Effects of Applying High Speed Congestion Control Algorithms in the Internet*

Xiuchao Wu
School of Computing, National University of Singapore
Computing 1, Law Link, Singapore 117590
Email:{wuxiucha}@comp.nus.edu.sg
November 3, 2007

ABSTRACT

In recent years, many high speed congestion control algorithms have been proposed for utilizing network pipes with large bandwidth-delay product, and some of them have also been implemented in popular operating systems. Since the Internet is and tends to be lightly-loaded and there are many bandwidth-greedy applications, high speed congestion control algorithms may be used by many flows of the Internet. Considering that the Internet is now very important, it is necessary to make sure that applying high speed congestion control algorithms will not hurt the existing applications, especially the interactive World Wide Web and streaming applications, such as Voice over IP.

In this paper, several influential high speed congestion control algorithms, HS-TCP, H-TCP, Cubic-TCP, and delay-based Compound-TCP, are evaluated with the focus on their effects to existing applications. Many simulations are carried out for answering whether it is safe to apply high speed congestion control algorithms in high speed wide area network and/or local area network of the Internet, and which algorithm performs the best.

Through this study, we find that it is safe and profitable to apply Cubic-TCP inside high speed local area network. Network utilization ratio can be improved and the experience of users, that communicate with remote hosts of the Internet through existing applications, is not adversely affected.

As for high speed wide area network, we find that with large amount of web traffics, routers should be provisioned with a large queue for accommodating their burstiness. With this prerequisite, all algorithms are not satisfying in high speed wide area network. Loss-based HS-TCP, H-TCP, and Cubic-TCP inevitably increase queue delay experienced by the existing applications, H-TCP also causes much higher packet loss rate, and these QoS (Quality of Service) degradation can be perceived by users. Delay-based Compound-TCP can well utilize the network without adversely affecting the existing applications only when congestion can be detected through queue delay. Considering that queue delay detection method of Compound-TCP is not scalable with the number of Compound-TCP flows, a new delay-based high speed congestion control algorithm, which is scalable with flow number, should be a promising solution for high speed wide area network of the Internet.

Simulation results of this study also indicate that end hosts should be a little more intelligent and use different high speed congestion control algorithms on different network pipes of the heterogeneous Internet.

*This work is supported by National University of Singapore under research grant R-252-000-203-112.

Categories and Subject Descriptors

C.2.5 [**Computer Systems Organization**]: Computer Communication Networks—*Local and Wide-Area Networks*

General Terms

Performance, Measurement.

Keywords

Congestion Control, High Speed Networks, Friendliness.

1. INTRODUCTION

TCP [22] is the de-facto standard protocol of the Internet for uni-cast reliable data transmission. But it is also well known that TCP's congestion control algorithm [8], AIMD (Additive Increase and Multiplicative Decrease), can not work well on network pipes with large bandwidth-delay product (BDP) [5][13]. On these network pipes, TCP becomes the performance bottleneck and network bandwidth is under-utilized. Based on this observation, many high speed congestion control algorithms, such as Scalable-TCP [14], HS-TCP [5], BIC-TCP [26], H-TCP [17], Cubic-TCP [23], Fast-TCP [11], TCP-Africa [15], and Compound-TCP [24], have been proposed in recent years.

It is very attractive to apply these high speed congestion control algorithms for providing high throughput to bandwidth-greedy applications, such as data backup, content distribution, and file sharing. Some of these algorithms have been implemented in popular operating systems. For example, Compound-TCP is implemented in Windows Vista and Linux is distributed with several algorithms which can be easily selected through a socket option. These high speed congestion control algorithms normally switch to fast mode when a flow finds that its sending window is larger than a threshold. In HS-TCP and Compound-TCP, this threshold is 38 and 41 segments respectively. Considering that network bandwidth increases quickly, the Internet is and tends to be lightly-loaded [21], and Peer-to-Peer communications are more selfish and irresponsible, these high speed congestion control algorithms may run over many network pipes of the Internet.

Flows driven by high speed congestion control algorithms are more aggressive and inevitably increase the load of the Internet. Since many tasks of our daily life and business are being carried out in the Internet, it is very important to make sure that applying high speed congestion control algorithms will not hurt the existing applications, especially the interactive Wide Wide Web (WWW) and streaming applications, such as Voice over IP (VoIP).

But the existing high speed congestion control algorithms are normally evaluated in the metrics of throughput, convergence speed, and fairness of flows driven by high speed congestion control algorithms. Although TCP friendliness had also been considered, the throughput of a concurrent long-lived FTP flow is normally the only metric. In addition, high speed congestion control algorithms are normally evaluated on network pipes without large amount of background web traffics, which do exist in the Internet. Hence, it is necessary to evaluate them on network pipes with large amount of background web traffics, and user experience of the existing applications, especially the interactive WWW and streaming applications, should be used as the friendliness metrics of high speed congestion control algorithms.

Such a study is carried out in this paper. Several influential high speed congestion control algorithms, HS-TCP [5], H-TCP [17], Cubic-TCP [23], and Compound-TCP [24], are evaluated and compared on simulated high speed wide area network and local area network of the Internet with the aim of answering whether it is safe to apply high speed congestion control algorithms in wide area network and/or local area network, and which algorithm performs the best. Different queue sizes are also used in simulations with the aim of investigating how to provision queue for well accommodating flows driven by high speed congestion control algorithms.

This paper is organized as follows. Section 2 first introduces these high speed congestion control algorithms to be evaluated. Section 3 then describes the methodology used in our evaluation. Simulation results are presented and analyzed in section 4. Related works are introduced in section 5, and the paper is concluded in section 6.

2. HIGH SPEED CONGESTION CONTROL ALGORITHMS

In this section, we briefly introduce high speed congestion control algorithms to be evaluated in this paper. The reader should refer to the original literature for more information.

2.1 HS-TCP

HS-TCP [5] updates its congestion window (*cwnd*) based on equation 1. Carefully designed logarithmic functions are proposed for $\alpha(cwnd)$ and $\beta(cwnd)$, whereby $\alpha(cwnd)$ and $\beta(cwnd)$ increase with *cwnd* for quickly probing network pipes with large BDP, and both increase and decrease slopes decrease with the increase of *cwnd* for converging quickly and avoiding large burst of segment loss.

$$cwnd = \begin{cases} cwnd + \frac{\alpha(cwnd)}{cwnd} & \text{per ack,} \\ \beta(cwnd) \times cwnd & \text{per loss.} \end{cases} \quad (1)$$

2.2 H-TCP

Equation 2 shows the rules used by H-TCP [17] for updating its *cwnd*. Here, $\beta = \min(\max(\frac{RTT_{min}}{RTT_{max}}, 0.5), 0.8)$ for high network utilization ratio. Its increase parameter is independent on the value of *cwnd*, and this independence from *cwnd* is helpful to convergence. As shown in equation 3, the parameter is also a quadratic increase function of Δ (the elapsed wall clock time since the last congestion event), and the threshold, $\Delta^L = 1$ second. This function makes H-TCP more scalable with RTT.

$$cwnd = \begin{cases} cwnd + \frac{2\times(1-\beta)\times\alpha(\Delta)}{cwnd} & \text{per ack,} \\ \beta \times cwnd & \text{per loss.} \end{cases} \quad (2)$$

$$\alpha(\Delta) = \begin{cases} 1 & \Delta \le \Delta^L, \\ 1 + 10(\Delta - \Delta^L) + (\frac{\Delta-\Delta^L}{2})^2 & \Delta > \Delta^L. \end{cases} \quad (3)$$

2.3 Cubic-TCP

Cubic-TCP [23] is an offspring of BIC-TCP [26]. As shown in equation 4, Cubic-TCP inherits the merit of BIC-TCP that, the closer *cwnd* is to W_{max} (the *cwnd* value when last congestion event occurs), the smaller its increase step is. The concavity of its window growth function can effectively reduce the burst of segment loss. In equation 4, β is the decrease parameter used when segment loss is detected and C is a carefully designed scaling factor. Just like H-TCP, for better RTT scalability, the window growth function of Cubic-TCP is also a function of the elapsed wall clock time since the last congestion event.

$$W_{cubic} = C(t - \sqrt[3]{W_{max} * \beta / C})^3 + W_{max} \quad (4)$$

2.4 Compound-TCP

Compound-TCP [24] is a delay-based high speed congestion control proposal from Microsoft and has been implemented in Windows Vista. Compound-TCP sender maintains two variables, *dwnd* and *cwnd*. Their sum ($win = cwnd + dwnd$) plays the same role of *cwnd* in other algorithms, the sending window probed through congestion control. *cwnd* is updated in the same way of regular TCP, and *dwnd* is updated based on queue delay information. Equation 5 shows these rules used by Compound-TCP.

$$cwnd = \begin{cases} cwnd + \frac{1}{cwnd} & \text{per ack,} \\ 0.5 \times cwnd & \text{per loss.} \end{cases} \quad (5)$$

$$dwnd = \begin{cases} dwnd + (\alpha \times win^k - 1)^+ & \text{per RTT, } \Delta < \gamma, \\ (dwnd - \zeta \times \Delta)^+ & \text{per RTT, } \Delta \ge \gamma, \\ (win \times (1-\beta) - \frac{cwnd}{2})^+ & \text{per loss.} \end{cases}$$

Here $(.)^+$ is defined as $max(., 0)$. That means a Compound-TCP flow will never receive less throughput than that received when the flow uses regular TCP. Δ is the number of packets backlogged at the bottleneck link and is estimated based on the same method used by TCP Vegas [2].

With these rules, when network is under-utilized (Δ is less than the threshold, $\gamma = 30$ segments, and there is no segment loss), sending window is increased binomially ($win = win + \alpha \times win^k$, per RTT) for probing network bandwidth quickly. After that, *cwnd* is continuously increased and *dwnd* reduces itself with the aim of maintaining γ packets at the bottleneck link and driving network to work at the knee [10]. After *dwnd* is reduced to *zero*, Compound-TCP

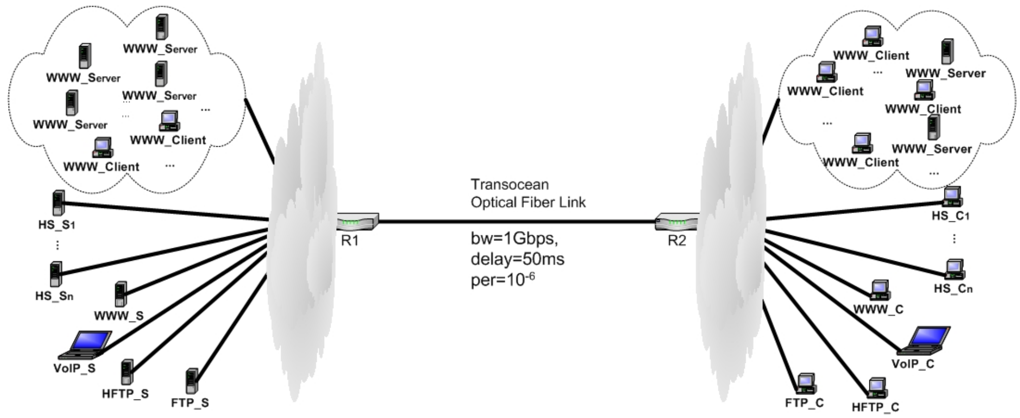


Figure 1: Wide Area Network Scenario: High Speed Congestion Control Algorithms on a Transocean Optical Fiber Link

acts as regular TCP and sending window is still increasing slowly until some segments are dropped or corrupted. When segment loss is detected, sending window is multiplicative decreased ($win = win \times (1 - \beta)$) for avoiding congestion collapse.

3. EVALUATION METHODOLOGY

Since experiments in the Internet may adversely affect users of the Internet, simulation is used to evaluate the effects of applying high speed congestion control algorithms in the Internet. NS-2 [1] is adopted due to its powerful functions and NS2-TCP-Linux [25] is patched for comparing these algorithms in a common TCP implementation.

The following subsections will describe the methods used for evaluating high speed congestion control algorithms on a simulated transocean optical fibre link and a simulated high speed campus network. By comparing their effects on the existing applications, we can answer whether is is safe to apply high speed congestion control algorithms in high speed wide area network and/or local area network of the Internet, and which algorithm performs the best.

3.1 Wide Area Network Scenario

The dumbbell network configuration in figure 1 is used for evaluating high speed congestion control algorithms in wide area network. The link between $R1$ and $R2$ is a simulated transocean optical fibre link whose bandwidth is 1Gbps, propagation delay is 50ms, and packet error rate is 10^{-6}. The other links are all highly reliable and very fast (bandwidth: 1Gbps, packet error rate: 0) so that the transocean optical fibre link is the only bottleneck.

3.1.1 *Background Web Traffics*

Between the two WWW node clouds of figure 1, HTTP sessions are generated according to the connection-based web traffic model [3]. The propagation delay of the links between WWW nodes and $R1/R2$ follows an uniform distribution whose range is 5 to 15ms. In the forward direction ($R1 \rightarrow R2$), 4000 HTTP sessions are generated per second, and 1000 HTTP sessions are generated per second in the reverse direction. Hence, similar to the real Internet, background web traffics consume quite large amount bandwidth of the simulated transocean optical fibre link.

3.1.2 *High Speed Flows*

N flows driven by high speed congestion control algorithms are established between HS_S$_i$ and HS_C$_i$. Unless specified, N equals to 4. *window_*, NS parameter for socket buffer, is set to 100000 packets so that the sending rate of these flows is solely determined by congestion control. In order to mitigate the presence of phase effects, as was done in [26], we set *overhead_*, NS parameter for node processing delay, to 8×10^{-6}. The propagation delay between HS_S$_i$/HS_C$_i$ and $R1/R2$ also follows the uniform distribution whose range is 5 to 15ms.

3.1.3 *Flows for Collecting User Experience*

In order to evaluate a high speed congestion control algorithm's effects on existing applications from user experience point of view, a client (WWW_C) continuously accesses a web server (WWW_S) which sends back responses with fixed sizes (1KB, 2KB, 4KB, 8KB, 16KB, 32KB, 64KB, or 128KB). The response time of these HTTP transactions is used to evaluate its effects on WWW.

A VoIP connection is established between VoIP_S and VoIP_C. We use the ITU G.711 PCM VoIP traffic as our voice source. The data rate during talk spurt is 87.2 Kbps, including the relative protocol headers. Each packet size is 218 bytes. The average burst time is 0.4s, average idle time is 0.6s, and the distribution follows an exponential ON/OFF model. One way delay from VoIP_S to VoIP_C and its jitter are used to evaluate a high speed congestion control algorithm's effects on VoIP, a typical streaming application. Considering that there may not be enough VoIP packets for accurately measuring a low packet loss rate, we use packet loss rate of the bottleneck link as the packet loss rate experienced by VoIP packets.

A FTP connection (HFTP) is established between HFTP_S and HFTP_C. It uses a TCP agent whose *window_* is set to 100000. Its throughput is used to evaluate a high speed congestion control algorithm's effects on long-lived FTP flows with support of window scale option [9] and without buffer-limitation. Another FTP connection (FTP) is established between FTP_S and FTP_C. It uses a TCP agent whose *window_* is set to 64. Its throughput is used to evaluate a high speed congestion control algorithm's effects on long-lived FTP flows with buffer limitation and/or without sup-

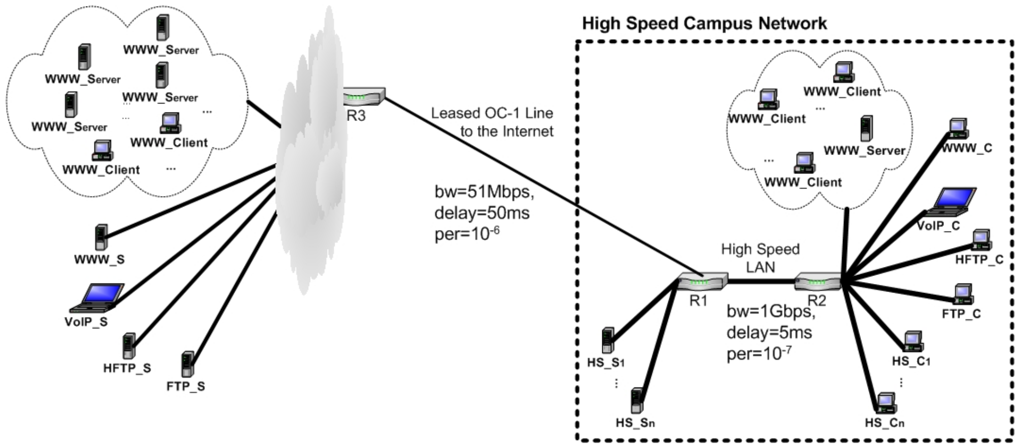


Figure 2: Local Area Network Scenario: High Speed Congestion Control Algorithms on a High Speed Campus Network

port of window scale option.

The propagation delay between the above nodes and $R1/R2$ is set to 10ms.

3.1.4 Experiments

In order to investigate how to provision queue for well accommodating flows driven by high speed congestion control algorithms, $R1$ uses DropTail queue and queue size is set to 0.02, 0.05, 0.1, 0.2, 0.5, or 1 BDP of the simulated transocean optical fibre link.

For each queue size, five experiments are carried out. HS-TCP, H-TCP, Cubic-TCP, and Compound-TCP are used by high speed flows in different experiments. For comparison, TCP (SACK [20]) is also used by these flows in the remaining one experiment. For each experiment, simulation runs for 830 seconds and figure 3 shows the start and end sequence of different kinds of flows.

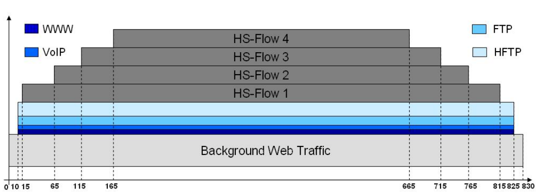


Figure 3: The start and end sequence of different flows

3.2 Local Area Network Scenario

As shown in figure 2, in local area network scenario, we simulate a high speed campus network (bandwidth: 1Gbps, delay: 5ms, packet error rate: 10^{-7}) that connects to the Internet through a leased OC-1 line (bandwidth: 51Mbps, delay: 50ms, packet error rate: 10^{-6}). In this scenario, we want to investigate whether it is safe to apply high speed congestion control algorithms inside a high speed local area network, and which algorithm performs the best. We focus on their effects on the experience of users that communicate with remote hosts of the Internet through existing applications, especially the interactive WWW and streaming applications.

3.2.1 Background Web Traffics

Between the two WWW node clouds of figure 2, background web traffics are also generated. In the forward direction ($R3 \to R1 \to R2$), 500 HTTP sessions are generated per second, and 125 HTTP sessions are generated per second in the reverse direction. The propagation delay of the links between WWW nodes and $R2$ follows an uniform distribution whose range is 1 to 3ms, and the propagation delay of the links between WWW nodes and $R3$ follows another uniform distribution whose range is 5 to 15ms.

3.2.2 High Speed Flows

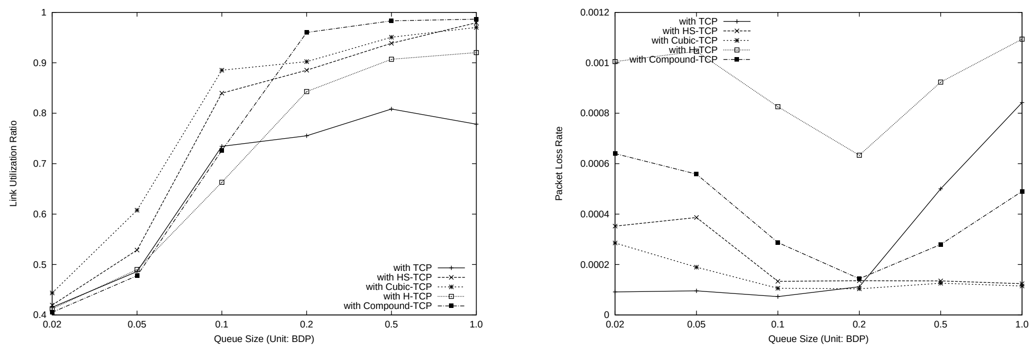
Just like the wide area network scenario, N flows driven by high speed congestion control algorithms are established between HS_S$_i$ and HS_C$_i$, and the default value of N is 4. The propagation delay between HS_S$_i$/HS_C$_i$ and $R1/R2$ follows the uniform distribution whose range is 1 to 3ms.

3.2.3 Flows for Collecting User Experience

WWW, VoIP, HFTP, and FTP connections are also established for evaluating a high speed congestion control algorithm's effects on the experience of users that communicate with remote hosts of the Internet through the existing applications. The propagation delay between client nodes and $R2$ is set to 1ms, and the propagation delay between server nodes and $R3$ is set to 10ms.

3.2.4 Experiments

DropTail queue is used by $R3$ and its size is set to 1 BDP of the leased OC-1 line. $R1$ also uses DropTail queue, and its queue size is set to 0.1, 0.2, 0.5, or 1 BDP of the simulated high speed campus network so that queue is not less than 50 packets. For each queue size, five experiments are also carried out. TCP (SACK), HS-TCP, H-TCP, Cubic-TCP, and Compound-TCP are used by high speed flows in different experiments. The start and end sequence of different kinds of flows is identical to wide area network scenario (shown in figure 3).



(a) Link Utilization Ratio

(b) Packet Loss Rate

Figure 4: Utilization Ratio and Packet Loss Rate of the Simulated Transocean Optical Fibre Link

4. SIMULATION RESULTS AND ANALYSIS

In this section, simulation results are presented and analyzed. Instead of the performance of flows driven by high speed congestion control algorithms, we focus on their effects on user experience of existing applications, especially the interactive WWW and VoIP. Performance of the whole network is also considered.

4.1 Wide Area Network Scenario

4.1.1 Simulation Results and Analysis

Figure 4 shows utilization ratio and packet loss rate of the simulated transocean optical fibre link, and figure 5 shows WWW, VoIP, and FTP user experience in this wide area network when 4 concurrent flows use HS-TCP, H-TCP, Cubic-TCP, Compound-TCP, or TCP.

Figure 4(a) shows that if queue is small (0.02 and 0.05 BDP), high speed wide area network can not be well utilized no matter which algorithm is used. The reason is that there are a large amount of bursty web traffics. When queue is small and these burstiness can not be accommodated, segments are dropped and sending rate is reduced even when the network is under-utilized. Figure 4(a) also indicates that high speed congestion control algorithms are really needed for utilizing network pipes with large BDP and large queue.

Figure 4(b) shows that H-TCP always causes the highest packet loss rate. The reason is that following the rules in equation 3, the window growth function of H-TCP is a convex function. When congestion occurs, its increase step can be quite large and many packets are dropped. The same reason can also explain the high packet loss rate of Compound-TCP when queue is small and congestion can not be detected through queue delay. Compound-TCP only achieves binomial window growth and its increase step can also be very large when congestion occurs. Figure 4(b) also shows that the concave Cubic-TCP and well designed HS-TCP can effectively reduce data burst and the number of segments dropped during a congestion event. One strange phenomenon is that packet loss rate of TCP and Compound-TCP is very high when queue is huge (0.5 and 1 BDP). We find that most of lost segments are dropped shortly after the join of new high speed flows. The reason is that TCP and Compound-TCP don't use any mechanism similar to Limited Slow Start [6] used by HS-TCP. The larger the queue is, the larger their increase step at the end of *slow start* phase is, and the more segments are dropped.

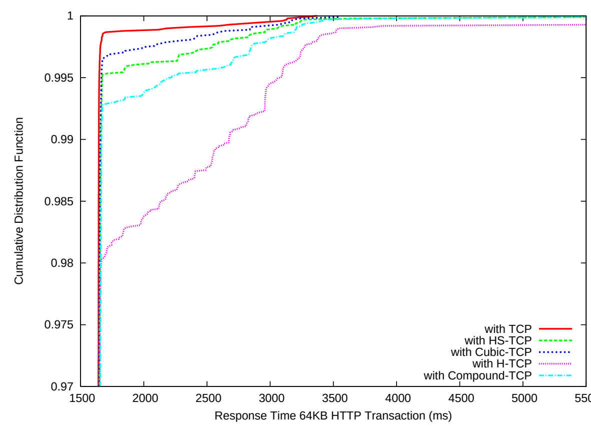
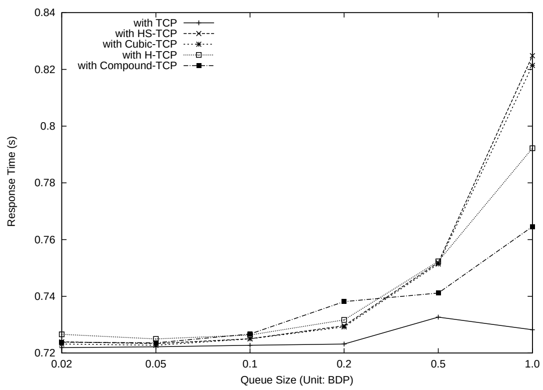
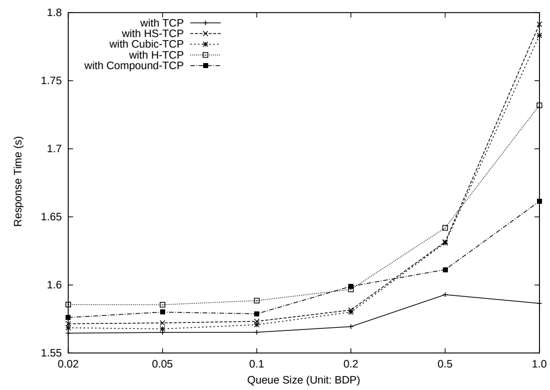


Figure 6: CDF of 64KB HTTP Transaction Response Time When Queue is 0.1 BDP

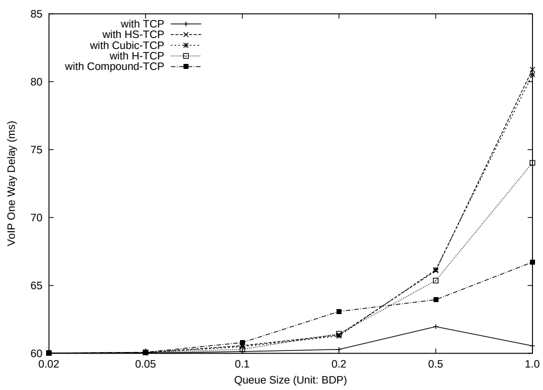
Irrespective of the size of response data, response time of HTTP transaction follows the same pattern. Due to space limitation, figure 5(a) and 5(b) only present average response time of HTTP transactions whose response data size is 4KB or 64KB. They show that loss-based high speed congestion control algorithms do increase the average response time of HTTP transactions a little. It is reasonable since these algorithms drive network to work near the *cliff* [10] during most of the time and HTTP packets experience long queue delay. As shown in figure 6, loss-based high speed congestion control algorithms, especially H-TCP, can cause user-perceptible increase of some HTTP transactions' response time. As for delay-based Compound-TCP, when queue is large, it can drive network to work around the knee and limit



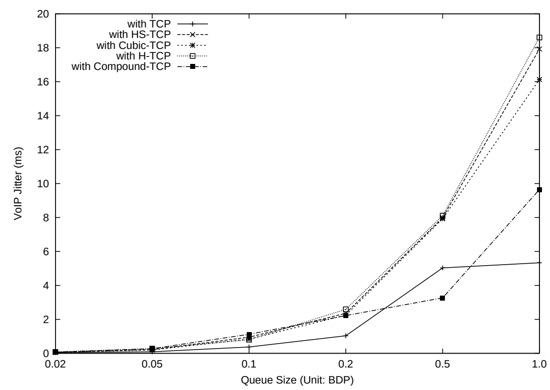
(a) WWW (Response Data Size=4KB)



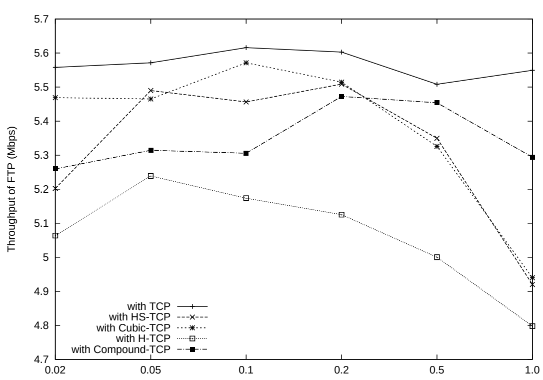
(b) WWW (Response Data Size=64KB)



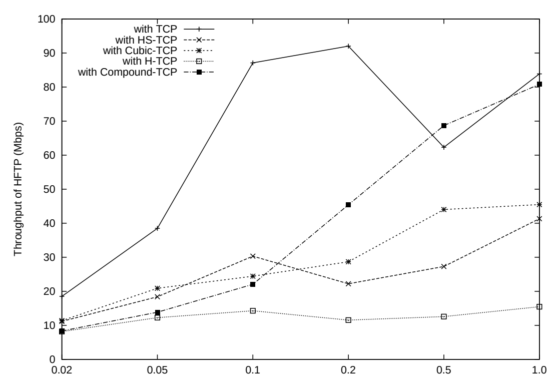
(c) VoIP (One Way Delay)



(d) VoIP (Jitter)

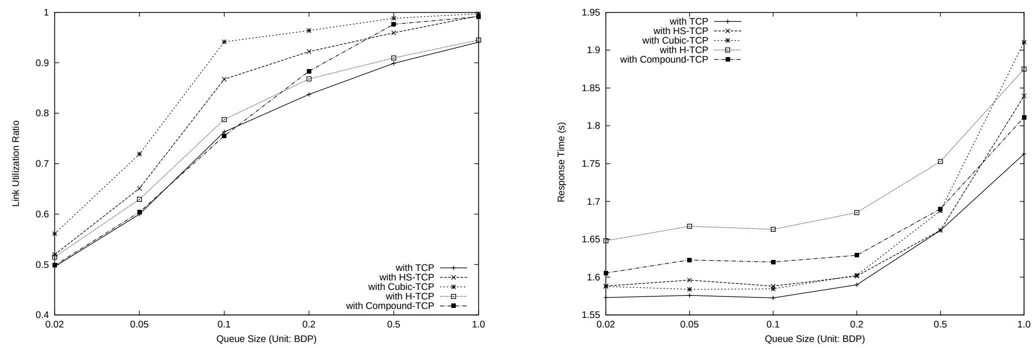


(e) FTP



(f) HFTP

Figure 5: WWW, VoIP, and FTP User Experience in Wide Area Network



(a) Link Utilization Ratio

(b) WWW (Response Data Size=64KB)

Figure 7: Twenty High Speed Flows on the Simulated Transocean Optical Fibre Link

the increase of HTTP transactions' response time. But when queue is 1 BDP, Compound-TCP still obviously increases the response time of HTTP transactions. The reason is that after *dwnd* is reduced to *zero*, Compound-TCP acts as regular TCP and slowly drives network to the cliff. When queue is huge, Compound-TCP drives network to leave the *knee* for a long time and the average response time of HTTP transactions is increased.

Figure 5(c) and 5(d) show the average one-way delay and jitter experienced by VoIP packets. Their pattern is similar to figure 5(a) and 5(b) due to the same reason. When queue is large, loss-based high speed congestion control algorithms obviously increase the delay and jitter suffered by VoIP packets, and this increase can be perceived by more sensitive VoIP users. As for delay-based Compound-TCP, when congestion can be detected through queue delay, it is also better than loss-based proposals in the two metrics.

Figure 5(e) and 5(f) show the throughput of FTP and HFTP connections. They indicates that Compound-TCP can be very friendly to long-lived TCP flows when queue is large enough so that Compound-TCP can detect congestion through queue delay. Loss-based high speed congestion control algorithms do decrease the throughput of FTP users and H-TCP performs the worst.

4.1.2 Discussion

According to previous section 4.1.1, routers of wide area network should be provisioned with large queue for accommodating the bursty web traffics of the Internet. With this prerequisite, loss-based high speed congestion control algorithms unavoidably increase response time of HTTP transactions and one way delay & jitter of VoIP packets. Their adverse effects on VoIP packets are quite obvious and may be perceptible to users. The throughput of concurrent FTP connections is also obviously reduced.

Among loss-based high speed congestion control algorithms, H-TCP performs the worst. It causes much higher packet loss rate and the largest jitter experienced by VoIP packets. In addition, its link utilization ratio is lower than other high speed congestion control algorithms and is even lower than TCP in some cases. Due the convexity of its window growth function, many segments are dropped during a congestion event and these large burst of segment loss harm cross traffics and H-TCP flows themselves. These results indicate that *convex window growth function is not suitable for loss-based high speed congestion control.*

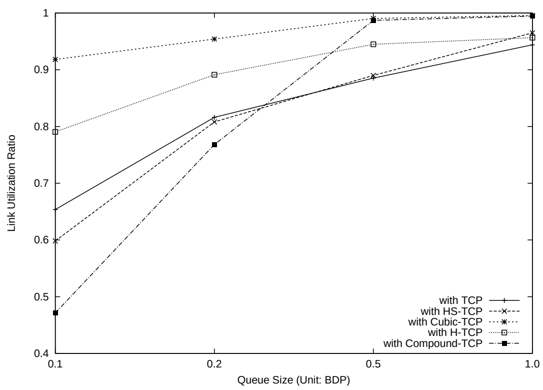
As for delay-based Compound-TCP, it is quite encouraging when queue is large ($\geq$0.2 BDP). It achieves the highest link utilization ratio and its adverse effects on user experience of the existing applications are also the slightest. But when queue is very large (1 BDP), the network still can leave the knee for a long time and queue delay experienced by interactive applications can be increased obviously. More severely, as shown in figure 4 and 5, when queue is not larger than 0.1 BDP, Compound-TCP can not detect congestion through queue delay, it only carries out binomial window growth, and the performance is very bad. Its utilization ratio is even lower than TCP, and the existing application are adversely affected. Considering that each Compound-TCP flow tries to maintain $\gamma = 30$ segments at the bottleneck, Compound-TCP is not scalable with flow number and its delay-based rules can not work in many cases. We had carried out the experiments in section 3.1.4 with 20 high speed flows. Figure 7 shows utilization ratio of wide area network and the response time of 64KB HTTP transaction. It indicates that until queue is larger than 0.5 BDP, Compound-TCP can not achieve high link utilization ratio and it is not better than loss-based high speed congestion control algorithms in the metric of HTTP transactions' response time. These results indicate that *a delay based high speed congestion control algorithm, which is scalable with flow number, should be a promising solution for high speed wide area network of the Internet.*

4.2 Local Area Network Scenario

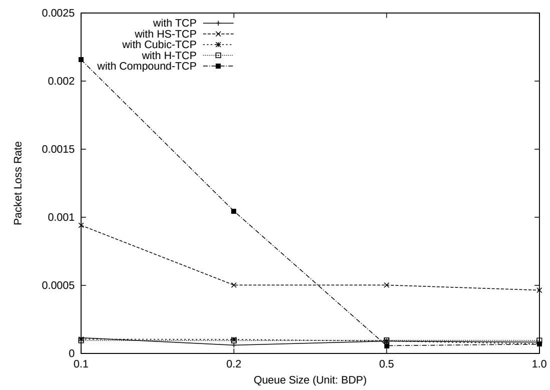
4.2.1 Simulation Results and Analysis

Figure 8 shows utilization ratio and packet loss rate of the simulated high speed campus network, and figure 9 shows the experience of WWW, VoIP, and FTP users that communicate with remote hosts of the Internet when 4 concurrent flows use HS-TCP, H-TCP, Cubic-TCP, Compound-TCP, or TCP.

Figure 8(a) shows that TCP can also achieve quite high



(a) Network Utilization Ratio



(b) Packet Loss Rate

Figure 8: Utilization Ratio and Packet Loss Rate of the Simulated High Speed Campus Network

network utilization ratio. The reason is that BDP of the simulated high speed campus network is not very large (1250KB) and RTT is short. It also shows that HS-TCP can not improve network utilization ratio and Compound-TCP is even worse than TCP when congestion can not be detected through queue delay. The reason is that the window growth functions of HS-TCP and Compound-TCP are related with RTT. They congest local area network too frequently and cause high packet loss rate (as shown in 8(b)). Their sending window is reduced frequently and their throughput is low. Compound-TCP are more liable to this problem since its decrease parameter is 0.5 when segment loss is detected. As for H-TCP and Cubic-TCP, their window increase functions are related with wall clock time. Their congestion frequency is effectively controlled and they achieve higher throughput than TCP. Cubic-TCP is better than H-TCP because of its concave window growth function. Its segment loss is less synchronized than H-TCP and the network can keep working around the cliff.

Figure 9(a) and 9(b) only present average response time of HTTP transactions whose response data size is 4KB or 64KB. Figure 9(c) and 9(d) show the average one-way delay and jitter experienced by VoIP packets. Compared with the propagation delay of leased OC-1 line, queue delay of local area network is very short. Response time of HTTP transactions and one way delay experienced by VoIP packets are not obviously increased by high speed congestion control algorithms. In some cases, they are even better than TCP.

When queue is less than 0.5 BDP, Compound-TCP can not detect congestion through queue delay and its binomial window growth function is a convex function. During each congestion, many segments are dropped. These frequent and large burst of segment loss cause the largest jitter experienced by VoIP users (as shown in figure 9(d)). And the throughput of FTP connections is also obviously reduced by Compound-TCP (as shown in figure 9(e) and 9(f)).

4.2.2 Discussion

According to previous section 4.2.1, *Cubic-TCP should be a good selection for high speed local area network*. Its wall clock time related window growth function can effectively reduce the frequency of congestion event. The concavity of its window growth function can reduce the burst of segment loss, drive the network to work around the cliff, and well utilize the bandwidth of high speed local area network. Since queue delay of local area network's router is normally quite short, their effects on the existing interactive applications can be imperceptible to users.

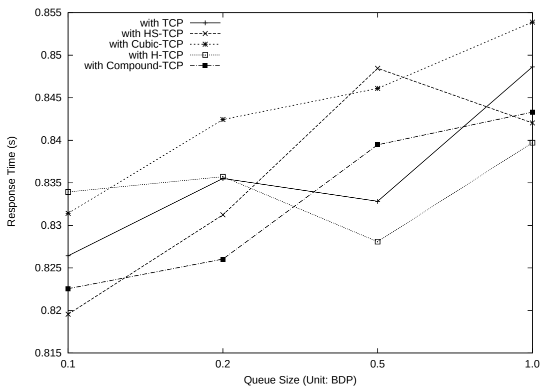
5. RELATED WORK

When high speed congestion control algorithms [26][17][23][24][12] are proposed , the authors also evaluate and compare their proposal with the existing algorithms. In [4], HS-TCP is evaluated on a simulated transocean optical fibre link, and Cubic-TCP is evaluated and analyzed on a testbed in [16]. In [19], BIC-TCP, Fast-TCP, HS-TCP, and H-TCP are systematically evaluated and compared on a testbed.

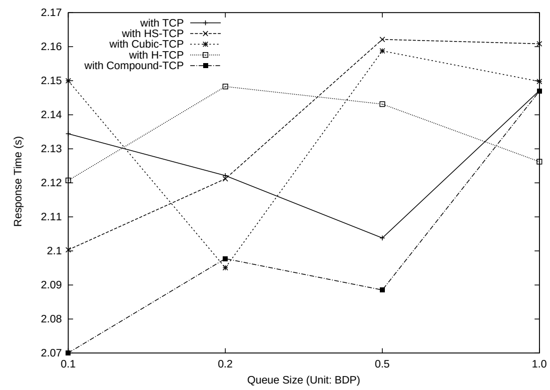
But these evaluations focus on the metrics of throughput, convergence speed, and fairness of flows driven by high speed congestion control algorithms. Although TCP friendliness had also been considered, the throughput of a concurrent long-lived FTP flow is normally the only metric. In [26], the authors suggested that it may be more realistic to evaluate the throughput of a long-lived FTP flow with buffer limitation and/or without support of window scale option. In addition, these high speed congestion control algorithms are normally evaluated on network pipes without large amount of background web traffics, which do exist in the Internet.

In [18], Compound-TCP is evaluated and compared with HS-TCP on several Internet paths (RTT: 8.5-151ms) that have large amount of background web traffics. In [7], BIC-TCP, Fast-TCP, HS-TCP, H-TCP, and Cubic-TCP are evaluated and compared on a emulated bottleneck link (bandwidth: 400Mbps, delay: 8-162ms) with different kinds of background traffics. But the authors mainly evaluate the effects of background traffics on flows driven by high speed congestion control algorithms.

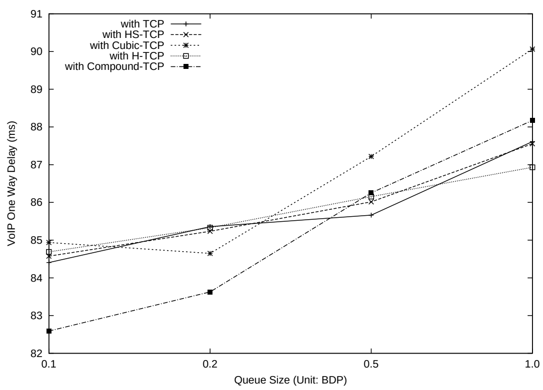
To the best of our knowledge, this paper is the first to study user experience of existing applications, especially the interactive applications, when high speed congestion control



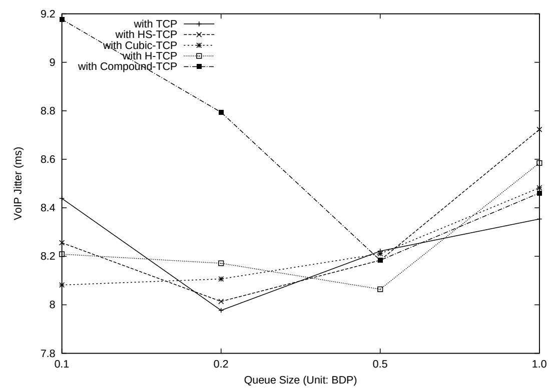
(a) WWW (Response Data Size=4KB)



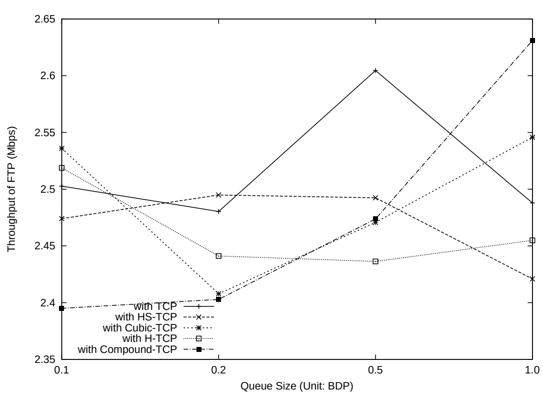
(b) WWW (Response Data Size=64KB)



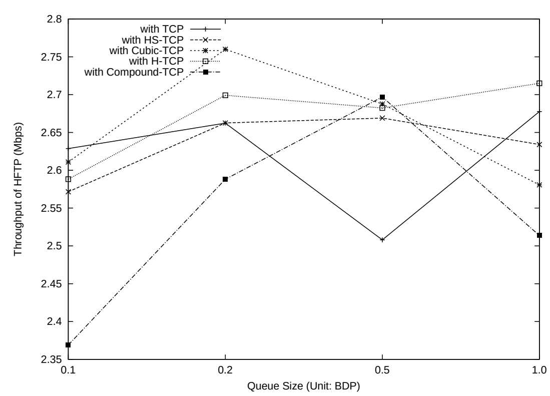
(c) VoIP (One Way Delay)



(d) VoIP (Jitter)



(e) FTP



(f) HFTP

Figure 9: WWW, VoIP, and FTP User Experience in Local Area Network

algorithms are used in high speed wide area network and local area network of the Internet

6. CONCLUSION

In this paper, several influential high speed congestion control algorithms, HS-TCP, H-TCP, Cubic-TCP, and delay-based Compound-TCP, are evaluated in simulated high speed wide area network and local area network of the Internet with the focus on their effects to the existing applications, especially the interactive WWW and streaming applications.

Through this study, we find that it is safe and fruitful to apply Cubic-TCP within high speed local area network. It can improve network utilization ratio and the experience of users, that communicate with remote hosts of the Internet through the existing applications, are not adversely affected.

As for high speed wide area network of the Internet, we find that with large amount of bursty web traffic, no matter the kind of congestion control algorithm, routers should be provisioned with a large queue. With this prerequisite, all high speed congestion control algorithms are not satisfying for wide area network. Loss-based algorithms inevitably increase queue delay experienced by the existing interactive applications, H-TCP also causes much higher packet loss rate due to its convex window growth function, and these QoS degradation can be perceived by users. Delay-based Compound-TCP can well utilize the network without adversely affecting the existing applications only when congestion can be detected through queue delay. Considering that the queue delay detection method of Compound-TCP is not scalable with flow number, a new delay-based high speed congestion control algorithm, which is scalable with flow number, should be a promising solution for high speed wide area network of the Internet.

Simulation results of this study also indicate that end hosts should be a little more intelligent and use different high speed congestion control algorithms on different network pipes of the heterogeneous Internet.

7. REFERENCES

[1] Ns-2 network simulator(ver.2). http://www.isi.edu/nsnam/ns/.

[2] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson. Tcp vegas: New techniques for congestion detection and avoidance. In *SIGCOMM*, 1994.

[3] J. Cao, W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M. Weigle. Stochastic models for generating synthetic http source traffic. In *IEEE INFOCOM*, 2004.

[4] E. de Souza. A simulation-based study of highspeed tcp and its deployment. Technical report, UC, Berkeley, May 2003.

[5] S. Floyd. Highspeed tcp for large congestion windows. RFC 3649, Dec. 2003.

[6] S. Floyd. Limited slow-start for tcp with large congestion windows. RFC 3742, Mar. 2004.

[7] S. Ha, L. Le, I. Rhee, and L. Xu. Impact of background traffic on performance of high-speed tcp variant protocols. *Computer Networks*, 51:1748–1762, 2007.

[8] V. Jacobson. Congestion avoidance and control. In *ACM SIGCOMM*, 1988.

[9] V. Jacobson, R. Braden, and D. Borman. Tcp extensions for high performance. RFC 1323, May 1992.

[10] R. Jain. A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks. *ACM SIGCOMM CCR*, 19:56–71, Oct. 1989.

[11] C. Jin, D. X. Wei, and S. H. Low. Fast tcp: motivation, architecture, algorithms, performance. In *INFOCOM*, 2004.

[12] K. Kaneko, T. Fujikama, Z. Su, and J. Katto. Tcp-fusion: A hybrid congestion control algorithm for high-speed networks. In *PFLDnet Workshop*, 2007.

[13] D. Katabi, M. Handley, and C. Rohrs. Internet congestion control for future high bandwidth-delay product environments. In *ACM SIGCOMM*, 2002.

[14] T. Kelly. Scalable tcp: improving performance in highspeed wide area networks. *ACM SIGCOMM CCR*, 33:83–91, 2003.

[15] R. King, R. Baraniuk, and R. Riedi. Tcp-africa: An adaptive and fair rapid increase rule for scalable tcp. In *INFOCOM*, 2005.

[16] D. Leith, R.N.Shorten, and G.McCullagh. Experimental evaluation of cubic-tcp. In *PFLDnet Workshop*, 2007.

[17] D. Leith, R. Shorten, and Y. Lee. H-tcp: A framework for congestion control in high-speed and long-distance networks. In *PFLDnet Workshop*, 2005.

[18] Y.-T. Li. Evaluation of tcp congestion control algorithms on the windows vista platform. Technical report, Stanford University, June 2006.

[19] Y.-T. Li, D. Leigh, and R. N. Shorten. Experimental evaluation of tcp protocols for high-speed networks. *IEEE/ACM TON*, 15, Oct. 2007.

[20] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. Tcp selective acknowledgment options. RFC 2018, 1996.

[21] A. Odlyzko. Data networks are lightly utilized, and will stay that way. *Review of Network Economics*, 2(3):210–237, Sept. 2003.

[22] J. Postel. Transmission control protocol - darpa internet program protocol specification. RFC 793, Sept. 1981.

[23] I. Rhee and L. Xu. Cubic: A new tcp-friendly high-speed tcp variant. In *PFLDnet Workshop*, 2005.

[24] K. Tan, J. Song, Q. Zhang, and M. Sridharan. A compound tcp approach for high-speed and long distance networks. In *INFOCOM*, 2006.

[25] D. X. Wei and P. Cao. Ns-2 tcp-linux: An ns-2 tcp implementation with congestion control algorithms from linux. In *ACM ValueTools - Workshop of NS-2*, 2006.

[26] L. Xu, K. Harfoush, and I. Rhee. Binary increase congestion control for fast long distance networks. In *INFOCOM*, 2004.