TCP sending rate control at Tera bits per second

E. Rodríguez-Colina, L. B. James,
R .V. Penty, I. H. White
University of Cambridge, Photonic Communications
Systems, 9 JJ Thompson Av. Cambridge, CB30FA, U. K.
er254@cam.ac.uk

K. A. Williams*
* Technische Universiteit Eindhoven, Den Dolech 2, 5600
MB Eindhoven, Netherlands

A. W. Moore $^{+}$,
$^{+}$ Queen Mary, University of London, Dept of Computer
Science, Mile End Road, E1 4NS, U.K.

Abstract — **An analysis of the sending rate of TCP control over Terabit per second rate links illustrates how future optical network characteristics, such as higher bitrates, network congestion, and larger data loads, would affect performance. We have implemented a model to allow increased sending rate for TCP. It is shown that even if the network bitrate is higher and the sending rate of TCP is scaled up, the throughput does not grow considerably, and latency remains one of the key parameters which must be reduced to improve performance.**

Keywords-component; TCP/IP performance over high bandwidth links, window scale option, latency reduction.

I. Introduction

The rapid increase in data traffic volumes and the bandwidth that can be provisioned by wavelength multiplexed optical channels present both challenges and opportunities in the data networking arena. The Transport Control Protocol (TCP) is one of the most commonly used protocols for data communications [1], but the extension of TCP to Terabit per second aggregate data rates has not been extensively studied. This work investigates the operation of future high capacity data links to explore the modifications which may be appropriate to ensure the robust operation of current protocols and infrastructure.

In this paper, it has been studied for the first time the effects of the window scaling up to 1.07GB at Tb/s line rates by comparing performance with and without the implementation of the scaling window. In the process, packet loss is analyzed under the window scaling implementation at Tb/s line rates.

II. Background

When we are talking about a reliable connection such as TCP, the performance of the transmission rate depends on how many segments of data have been received and acknowledged. Every time the sender receives an acknowledgement, the TCP sending rate grows. Thus the sending rate of each TCP transmission varies according to the number of acknowledgments received. In the case of packet loss or congestion, the transmission rate is decreased by the sender. These variations take the form of an additive increase, and multiplicative decrease (AIMD) scheme [2].

The conventional growth of the sending rate or scaling of the "window" utilizes an algorithm known as "slow start" [3] which provides exponential increases up to 65KB, which is known as the slow start threshold. After this threshold the scaling becomes linear, a process known as "congestion avoidance" [3].

The slow start operates by observing that the rate at which new packets should be injected into the network is the rate at which the acknowledgments are returned by the receiver. The congestion window is the control flow imposed by the sender and the advertised window is the flow control imposed by the receiver.

It is possible within TCP to scale the slow start threshold up to 1.07GB using the "window scale option" [4]. The window scale extension expands the definition of the TCP window to 32 bits and then uses a scale factor to carry this 32-bit value in the 16-bit Window field of the TCP header. The scale factor is carried in an option field of the TCP header. Two approaches to window scaling are feasible. This first approach sends the scaling information only in a synchronization segment and the window scale is therefore fixed in each direction when the connection is opened. A second approach allows the specification of the window scale in every TCP segment. Fixing the scale when the connection is opened has the advantage of lower overhead but the disadvantage that the scale factor cannot be changed during the connection. [4]

As data rates approach and exceed Tb/s however, the feedback closed loop requirements of TCP are expected to incur a severe burden in terms of latency. In this work, it was studied how window scaling may be implemented to enhance network performance at line rates of up to 10 Tb/s, addressing the dependence on line rate. The goodput, the amount of data successfully transmitted, is quantified along with transmission time for the example of 8 MB packet transmission. The improvement in link efficiency facilitated through window scaling is characterized through the implementation of an adapted SSFNet simulation in terms of maximum window for the sending and receiving rates with exponential growth.

III. Implementation

The role of window scaling was explored by implementing a customized option for the SSFNet simulator [6]. This option was not otherwise available. While the "window field" of the

E. Rodriguez-Colina is member of St Edmund's College and is sponsored by the Council of Science and Technology of Mexico (CONACYT)

TCP header was not directly modified, the sending rate algorithm was adjusted to allow it to grow to the maximum value that is a window of 1,073,725,440 bytes (65535 x 2^{14}). The 65535 bytes is the previous slow start threshold and is the conventional value for all versions of TCP; the 2^{14} number comes from the options section of the TCP header which is the window scale factor and is 3 bytes long, the last byte being the shift count and is set to 0 when scale option is applied.

To facilitate this modification, the sending rate and the receiving rate must also be able to facilitate the scaling and therefore the receiver buffer was modified to allow as many packets as the line physical rate permitted. This avoids reception side restrictions and to allow the unambiguous interpretation of simulations in terms of the sending rate behavior.

The fact that we modify directly the behavior of the growth is actually different from the original idea of the window scale option only in the way the start of the communication is executed. The scaling is not advertised between the receiver and server during the synchronization of the communication. The scaling is allowed for all the transmissions by modifying the congestion window threshold and the advertised window.

Even if the implementations are different in the initial period of the communication (TCP synchronization); the scaling of the window is the same for the window scale option and our scaling up implementation during the slow start growth.

IV. RESULTS

Five different line rates were simulated with a client server link of 1 microsecond delay. The one and ten Tera bits per second line rates simulation cases, show and improvement in the performance with the scaling up window implementation. See Table 1. However the improvement could be better, except for other important factors which are contributors to the total performance. Thus it should be considered the fact that every amount of data to be transfer is segmented by TCP. This segmentation is directly related to the round trip time and the acknowledgments.

It was found through simulations that the performance of TCP for high data rates is dominated by the latency and the rate at which transmissions may be sent. This sending rate varies with the number of acknowledgements received and their incidence. An important consideration is the wait to receive and acknowledged for a sent segment which is dependent on the physical line delay.

While it is true that the latency imposed by the physical layer is small the final performance will be affected by the feedback closed loop generated between the transmission of packets and their acknowledgments. This loop is observed every time that a TCP connection is established, thus the time that it takes for an entire file to be transmitted would be nearly a constant number dependent on the physical line delay when the bit rate is equal to or higher than 1 Tb/s and there are not retransmissions due to packet loss; see Table 1. In this case the limit is the number of packets (segments of the file) that can be sent per unit of time and are controlled by the congestion window of TCP and the round trip time of the feedback loop created by the flow between the packets and their acknowledgments.

TABLE I. PERFORMANCE WITH AND WITHOUT SCALING UP THE SENDING RATE IMPLEMENTATION

			Scaling window implemented		
Line rate (bits/sec)	Goodput (Gbits/sec)	Transmission time (µsec)	Goodput (Gbits/sec)	Transmission time (µsec)	
1 Giga	0.97	8221.07	0.97	8221.07	Line rate limited
10 Giga	9.66	828.31	9.66	828.31	
100 Giga	86.06	92.96	86.06	92.96	Latency limited
1 Tera	198.27	40.35	279.31	28.64	
10 Tera	199.83	40.03	285.06	28.06	
Data transfer 8 Mbits per transmission					
Physical line delay 1 µsec					

The calculation of the time required to transmit the entire file depends on factors such as: segmentation, the physical line delay and network congestion, plus round trip time and the adaptive algorithms of the TCP window.

Table 1 shows the results of a transmission with 8 Megabits file size using TCP; the first column to the left hand side shows the bit rates tested, the second and third columns show the results of the simulation without the scaling up implementation and the fourth and fifth columns show the results with the implementation of the scaling window.

In the table we refer to "goodput" as the measurement of the data rate successfully transmitted; the aggregated throughput minus the overhead and retransmissions.

As can be seen, the transmission time becomes a constant at approximately 40usec; this is because the physical delay of the link is determining the connection performance. When the simulations run with the window scaling factor increased 1.07GB, the transmission time stabilizes at approximately 28usec for both speed cases of 1 Tb/s and 10 Tb/s. This time cannot be reduced significantly because it is the product of the physical delay multiplied by the number of packets transmitted (equal to the number of segments of the full amount of data to be transferred).

A comparison of the goodput values in Table 1 reveals that the scaled up window implementation increases the performance by approximately 40%.

In Fig. 1 it can be seen the behavior of the congestion window size for the scaling up window and for the previous growth of the congestion window, without the scaling up implementation.

As is also seen, both with and without the scaling up implementation increase approximately exponentially up to 1.5usec although in contrast with the previous behavior it is clear that the exponential increase persist for the scaling up option because now it can increase exponentially to reach the new limit causing the time to transmit the total amount of data to be reduced from 40.35usec to 28.64usec.

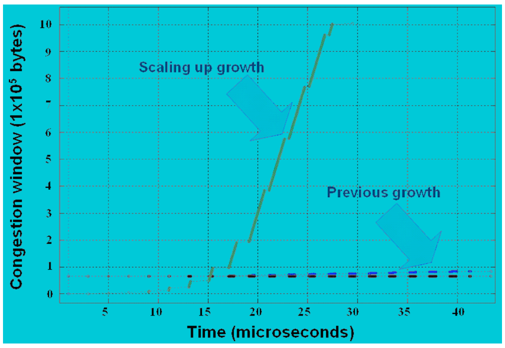

Figure 1. Congestion window growth for the scaling up and withouth scaling up implementation

With the previous limit of 65KB (red-dashed line) in Fig. 1, the growth of the window change from exponential to linear (blue-dashed line) that is known as "congestion avoidance". The congestion avoidance is activated in the presence of packet loss or congestion of the network as well, not only by the 65KB or 1.07GB thresholds.

However the improvement may be conditioned to a good quality link because in the case of packet loss the goodput performance of the link may be affected in practice by the retransmissions. These retransmissions are the natural control of TCP to maintain reliable communication in case of packet loss or congestion.

It was decided to test the new implementation with several packets passing through the link and simulated packet losses with the use of a random number generator with an exponential distribution. This function allows us to induce a packet loss every certain number of packets transmitted.

Performance is compared for a range of packet error rates in Fig. 2 both with and without the scaling modification, for different packet error rates. It was also investigated for bit rate links of 1 Tb/s and 10 Tb/s.

The simulations show that; the goodput has not been compromised by a packet error rate that is better than ten to the power of minus four (1E-04); 100 files were tested transferring each 8MB of size where some packet losses were generated over the link.

The 8MB file transferred is expected to be more susceptible to packet loss because of its considerable size but TCP gives stable performance although the final sending rate is reduced by the retransmissions generated.

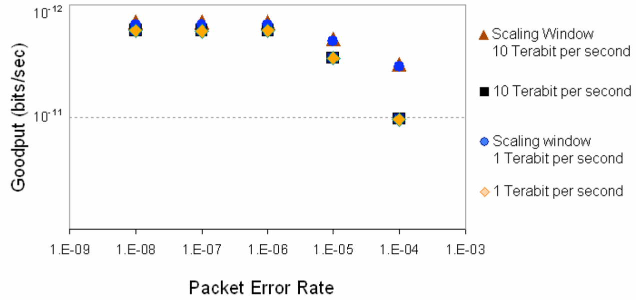

Figure 2. Goodput vs. packet error rate for the scaling window implementation

V. Conclusions

It can be concluded from the results that scaling the TCP window improves the performance of the communication link and is robust in the face of packet loss for the types of transmission and link rates which we have considered.

TCP was simulated with the scaling up sending rate with the slow start threshold set to 1.07GB which has not been deeply studied at Tb/s before. The simulation results show some of the limitations of TCP operating at Tera bits per second rates.

It was found that a TCP transfer at 1Tb/s takes approximately the same time as 10 Tb/s; because of the segmentation and the latency receiving the acknowledgments. An overview of how the data segments and their acknowledgments can determine the performance of the network is presented in this paper.

TCP has been working well, covering the requirements of the data communications although some modifications to the control algorithms of the window must be reconsidered.

VI. Future Work

From our point of view latency remains one of the key parameters which must be reduced to improve the performance.

Acknowledgments

We would like to thank Madeleine Glick from Intel Research Cambridge, UK, and Pablo Vidales from Deutsche Telekom Laboratories, TU Berlin for their help.

References

[1] C. Cameron, H. Le Vu, J. Choi, S. Bilgrami, M. Zukerman and M. Kang "TCP over OBS -fixed-point load and loss" Optics Express, OSA, Vol. 13, No. 23/9172, 2005

[2] Cheng Jin, David Wei, at al, "Fast TCP From theory to Experiments", IEEE Network January/February, 2005

[3] A. Detti, M. Listanti, "Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks", INFOCOM 2002

[4] V. Jacobson, R. Braden, D. Borman, "TCP Extensions for High Performance", Network Working Group, RFC 1323

[5] W. Richard Stevens, "TCP/IP Illustrated, Volume 1: The Protocols", ISBN 0-201-63346-9, Addison-Wesley

[6] "SSFNet simulator", http://www.ssfnet.org/homePage.html