Jul. 2002.

[93] G. Wang, Y. Xia, and D. Harrison, "An NS2 TCP evaluation tool suite," Internet Draft, draft-irtf-tmrg-ns2-tcp-tool-00, Apr. 2007.

[94] J. Wang, D. X. Wei, and S. H. Low, "Modelling and stability of FAST TCP," in *Proceedings of IEEE Infocom 2005*, vol. 2, Miami, USA, Mar. 13-17, 2005, pp. 938–948.

[95] R. Wang, K. Yamada, M. Y. Sanadidi, and M. Gerla, "TCP with sender-side intelligence to handle dynamic, large, leaky pipes," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 2, pp. 235–248, 2005.

[96] D. X. Wei, P. Cao, and S. H. Low, “Time for a TCP benchmark suite?” Online, Aug. 2005. [Online]. Available: http://www.cs.caltech.edu/~weixl/research/technicals/benchmark/summary.ps

[97] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, “FAST TCP: motivation, architecture, algorithms, performance,” *IEEE/ACM Transactions on Networking (ToN)*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.

[98] M. C. Weigle, P. Sharma, and J. Freeman, “Performance of competing high-speed TCP flows,” in *Proceedings of IFIP Networking 2006*, Coimbra, Portugal, May 15-19, 2006.

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

[100] L. Xu, K. Harfoush, and 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.

Publications

[J] **Journal papers**

[J1] S. Molnár, B. Sonkoly, and T. A. Trinh, "A comprehensive TCP fairness analysis in high speed networks," *Computer Communications*, vol. 32, no. 13-14, pp. 1460–1484, Aug. 2009.

[J2] T. A. Trinh, B. Sonkoly, and S. Molnár, "On the fairness characteristics of FAST TCP," *Annals of Telecommunications*, vol. 65, no. 1, pp. 73–85, Feb. 2010.

[J3] B. Sonkoly, B. Simon, T. A. Trinh, and S. Molnár, "A research framework for analyzing high speed transport protocols based on control-theory," *International Journal of Network Protocols and Algorithms*, vol. 1, no. 2, pp. 1–26, Dec. 2009.

[J4] B. Simon, B. Sonkoly, and S. Molnár, "Nagysebességű TCP protokollok együttműködésének modellezése," *Híradástechnika folyóirat*, LXII:(8), pp. 2–5, Aug. 2007. (in Hungarian)

[J5] T. Henk, R. Szabó, S. Molnár, B. Sonkoly, M. Csernai, A. Gulyás, Z. Heszberger, L. Gyarmati, and T. A. Trinh "A jövő Internetjének kutatásai," *Híradástechnika folyóirat*, LXIV, Különszám, pp. 23–33, 2009. (in Hungarian)

[J6] S. Molnár, and B. Sonkoly, "Congestion Control and Network Management in Future Internet," *Híradástechnika folyóirat*, Invited Paper, LXIV:(4), Special Issue, pp 14–22, 2009.

[J7] Sz. Malomsoky, Sz. Nádas, and B. Sonkoly, "Performance Evaluation of UMTS Terrestrial Radio Access Networks," *Híradástechnika folyóirat*, LVII:(7), pp. 9–19, Jul. 2002.

[J8] Sz. Malomsoky, Sz. Nádas, and B. Sonkoly, "UMTS hozzáférési hálózatok teljesítőképesség-vizsgálata," *Híradástechnika folyóirat*, LVII:(9), pp. 19–29, Sep. 2002. (in Hungarian)

[C] **Conference papers**

[C1] T. A. Trinh, B. Sonkoly, and S. Molnár, "Revisiting FAST TCP fairness," in *Proceedings of 18th ITC Specialist Seminar on Quality of Experience*, Karlskrona, Sweden, May 29-30, 2008.

[C2] B. Sonkoly, T. A. Trinh, and 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.

[C3] B. Simon, B. Sonkoly, and S. Molnár, "Modelling the inter-operation of high speed TCP protocols," in *Proceedings of 7th European Conference of Young Research and Science Workers in Transport and Telecommunications (TRANSCOM 2007)*, Žilina, Slovak Republic, Jun. 25-27, 2007.

[C4] B. Sonkoly, T. A. Trinh, and S. Molnár, "Understanding HighSpeed TCP: A control-theoretic perspective," in *Proceedings of The Third IASTED International Conference on Communications and Computer Networks (CCN 2005)*, Marina del Rey, CA, USA, Oct. 24-26, 2005.

[C5] T. A. Trinh, B. Sonkoly, and 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.

[C6] T. D. Dang, B. Sonkoly, and S. Molnár, "Fractal Analysis and Modelling of VoIP Traffic," in *Proceedings of NETWORKS 2004*, Vienna, Austria, Jun. 13-16, 2004.

[O] **Others**

[O1] B. Sonkoly, T. A. Trinh, and S. Molnár, "Benchmarking high speed TCP fairness," Budapest University of Technology and Economics, BME, Tech. Rep., Oct. 2007. [Online]. Available: `http://hsnlab.tmit.bme.hu/projects/tcp/`

[O2] S. Molnár, T. A. Trinh, and B. Sonkoly, "A Performance Study of HighSpeed TCP," Technical Report, COST 279TD(04)012, Paris, France, May. 27-28, 2004.