Modeling Scalable TCP friendliness to NewReno TCP

Mohamed Tekala and Robert Szabo

Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics,
H-1117, Budapest, Magyar Tudosok krt. 2

Summary
Scalable TCP is a simple change to the traditional TCP congestion control algorithm which dramatically improves TCP performance in high speed wide area networks. The focus of this contribution is to model the performance of Scalable TCP algorithm sharing bottleneck link with a NewReno TCP. The analysis is performed during congestion avoidance phase, for both TCP flows, in steady state and near to the steady state. This model follows the existing model of NewReno TCP and is further expanded to predict the performance of congestion avoidance phase of the two TCP variants competing on a bottleneck link. The model is used in situations where congestion occurs due to buffer overflow (synchronous loss) and random loss (asynchronous loss). The applied model produced excellent information on how bottleneck link distributed over different TCP flavors. It predicts the Scalable TCP friendliness to NewReno TCP.
Simulation measurements were carried out in order to verify the model accuracy. Long lived TCP flows traveling along high speed-wide area networks were used. The results of the applied model and the performed simulation measurements were in good agreement.

Keywords:
NewReno TCP, Scalable TCP, TCP friendliness, Mathematical analysis, Numerical results.

1. Introduction

The deployment of high-speed links for high-delay communication has posed a serious challenge for the AIMD algorithm used for congestion control in TCP. Several researchers have worked to improve the performance of TCP in high speed wide area networks. Some researchers modified the congestion response function of TCP itself. High-Speed TCP [1], Scalable TCP [2], Fast TCP [3] and BIC TCP [4] are some of the examples in this field of research.
A number of recent analytical models characterize the TCP performance as a function of round trip time (RTT) and packet loss rate. These models have provided a better understanding of the sensitivity of TCP performance to network parameters [5], [6] and [7].

The aim of this work is to investigate the effect of Scalable TCP on the NewReno TCP flows when they share a bottleneck link. In order to understand the relationship between throughput of Scalable TCP and NewReno TCP (in case of packet drop due to buffer overflow and/or random link failure), we tend to conduct a mathematical analysis that predict the performance of the two TCP algorithms sharing bottleneck link. The DropTail queuing system is used to manage the shared buffer (the buffer size is set to the bandwidth-delay product). The proposed model divided the congestion avoidance phase into a set of cycles, where a cycle is defined by the duration between two consecutive packet losses; hence, the model can estimate the performance of the TCP during a cycle.
Simple approximation derivation of the mode is presented in this communication. This derivation was done under the assumptions that the congestion signal is subject to the buffer overflow (synchronous loss) and random link failure (asynchronous loss). The analysis presents expressions for the throughput of the two TCP versions that share the bottleneck link. The throughput equations are used to estimate the percentage of each flow in the bottleneck link.
In order to evaluate our model, we examined the performance of the TCP congestion avoidance in network simulator [8] for the congestion signals due to buffer overflow and/or link failure. The numerical results are in a good agreement with our proposed model.

2. Related works

Mahdavi et al. presented a mathematical analysis for the performance of NewReno TCP; the model predicts the bandwidth occupied by TCP connection [9]. The performance of High-Speed TCP, and the impact of its use on the present implementation of NewReno TCP on different network conditions was studied by Agarwal and coworkers [10]. A comparison between the performance of TCP Vegas and NewReno TCP was presented by Mo et al. [11]. Several exiting analytical models of TCP variants are compared in an overview by Trajkovic et al. [12].
Scalable TCP [2], which is specifically designed for use in networks with a high bandwidth-delay product, updates the congestion window using fixed parameters.

Manuscript received March 5, 2007
Manuscript revised March 25, 2007

In our previous work [13], the performance of Scalable TCP under different network environments and its friendliness to NewReno TCP were numerically investigated. It was clear that Scalable TCP badly outpaces NewReno.
The previously mentioned results [13] showed the necessity to model the performance of Scalable TCP in presence of NewReno TCP in order to predict the fraction of bandwidth occupied by each TCP connection.

2.1 NewReno TCP

Many research efforts have been devoted to the network congestion against an increase of network traffic in the Internet. TCP's algorithm is referred to as additive increase multiplicative decrease (AIMD) and is the basis for steady state congestion control. TCP increases the congestion window by one packet per window of data acknowledged, and halves the window for every window of data containing a packet drop. The TCP congestion control can be roughly expressed with the following equations:

Slow start:

$$W = W + 1 \qquad Ack \tag{1}$$

Congestion avoidance:

$$W = \begin{cases} W + \frac{1}{W} & Ack \\ \frac{W}{2} & Drop \end{cases} \tag{2}$$

2.2 Scalable TCP

Scalable TCP is a simple sender-side alteration to the TCP congestion window update algorithm. It offers a robust mechanism to improve performance in high speed wide area networks using traditional TCP receivers. Scalable TCP utilizes fixed increase and decrease parameters for updating its congestion window.
The generalized Scalable TCP window update algorithm responds to each acknowledgment received in a round trip time in which congestion has not been detected with the update

$$W = W + \alpha \tag{3}$$

where α is a constant ($0<\alpha<1$).
In the event of congestion the congestion window is multiplicatively decreased as follows:

$$W = \beta \cdot W \tag{4}$$

Where, β is a constant with $0<\beta<1$.
A proposed setting for the constants (α and β) is: α=0.01 and β=0.875. Throughout the rest of this communication we adhere to this parameter setting. Further details of the Scalable TCP algorithm are available in [2].

3. Mathematical Model

The main purpose of modeling STCP friendliness is to predict the bandwidth distributed on the two TCP variants sharing bottleneck. In order to accomplish this, we must introduce a system model first and then investigate the performance of STCP sharing bandwidth with NewReno TCP, with and without random losses by making some gross simplification.

3.1 System model

Throughout our analysis, the model presented by Mahdavi et al. [9] was followed and applied to the described multiplicative increase-multiplicative decrease (MIMD) type of Scalable TCP. The assumptions of the system model are briefly summarized as follows:

- Data sources are infinite, there are always data to be send packets of the maximum segment size (MSS);
- There is a single bottleneck with a capacity of μ packets per second and buffer size B (in packets). The link uses FIFO queuing (Drop Tail) unless otherwise specified;
- Random loss is introduced in addition to buffer overflow;
- Two streams share the bottleneck link. The first one is the Scalable TCP (we note it by STCP), it is congestion window denoted by W_S. The second one is the NewReno TCP and it's congestion window denoted by W_R;
- Round trip time (RTT) is defined as the sum of two parts. First is the fixed (propagation) delay T which includes all fixed delays of processing, transmission, and propagation. A variable delay denoted $D(t)$ is also introduced to represent the random queuing delay (service time); the total RTT is thus $T + D(t)$, or $T_i + D(t)$ for the i^{th} connection when the distinction is necessary;
- Throughput η is the total amount of data sent through the network per unit time, where η_S and η_R are the throughput of STCP stream and NewReno TCP stream respectively.

3.2 Link failure loss

TCP treats the loss of packets as a signal of network congestion and reduces its window when this occurs.

Losses not due to congestion are mostly caused by transmission errors. If we assume that the probability for a packet to be lost is q, then on average 1/q packets are successful transmitted before a loss occurs. Under these assumptions the congestion window traverse a perfectly a periodic oscillation.

Scalable TCP:

Let the maximum value of the window when a loss occurs to be W_S packets. By the definition of congestion avoidance, it is known that during the congestion avoidance phase; the minimum window must be βW_S packets. If the receiver is acknowledging every packet, then the window opens by $\frac{\alpha(1+\beta)}{2}W_S$ packets per round trip time. So that, each cycle must take a period of $\tau_l = \frac{2(1-\beta)}{\alpha(1+\beta)} \cdot RTT$ seconds. The total number of packets delivered between losses is the area under the saw-tooth diagram (shown in Fig. 1), which is defined by the following Equation.

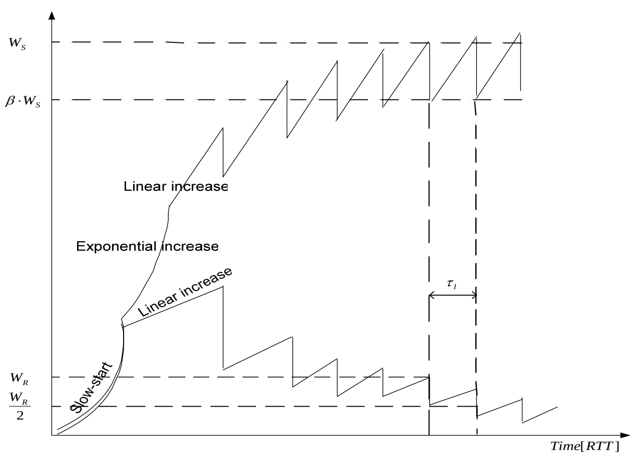

Fig. 1 W growth as a function of time

$$\eta_S = \beta W_S \frac{\tau_l}{RTT} + \frac{1}{2}(1-\beta)W_S \frac{\tau_l}{RTT} = \frac{1-\beta}{\alpha} W_S \quad (5)$$

By assumption, each cycle also delivers 1/q packets, and then the maximum congestion window of STCP when the loss occurs is

$$W_S = \frac{\alpha}{q(1-\beta)} \quad (6)$$

To estimate the amount of bandwidth saturated by the TCP flow using the bandwidth equation:

$$BW_S = \frac{MSS \cdot \eta_S}{\tau_l} = \frac{MSS(1-\beta)}{\alpha \tau_l} W_S \quad (7)$$

By substituting for W_S and simplifying we get:

$$BW_S = \frac{MSS \cdot \alpha (1+\beta)}{2q(1-\beta)RTT} \quad (8)$$

New Reno TCP:

Similarly to the treatment of Scalable TCP, assume the maximum value of NewReno congestion window to be W_R. Then by the definition of Congestion Avoidance we know that during equilibrium, the minimum window must be $\frac{W_R}{2}$ packets. If the receiver is acknowledging every packet and the window opens by one packet per round trip time, then each cycle must be $\tau_l = \frac{W_R}{2} \cdot RTT$ seconds. The total data delivered between losses is the area under the saw-tooth, which is:

$\eta_R = \left(\frac{W_R}{2}\right)^2 + \frac{1}{2}\left(\frac{W_R}{2}\right)^2 = \frac{3}{8}W_R^2$ packets per cycle.

Assuming that, the link delivers approximately 1/q packets followed by one drop. Hence, the maximum congestion window when loss occurs will be:

$$W_R = \sqrt{\frac{8}{3q}} \quad (9)$$

To estimate the bandwidth utilized by NewReno TCP flow we use the bandwidth equation:

$$BW_R = \frac{MSS \cdot \eta_R}{\tau_l} = \frac{MSS \frac{3}{8} \cdot 2W_R^2}{RTT \cdot W_R} \quad (10)$$

Substituting Eq. 9 into Eq. 10 and upon simplification:

$$BW_R = \frac{MSS}{RTT\sqrt{\frac{2q}{3}}} \quad (11)$$

Based on Equations 8 and 11, the model predicts the performance of the two TCP algorithms competing on bottleneck link during or around the steady state condition. Hence, the total link bandwidth occupied by the two flows is given by:

$$BW = \frac{MSS \cdot \alpha(1+\beta)}{2q(1-\beta)RTT} + \frac{MSS}{RTT\sqrt{\frac{2q}{3}}} \quad (12)$$

The fraction of the bandwidth on a bottleneck link for each TCP were obtained by using Equations (8) and (10) with substitution of the increase and decrease parameters

(α=0.01, β=0.875 and a set of value for the probability of packet drops q). Table 1 shows the fraction of bandwidth utilized by each flow competing on bottleneck link. It is clear that for high loss probability, NewReno TCP maintains higher bandwidth, in contrary lower bandwidth for STCP was observed due to high packet loss. On the other hand, STCP provides higher performance when probability of loss is very low $(q \le 1.5e-5)$. Also, there is sufficient number of asynchronous losses where the shared bandwidth can be fairly distributed between connections (see Table 1).

Table 1: Analytical STCP friendliness to NewReno TCP

TCP's	Partial utilization Probability of loss q=					
	1.5e-2	1.5e-3	1.5e-4	1.5e-5	1.5e-6	1.5e-7
New-Reno	0.7	0.42	0.19	0.06	0.02	0.007
STCP	0.3	0.57	0.81	0.93	0.97	0.99

3.3 Buffer overflow

In the DropTail queuing system the bottleneck link buffer fully occupied and there are μRTT packets in flight. The typical evolution is cyclical during the congestion avoidance phase. In this phase window-size grows slowly to a maximum value of W_l, at which point there is a packet loss. This type of loss is a synchronous loss. Connections on the network sharing bottleneck link suffer from this type of loss.

Suppose that, there are two flows competing on bottleneck link (STCP and NewReno TCP); according to their algorithm behavior, the total decrease of the window value when loss occurs at period k is $(1-\beta)W_S[k]+\frac{W_R[k]}{2}$.

The minimum values of NewReno and Scalable TCP congestion windows during period k are $\frac{W_R[k]}{2}$ and $\beta W_S[k]$, respectively.

The maximum value of congestion window where the loss occurs is $W_R[k+1]$ for NewReno TCP and $W_S[k+1]$ for STCP. In most of the cycles $W_R[k] > W_R[k+1]$ and $W_S[k] < W_s[k+1]$, this was repeated until the connections reach the steady state phase defined by:

$$k \approx \infty \Rightarrow \begin{cases} U_S \approx 1 \\ U_R \approx 0 \end{cases} \tag{15}$$

Where U_S and U_R are the fraction of the bandwidth utilized by STCP and NewReno TCP, respectively.

We assumed that, in the two TCP versions the receiver is acknowledging every packet and upon the increase parameter of the two connections, then the average of the total increase rates of the two congestion windows during period k is $\frac{\alpha(1+\beta)W_S[k]}{2}+1$ packets per round trip time.

Hence, the duration between two consecutive packets loss for TCP connections is defined by:

$$\tau_l[k] = \frac{(1-\beta)W_S[k]+\frac{W_R[k]}{2}}{\frac{\alpha(1+\beta)W_S[k]}{2}+1}\cdot RTT = \frac{2(1-\beta)W_S[k]+W_R[k]}{\alpha(1+\beta)W_S[k]+2}\cdot RTT \tag{16}$$

Scalable TCP:

Based on the previously mentioned description, the total packets delivered by Scalable TCP during period k are simply obtained by the area under the saw-tooth curve of the period (shown in Fig. 1), which is defined by:

$$\eta_S[k] = \beta W_S[k]\frac{\tau_l[k]}{RTT}+\frac{1}{2}(1-\beta)W_S[k]\frac{\tau_l[k]}{RTT} = \frac{(1+\beta)W_S[k]\tau_l[k]}{2RTT} \tag{17}$$

NewReno TCP:

The congestion window of NewReno TCP is affected by the synchronous loss, so that, the evaluation of the performance of NewReno TCP is effected by STCP flow where the increase-decrease parameters make sense. The total packets delivered are the area under the saw-tooth (Fig. 1), which is defined by.

$$\eta_R[k] = \frac{7W_R[k]\tau_l[k]}{8RTT}$$

(18)

The total throughputs of the two flows are given by the sum of the two TCP throughput Equations (17) and (18):

$$\eta = \left(\frac{(1+\beta)W_S[k]}{2}+\frac{7W_R[k]}{8}\right)\cdot\frac{\tau_l[k]}{RTT} \tag{19}$$

For two flows competing on bottleneck link, where the round trip time is the same and the probability of loss is very small (q=1.5e-7), where we can assume that the losses occur due to the buffer overflow and substituting the value of α=0.01, β=0.875. We can predict the fraction of the bandwidth used by Scalable TCP connection by dividing the throughput of STCP by the total throughput and substituting the values mentioned above yield:

$$U_S = \frac{\eta_S}{\eta} = 0.992 \quad (20)$$

Following the same instructions, we predict the fraction of the bandwidth used by NewReno TCP connection:

$$U_R = \frac{\eta_R}{\eta} = 0.007 \quad (21)$$

4. Numerical Results

During the process of numerical analysis, we applied a dumbbell network with a single bottleneck link, as shown in Fig. 2. All traffic of the different scenarios listed below pass through a 100 Mbps or 1 Gbps shared bottleneck link. The two routers of the topology used FIFO queuing and DropTail buffer management; their buffers were set to the bandwidth-delay product of the setup. The bandwidth between the hosts and the routers was 10 Gbps, and the link delay between the router and hosts was 1 ms. The packet size was set to 1,500 bytes; the maximum allowed window size was large enough (83,000 packets) to saturate the bottleneck link. On top of the TCP agents, we used FTP application to transmit large data. All FTP's started at time zero, and the simulation time was 100 sec.

We simulated a network with two connections with the same round-trip time, where one is a Scalable TCP connection and the other is a NewReno TCP connection, in an attempt to verify our model.

The different traffic mix scenarios are discussed in details below.

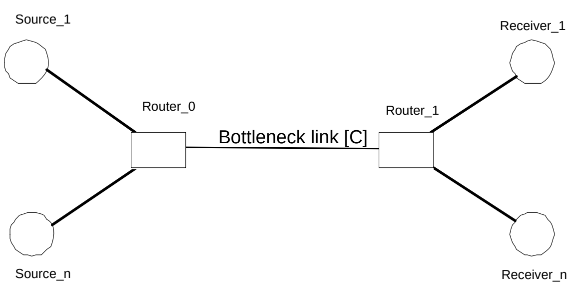

Fig.2. Network topology with a single bottleneck link

4.1 Lossy link condition

The focus of this set of experiments was to observe the behavior of the STCP and NewReno TCP flows when subjected to systemic losses. The simulator error model was used to simulate losses on bottleneck link. This loss model was set to drop a packet with a defined average drop rate. A set of simulations were applied with different RTT's, where both TCP connections have the same RTT during simulation experiment. These investigations were done for a high-speed environment (μ=1Gbps) and low-speed environment (μ=100Mbps) with different probability of packet drops.

Simulation results demonstrate that random losses effects on the performance of both TCP connections.

In some cases Scalable TCP out-placed NewReno TCP under the affection of low packet loss. A congestion window plot and the corresponding throughput for 1Gbps and 10ms round trip time when the probability of packet loss (q=1.5e-4) are shown on Figs. 3 and 4, respectively.

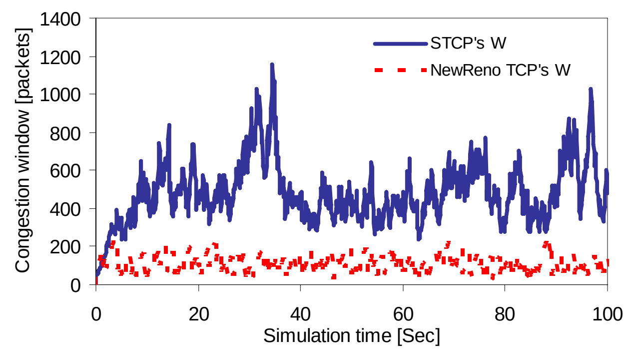

Figure 3: W vs. time under random loss (μ=1Gbps, T=10msec, q=1.5e-4)

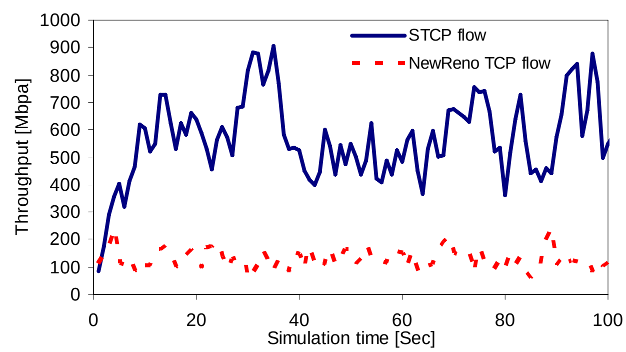

Figure 4: Throughput under random loss (μ=1Gbps, T=10ms, q=1.5e-4)

Fig. 5 presents the utilization achieved by the two TCP mixed flows competing on bottleneck link. One line is the fraction of the measured utilization of NewReno TCP; this is compared by the analytical result presented on the same figure. The other line is the fraction of the measured utilization of STCP flow and the fourth line is the analytical fraction of the utilization of STCP flow obtained by our model (as can bee seen on the legend of Fig. 5).

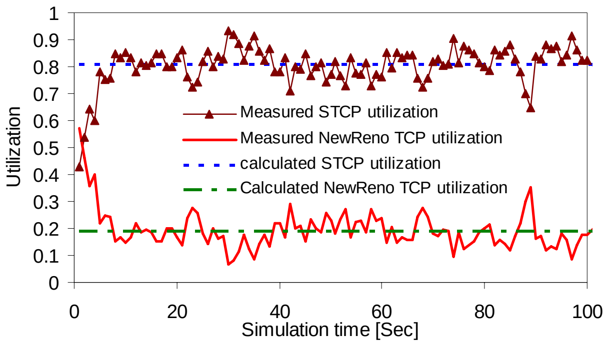

Figure 5: Utilization vs. time (μ=1Gbps, T=10ms, q=1.5e-4)

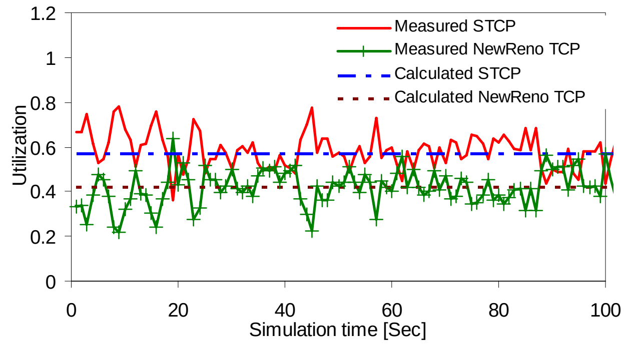

Figure 5: Utilization vs. time (μ=100Mbps, T=10ms, q=1.5e-3)

On the other hand, there are cases where the bandwidth is better shared amongst the TCP versions. This is because there are sufficient numbers of asynchronous losses to fairly distribute the bandwidth over the shared connections. A 100 Mbps and 10ms round trip time under a high asynchronous loss (q=1.5e-3) scenario was tested. Figure 6 shows the congestion windows as a function of time for both TCP flows, it reveals that STCP friendlier to NewReno TCP flow. The corresponding throughput and bandwidth utilization are shown on Figures 7 and 8, respectively. It is clear that, by controlling the selected loss ratio fairness could be achieved.

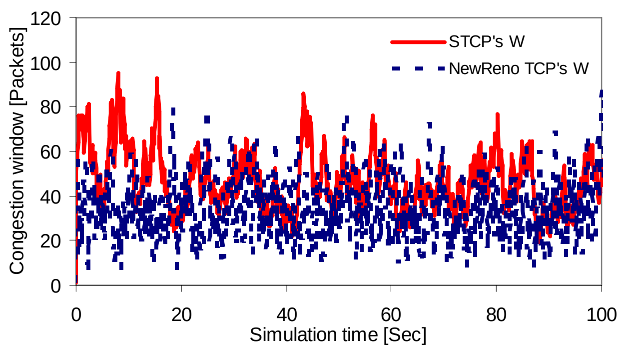

Figure 6: W vs. time (μ=100Mbps, T=10msec, q=1.5e-3)

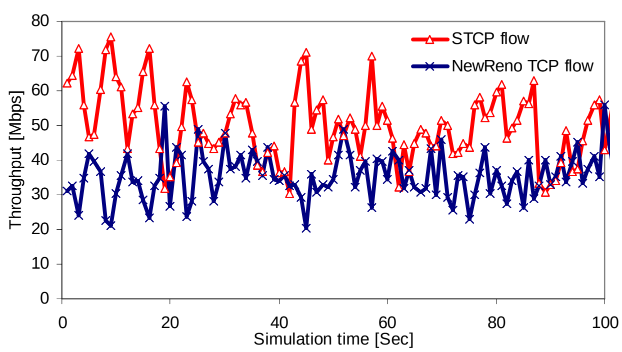

Figure 7: Throughput (μ=100Mbps, T=10ms, q=1.5e-3)

Once again Figures 5 and 8 demonstrate that the model is accurate to predict the performance of the two TCP versions sharing bottleneck link.

To further quantify our proposed model, Table 2 shows a comparison of partial link utilization between Scalable TCP and NewReno TCP flows. It is clear that the model is fairly accurate to predict the performance of two TCP versions sharing bottleneck link (compare numerical results with Table 1).

Table.2 STCP friendliness to NewReno (μ=1Gbps, RTT=100ms)

TCP's	Partial utilization Probability of loss q=					
	1.5e-2	1.5e-3	1.5e-4	1.5e-5	1.5e-6	1.5e-7
New-Reno	0.79	0.37	0.18	0.08	0.02	0.01
STCP	0.2	0.63	0.82	0.92	0.97	0.99

4.2 Synchronous loss

These set of experiments make it possible to observe the behavior of the STCP and NewReno TCP flows when subjected to buffer overflow losses (synchronous losses). A set of simulations with different RTTs, where both TCP connections have the same RTT during simulation experiment, were used. DropTail queuing system was applied to control the bottleneck link buffer, so that the loss, occurred when the bottleneck link was fully utilized and the buffer size is full. We investigated a high-speed environment (1 Gbps).

Simulation results demonstrate that Scalable TCP out-placed NewReno TCP, a congestion window plot and the corresponding throughput for bottleneck link of 1 Gbps and 100 ms round trip time are shown on Figs. 6 and 7, respectively.

Fig. 8 presents the utilization achieved by the two TCP flows sharing the bottleneck link. It can be seen that our proposed model gives good prediction to the STCP friendliness with NewReno TCP.

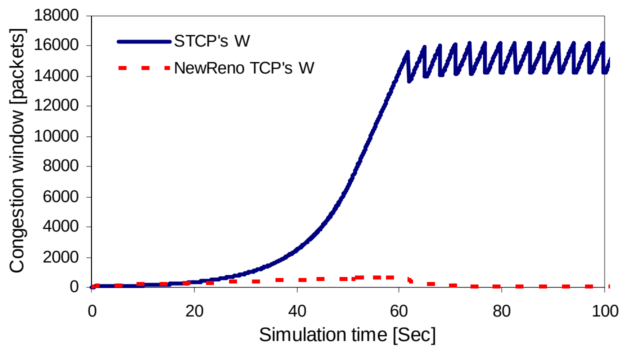

Figure 6: W vs. t under synchronous loss (μ=1 Gbps, T=100 ms)

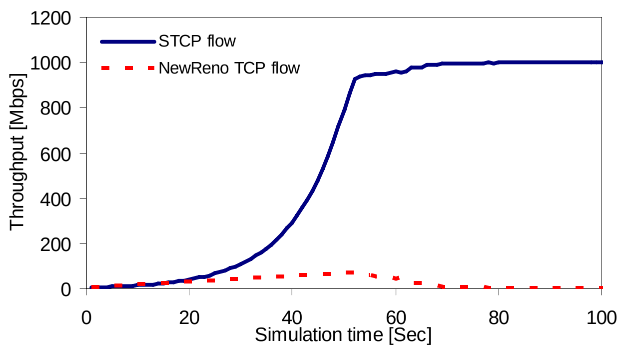

Figure 7: Throughput under synchronous loss (μ=1Gbps, T=100 ms)

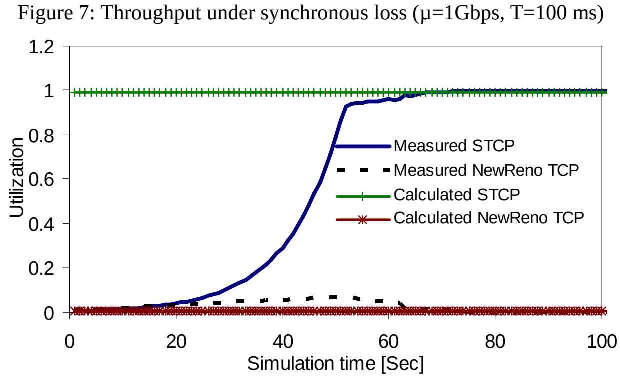

Figure 8: utilization vs. t under synchronous loss (μ=1 Gbps, T=100 ms)

Table 2 illustrates the behavior of Scalable TCP friendliness to NewReno TCP when bottleneck link was 1 Gbps, with different round trip-times under synchronous loss affection.

Table.2 STCP friendliness to New Reno under synchronous loss (μ=1 Gbps)

TCP's	Utilization				
	10ms	30ms	60ms	100ms	200ms
NewReno	0.005	0.005	0.003	0.002	0.005
STCP	0.99	0.99	0.99	0.99	0.98

5. Conclusions

In this contribution, the performance of Scalable TCP and NewReno TCP sharing a bottleneck link have been analyzed. Steady state in a congestion avoidance phase was applied in this proposed model. Both mathematical analysis and simulations show that the STCP was the dominant protocol on the bandwidth when sharing bottleneck link with NewReno TCP. Our proposed model predicts the performance of the two TCP algorithms when the loss occurs due to buffer overflow and random loss (link failure). The results show that, the Scalable TCP performs better than NewReno TCP at high speed long delay connections. The major drawback in NewReno TCP is that the congestion window decreased to a half of its value in response to congestion event and its congestion window grows in increments of one for each RTT cycle. This leads to slow recovery from a congestion event when the congestion window was very large. In contrary, Scalable TCP functions with less decrease and faster increase in congestion window, which means less time for recovery process and as a consequence increase in average utilization. In presence of systemic losses, Scalable TCP works more friendly with New-Reno TCP.

Acknowledgements

This work is partly supported by the Inter-University Center for Telecommunications and Informatics (ETIK), Hungary.

References

[1] Sally Floyd, Sylvia Ratnassamy, and Scott Shenker, Modifying TCP's Congestion Control for High Speeds May 5, 2002.
[2] Tom Kelly. Scalable TCP:Improving Performance in Highspeed Wide Area Networks, submitted for publication, 2002, available at: http://www.lce.eng.cam.ac.uk/~ctk21/scalable/.
[3] C. Jin, D. X. Wei, and Steven H. Low, FAST TCP: motivation, architecture, algorithms, performance, IEEE INFOCOM, 2004.

[4] L. Xu, K. Harfoush and I. Rhee. Binary Increase Congestion Control (BIC) for Fast Long-Distance Network. In proceedings of the IEEE INFOCOM, March 2004.
[5] M. Goyal, R. Guerin, and R. Rajan. Predicting TCP throughput from non-invasive network sampling. INFOCOM 2002, New York, June 2002.
[6] V. Misra, W. Gong, and D. Towsley. Stochastic differential equation modeling and analysis of TCP-Windowize behavior. In PERFORMANCE'99, Istanbull, Turkey, October 1999.
[7] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation, In SIGCOMM 98, 302-314, Vancouver, Canada, September 1998
[8] The network simulator ns-2, http://www.isi.edu/nsnam/ns/.
[9] M. Mathis, J. Semke, and J. Mahdavi.. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. Computer Communication, 27(3), July 1997.
[10] Evandro De Souza and Deb Agarwal. A High-Speed TCP Study: Characteristics and Deployment Issues, 2003.
[11] Jeonghoon Mo, Richard J. La, Venkat Anantharam, and Jean Walrand. Analysis and Comparison of TCP Reno and Vegas. IEEE INFOCOM 99, New York, USA, 1999.
[12] Inas Khalifa, and Ljiljana Trajkovic. An overview and comparison of analytical TCP models, ISCAS, 5:469-472, May 2004.
[13] Mohamed Tekala and Robert Szabo, Evaluation of Scalable TCP. AICCSA-05, 3rd IEEE conference in Egypt, January 2005.

Mohamed Tekala Received the B.S. degree in computer engineering from higher Institute of electronics, Ben Wlide, Libya in 1986. From 1987 to 1992 he was in computer department in cement factory in Al khoms, Libya. In 1993 he moved to the Nasser University in Al khoms. The M.S. degree in a field of computer network from Prague Technical University in Czech Republic 1997 from 1997 to 2001 he return to Nasser University as a lecturer and researcher. From 2001 he is Ph. D. student in Budapest Technical University in Hungary.

Robert Szabo received his MSc in electrical engineering from Budapest University of Technology (BME) in 1996. He followed his studies in the same institution and earned his PhD in 2002. Since 1999 he works in the High Speed Networks Laboratory (HSNLab) at the Department of Telecommunication and Media Informatics, BME. His is currently an associate professor and head of the HSNLab. In 2003 he obtained the Bolyai scholarship of the Hungarian Academy of Science. He is the president of the Telecommunications Section of the Scientific Association for Infocommunications, Hungary, since 2005. His main research interests are architectures, protocols and performance of communication networks. He is the co-author of more than 10 journal and 30 conference papers.