Revisiting FAST TCP Fairness

Tuan A. Trinh, Balázs Sonkoly, Sándor Molnár

***Abstract*—Fairness of competing TCP flows is an integral and indispensable part of transport protocol design for next generation high bandwidth-delay product networks. In this paper we revisit FAST TCP fairness behavior based on a comprehensive performance evaluation study. We demonstrate that FAST TCP with proper parameter settings can always achieve fair behavior with HighSpeed TCP and Scalable TCP. We also show that this behavior is rather robust property of the protocol concerning different traffic mix or network topology. The dynamic behavior of reaching the fair equilibrium state can be different which is demonstrated in the paper. Our study also emphasizes the important need for finding a dynamic sensitive fairness metric for performance evaluation of transport protocols for next generation high bandwidth-delay product networks.**

***Index Terms*—High Speed Networks, FAST TCP, Fairness Analysis.**

I. Introduction

The advance in technologies, newer and newer forms of applications ensure improved and diversified services to the end-users but they also bring new challenges for the network designers. As a result, there is a genuine need for next generation transport protocols that can efficiently utilize the resources and that can operate in these new and diverse network environments.

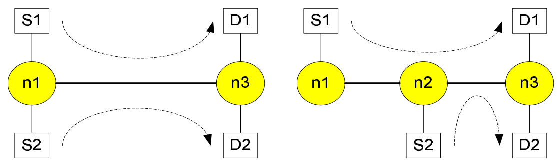
This question has recently received considerable attention from the research community [1], [2] and a number of solutions have been proposed (HighSpeed TCP [3], Scalable TCP [4], FAST TCP [5], and others). Roughly, these protocols can be divided into two classes: loss-based and delay-based. Loss-based versions share similar features with traditional TCP (TCP Reno) whereas delay-based TCP (FAST TCP) is an extension of TCP Vegas [6]. There is a considerable research regarding the modeling and analysis of high speed TCP versions, e.g., [3]–[5], [7]–[9]. It is widely accepted that one of the most important issues with these protocols is operability and deployability. This directly leads to the question of fairness. In fact, this question is tackled by research community for quite a long time (see e.g., [4], [5], [10]–[14]) and a number of fairness metrics have been proposed, such as Jain's index, max-min fairness, proportional fairness, utility-based fairness, etc. These metrics are different, but they share a common aspect. They all concern with the long term average of the flows and their stable/equilibrium performance. The main weakness of these metrics is the lack of attention to the dynamic of the flows. In this paper, we revisit FAST TCP, a delay-based TCP version that is designed as a transport protocol for next generation networks, especially high bandwidth-delay product networks. We illustrate and show some surprising benefits of this approach, in particular FAST TCP, in terms of fairness. This paper is devoted to give important and demonstrative results gained from a comprehensive analysis [15].

T. A. Trinh, B. Sonkoly and S. Molnár are with High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar tudósok körútja 2, H–1117, Budapest, Hungary. E-mail: {trinh, sonkoly, molnar}@tmit.bme.hu

This paper is organized as follows. In Section II, the simulation environment and the important parameters are presented. Section III shows the fairness issues of FAST TCP with other loss-based TCP versions from a flow-level perspective. Observations of the deficiencies of the available fairness metrics are also provided in Section III. Section IV provides a comprehensive packet-level analysis of the observed phenomena, especially the impact of starting time on long-term fairness. In Section V, a brief discussion on the performance in more complex topologies is given. Section VI concludes the paper.

II. Simulation environment

The fairness analysis of competing high speed TCP protocols and the validation of the analytical results are carried out in the Ns-2 [16] simulation environment. Our simulation scripts regarding different network scenarios can be found in [15]. 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 FAST TCP is used from [17].



(a) dumb-bell (b) parking-lot

Figure 1. Network topology

The examined dumb-bell topology containing one bottleneck link is shown in Figure 1(a). The queueing mechanism corresponding to the bottleneck link is drop-tail. We do not consider the impacts of the buffer size (B) 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 1(b)), where the impacts of different round-trip times (RTT) can be revealed. Here, only the second link is congested. In case of these scenarios, a simulation contains two competing flows starting at different time instances and performing an infinite FTP download (permanent TCP connection). 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.

During the evaluation, the default parameter set of the protocols is used (see [3] and [4]). HighSpeed TCP (HSTCP) and Scalable TCP (STCP) apply the Limited Slow-Start (LSS) mechanism [18], as well. The parameters of the simulations are summarized in Table I.

Table I
PARAMETERS

Network parameters	
capacity	1 Gbps
RTT	100 ms
packet size	$1,500$ bytes
Buffer size	
dumb-bell	$8,333$ pkts
parking-lot	$25,000$ pkts

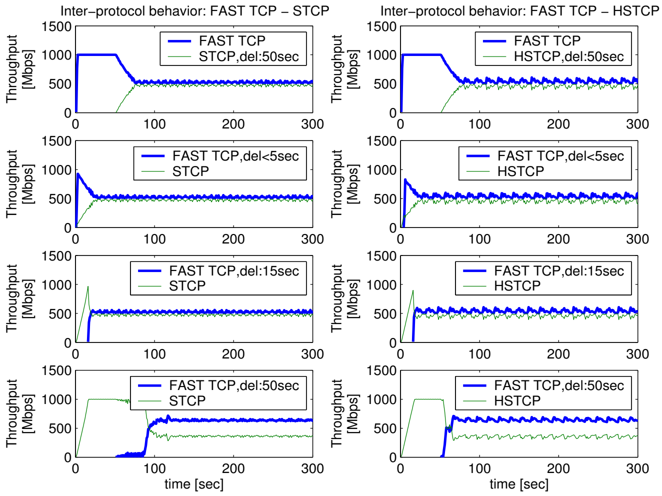
Sampling periods	
`cwnd`, queue	0.01 s
throughput	1 s
FAST TCP – dumb-bell	
$\alpha = \beta$	$4,166$ pkts
FAST TCP – parking-lot	
$\alpha = \beta$	$12,500$ pkts

FAST TCP seeks to restrict the number of its packets queued through the network path between an upper and a lower bound. The appropriate setting of parameters α and β regarding the bounds is crucial. We use only one parameter (α) setting the bounds as $\alpha = \beta$. The control mechanism is based on the comparison between the observed RTT and the `baseRTT` which is an approximation of the round-trip propagation delay (when there is no queueing). The α parameter of FAST TCP flows is chosen so that the total number of outgoing packets of the flows is smaller than the buffer size to avoid losses due to buffer overflow for these FAST TCP connections. In case of two flows, then the α parameter of FAST TCP is set to $B/2$.

III. Flow-level study

In this section, two competing flows (one FAST TCP flow and one HSTCP or Scalable TCP flow) are examined at the flow-level and the dynamics of average throughput and fairness metrics are analyzed. The deficiencies of the available fairness metrics are also revealed. Recent researches [13], [15] showed that competing high speed transport protocols (such as Scalable TCP, HSTCP) can not always achieve fair equilibrium state or the convergence time can be too long. It was also revealed that the starting time of the flows can have relevant impact on the performance. We show that FAST TCP with appropriate parameters can exhibit fair or almost fair behavior. Here, we focus on the inter-protocol properties, however, FAST TCP with appropriate parameters shows fair behavior with other FAST TCP flows, as well.

First, the dumb-bell topology is analyzed. The dynamics of the bandwidth share for scenarios with different starting delays are presented in Figure 2. It can be observed in all presented



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

Figure 2. Performance of competing flows

results that the equilibrium bandwidth share is approximately fair. On the one hand, when FAST TCP starts first, the bottleneck bandwidth is always shared fair in equilibrium state. On the other hand, in case of later entering FAST TCP flow, the equilibrium state is only near to the fair state (almost fair), and it is reached after a transient period (which length depends on the starting delay). This bias in the equilibrium state can be caused by the estimation error of `baseRTT` (the FAST TCP flow does not experience empty buffer).

For the examined scenarios, available fairness metrics can also be derived. Here, we calculate three important ones that can be used in our analysis. Let $\overline{BW}_1$ and $\overline{BW}_2$ be the average bandwidth share of the two sources, respectively.

Relative fairness (used in [3]) can be defined as follows:

$$RF = \frac{\overline{BW}_1}{\overline{BW}_2}.$$

Jain's index [19] is a normalized metric in the $[0.5, 1]$ interval and can be defined as follows:

$$JI = \frac{(\overline{BW}_1 + \overline{BW}_2)^2}{2(\overline{BW}_1^2 + \overline{BW}_2^2)}$$

The bandwidth share is fair if $JI \to 1$ and unfair behavior can be observed if $JI \to 0.5$.

Another normalized asymmetry metric proposed in [20] can express which flow is more aggressive. This index is defined as

$$BI = \frac{\overline{BW}_1 - \overline{BW}_2}{\overline{BW}_1 + \overline{BW}_2}.$$

The closer BI is to 0, the more fair bandwidth share can be observed. $BI \to 1$ shows the dominance of the first flow while $BI \to -1$ shows the dominance of the second one.

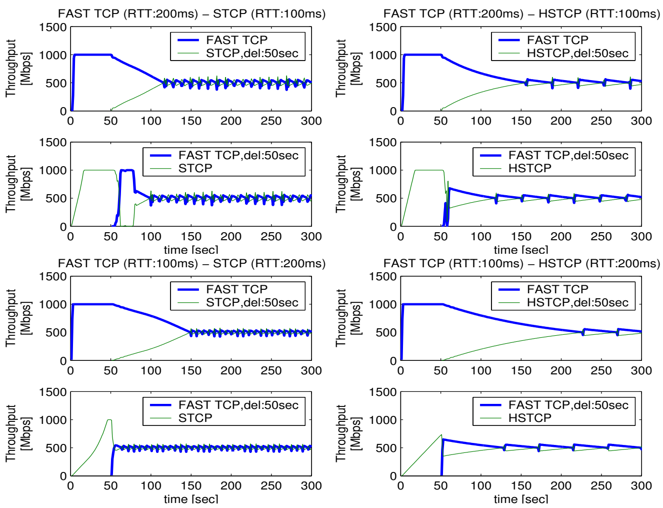
Table II summarizes the fairness indices of the protocols calculated from the simulations. The results confirm quantitatively the qualitative statements based on throughput diagrams.

Table II
FAIRNESS INDICES (DUMB-BELL TOPOLOGY)

Flow 1 Prot.	Flow 2 Prot.	del: 50sec			del: 15sec			del < 5sec		
		BI	JI	RF	BI	JI	RF	BI	JI	RF
FAST TCP	STCP	0.051	0.997	1.106	0.051	0.997	1.107	0.051	0.997	1.107
STCP	FAST TCP	-0.273	0.930	0.569	-0.051	0.997	0.904	-0.051	0.997	0.905
FAST TCP	HSTCP	0.081	0.993	1.177	0.081	0.994	1.176	0.081	0.994	1.176
HSTCP	FAST TCP	-0.287	0.924	0.555	-0.081	0.994	0.851	-0.081	0.993	0.848

These metrics are mainly capable of characterizing the equilibrium behavior and can not capture the dynamic properties of the interaction, thus further analysis both in packet-level and flow-level is necessary.

Second, we examine the behavior of FAST TCP in the parking-lot topology where the fairness of competing high speed TCP flows with different round-trip times (RTT) can be analyzed. The results are promising and here, some examples are given as illustration. We found that FAST TCP with longer path and round-trip time can also achieve good performance and shows fair behavior with Scalable TCP and HSTCP. Another attractive property of the protocol is shown when the round-trip time of the FAST TCP flow is shorter. FAST TCP does not starve the other flows and fair equilibrium states are achieved. Demonstrative results are shown in Figure 3. The upper parts of the figure correspond to scenarios when the RTT of the FAST TCP flow is greater, while the lower parts present the results for the reverse case. When FAST



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

Figure 3. Performance – parking-lot topology

TCP flow starts first, the characteristics of the interaction is similar to the behavior exhibited in the dumb-bell topology: after a transient phase, FAST TCP gives up approximately the half of the shared link bandwidth. However, the length of the transient period differs. In case of later entering FAST TCP flow, the fair (or almost fair) equilibrium state is realized again, however, in the transient phase other phenomena can also be observed. The later entering FAST TCP flow with longer RTT can only achieve the fair equilibrium at the cost of timeout of the other flow. In our simulation environment, HSTCP and Scalable TCP can respond to this event in a different manner, thus, the transient phases show different properties.

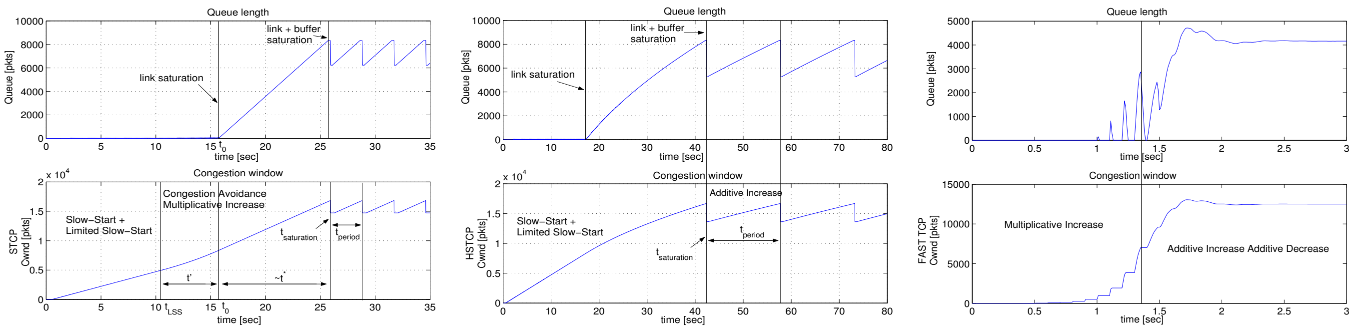
IV. PACKET-LEVEL ANALYSIS

This section is devoted to reveal and explain the phenomena experienced at the flow-level based on a comprehensive packet-level analysis. First, the initial behavior of individual flows is summarized in order to gain a basic knowledge of the characteristics of different congestion control principles. Second, the long-term behavior of competing high speed TCP protocols is investigated. We apply spectral analysis of `cwnd` and queue processes to get an insight into the dynamic properties of the interaction.

A. 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. Here, the investigation is carried out considering the simple dumb-bell topology. We introduce a new performance metric, namely, the saturation time, as the length of this transient phase. This metric can be defined for a loss-based protocol as the time from the starting till the first packet drop. In Figure 4(a), 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 instance 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) operating regime (see Figure 4(c)).

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



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

Figure 4. Saturation time and equilibrium behavior

increase (HSTCP) or the multiplicative increase (Scalable TCP) phase, before the first packet drop. FAST TCP 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.

1) Dynamics of Slow-Start: The Slow-Start mechanism is analytically tractable and relevant parameters can easily be derived. Here, we only summarize the main results (for further details, see [15]).

The Slow-Start phase lasts till `cwnd` reaches the threshold `ssthresh` or a packet loss can be detected. This time instance is $t_{ss} = \min\{t_{th}, t_{drop} + t_{delay}\}$, where $t_{th} = R_0 \log_2 \texttt{ssthresh}$ and t_{drop} is the solution of the following equation:

$$\frac{2^{t/R_0}}{\lg 2} - Ct + \left(Ct_0 - \frac{2^{t_0/R_0}}{\lg 2} - B \right) = 0, \qquad (1)$$

where R_0 is the round-trip propagation delay, C is the link capacity and B is the buffer size. 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 BDP (bandwidth-delay product) and the buffer size at the saturation time:

$$t_{drop} \approx R_0 \log_2(R_0C + B) = R_0 \frac{\lg(R_0C + B)}{\lg 2}. \qquad (2)$$

t_{delay} corresponds to queueing delay and one-way propagation delay.

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`.

2) Dynamics of Limited Slow-Start: LSS operates in congestion avoidance mode in the Ns-2 implementation till the first packet drop. LSS 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 are used. With this algorithm, a faster convergence can be achieved when the source's sending rate is far from the equilibrium value. 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 ($\texttt{max\_ssthresh} = 100$). It can easily be derived analytically [18] that $\log_2(\texttt{max\_ssth}) + \frac{\texttt{cwnd}-\texttt{max\_ssth}}{\texttt{max\_ssth}/2}$ RTT is needed to reach `cwnd` (that is greater than `max_ssthresh`). The first term corresponds to the standard Slow-Start phase and max_ssth stands for $max\_ssthresh$.

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 and the increment of Scalable TCP's or HSTCP's algorithm will be equal triggering the end of this phase. On the one hand, in case of individual Scalable TCP flow, this state can be expressed by

$$W_{LSS} = \frac{\texttt{max\_ssth}}{2} \frac{1}{a},$$

where a is the increase parameter of Scalable TCP. The details of the derivation can be found in [15]. 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, for HSTCP flow, W_{LSS} can be derived from the following equation:

$$a(W_0) = \texttt{max\_ssth}/2,$$

where $a(W_0)$ 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.

3) Scalable TCP – saturation time.: In case of Scalable TCP, the end of LSS phase can be expressed as follows (see [15] for details):

$$t_{LSS} = R_0 \frac{\lg \texttt{max\_ssth}}{\lg 2} + R_0 \frac{W_{LSS} - \texttt{max\_ssth}}{\texttt{max\_ssth}/2} \approx 10.46 \text{ sec}, \qquad (3)$$

where W_{LSS} is the value of congestion window triggering the end of LSS. After Limited Slow-Start, the multiplicative increase mechanism of the protocol operates. During this period, the congestion window is increased from W_{LSS} to the BDP (R_0C). Thus, the link saturation time can easily be determined:

$$t_0 = t_{LSS} + t' = t_{LSS} + R_0 \frac{\lg \frac{R_0C}{W_{LSS}}}{\lg(1+a)} \approx 15.6\,\text{sec}. \qquad (4)$$

The time till the first packet drop can also be determined by solving differential equations describing the dynamics of congestion window and the behavior of the queue. Instead of solving complicated differential equations (with varying delays and recursive arguments), a simple approximation can be applied. In this phase, the congestion window is increased from $W_0 = R_0C$ to $R_0C + B$ according to the multiplicative increase mechanism. Approximating the increase of the queueing delay by a linear function, the round-trip time can be treated as a constant with a mean value: $\tilde{R} = R_0 + B/2C$. Thus, the saturation time can be expressed as follows:

$$\hat{t}_{saturation} = t_0 + t^* = t_0 + \tilde{R} \frac{\lg \frac{R_0C+B}{R_0C}}{\lg(1+a)} \approx 26.05\,\text{sec}. \qquad (5)$$

The analytically derived parameters and the approximation of saturation time meet well the simulation results presented in Figure 4(a).

4) 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. The link saturation time can be expressed as follows:

$$t_0 = R_0 \frac{\lg \mathtt{max\_ssth}}{\lg 2} + R_0 \frac{R_0C - \mathtt{max\_ssth}}{\mathtt{max\_ssth}/2} \approx 17.13\,\text{sec}. \qquad (6)$$

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

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

where

$$t^* = \tilde{R} \frac{R_0C + B - R_0C}{\mathtt{max\_ssth}/2} = \tilde{R} \frac{B}{\mathtt{max\_ssth}/2}. \qquad (7)$$

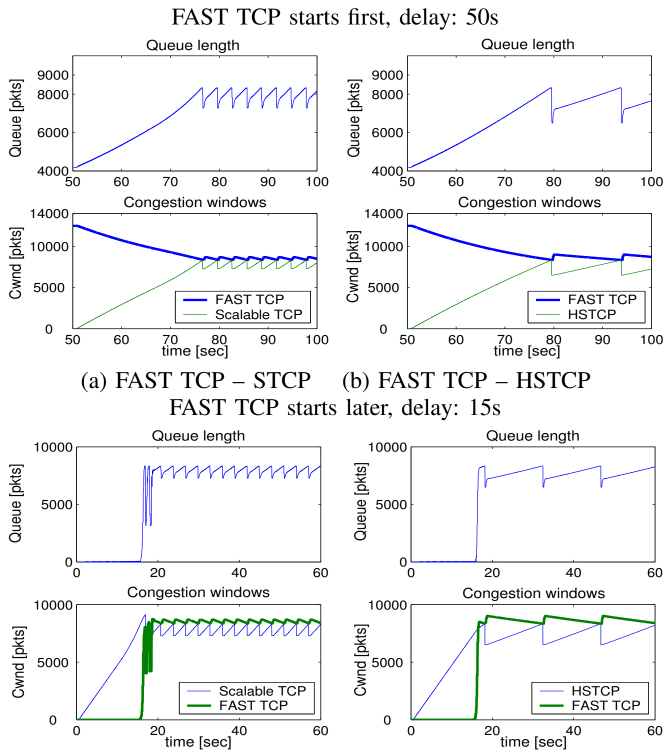
Our parameters yield that $\hat{t}_{saturation} = 42.13$ sec which meets well the simulation results (see Figure 4(b)).

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. This fast convergence of the `cwnd` is shown in Figure 4(c). If queueing delay exceeds a threshold (which is a constant parameter of the protocol in the Ns-2 implementation), the additive increase additive decrease control algorithm is used instead of multiplicative increase. In our simulation environment, FAST TCP (with $\alpha = 4166$) reaches the equilibrium approximately after 2 seconds.

B. Equilibrium behavior

In this section, the packet-level characteristics of the long-term behavior of competing high speed TCP flows are investigated.

Firts, we focus on scenarios when FAST TCP source starts the transmission and the other flow enters into the network when the first one has achieved maximal sending rate. The simulation results corresponding to starting delay of 50 sec are presented in Figure 5 for Scalable TCP (a) and HSTCP (b), respectively. During the transient phase, the additive decrease



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

(c) STCP – FAST TCP (d) HSTCP – FAST TCP

Figure 5. Competition of 2 flows

algorithm of FAST TCP interacts with the control mechanism of the other protocol (limited slow-start and multiplicative increase in case of Scalable TCP and limited slow-start in case of HSTCP). Following the proposed schemes for choosing an α parameter of FAST TCP, `cwnd` processes converge to an equilibrium state corresponding to approximately fair bandwidth share. The length of the transient phase is determined by the congestion control algorithms. After the transient period, a common periodic behavior is shown by the sources. When Scalable TCP reduces its congestion window, FAST TCP can increase the number of packets in the bottleneck queue performing an additive increase based on queueing delay. During the second part of a period, the multiplicative increase of Scalable TCP interacts with the additive decrease of FAST TCP. Thus, the periodic behavior is affected by the interaction of AIAD – MIMD algorithms. 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 AIAD and AIMD mechanisms results in a longer time period.

Second, the FAST TCP source enters later into the network and try to catch the half of the capacity of the bottleneck link. Recent researches [13], [15] showed that a Scalable TCP flow in equilibrium state can starve other flows starting later (including other Scalable TCP flow). FAST TCP with α parameter chosen as suggested in [5] achieve significant bandwidth share against Scalable TCP and HSTCP, too. The simulation results corresponding to 15 sec delay are presented in Figure 5(c) and (d). In these scenarios, after a very short transient period, the congestion windows settle down again around an equilibrium state.

A significantly different behavior can be experienced at the packet-level increasing the starting delay of the FAST TCP flow. As an illustration, the simulation results of the competition of Scalable TCP and FAST TCP flow corresponding to 50 sec delay are shown in Figure 6. This behavior can be examined in the frequency domain, too. The power spectral density (PSD) functions of the `cwnd` process of Scalable TCP and FAST TCP and the bottleneck queue process are also shown in Figure 6. 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, it performs multiplicative increase algorithm. 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 exponential increase is performed until the next loss. After a long and oscillating transient phase, the previously seen common periodic equilibrium behavior 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 MIMD oscillation of the transient phase. These two frequency spikes mainly determine the dynamics of the bottleneck queue. In case of interaction with HSTCP, a similar behavior can be observed. We observed that the length of the transient period depends on the starting time and other parameters of FAST TCP, as well.

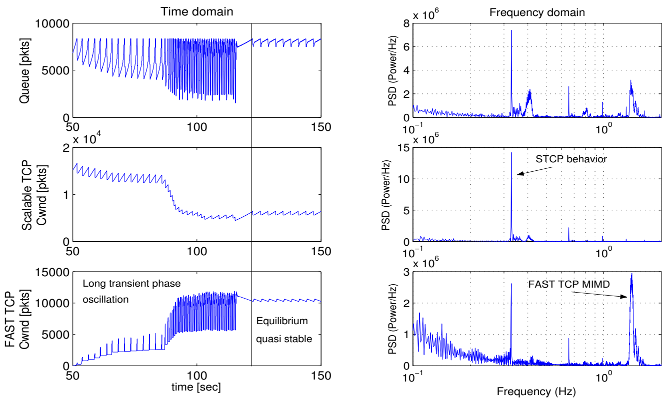


Figure 6. STCP – FAST TCP, delay: 50s

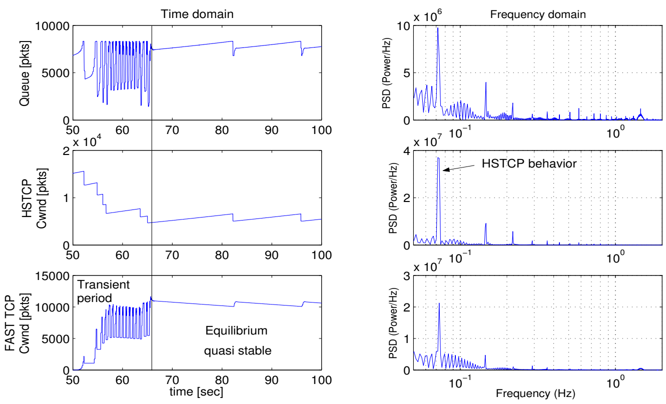


Figure 7. HSTCP – FAST TCP, delay: 50s

FAST TCP can also achieve good performance and fair behavior against Scalable TCP or HSTCP traffic aggregate. On the one hand, we found that later entering FAST TCP flow can occupy the 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 (AIAD). On the other hand, the interaction with Scalable TCP traffic aggregate can have different characteristics. To illustrate the point, we let a traffic aggregate that contains 10 Scalable TCP flows (Figure 8) and 10 HSTCP flows (Figure 9) mixed with a FAST TCP source entering with a delay of 50 sec. Here, the characteristics of the equilibrium behavior is determined by the MIMD mechanisms of the protocols and FAST TCP is not able to reach a stable (or quasi stable) state and shows oscillation. However, the achieved throughput approximates the half of the bottleneck capacity (as it is targeted by the parameter setting of FAST TCP) resulting in good performance.

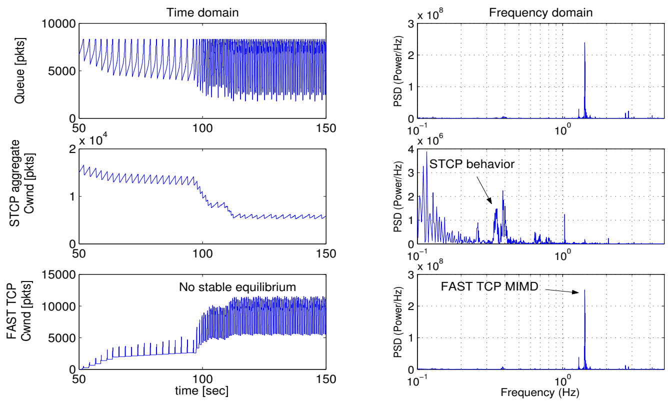


Figure 8. STCP aggregate – FAST TCP

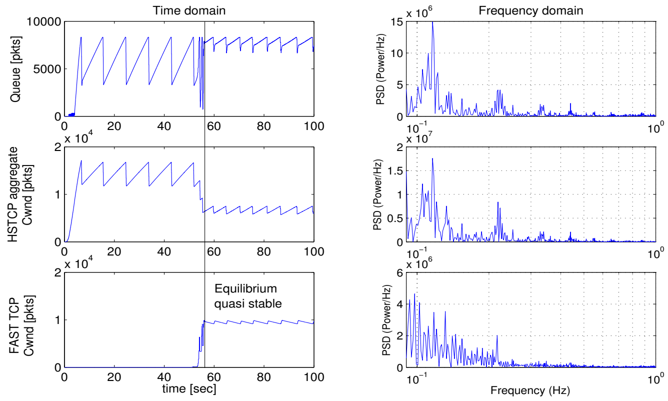


Figure 9. HSTCP aggregate – FAST TCP

V. Discussion

Surprisingly, the performance of FAST TCP shows degradation in complex parking-lot topology with multiple congested links. Here, we show some demonstrative results (for further details, see [15]).

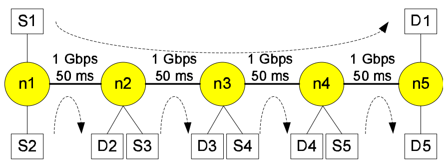


Figure 10. Complex parking-lot topology with four congested links

A complex parking-lot topology with five nodes is shown in Figure 10, where all links are congested. Here, one FAST TCP flow traverses across the backbone containing four congested links and four flows of a loss-based protocol use single links, respectively. Here, the parameters of FAST TCP are set to the same value as it was used in the dumb-bell scenarios.

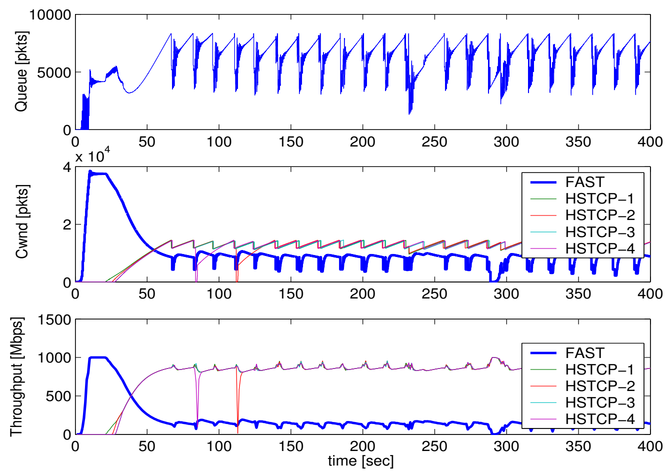


Figure 11. Parking-lot: 5 nodes, FAST TCP – HSTCP (FAST TCP with longer RTT starts first)

As an illustration, the competition of a single FAST TCP flow traversing the backbone and four HSTCP flows using separate links is shown in Figure 11. 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. Obviously, the performance of FAST TCP can be enhanced increasing the α parameter of the protocol. However, this can yield unstable network behavior with degraded link utilization.

VI. Conclusion

In this paper, we revisited FAST TCP, a delay-based TCP version proposed for next generation networks. We illustrated and showed some surprising benefits of this approach in terms of fairness.

Our main findings are the following: *In contrast to loss-based protocols, FAST TCP with appropriate parameters can always show fair (or almost fair) behavior beside HSTCP and Scalable TCP flows.* Concerning the dynamics of TCP starting times the fair or almost fair state is achieved by different ways:

- If 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. *More specifically, we have found that FAST TCP can achieve good utilization against Scalable TCP or HSTCP traffic aggregate. Moreover, FAST TCP can exhibit these good properties operating in more complex network environments (parking-lot topology).* We should also note that this property holds for a certain range of the parameter alpha 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.

We have also revealed the drawbacks of currently used fairness metrics and showed the urgent need to find metrics which can reflect the dynamic protocol behavior. In the future, we plan to continue the analysis with other high speed protocols and more complex network environments. Our aim is to define a dynamics sensitive fairness metric for performance evaluation of transport protocols for next generation high bandwidth-delay product networks.

References

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

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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, 29 Aug. - 2 Sep., 2005.

[9] 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.

[10] 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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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/Tcp

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

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

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

[19] 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.

[20] 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.