# A Simulation Based Comparison Between XCP and HighSpeed TCP

Gleb Chuvpilo and Jae Wook Lee
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
{chuvpilo, leejw}@lcs.mit.edu

## Abstract

*TCP congestion control performs poorly when the per-flow available bandwidth is very large. The problem is caused by the fact that the TCP increase rule is indifferent to the spare bandwidth in the network keeping the increase /decrease factor constant. Thus, it might take TCP more than thousands of RTTs to ramp up to full utilization. Several new protocols are proposed to deal with this problem, and they all claim to be able to scale well from megabits to gigabits per second. In this work, we perform a comparison study between two proposed protocols - eXplicit Control Protocol(XCP)[7] and HighSpeed TCP [5]. They differ in performance and deployment tradeoffs. In particular, we focus on deployment issues dealing with three questions. First, we discuss how safely both protocols are deployable into the current Internet. Second, we examine the argument of HighSpeed TCP's gradual depolyment path. Lastly, we also investigate the effects of buffer size on deployment, and its implication in the future Internet. In performing this study, we base our conclusion on simulation results with both simple and complex topologies.*

## 1 Introduction

The Transmission Control Protocol (TCP) dates back to 1981, and its behavior was optimized for the common case at the time, when available network bandwidth was measured in tens, or at most hundreds of megabytes per second. Undoubtedly, TCP congestion control mechanism has worked very well at least until recently, and has been a cornerstone for the Internet to florish. However, as people want larger (i.e. Gb/s) end-to-end throughput, network architects face challenging tasks to meet this needs with minimized costs. From technological standpoint, with the advent and wide deployment of optical carriers network capacity has grown by two to three orders of magnitude, and the TCP increase rule is starting to become inefficient at such link speeds. Indeed, in environments with high network bandwidth, it takes TCP more than thousands of round-trip times to reach full link utilization. Therefore, it is becoming increasingly important to study new approaches to transport protocols which would demonstrate good scaling from megabits to gigabits per second. There are a few proposed solutions to the problem, including XCP [7], QuickStart [6], and HighSpeed TCP [5]. Which protocol will be deployed in the future Internet is decided based on both the performance of the protocol and its ease of depolyment. In this paper we conduct a comparative study between XCP and HighSpeed TCP with an emphasis on depolyment issues and examine tradeoffs with both the network of a simple dumbbell topology and that of more complex and realistic environment setup.

### 1.1 Contributions

In this paper we examine three specific issues that involve the deployment of these two protocols.

1. **Are XCP and HighSpeed TCP safe to run in the Internet** The two protocols involve major changes to the dynamics of current congestion control. Both protocols replace the AIMD increase-decrease rule by a new one. In this paper, we first check the claims of the authors of XCP and HighSpeed TCP that their protocols are safe to deploy in the Internet; for instance, the new increase-decrease rules are responsive enough and will not drive the network into congestion collapse.

   We simulate XCP and HighSpeed TCP in a network of 21 nodes and 48 simplex links. Our simulations include hundreds of web sessions and FTP senders and reverse traffic which emulate more realistic network environments. Our simulations which are larger than the simulation presented in both the XCP and the HighSpeed TCP papers show safe and benign behavior in terms of efficiency, drop rates, and so on.

2. **Is HighSpeed TCP's gradual deployment path safe?** HighSpeed TCP was developed to address the difficult deployment issue involving XCP. HighSpeed TCP trade off optimal fairness and efficiency for a relatively easy gradual deployment path. Sender who have high capacity access nodes need only change their TCP stack to achieve a large end-to-end throughput. No router modification is required. In this paper, we examine the plausibility of this gradual deployment path and how the existence of some HighSpeed TCP senders affect the performance perceived by traditional TCP flows.

   Our results show that HighSpeed TCP is unfair to TCP and grabs more bandwidth. This behavior is benign as long as it happens only in very high bandwidth environments where TCP cannot acquire all of the bandwidth. However, our simulations show that HighSpeed TCP flows steal the bandwidth form TCP in moderate bandwidth environments which are common in the current Internet. In these moderate bandwidth networks, TCP can easily acquire the spare bandwidth and achieve high utilization. However, the replacement of some of the flows along these path HighSpeed TCP causes a drastic decrease in the bandwidth of the TCP flows which have not yet upgraded to HighSpeed TCP.

3. **The impact of buffer space on performance** the current recommendations require router manufacturers to provide a buffer whose size is comparable to delay bandwidth product of the path. As the link capacity increases to a tens of Gb/s, providing such a huge buffer would drastically increase the cost of the routers and impose technological problems such as heat dissepation and increased memory access latency.

   Our results show that both XCP and HighSpeed TCP achieve a reasonably good performance($> 90\%$) with the buffer size of 10% of the link capacity. If the requirement becomes even tighter, 1% of the capacity for example, in the future, resources-saving protocols like XCP may have an advantage over those protocols requiring more resources. If the benefit from saved resources (and performance as well in most cases) outweighs the deployment cost, it could be a viable alternative to replace the current TCP protocol.

The rest of the paper is organized as follows: Section 2 explains the design, implementation, and performance of the protocols in question as proposed solutions to the TCP problem in dealing with high-bandwidth networks. Section 3 is central to our contributions; We examine three deployment questions presented in the previous section in details, and also cover performance issues of the two protocols briefly. Finally, section 4 concludes the results of our work and suggests possible future works.

## 2 Background

### 2.1 XCP

XCP is a window-based protocol that generalizes the Explicit Congestion Notification (ECN) and introduces the new concept of decoupling utilization control from fairness control. XCP has several broad goals, including stability, fair bandwidth allocation, high utilization, small standing queue size, and near-zero packet drops. More specific goals are minimization of oscillations, efficiency for high bandwidth-delay connections, minimization of the transfer delay of short flows, ans fairness between flows with different RTTs.

XCP provides a joint design of both routers and end systems. According to the protocol, sendes maintain two parameters, congestion window size `cwnd` and round trip time `rtt`, and communicate them to routers by inserting them in packet headers (called congestion headers). The header also contains the feedback field, and it is initialized to the desired increase in bytes in the `cwnd` per ACK. Routers can modify the feedback field to decrease `cwnd` if they are congested, but they can never increase it. Therefore, the feedback field of the packet arriving at the sender will have the information from the bottleneck of the connection. Now what the receiver has to do is just simply send back the feedback field to the sender. When each ACK arrives at the sender, `cwnd` is updated by the value of the feedback field.

Routers deal with efficiency and fairness separately. The Efficiency Controller looks at the aggregate traffic and computes the desired change in the number of arriving bytes in a control interval (i.e., an average RTT), based on the spare bandwidth and persistent queue. The Fairness Controller uses AIMD to allocate the increase or decrease on a per-packet basis. The router, therefore, does not need to maintain per-flow state, and moreover, the computational resources needed are very moderate – a few additions and three multiplications per packet. Thus, it can be readily implemented in high-speed routers.

### 2.2 HighSpeed TCP

HighSpeed TCP is a modification to TCP's congestion control mechanism for use with TCP connections with large congestion windows.[5] In the network with a high bandwidth-delay product, it imposes unrealistic constraints for a standard TCP response function[1] to increase *cwnd* so that it can fully utilize the available bandwidth.

In order to address this fundamental limitation of TCP and of the TCP response function, HighSpeed TCP modifies the TCP response function for regimes with higher congestion windows. In short, as the current window size increases, the additive factor ($A$) also increases and the decrease factor ($B$) gets smaller. The overall effects of this modification is grabing the available bandwidth more aggresively and making the oscillation of TCP's window size smoother. Because HighSpeed TCP's modified response function would only take effect with higher congestion windows, HighSpeed TCP does not modify TCP behavior in environments with mild to heavy congestion, and therefore does not introduce any new dangers of congestion collapse.

HighSpeed does not address the mechanism required to enable best-effort connections to start with large initial windows. Starting with a large initial window requires some forms of explicit feedback from all of the routers along the path. On the other hand, HighSpeed TCP does not need any modification of the functionality of existing routers, which helps deploy the protocol.

## 3 Evaluation of the Protocols

We divide research questions related to the protocols into two categories. The first catetory is performance-oriented, and the second is deployment-oriented. Sometimes their boundary is blurred, but we can separate them with the following guidelines.

Performance-oriented questions address performance gain by a protocol in terms of conventional measures such as utilization and fairness. On the other hand, deployment-oriented questions ask how easily the protocol can be integrated into the real Internet. These questions include deployment costs, negative side effects, and robustness of the protocol, for example.

Because robustness in heterogeneous environments is often valued over efficiency of performance in well-defined environments[4], it is necessary to carefully evaluate the depolyment costs of a protocol and the plausibility of its deployment strategy as well as performance. Hence we primarily focus on deployment-related issues of the two proposed protocols in the following three sections, and partially explore performance tradeoffs in section 3.4.

### 3.1 Safety

The first deployment-related question is whether the proposed protocols actully *work* in the current Internet. Since both protocols make fundamental changes to the response function of TCP protocol, we cannot guarantee that they will be functioning in the real network environments as well.

The authors of HighSpeed TCP performed a series of simulations to prove its safety for depolyment. However, all the simulation results made public are based on a simple dumbbell topology with varying parameters.[1] XCP paper addresses issues such as multiple bottlenecks and existence of cross traffic by including simulation results with a more complex topology, called parking lot topology, but their simulation has the same limitation from complexity standpoint.[7] More specifically, they still have a small number of nodes (less than 10), don't have any loop or multiple paths, and lack randomness in topology and type of traffic.

Therefore, we generated a reasonably complex random topology using *tiers* with minor modification as shown in figure 1. [2][2] Important figures are summarized in table 1.

| | |
|---|---|
| Number of Nodes | 21 |
| Number of Links (Simplex) | 48 |
| Number of Loops | 4 |
| Number of Highspeed Flows | 5 |
| Number of Web Servers | 2 |
| Number of Web Clients | 10 |
| Number of Web Sessions | 300 |
| Number of Perturbation FTP Flows | 200 |
| Highspeed path BW | *variable* |

**Table 1.** Characteristics of the Random Topology

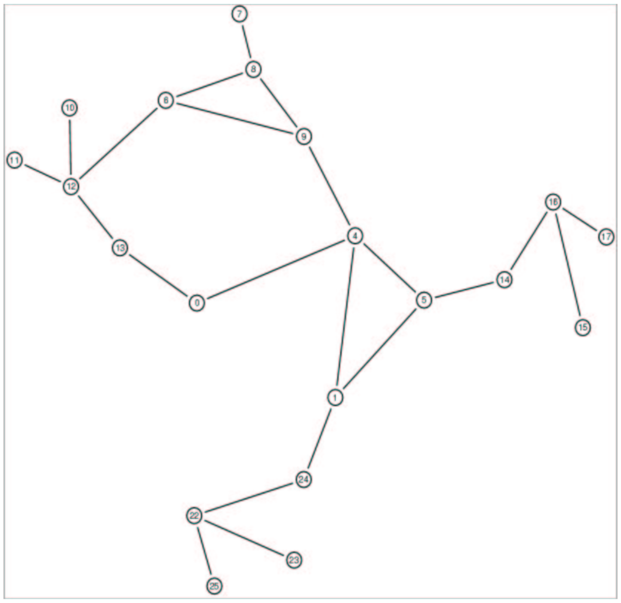
In the topolgy, the path between node 24 and node 5 is the highest bandwidth links in the network, and the link bandwidth decreases as we move toward a leaf node. The latency and the bandwidth of each link were randomly assigned by *tiers*. Web servers are placed at node 9 and node 14, and perturbation FTP flows are distributed randomly across the entire network.

Figure 2 shows that the behavior of both XCP and HighSpeed TCP scales in a benign manner even in our complex topology by achieving over 90% utilization across the high bandwidth links. One thing to mention is that standard TCP performs slightly better than HighSpeed TCP in case of 800 Mbps bandwidth links, but our focus here is safety of the protocol in this complex topology rather than its performance. For example, we observe that the protocol is reasonably responsive to a congestion event and that there is no undue congestion collapses introduced by the protocol. Our simulation with simple dumbbell topology confirms that HighSpeed TCP outperforms standard TCP in

---

[1] TCP response function is the function mapping the steady-stae packet drop rate to TCP's average sending rate in packets per round-trip time.

[2] We modified a random topology generated by tiers mainly for reducing simulation time and adding complexity.



Web Server {9, 14} → Web Client {6, 7, 10, 11, 12, 13, 15, 17, 23, 25}

FTP Sender-Receiver pairs (1 pair corresponding to 20 flows)

| Flow ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Sender | 12 | 10 | 23 | 17 | 13 | 13 | 9 | 11 | 12 | 22 |
| Receiver | 8 | 17 | 16 | 22 | 16 | 7 | 17 | 22 | 14 | 16 |

**Figure 1.** A Random Complex Topology for Simulation

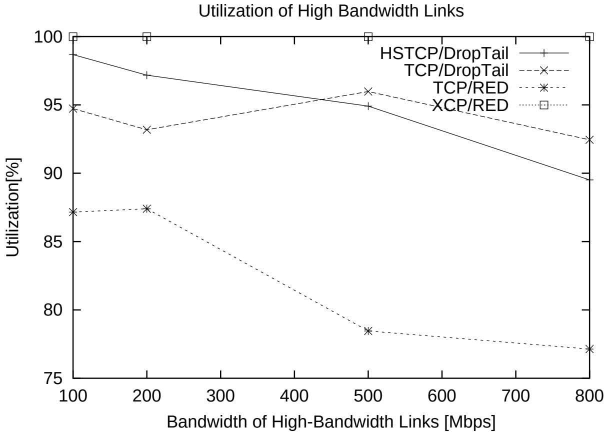


**Figure 2.** Utilization across the High Bandwidth Links (RTT=40ms, QSize=BW*RTT)

higher bandwidth links such as 1Gbps, 2Gbps, and so on. However, in this complex topology we could not simulate such a high bandwidth link due to limited time.

Even if we did not find any adverse effects of both protocols from effieciency standpoint, we are not concluding that it is guaranteed to be safe in any other topology and with any type of traffic. Rather, it is just a first step to evaluate their robustness in more realistic environments, and furthur investigation needs to be done on this issue.

## 3.2 Gradual Deployment of HighSpeed TCP

In general, HighSpeed TCP is known to have an advantage over other proposed protocols pursuing the same goal considering deployment costs, because unlike XCP[7] and QuickStart[6] it does not require an explicit feedback from the routers on the path.

However, this advantage might be weakened if there are side effects which negatively affects the performance of the existing Internet infrastructure. In this section, we assess the gradual depolyment strategy of HighSpeed TCP based on our simulation results. Especially, we investigate interaction between HighSpeed TCP and conventional TCP flows.

### 3.2.1 Simple Dumbbell Topoloogy

In order to observe the interaction clearly, we chose the simplest simulation setup to the first step. Figure 3 shows a simple dumbbell topology with only two flows – one HighSpeed TCP flow and one standard TCP flow. We followed the parameter setting for HighSpeed TCP simulations in [1].

As shown in figure 4 (A), TCP flows are fair to each other and eventually get to the fair share of available bandwidth. However, if we replace one of the flows by a HighSpeed

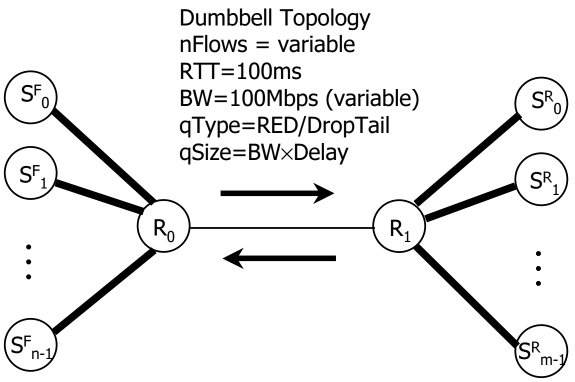


**Figure 3.** Simple Dumbbell Topology for Simulation

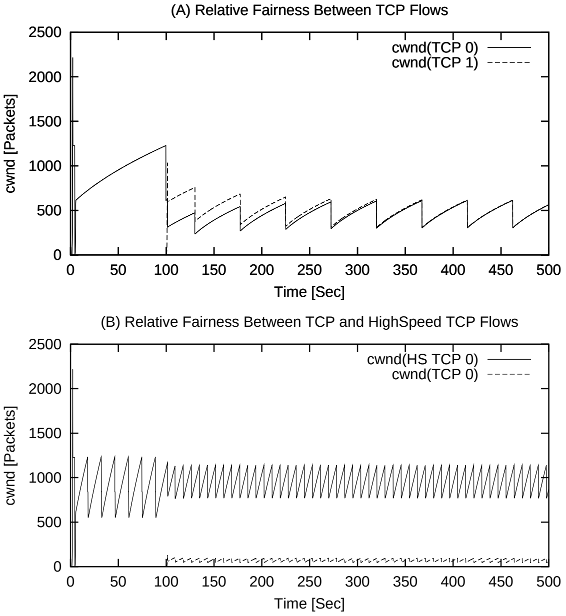


**Figure 4.** Relative Fairness Between Flows (RTT=100ms, BottleneckBW=100ms, DropTail)

TCP flow in our setup, fairness between flows disappears as in figure 4 (B). Instead, a HighSpeed TCP takes over ten times more bandwidth than the standard TCP flow. An important point is that this behavior is harmless if it happens only in very high bandwidth environments where TCP cannot acquire all of the available bandwidth. Figure 4 presents that this bandwidth is not an extra bandwidth which standard TCP flows cannot grab, but a fair share of the standard TCP flow.

The author of HighSpeed TCP protocol mentioned this fairness implication of the HighSpeed response function in her paper.[5] She predicts that this degree of unfairness is likely to occur in the network whose bandwidth is about 1.6 Gbps, and there are not many of TCP connections effectively operating in this regime today. However, according to our simulation result, this amount of unfairness can happen even in a low or moderate bandwidth network like 50Mbps, which falls on the plausible operating bandwidth range of today's Internet. Note that our simulation setup is not a pathodological case at all, and all the parameters are in a reasonable range. This implies that this kind of situation may appear in the real Internet environments, and possibly undermines the argument of gradual deployment of HighSpeed TCP.

### 3.2.2 More Complex Topology

To understand better how this phenomenon manifest itself in a real situlation, we designed a simulation based on the complex topology presented in section 3.1. To the beginning, we randomly place multiple standard TCP sender-receiver pairs(e.g. 30 pairs) with long flows, then run simulation once to get the rank of high end-to-end throughput flows. Once we get the profile of throughputs, we replace a subset of TCP flows with a high throughput[3] to HighSpeed TCP flows.

Due to time limitation we did not perform the simulation. However, by *gedanken* experiments based on the results in section 3.2.1, we expect that the HighSpeed TCP flows would steal the bandwidth of the existing TCP flows, resulting in ”the rich gets richer, and the poor gets poorer” situation. This phenomenon would negatively affect the performance perceived by traditional TCP flows.[4]

## 3.3 Impact of Buffering

Buffer space at the routers is important resource to provide efficiency to network infrastructure. The current recommendation require routers to have a buffer with the size comparable to delay-bandwidth product. However, as we migrate to high delay-bandwidth region, it becomes more and more infeasible to have a buffer in the routers on a path,

---

[3] It complies to the real-world situation to upgrade high-throughput TCP flows to HighSpeed TCP flows, because HighSpeed TCP does not help improving the performance of low-throughput flows very much. Rather, this protocol motivates the users with a high throughput to acquire the available bandwidth faster. Of course, this argument reflects only technological point of views, but probably not political or economical.

[4] In order to quantify users' satisfaction, the concept of utility function was introduced.[8] Then, we can design the network infrastructure to optimize the aggregate utility.

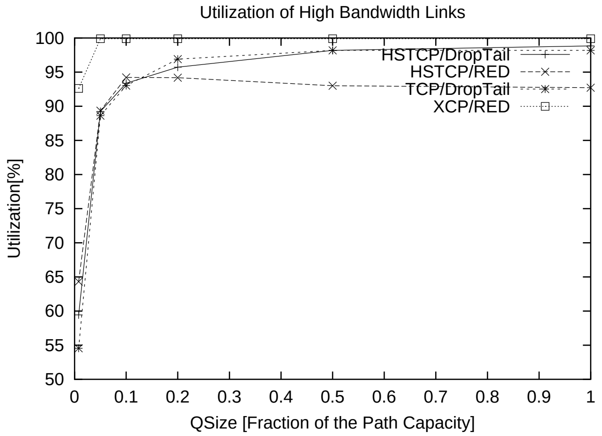


**Figure 5.** Utilization across the High Bandwidth Links (RTT=40ms, QSize=BW*RTT, HighSpeedBW=100Mbps)

whose size is comparable to the capacity of the path (or even a fraction of the capacity) in the routers.

As we expect, there is a tradeoff between buffer size(resources) and utilization(performance). In addition, our simulation with the complex topology shows that each protocol has diffenrent level of requirements on buffersize.

Figure 5 depicts how utilization of the high bandwidth link varies using the complex topology presented in figure 1. According to the simulation results, TCP, XCP, and HighSpeed TCP work well($> 90\%$ link utilization) in this specific simulation when we have the buffer size of more than 10% of path capacity.

In this specific situation, the performance of HighSpeed TCP flows versus buffer size is comparable to that of standard TCP flows. XCP flows are the most insensitive to variation of buffer size. This implies that the issue of buffer space can be a motivation to adopt resource-saving protocols (like XCP in this example) in the future Internet.

### 3.4 Performance

We also studied performance tradeoffs as a side track along with deployment tradeoffs. Utilization issue was addressed in 3.1. As far as fairness is concerned, each protocol is basically fair to itself; Figure 6 suggests that XCP, HSTCP, and standard TCP have a comparable fairness index ($> 0.97$) according to Jain's definition.[3] In his definition, the perfect fairness corresponds to the index value of 1, and the worst fairness to $1/n$ where $n$ is the number of flows. Fairness between heterogeneous flows is partly covered in the section 3.2.

For dynamics, we have found an interesting tradeoff. In many cases, HighSpeed TCP flows converges slower to fairness than standard TCP flows. Figure 7 in Appendix depicts

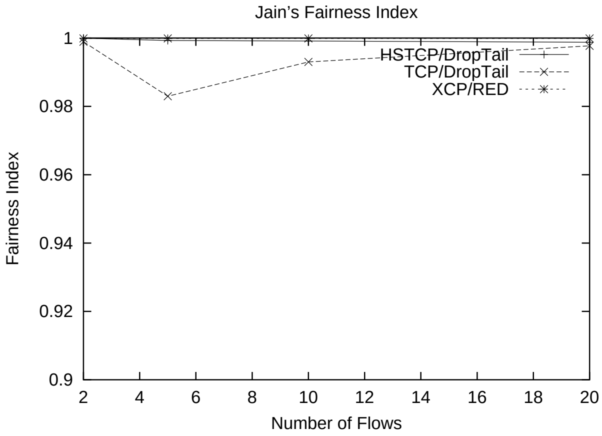


**Figure 6.** Jain's Fairness Index over a Simple Dumbbell Topology (RTT=100ms, QSize=BW*RTT, HighSpeedBW=200Mbps)

this phenomenon. On the other hand, HighSpeed TCP flows are much faster in grabbing available bandwidth than standard TCP flows as shown in the figure 8 in Appendix. Both plots are generated with a simple dumbbell topology as in figure 3. Here we claim that there is a tradeoff between convergence to fairness and convergence to full utilization, but it is still an open problem whether it is correct or not. We do not go for any further arguments on the issue, because it is not the primary focus of this project.

## 4 Conclusions and Future Work

As networks with a high throughput (e.g. optical link) and/or a high latency (e.g. satellite link) become more prevalent, increase/decrease function of conventional TCP protocol does not work efficiently for such high-bandwidth-delay-product networks. There have been efforts to address this problem recently, and new protocols have been proposed such as XCP, HighSpeed TCP, and QuickStart TCP, for example.

In this study, we performed a simulation-based comparison study between XCP and HighSpeed TCP, focusing on deployment issues. As far as methodology is concerned, we understood the properties of each protocol by running various simulations on a simple dumbbell topology first, and also explored its behavior in more realistic simulation setup.

Here we reiterate our conclusion as following:

First, the behavior of both XCP and HighSpeed TCP scales in a benign manner in our specific simulation setup with a complex topology, heterogeneous perturbation flows, various link bandwidth, latency, and buffer size. We did not find any adverse effects of the protocols, but in order to guarantee better safety for depolyment, further simulations need to be done.

Second, HighSpeed TCP is known to be more easily deployable, because it requires only end-to-end modification of proven TCP protocol. However, under a certain condition HighSpeed TCP flows can starve standard TCP flows, even in a low or moderate bandwidth network where TCP flows could achieve a high efficiency without existence of HighSpeed TCP flows. This phenomenon may undermine the arguments for gradual deployment strategy of the protocol.

Third, since buffer space in routers is expensive resources, buffering cost is getting more and more important issues in deploying a new protocol in high bandwidth-delay networks. TCP, XCP, and HighSpeed TCP perform reasonably well ($> 90\%$ utilization) with buffer size of 10% of the link capacity. In the future, this condition may become tighter such that we are allowed to have a even smaller fraction of the link capacity as a buffer size. This could be another motivation to deploy a resources-saving protocol like XCP. Of course, there is a tradeoff, and we have to compare deployment cost of XCP with its benefit from performance and saved resources.

We also covered performance implication of each protocol briefly in this work. As a future work, it would be an interesting study to compare these two protocols with QuickStart TCP and other protocols which aim to achieve the same goal.

## References

[1] Highspeed tcp simulation reports. http://www-itg.lbl.gov/ evandro/hstcp/simul/simul.html.

[2] The network simulator ns-2: Topology generation. http://www.isi.edu/nsnam/ns/ns-topogen.html#tiers.

[3] D. Chiu and R. Jain. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks. *Journal of Computer Networks and ISDN*, 17, 1:1–14, June 1989.

[4] S. Floyd. Congestion control for high-bandwidth-delay-product networks: Xcp vs. highspeed tcp and quickstart, September 2002.

[5] S. Floyd. HighSpeed TCP for Large Congestion Windows, August 2002. IETF Internet draft.

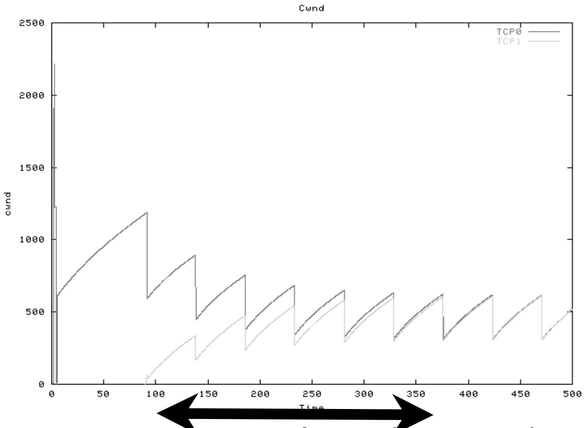
[6] A. Jain and S. Floyd. Quick-Start for TCP and IP, August 2002. IETF Internet draft.

[7] D. Katabi, M. Handley, and C. Rohrs. Internet congestion control for future high bandwidth-delay product environments, 2002.

[8] S. Shenker. Fundamental design issues for the future internet. *IEEE Journal on Selected Areas in Communication*, 13(7), September 1995.

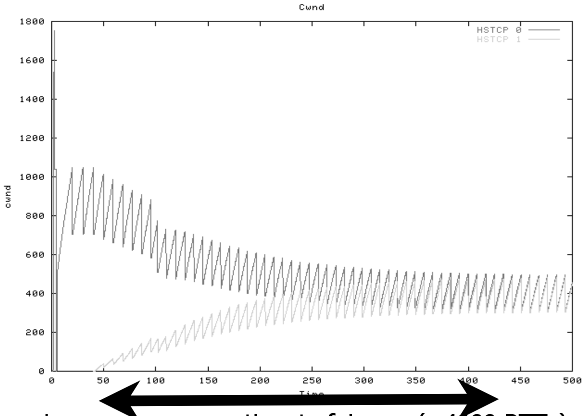
## Appendix

**TCP**



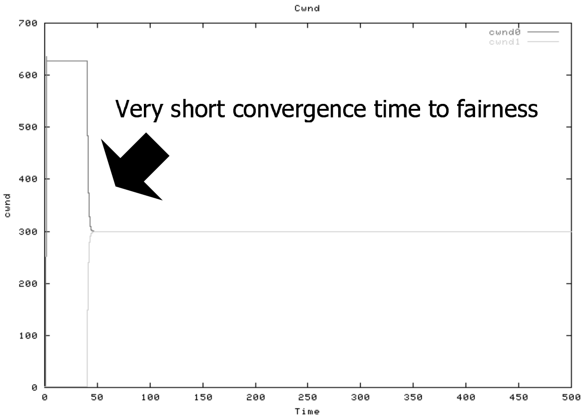
Long convergence time to fairness (~1700 RTTs)

**HSTCP**



Long convergence time to fairness (~4000 RTTs)

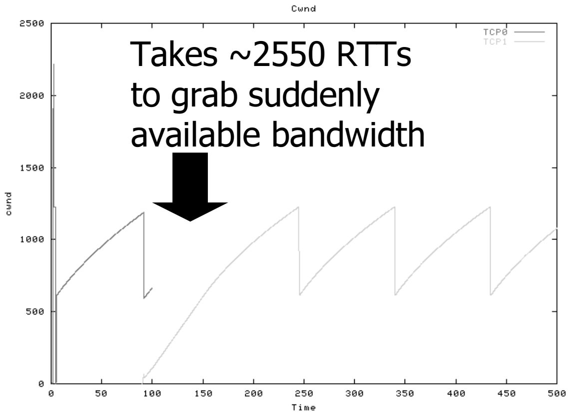
**XCP**



Very short convergence time to fairness

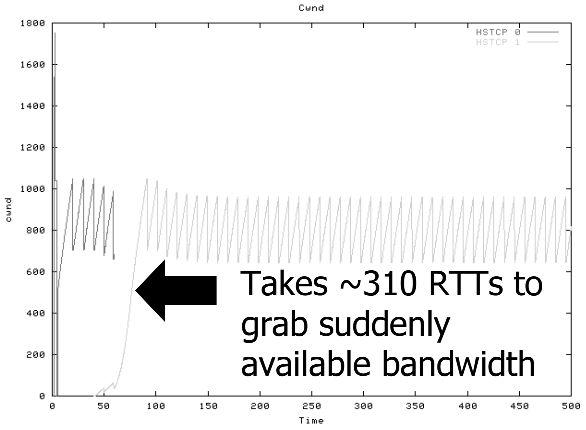
**Figure 7.** Dynamics 1: Convergence to Fairness (RTT=100ms, QSize=BW*RTT, HighSpeedBW=50Mbps)

**TCP**



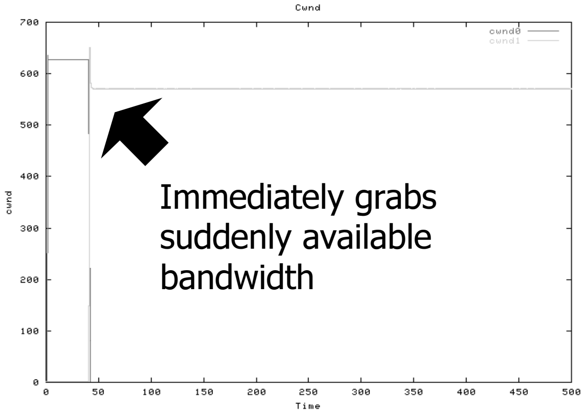
Takes ~2550 RTTs to grab suddenly available bandwidth

**HSTCP**



Takes ~310 RTTs to grab suddenly available bandwidth

**XCP**



Immediately grabs suddenly available bandwidth

**Figure 8.** Dynamics 2: Fetching Suddenly Available Bandwidth (RTT=100ms, QSize=BW*RTT, HighSpeedBW=50Mbps)