**International Journal of Computer Networks and Communications Security**

VOL. 3, NO. 11, NOVEMBER 2015, 417–420
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

# Network Congestion Avoidance Using Monte Carlo Sampling

**Saleem-ullah[1], Tareef Ali Khan[2], M. Abubakar Siddique[3], Aqeel ur Rehman[4]**

[1, 2] Department of Computer Science & IT, The Islamia University of Bahawalpur, Pakistan

[3, 4] College of Computer Science, COMSATS Institute of IT, Vehari, Pakistan

*E-mail: [1]saleemullah@iub.edu.pk*

**ABSTRACT**

In this paper we have study Monte Carlo Sampling (MCS) method and applying this method on heterogeneous network traffic on intranet. Resource management and admission control have been notified to bursty traffic. We have introduced new approach based on MCS to overcome network congestion. Simulation results shows that our proposed mechanism is better in terms of throughput, utilization and appropriate congestion window size while comparing with previous scheme based on fuzzy networks.

Keywords: *Monte Carlo sampling, Packet loss, throughput, Congestion Window.*

## 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] 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 [4-7]. We tend to conduct a mathematical analysis that predicts the performance of the two TCP algorithms sharing bottleneck link. The Drop Tail 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 for the congestion signals due to buffer overflow and/or link failure.

## 2 WHY MONTE CARLO SAMPLING?

The main advantages of using a Monte Carlo simulation are: It provides approximate solutions to many mathematical problems concerning Systems that involve an element of randomness that can be modeled via a distribution. It also provides a frame work for statistical sampling of inputs, allowing results from numerical experiments to be aggregated in a meaningful way. Its use is quite simple and straightforward as long as convergence can be guaranteed by the theory. For optimization problems, it often can reach global optima and overcome local extrema.

## 3 WHY MONTE CARLO SAMPLING?

We know that TCP uses ACKs to ensure synchronization, stability and flow control. In this case if ACKs are delayed or dropped due to reverse channel characteristics, the performance of the TCP degrades even though data packets are transmitted successfully.

Therefore we have proposed a new approach based on MCS to improve TCP performance in terms of link utilization and throughput. Following figure is a network congestion avoidance decision making model based on MCS technique.

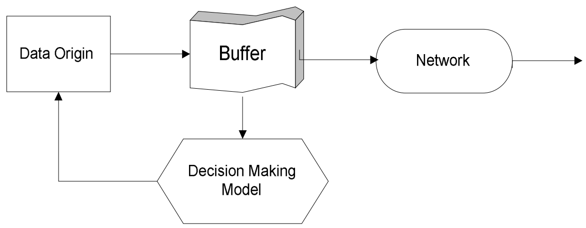


*Fig. 1. Monte Carlo based Decision Model*

***Defining Initial parameters***

*Table 1: parameters used in MCS Model*

| Model Parameters | Description |
|---|---|
| $S_d$ | Amount of data generated by the source |
| $S_b$ | Size of buffer |
| **T** | Time interval |
| $B_0$ | Amount of data out of buffer at time T |
| $B_i$ | Amount of data in the buffer at time T |

After defining the parameters we suppose that $S_d[i]$ is the total amount of data arriving during $T_i$-$T_i$-$1=T$ time interval. From this we can get the current data quantity as;

$$B_i = \min\{\max\{B_{i-1}+S_d[i]-B_o,0\},S_b\} \quad 1$$

Where $B_i$ is the amount of data in buffer at time $T_i$-$1$, in this case we suppose that buffer threshold is $S_b(th)$, packet loss and congestion will occur if the value of $B_i>S_b(th)$, then decision model will send a feed back to the source and source will decrease its window size otherwise vice versa. From this analysis we can define new statistical decision model as

$$I\{B_{i+1} > \mathrm{S}_{bth}\} > 1-\zeta \quad 2$$

Where $\zeta$ determines the trust level, which is actually a probability of congestion when $B_i>S_b(th)$. By using these statistics we can use MCS algorithm to predict data amount entering into buffer, and we can also get the prediction value and data quantity $S_b[i+1]$ at the same time by using equation 1.

Monte Carlo Predictor and filtration algorithm
The state and time varying system model is described as

$$\left.\begin{aligned} S_k &= f_k(S_{k-1},\mu_k) \\ O_k &= g_k(S_k,\lambda_k) \end{aligned}\right] k \in Z+ \quad 3$$

$K$ is time index, $O_k$ and $S_k$ are monitoring and state vectors respectively at time $k$, while $\mu_k$ and $k$ are noise vector and Gaussian respectively with its independent and unique distribution,
Then its probability initial parameters are as follows

$$I(S_0 \mid O_0) = I(S_0) \quad (4)$$

From this equation we can write the prediction equation as

$$I(S_k \mid O_{1:k-1}) = \int I(S_k \mid S_{k-1}).I(S_{k-1} \mid O_{1:k-1})d_{S_{k-1}} \quad (5)$$

and its dynamics are as

$$I(S_k \mid O_{1:k}) = \frac{I(O_k \mid S_k).I(S_k \mid O_{1:k-1})}{\int I(O_k \mid S_k).I(S_k \mid O_{1:k-1})d_{S_k}} \quad (6)$$

When applying algorithm in practice and avoid other lack of particles we can convert the above equation in to the probability of finite sampling points, therefore it is necessary to select the importance function and use resampling technique because MCS method is based on random sampling iterations. Since probability of importance function is same with and the probability distribution.

### 3.1 Algorithm

Since MCS method is based on random sampling operations can turn the integral operation into finite sampling values, thus we can convert the equation 5 into transition probability of finite sampling points. It is also necessary to select importance function when applying algorithm in practice to avoid lack of samples. To implement this algorithm based on prediction and estimation, re-sampling requires some monitoring data before time k, and then importance function can be represented as

$$q(S_{0:k} \mid O_{1:k}) = q(S_0)\prod_{j=1}^{k} q(S_j \mid S_{0:j-1}, O_{1:j}) \quad (7)$$

If we suppose this state confirm to Markov chains and variables are independent with the current state then the recursive weight value formula can be easily calculated. By using equation

5 with re-sampling we can obtain N random sample point to express the probability as

$$I(S_{k-1} \mid O_{1:k-1}) = \sum_{i=1}^{N} q(S_{k-1} - S_{k-1}^{i}) \quad (8)$$

And its update state can be expressed as

$$I(S_k \mid O_{1:k}) = \sum_{i=1}^{N} q(S_k - S_k^{i}) \quad (9)$$

Sample points can be obtained from the equation 3

## 4 SIMULATION TOPOLOGY, RESULTS AND OBSERVATION

In our simulation experiments we have consider two types of traffic generation i.e. one way and two transfers. In one way no competing data traffic on reverse paths except acknowledgements. While in two way transfers there are two TCP transfers simultaneously moving on the reverse paths, i.e. acknowledgement transfer for one connection and data for another. For example; user uploading some pages or other information on a web server while at the same time downloading data from web.

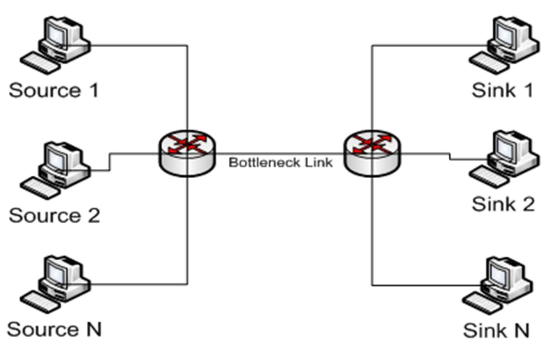


*Fig. 2. Simulation Topology*

Arrangements of these experiments make it easy to observe the behavior of both flows in case of synchronous losses. Bench of simulations with different and same delays were observed during simulations. In figure 3 and figure 5 we can see that proposed mechanism outperforms TCP NewReno in terms of congestion window and throughput (plot of 1GB link with delay of 100ms) respectively.

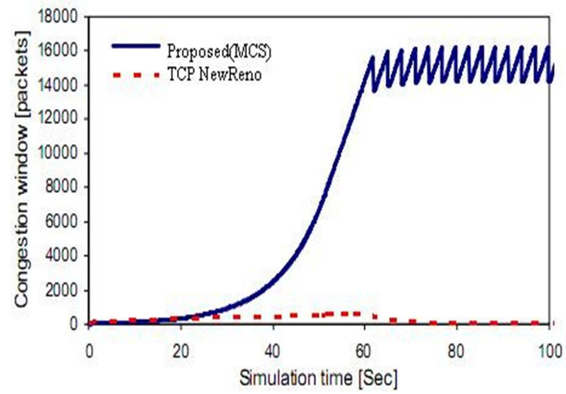


*Fig. 3. Effect of Congestion window (one way Trf.)*

Simulation results demonstrate that random losses effects on the performance of both TCP connections. In some cases proposed model out-placed NewReno under the regards of low packet loss. A congestion window plot and throughput by using formula 2, which is ( $I\{B_{i+1} > S_{bth}\} > 1 - \zeta$ ) for 1Gbps and 10ms round trip time shown on Fig 4 and figure 6 respectively.

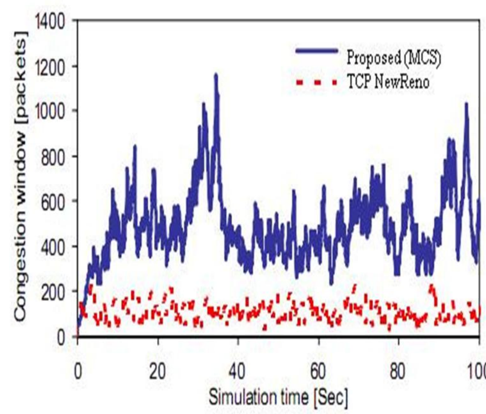


*Fig.4. Effect of congestion window(2-way trf.)*

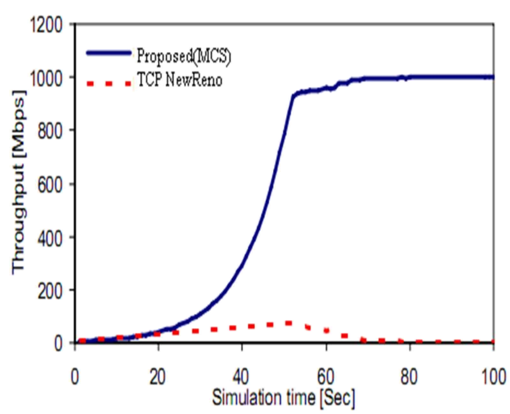


*Fig. 5. Throughput comparison (one way Trf.)*

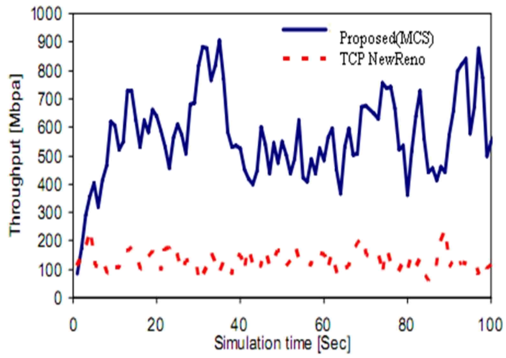


*Fig. 6. Throughput comparison (two ways Trf.)*

## 5 CONCLUSION

In this contribution, the performance of proposed MCS and NewReno TCP sharing a bottleneck link has been analyzed. Steady state in a congestion avoidance phase was applied in this proposed model. Both mathematical analysis and simulations show that the MCS outperform 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 MCS model 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, MCS parameters 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, MCS model works more friendly and efficiently with New-Reno TCP.

## 6 REFERENCES

[1] Wei Xia, and Wei Zhang, “End-to-end solution of TCP protocol in high speed network,” Computer Research and Development (ICCRD), 2011 3rd International Conference, 2011, pp. 284 – 288, DOI: 10.1109/ICCRD.2011.5764021

[2] [2] R .El Khoury, E. Altman, and R. El Azouzi, “Analysis of scalable TCP congestion control algorithm,” Computer Communications vol. 33 -2010, pp. 41–49.

[3] [3] Fei Ge, Sammy Chan, Lachlan L. H. Andrew, Fan Li, Liansheng Tan, and Moshe Zukerman, “Performance effects of two-way FAST TCP,” Computer Networks vol. 55-2011, pp. 2976–2984.

[4] [4] S. P. Kim, and K. Mitchell, “Ananalytic model of TCP performance over multi-hop wireless links with correlated channel fading,” Performance Evaluation vol. 64-2007, pp. 573–590.

[5] [5] R. E. Kooij, R. D. van der Mei, and R. Yang, “TCP and web browsing performance in case of bi-directional packet loss,” Computer Communications vol. 33-2010, pp. 50–57.

[6] [6] Sakib A. Mondal, and Faisal B. Luqman, “Improving TCP performance over wired–wireless networks,” Computer Networks vol. 51-2007, pp. 3799–3811.

[7] [7] Qiang Fu, Jadwiga Indulska, Sylvie Perreau, and Liren Zhang, “Exploring TCP Parallelization for performance improvement in heterogeneous networks,” Computer Communications vol. 30-2007, pp. 3321–3334.