# Performance evaluation of TCP-Reno, TCP-Newreno and TCP-Westwood on Burstification in an OBS Network

**Ratna Pavani.K$^1$ and N.Sreenath$^2$**

1. **Department of Computer Science, Indira Gandhi College of Arts and Science, Puducherry, ratnapavanik@gmail.com**
2. **Department of Computer science and Engineering, Pondicherry Engineering College, Puducherry, nsreenath @pec.edu**

*Abstract-* **In a TCP/OBS network the burst is formed with TCP data aggregated at the ingress node based on time dependent mechanism or quantity dependent mechanism. In the former scheme the size of the burst is dependent on the time allotted for burstification, where as in the latter one the burst is formed with a precise amount of data. Hence time is not a major constraint in quantity based mechanism for formation of burst. When the type of data sent is non-real-time and if TCP sender is slow, when Quantity based mechanism is preferred, TCP source may not generate data since the time of burstification is long and the data generated by TCP is waiting for burstification. This impediment in aggregation of data is due to delay in formation of data by the TCP source. As a result, burstification cannot be done at a time and OBS edge node is compelled to wait for completion of burstification process. The delay caused due to burstification at the ingress node may influence the performance of TCP variants as it influence the calculation of RTT and consequently size of the congestion window. In this backdrop an experimental study was made to compare the performance of TCP variants like TCP-Reno, TCP-Newreno and TCP-Westwood using Network simulator NS-2.**

***Keywords:*** *Transmission Control Protocol (TCP), Optical Burst Switching (OBS) Network, TCP-Reno, TCP-Newreno, TCP-Westwood, Network Simulator version-2 (NS-2)*

## I. Introduction

Ever Increasing demand for higher bandwidth due to higher utilization of applications such as remote data access, on demand video, multimedia messages and message streaming has motivated for the development of alternative to the existing electronic networks. Wavelength Division Multiplexing (WDM) with fiber optics is a pertinent technology that can handle the rising demand for bandwidth. In these networks, the huge bandwidth offered by a fiber is managed by isolating it into number of wavelengths, each acting as an autonomous communication channel at a data rate of 10Gbps [1]. These networks can provide about 50 Tbps bandwidth for a single fiber. Low attenuation of signal, extremely lower error bit rates, and minimum signal distortion as light rarely radiates away from fiber, are the main characteristics of WDM networks. Optical Circuit Switching (OCS), Optical Packet Switching (OPS) and Optical Burst Switching (OBS) are three accepted switching paradigms in all-optical WDM networks. If an end-to-end lightpath is established between the source node and the destination node for the entire session to avoid optical -to-electronic (OEO) conversation at the intermediate node in a network then that type of switching technology is called OCS. Setting up a lightpath causes ineffective usage of resource during high Internet traffic in an OCS network because the lightpath that is established during this session may remain active for days, weeks or sometimes months, during which there may be inadequate traffic that uses this bandwidth. This wastage of bandwidth can be avoided in OPS networks.

In OPS, IP packets are directly switched in the optical domain [2]. These packets are sent along with their headers without any prior setup into the network. In the core network, buffering of optical packets takes place using fiber delay lines (FDL) while the header undergoes optical to electronic conversion. In OPS networks with fixed length packets, packet synchronization becomes necessary to minimize contention which is difficult to implement [3]. Another major aspect of concern in OPS networks is, while using optical buffers like FDLs they are severely limited by physical space. In order to hold an optical fiber for a few microseconds, a kilometer of optical fiber will be required. Practical implementation of OPS stresses on fast switching times, though semiconductor optical amplifiers based switches have lower switching times, they are quite expensive and the fundamental architecture uses optical couplers which result in higher power losses [4].

To overcome with the problem of optical buffering and optical processing and still attain switching in optical domain, OBS networks have been proposed. OBS is considered as a balance between the coarse-grained OCS and fine-grained OPS networks [5]. There are three components in an OBS network, an ingress node, an egress node and a network of core nodes. Ingress nodes and egress nodes can be together termed as edge nodes. In OBS, a burst is the fundamental switching entity. The edge nodes must gather the IP packets and assemble them into bursts called as burstification. Packets that are destined to the same egress node and that require same level of service are put into burst assembly queue. To circumvent buffering and processing of the optical data burst at core nodes, a control

978-1-4799-0801-1/12/$31.00 ©2012 IEEE

packet also called burst header packet that contains the information about the length and arrival time of the data burst is sent ahead of data burst with an offset time. This offset time or time gap between the control packet and the data burst is sufficient to process the burst header packet and configure the switches at the core nodes. The switches along the route are configured only when the data burst arrives to enable the burst to cut through an all-optical path. At the egress node the data burst is disassembled back into IP packets.

Tell-and-wait (TAW) and Tell-and-Go (TAG) are two types of protocols mainly used for resource reservation in the core nodes. In the former protocol, the control packet travels along the entire path from ingress node to egress node making reservation for the data burst. After successful reservation of the resource, an acknowledgement is sent back by the destination, which is followed by the data burst. If at any intermediate node the reservation fails along the path, a negative acknowledgement is sent back to release the acquired resource. In TAG protocol, data burst does not wait for an acknowledgement from the destination. The control packet first travels and reserves bandwidth on the core network, but the data burst is sent after an offset time even before the control packet reaches the destination. In case the control packet fails to make the reservation along the path, the control packet along with the corresponding data burst is simply dropped or an appropriate contention resolution policy is considered. Just-enough-time (JET) and Just-in-time (JIT) are two most important signaling mechanisms in TAG protocol. In JIT, the wavelength that is reserved by the control packet for the data burst is torn down using an explicit control message. The control packet and the data burst travel on different wavelengths. The control packet needs to inform the core node only about the wavelength on which the data burst is intended to arrive. In JET, the bandwidth is reserved only for the duration of the data burst; no explicit message is required to release the acquired resource. This enhances the utilization of the wavelength but increases the processing time of the control packet.

Due to bufferless nature of OBS networks, and one way signaling scheme, the OBS networks will suffer from Random Burst Losses (RBL) even at minimum traffic loads [6]. These RBL will be interpreted as network congestion by TCP layer. For instance, if a burst that has multiple packets from a sending TCP is dropped due to contention at minimum traffic loads, the TCP sender times out , which leads to false congestion detection by TCP. This false congestion detection is referred to as False Time Out (FTO) [7]. When TCP detects this false congestion, it will start congestion control mechanism, which will reduce the size of the CW.

Functioning of TCP over IP can be reminded here. TCP is a connection oriented end-to-end protocol that provides reliable services to the applications.TCP receives information from upper layers and organizes them in form of packets/segments. TCP packet is encapsulated in an IP datagram. This IP datagram contains TCP header and TCP payload. Maximum segment size which is the largest amount of data that TCP sender is willing to send in one segment is specified by TCP. Congestion control is the main function of TCP. To begin with, TCP sender adjusts the number of segments sent in each round to the buffer size advertized by the TCP receiver to prevent overflow. This is done by using a window called congestion window (CW). To ensure that the data transfer is reliable and to avoid congestion in the network, TCP uses acknowledgements from the receiver for every segment sent. Once the packet is prepared for transmission, TCP buffers a replica of the packet and sets up the timer for the packet. This timer is set to Retransmission Time Out (RTO) value which can be computed to estimate the Round Trip Time (RTT). If the TCP sender does not receive an acknowledgement and if timer expires or if TCP sender receives triple duplicate acknowledgement then TCP understands that the network is in congestion and starts congestion control mechanism. Slow start, Congestion avoidance, Fast retransmit and Fast recovery are four phases of congestion control mechanism.

The basic assumption of various flavors of TCP is that the underlying physical medium is electronic and packets are experiencing delay due to IP routers, but with TCP over OBS networks there will be a change in RTT calculation, RTO and size of CW due to the underlying network. In this paper a study of three TCP variants namely TCP-Reno, TCP-Newreno and TCP-Westwood and their variation in behavior with respect to burst losses when there is a delay due to burstification. This paper is organized as follows; in section II analysis of TCP Variants over OBS network is done. In section III, the results of simulation are discussed. In section IV the conclusion based on simulation results is presented.

## II. TCP Variants over OBS network

TCP variants like TCP-Reno, TCP-Newreno that are used for simulation in our paper consider loss in the network as an indication of congestion and start congestion control mechanism, but, they differ in their fast retransmit and fast recovery phases from other variants of TCP. TCP-Reno avoids the need to wait for a time-out when there is a packet loss and implements fast retransmit algorithm [9, 10].During transmission if there is a triple duplicate acknowledgement; TCP-Reno understands it to be a packet loss and enters a fast recovery phase by retransmitting the lost packets. Slow start threshold is set to half of the window size in fast recovery phase and the size of CW is set to slow start threshold plus three. By this the size of the CW is inflated to the value of segments that have reached the receiver. If window size is adequate TCP sender can send new segment.TCP sender sets the CW to slow start threshold and exists fast recovery when a new acknowledgement comes in from the receiver. If several packets are lost in the same window then the TCP sender enters fast retransmit and fast recovery by halving the CW each time there is a packet loss. Halving the window continuously for each packet loss will lead to a time-out and degrades the performance of TCP-Reno.

TCP-Newreno is a slight modification of TCP-Reno [11, 18]. It modifies the fast retransmit and fast recovery phases of TCP-Reno when there are multiple packet losses in the same window. Implementation of fast retransmit algorithm of TCP-

Newreno is analogous to that of TCP-Reno. When there is a triple duplicate acknowledgement, TCP-Newreno retransmits the lost packet and starts fast recovery phase. Here, sender retransmits the remaining packets in as many RTTs as the number of packets in a window, thereby retransmits one packet per RTT. In case of partial acknowledgement, TCP-Newreno transmits the next packet. TCP sender does not wait for a triple duplicate acknowledgement to detect the loss of other packets in the same window. Therefore when there are multiple packet losses in the same window, TCP-Newreno performs better than TCP-Reno. The performance of TCP-Newreno is vulnerable by the fact that it takes one RTT to detect a packet loss. The loss of other segments can only be detected when the acknowledgement for the first retransmitted segment is received. But TCP variants like TCP-Westwood and TCP-Vegas [12] use estimated delay along the path to identify congestion unlike TCP-Reno and TCP-Newreno which uses segment loss to identify congestion.

The TCP-Westwood algorithm is based on end-to-end assessment of the bandwidth existing along the TCP connection path [13, 14]. This assessed value is obtained by filtering the stream of acknowledgements that are returning from the TCP receiver. This value is adaptively used to set the size of CW when network is congested. In case of triple duplicate acknowledgement, the CW and slow start threshold are set to the value of assessed bandwidth that is measured during the RTT of the acknowledgement. When the coarse timer experiences a timeout, the slow start threshold is set to its previous value and the size of the CW is set to 1. For every acknowledgement received successfully, the CW is linearly increased and the assessment of available bandwidth is computed. The algorithm is as follows-

**Step1:** When triple duplicate acknowledgements are received:
*Slow start threshold* =max (2, (*estimated window bandwidth* * *RTT minimum*) / *size of the segment*);
*CW* = *slow start threshold*;
**Step2:** When coarse timer experiences timeout: *Slow start threshold* = max (2, (*estimated window bandwidth* * *RTT minimum*) / *size of the segment*);
*CW* = 1;
For the algorithm above, TCP-Westwood additively increases the CW as Reno, when acknowledgements are received. During the congestion, TCP-Westwood employs an adaptive setting of CW and slow start threshold so that it follows an **Additive-Increase/Adaptive-Decrease (AIAD)** pattern [14]. The Adaptive decrease mechanism used by the TCP-Westwood improves the constancy when compared to standard TCP multiplicative decrease algorithm and increases the fair allocation of existing bandwidth.

When TCP is used over OBS network the data from TCP is collected by the ingress node and the data burst is formed. This formation of burst can be classified into two ways: time dependent and quantity dependent. In the former mechanism the burst is formed with the data that is collected in a fixed time. The transfer of burst is done without considering the size of the burst. In latter mechanism, the formation of burst is constrained by the size of the burst. Hence time taken to form the burst can be long. Apart from time based and quantity based there is also one more technique that combines the features of both the mechanisms. In this hybrid mechanism, the burst is either transferred if it meets the required amount of data or if the time taken for burstification procedure is complete. With quantity based method the burst is created with a bound on amount of data, hence the burst is formed only if definite amount of data is collected irrespective of the time needed to form the burst.

When the type of data used is non-real-time and TCP source is considered to be slow, it may be preferred to use quantity based mechanism. Because in case of time based mechanism, the data in the burst may be very low and the burst transmission process may possibly reserve wavelength unnecessarily leading to contention for some other bursts. At times when we use quantity based mechanism the TCP source may not produce sufficient data and consequently the time of burstification is long. In these circumstances the data generated by TCP is waiting for burstification and so the acknowledgement may not be received by the TCP. Owing to this condition the RTT at the source may time-out. Source TCP may presume this situation as congestion in network and react to it by lowering the CW size. This problem is analogous to multiple packet loss issue in TCP-Reno and TCP-Newreno or the way the congestion is dealt in TCP-Westwood. In TCP-Newreno, when there is multiple packet loss it takes one RTT per packet to recover from slow start phase and in case of TCP Westwood which uses adaptive increase in its CW size to increase it to an optimal value and recover from slow start phase without delay. Hence it can be assumed that TCP Westwood with its adaptive increase of CW mechanism from slow start phase after network congestion will have a superior performance over TCP-Newreno and TCP-Reno in an OBS network. In this scenario a simulation is being done using Network simulator version 2.27 (NS-2.27) to evaluate the TCP variants, TCP-Reno, TCP-Newreno, and TCP-Westwood. NS-2.27 with OBS patch [15] with random uniform burst distribution algorithm. Topology used is NSFNet. There are 14 optical core nodes in our topology with 28 TCP/IP nodes and 10 TCP connections. The IP packets are aggregated into burst at the edge nodes and transmitted all optically from source to destination. Packet processing in the core network is done by the optical classifier. The packets that are assembled in a single burst are defined in Burst size. The size of the burst is varied to evaluate the performance of the three TCP variants. Burst size varies from a minimum of 10 packets per burst to a maximum of 11000 packets per burst. This burst travels in the core network with a hop delay of 0.01ms. With 100 assembly buffers at the ingress node the simulations are being done. Burstification period is varies between 0.001 and 1.0 to estimate its affect on throughput of TCP variants. To eliminate the problems in all optical processing of packet headers, the data plane and the control plane are separated in OBS simulator. MAX-PACKET-NUM is a variable used in the simulation to count the number of packets in a burst. JET signaling mechanism is used in this network, where the control packet tries to reserve resources for the burst just sufficient enough for transmission on each link it traverses. The control packet has all the vital information so that each intermediary optical switch in the core OBS network can transfer the data burst and also configures its switching matrix in order to switch the burst all-optically. The conversion of electrical-optical-electrical is taken care by the edge node in the core OBS network. They generate and forward the control packets followed by the data burst.

The node entrance has classifier that separates TCP segments form the optical bursts. Latest Available Unused Channel with Void Filling (LAUC-VF) [16] and Minimum Starting Void (Min-SV) [17] are the scheduling algorithms that are presently implemented in OBS. In this simulation burstification period and burst size are used as parameters to evaluate the throughput of TCP-Reno, TCP-Newreno and TCP-Westwood.

## III. Results

The topology used for the simulation is NSFNet with 14 core nodes. This Network consists of 14 optical core nodes and 28 electrical nodes. This core network in our simulation is modeled as a single network with 1Gbps bandwidth and 10ms propagation delay. The access links have bandwidth of 155Mbps with link propagation delay of 1ms. Maximum number of packets in a burst varies between from 10 to 11000.

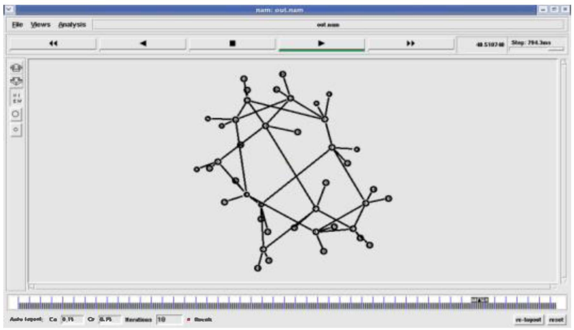

Figure1: NSF Topology with 14 optical core nodes and 28 electric nodes

| | |
|---|---|
| Topology : | NSFNet |
| Number of optical core nodes: | 14 |
| Number of electronic nodes: | 28 |
| Number of TCP/IP connection: | 10 |
| Packets per burst : | 10 to 11000 |
| Max lambda: | 20 |
| Link Speed: | 1GB |
| Hop delay: | 0.01ms |
| Burstification period: | 0.001ms to 1.0ms |

Table: 1 Simulation Parameters

Figure2 shows the simulation results on TCP-Reno. It is a graph showing the performance of TCP-Reno when there is a variation in burst time out (BTO) value. BTO is the time out value of burstification, that is, the delay between the arrival of the packet and formation of the optical burst. While simulating with varying burst sizes the performance of TCP-Reno is evaluated for BTO values varying from 0.001ms to 1.0ms. When the BTO is 1.0 there is an increase in throughput of TCP-Reno when compared to BTO value 0.001ms. It can be inferred from this result that if BTO value is decreased there will be degradation in the performance of TCP-Reno. As TCP-Reno is basically designed for electronic networks, in OBS networks it can infer that bigger BTO will stabilize the performance of TCP-Reno.

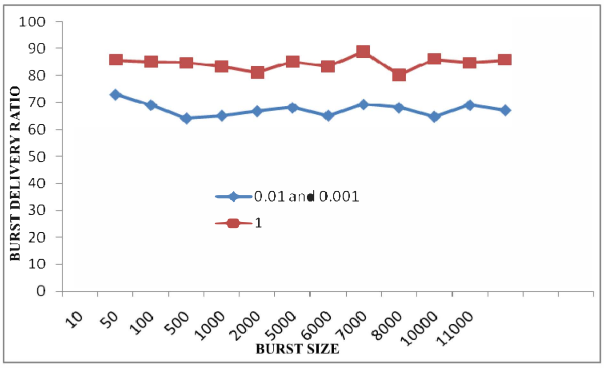

Figure: 2 Throughput of TCP-Reno with varying BTO

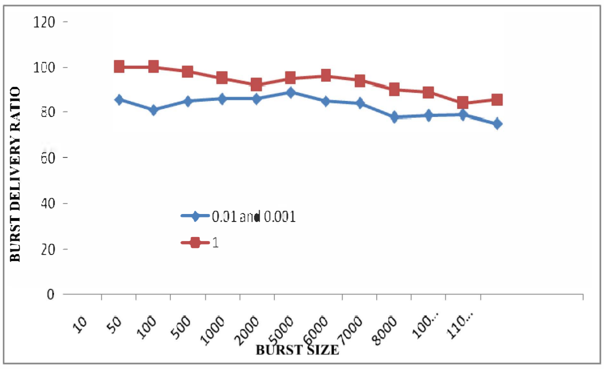

Figure: 3 Throughput of TCP-Newreno with varying BTO

TCP-Newreno does slight modification to the performance of TCP-Reno in congestion control mechanism. Since both the variants are actually designed for electronic networks their performance will be affected by BTO value also, apart from other factors like random contention and false time out which are common in OBS networks. In Figure3 the simulation results of TCP-Newreno are plotted with varying BTO values. There is a slight fall in the throughput of TCP-Newreno when the burst size is more than 11000 and BTO 0.001ms owing to congestion in the network.

But when it comes to simulation results of TCP-Westwood from figure4, it can be observed that there is not much variation in the throughput when there is change in BTO. Like TCP-Reno or TCP-Newreno TCP-Westwood is not influenced by a change in BTO. It is also analyzed, that there is no degradation of performance when there is an increase in burst size even with lower values of BTO.

The higher performance of TCP-Westwood can be seen in figure 5. Even after 11000 bursts there is a growth in the graph of TCP-Westwood in comparison with the result of TCP-Reno and TCP-Newreno which show a decline in burst delivery ratio.

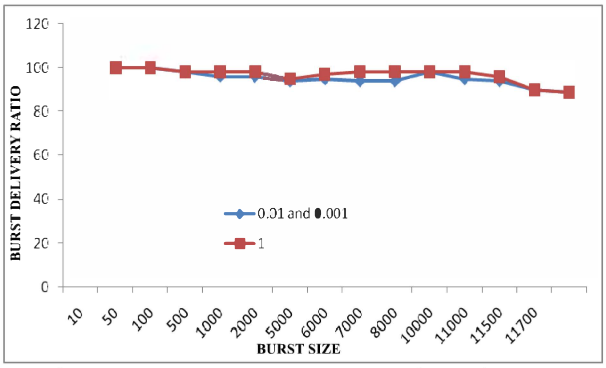

Figure: 4 Throughput of TCP-Westwood with varying BTO

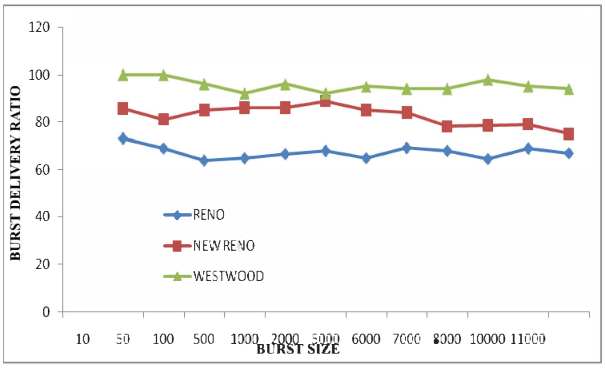

Figure: 5 Comparison of TCP-Reno, TCP-Newreno and TCP-Westwood when to is 0.001

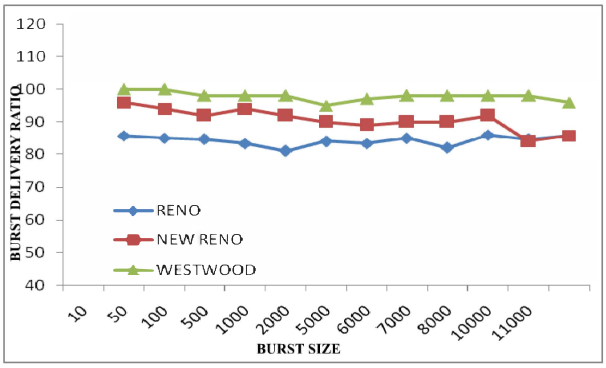

Figure: 6 Comparison of TCP-Reno, TCP-Newreno and TCP-Westwood when to is 1.0

Figure: 6 Comparison of TCP-Reno, TCP-Newreno and TCP-Westwood when to is 1.0. Even with higher BTO the performance of TCP-Westwood is unaltered, where as there is a slight change in the performance of TCP-Newreno after 10000 bursts due to congestion in the network.

## IV. Conclusion

In an all optical network when simulation is done as per table1, the performance of TCP-Reno, TCP-Newreno and TCP-Westwood was evaluated. Our results show that with a decrease in BTO affect the performance of the above three TCP variants. When BTO value is as low as 0.01ms TCP-Westwood performed better than TCP-Reno and TCP-Newreno.

Since the available bandwidth and speed of optical networks is very high in comparison with electrical networks our simulation results show that the TCP-Westwood with its adaptive decrease policy will utilize the available bandwidth to its maximum. In future a detailed study of other variants of TCP need to be analyzed which can perform better over OBS. Some changes can also be made in the existing setup of the core OBS to improvise it to the existing TCP variants.

## References

[1] T.Venkatesh and C.SivaRamMurthy, “An analytical approach to optical Burst Switched Networks”, e- ISBN 978—4419-1509-2 Springer Newyork Dordrecht Heidelberg London,2010.

[2]Mukherjee.B, “Optical WDM networks”, Springer, USA 2006.

[3]Blumenthal, D.J., Prucnal.P.R., Sauer, J.R, “Photonic packet switches: Architectures and experimental implementation”, proceedings of IEEE 92(11), 1650-1667,1994.

[4]Ramaswam.R, Sivarajan.K.N, “optical networks a practical perspective” USA 1998.

[5]Vokkarane.V.M, Jue.J.P, “Introduction to Optical burst switching”, Springer, USA, 2004.

[6]Qiong Zhang, Vinod M. Vokkarane, Yuke Wang, and Jason P. Jue, “TCP over optical burst- switched networks with controlled burst retransmission,” Photonic Network Communications December 2011, Volume 22, Issue 3, pp 299-312.

[7]Xiang Yu, Chunming Qiao and Yong Liu, “TCP I mplementations and False Time out Detection in OBS Networks,” Proceedings of IEEE INFOCOM, pp. 358–366 2004.

[8]W. Stevens, “TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms,” RFC 2001, 1997.

[9]Jeonghoon Mo, Richard J.La, Venkat Anantharam, and Jean Walrand, “Analysis and Comparison of TCP Reno and Vegas,”1999 IEEE.

[10]Andrea Detti, Marco Listanti, “Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks,” Proc IEEE INFOCOM 2002.

[11] S. Floyd, T. Henderson, A. Gurtov, ”The NewReno Modification to TCP's Fast Recovery Algorithm”, Network Working Group, RFC 3782, April 2004.

[12]Luigi A. Grieco and Saverio Mascolo, “Performance Evaluation and Comparison of Westwood+, New Reno, and Vegas TCP Congestion Control,” ACM SIGCOMM computer communications review April 2004.

[13] Mascolo, S., Casetti, C., Gerla, M., Sanadidi, M., Wang, R., “TCP Westwood: End-to-End Bandwidth Estimation for Efficient Transport over Wired and Wireless Networks,” in Proceedings of ACM Mobicom 2001, (Rome, Italy, July 2001).

[14] Grieco, L. A., and Mascolo, S., “Westwood TCP and easy RED to improve Fairness in High Speed Networks,” in Proceedings of IFIP/IEEE Seventh International Workshop on Protocols for High-Speed Networks, PfHSN02, (Berlin, Germany, April, 2002).

[15]Guray Gurel, Onur Alparslan and Ezhan Karasan, “nOBS: an ns2 based simulation tool for performance evaluation of TCP traffic in OBS networks,” *Annals of Telecommunications*, vol. 62, no. 5-6, pp. 618-632, May-June 2007.

[16] Xiong, Vandenhoute, Cankaya, “Control Architecture In Optical Burst-Switched WDM Networks,” IEEE Journal on Selected Areas in Communications, 18, no 10, pp. 1838–1851, Oct. 2000.

[17] Xu, Qiao, Li, Xu, " Efficient Channel Scheduling Algorithms In Optical Burst Switched Networks," Proceedings of IEEE, Infocom'03, 3, pp. 2268 - 2278, 30 March-3 April 2003.

[18]V. Bimal, T. Venkatesh, C. Siva Ram Murthy, "A stochastic model for the behavior of multiple TCP NewReno sources over optical burst switching network," Photonic Network Communications October 2011, Volume 22, Issue 2, pp 109-122