

Evaluating the Burst Delivery Ratio of TCP Variants with Varying Burst Sizes and Offset Times in an all-Optical Burst Switching Network

Ratna Pavani.K

Department of computer science, Indira Gandhi
College of Arts and Science, Puducherry,

N.Sreenath

Department of Computer science, Pondicherry
Engineering College, Puducherry,

ABSTRACT

To protract next generation Internet, TCP over OBS is a potential transport paradigm. Since the popular variants of TCP were designed to work with IP networks, they behave differently over OBS networks. Packets from various IP sources assemble into a Data burst and it is transmitted from ingress node to egress node cutting through the core nodes all-optically. Data burst is always preceded by a control packet with an offset time. Owing to bufferless nature of OBS burst losses may occur due random contention. In this scenario an experimental study was made to evaluate the performance of two popular TCP variants, TCP-Reno and TCP-Vegas using NS-2.

Keywords

Transmission Control Protocol (TCP), Internet protocol (IP), Optical Burst Switching (OBS) Network, TCP-Reno, TCP-Vegas, Network Simulator version-2 (NS-2)

1. INTRODUCTION

In the past decade there has been an unparalleled increase in the demand for high bandwidth applications and services. This increase is due to significant growth in the number of Internet users and raise in bandwidth intensive applications such as video conferencing, voice-over-IP, interactive video-on-demand [1]. To clamp the ever growing demand of bandwidth copper cables were reinstated with optical networks. An optical fiber is capable of supporting a bandwidth demand up to 50Tbs. Low attenuation of signal, exceptionally lower error bit rates, and minimum signal distortion as light rarely radiates away from fiber, are the primary features of WDM networks. Optical Circuit Switching (OCS), Optical Packet Switching (OPS) and Optical Burst Switching (OBS) are three established switching paradigms in all-optical WDM networks. If an end-to-end lightpath is established between the source and the destination nodes for the entire session to evade optical -to-electronic (OEO) conversation at the intermediary node in a network then that sort of switching technology is called OCS. Setting up a lightpath results ineffective usage of bandwidth during high Internet traffic in an OCS network since the lightpath that is established during this session may remain alive for days, weeks or sometimes months, during which there may be inadequate traffic that exploits this bandwidth. In OPS networks this wastage of bandwidth can be avoided.

In an OPS network, IP packets coming from various sources are directly switched in the optical domain [2]. These packets are sent along with their headers into the network without any prior setup. In the core network, buffering of optical packets takes place by means of fiber delay lines (FDL) as the header

undergoes optical to electronic conversion. In OPS networks with fixed length packets, packet synchronization turns out to be crucial to minimize contention which is difficult to execute [3]. One more major facet of concern in OPS networks is, while using optical buffers like FDLs they are severely constrained by physical space. To hold an optical fiber for a few microseconds, a kilometer of optical fiber will be essential. Pragmatic implementation of OPS stresses on fast switching times, despite the fact that semiconductor optical amplifiers based switches have lower switching times, they are quite expensive and the fundamental architecture employs optical couplers which result in higher power losses [4].

To trounce with the problem of optical-buffering and optical-processing and still attain switching in optical domain, OBS networks have been proposed. OBS is recognized as a balance between the coarse-grained OCS and fine-grained OPS networks [5]. The three components of OBS networks are an ingress node, an egress node and a network of core nodes. Ingress nodes and egress nodes can be collectively termed as edge nodes. In OBS, a burst is the basic switching component. The edge nodes have to congregate IP packets and assemble them into bursts called as burstification. Packets that are intended to the same egress node and that need same level of service are put into burst assembly queue. To avoid buffering and processing of the optical data burst at core nodes, a control packet also called burst header packet that holds 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.

Just-enough-time (JET) and Just-in-time (JIT) are two most important signaling mechanisms in tell-and-go (TAG) protocol. In JIT with an explicit control message, the wavelength that is reserved by the control packet for the data burst is torn down. The data burst and the control packet travel on different wavelengths. The control packet needs to notify the core node only about the wavelength on which the data burst is proposed to arrive. In JET the bandwidth is reticent only for the duration of the data burst. No explicit message is required to liberate the acquired resource. This amplifies the usage of the wavelength but increases the processing time of the control packet.

Transmission control protocol (TCP) is the de facto standard that account for 90% of internet traffic [6] when exploring a new networking model for the future Internet like optical burst switching (OBS). At this point it has to be reminded that TCP has been subjected to a significant amount

of research over the past years and many variants of TCP were suggested for adjusting to new network scenario with contrasting transmission characteristics while analyzing TCP congestion control mechanisms [7]. In the recent years lot of suggestions were made to modify TCP to adapt it to web based applications such as HTTP, SMTP, peer-to-peer file sharing [8], [9], and grid computing [10] that account for a majority of data traffic in the Internet. Hence understanding and improvising the performance of TCP implementations over OBS networks is critical. The fundamental assumption of various TCP flavors is that they are working on an electronic medium and packets experience a delay due to congestion in IP routers. TCP has been sorted into the following three categories i.e., Loss-based, Delay-based and Explicit notification-based.

This paper is organized as follows; section II presents literature survey with respect to variants of TCP, Section III describes impact of OBS over TCP variants and motivation to the proposed work. In Section IV discusses the system performance. Results along with conclusion and future work are in Section V.

2. LITERATURE SURVEY

TCP is in-charge of controlling end-to-end communications using the services provided by the network layer which is generally IP. So there is a need to describe the behavior of TCP as some of the characteristics of TCP have a significant impact when used over OBS networks.

TCP's congestion control: TCP sender sends data in chunks called segments which are acknowledged (ACK) by the TCP receiver. Based on the size of the Congestion Window (CW) and the threshold value a certain number of segments are allowed to be sent using a window. For every successful ACK, size of the CW is incremented. There are various phases in the congestion control of the TCP which mainly controls the rate of transmission in the network to avoid congestion and to retransmit the lost packets. To begin with, TCP has a low CW typically one segment and its size is increased by one segment every time positive ACK is received. This period is called slow start (SS). During SS for every successful ACK the CW grows exponentially either till a packet is lost or if the size of CW equals predefined threshold value.

When TCP experiences congestion due to loss of packets the size of the CW is reduced to initial state and TCP enters congestion Avoidance (CA) phase. During this phase, the size of the window (W) is increased by 1/W each time an ACK is received. This growth continues until maximum window size is reached or a packet loss is detected. TCP receiver sends a duplicate ACK if a segment arrives out of order. Duplicate ACKs are used by the TCP sender to infer about the lost segments in the Fast Retransmit (FR) phase. When a sender receives triple duplicate ACK, it is then considered as a sign of network congestion. TCP sender instantly retransmits the lost packets without waiting for the Retransmission Time Out (RTO). After retransmission of the lost segment Fast Recovery algorithm is initiated by the TCP sender. TCP implementation of the fast recovery algorithm is dependent on the type of the TCP variant [11].

Owing to the bufferless nature of OBS core network and the one-way signaling mechanism the OBS network endures from random burst losses due to contention even at low traffic loads. There are several contention resolution schemes that can condense random burst loss. These schemes include Fiber Delay Lines (FDL's) [12], wavelength conversion (WC)[13],

segmentation [14], and deflection [15]. In this paper a study has been made to analyze the performance of loss based TCP-Reno with delay based TCP-Vegas over bare bone OBS networks using NS-2.

3. MOTIVATION – IMPACT OF OBS OVER THE PERFORMANCE OF TCP VARIANTS

TCP-Reno is principally a loss-based TCP variant that considers a packet loss as an indication of network congestion and follows an additive increase multiplicative decrease (AIMD) window-based congestion control mechanism. AIMD has four stages of congestion control- SS, CA, FR and fast recovery [16]. When we have multiple packet losses in a single window the performance of TCP-Reno is significantly degraded as TCP-Reno can only detect single packet loss at a time. The information about the second packet loss reaches only after the acknowledgement of the first packet is received by the TCP sender after one round trip time (RTT). Delay based variant TCP Vegas improvises TCP-Reno in the SS, CA and retransmission stages [17, 18].

TCP-Vegas operate two parameters namely Estimated Throughput (ET) and Actual Throughput (AT). TCP-Vegas estimate the congestion in the network by computing the delay of the packet transmission in terms of RTT. The Initial-RTT is first determined by TCP-Vegas and is considered as minimum measured RTT which is used to identify the propagation delay and queuing delay. The calculation of ET is computed as

$ET = CW / \text{Initial-RTT}$ (CW = current CW) and AT is derived as $AT = CW / \text{RTT}$.

TCP-Vegas compute the variation between ET and AT to regulate its present CW size as follows by using two parameters called α and β .

$$\begin{aligned} \text{Difference/ Variation} &= (\text{Expected} - \text{Actual}) \text{Initial-RTT, where Difference} > 0 \\ &= ((CW/\text{Initial-RTT})(CW/\text{RTT})) * \text{Initial RTT} \\ &= CW * (1 - (\text{Initial-RTT}/\text{RTT})) \end{aligned}$$

With the help of α and β TCP-Vegas controls the size of CW as follows,

$$\begin{aligned} CW &= CW + 1 && \text{iff difference} < \alpha \\ CW &= CW && \text{iff } \alpha \leq \text{difference} \leq \beta \\ CW &= CW - 1 && \text{iff difference} > \beta \end{aligned}$$

If α is larger than the difference, the CW is linearly increased in the next round and if β is less than the difference, the CW is linearly decreased. TCP-Vegas constantly tries to sustain the backlog of packets between α and β . TCP-Vegas approximate that the network is congested if the AT is less than ET, hence decreases the transmission rate. In SS state the size of CW in TCP-Vegas is initialized to two segments and is exponentially increased for every RTT. Slow start threshold is estimated by the value of γ . The value of the CW remains unaltered between two consecutive rounds to make a valid assessment between ET and AT. When an acknowledgment is received TCP-Vegas sender records the clock to evaluate the estimated RTT using the current time and the timestamp recorded for the associated packet.

Each time a duplicate acknowledgement is received it validates the difference in current time and the timeout of the packet is larger than the RTT; then it immediately retransmits

the segment without waiting for triple duplicate acknowledgements or a coarse timeout [17]. Whenever a non duplicate acknowledgement is received, TCP-Vegas checks for its timeout value from the time the packet was sent and retransmits without waiting for the duplicate acknowledgement if the segment time exceeds the timeout value. In this way when there are multiple packet losses TCP-Vegas outperforms TCP-Reno. At this situation it is understood that both the TCP variants discussed are predestined on the supposition that the network congestion can be effectively indicated either by packet loss, prolonged RTT or combination of both [19].

The buffer less nature of OBS networks can result in random burst contention losses (RBL) that impose a significant impact on the upper-layer protocols like TCP particularly for those that take packet loss as the only indication of network congestion. In case of TCP-Reno when a burst loss occurs multiple packets in the burst are lost. The packets in the burst may be from the same TCP source. In such case the performance of TCP will radically deteriorate, whereas TCP-Vegas adjust the RTT based on the difference between ET and AT. So the enticement of the work is to evaluate the performance of TCP-Reno and TCP-Vegas over OBS networks and asses their performance with varying burst sizes and offset time.

4. SYSTEM PERFORMANCE

The system was designed using NS-2 with modified OBS patch [20]. Random uniform distribution algorithm was used for burstification of IP packets into bursts. Topology used is NSFNet. There are 14 optical nodes and 28 electrical nodes with 10 TCP/IP connections. The values used for simulation are shown in Table: 1.

Packets in the core network are processed by optical classifier. Since the next hop for a packet is in the optical domain optical classifier forwards the packet for burstification, OBS segregates the data and the control planes in the optical and electronic domain respectively. This purges the problem in all-optical processing of packet headers. MAX-PACKET-NUM variable in simulation environment specifies the number of IP packets contained in a single burst. In this simulation we have varied the size of MAX-PACKET-NUM from 10 packets to 10000 packets per burst. JET signaling mechanism is used in the core network to configure control packet's information so that data burst traverses from ingress node to egress node cutting through the switching matrix all-optically without undergoing optical-electrical-optical conversion. The edge nodes generate and forward the control packets trailed by the data burst. The classifier at the node entrance separates TCP segments from optical bursts.

Table: 1 Simulation Parameters

Topology Used	NSFNet
Total Number of optical core nodes	14
Total Number of electronic nodes	28
Total Number of TCP/IP connection	10
Packets per burst varies from	10 to 10000
Max lambda value	20
Maximum Link Speed	1GB
Hop- delay	0.01ms
Burstification period Value	0.001ms , 0.01ms

Latest Available Unused Channel with Void Filling (LAUC-VF) [21] and Minimum Starting Void (Min-SV) [22] are the two scheduling algorithms implemented in OBS core network. The core network consists of 1Gbps links with 10 ms propagation delay. The access links have 1ms link propagation delay with 155Mbps bandwidth. In our simulation we have altered the offset time and the size of the burst at the ingress node to obtain the performance of TCP variants over OBS network.

5. SIMULATION RESULTS

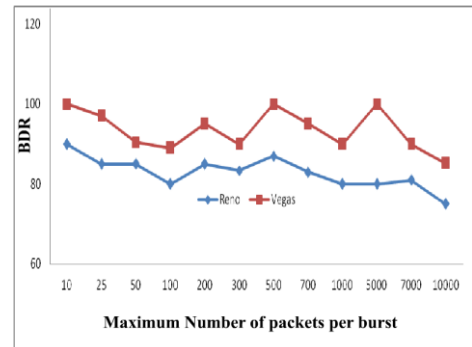


Figure1: Simulation results of TCP-Vegas and TCP-Reno with offset time 0.01

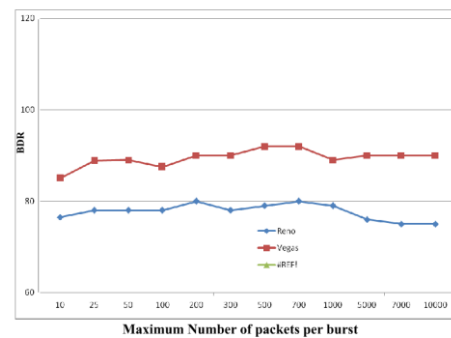


Figure2: Simulation results of TCP-Vegas and TCP-Reno with Offset time 0.001

In Figure1 Maximum-flow-queue is 100. The offset time is set to 0.01. To process TCP segment random uniform distribution algorithm is used. FTP traffic generator is used to generate traffic from TCP source to TCP destination. To consider traffic in one direction only and avoid ACK to be burstified we have set TCP-ACK to 1 so that TCP-ACK packets are not burstified. Each simulation with varying parameters was run for a period of 15 minutes. Both the TCP variants were tested with varying burst sizes to calculate burst delivery ratio (BDR). In the above simulation the throughput of TCP-Vegas is slightly better than TCP-Reno when burst size is less than equal to 7000 packets per burst. After the burst size of 7000 packets there is a slight decline in the performance of TCP-Vegas. This can be presumed because of excessive negative delay due to minimization of processing speed to 0.01ms and increased burst sizes. When simulation was done with 10000 packets per burst and with maximum-flow-queue of 100 there is a fall in the performance of TCP-Vegas and TCP-Reno. When burst size is below 1000 packets the performance of both these flavors is significant.

In figure2 both the variants showed lower BDR when the offset time is reduced. When burst offset time decreased to 0.001ms, the performance of both the variants has lowered. It is observed that TCP-Vegas remained consistent with approximately 70% BDR there is a diminishing trend in the performance of TCP-Reno after 1000 packets. There is an amount of negative delay from this point onwards and there could be multiple retransmissions which may be due to random contention and false time out.

6. CONCLUSION

In an all-optical network with high bandwidth and faster transmission speeds, a TCP variant that can clutch maximum data is optimal. In this work simulations have been done with varying burst sizes and offset time to obtain BDR of TCP-Reno and TCP-Vegas. TCP-Vegas performed better than TCP-Reno. When time delay was 0.01ms the performance of TCP-Vegas was better than TCP-Reno. TCP-Vegas with this offset delay resulted in 100 percent BDR when the burst size was 7000. As the burst size increased during simulations we have noticed that there are random burst losses in both TCP-Vegas and TCP-Reno. Throughput of TCP-Vegas is better in comparison with TCP-Reno on an OBS network. When we have altered the offset time to 0.001ms TCP-Vegas showed constant BDR. TCP-Vegas BDR reduced when the offset delay was reduced and there were burst losses due to false time outs and random contentions but throughput remained around 90 percent. Whereas the performance of TCP-Reno was affected and after burst size of 5000 packets BDR fell below 70 percent with lot of negative delay and random burst losses.

7. FUTURE WORK

Since both the variants assume that the underlying network is electronic and considers that loss of packet is due to network congestion, their performance over OBS network was evaluated and the graphical results are presented for analysis. With the available results we evaluate that TCP-Vegas adapts itself better to OBS network than TCP-Reno. As a future work we propose to work on other variants of TCP by changing some of the parameter values and other factors in network such as traffic and CW size.

8. REFERENCES

- [1]. Sagar H. Sodhatar, Rohit B. Patel, Janardana V. Dave, "Throughput Based Comparison of Different Variants of TCP in Optical Burst Switching (OBS) Network", 2012 International Conference on Communication Systems and Network Technologies.
- [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]. V.Bimal, T.Venkatesh and C.SivaRamMurthy, "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

- [7]. B. Shihada and P-H. Ho, "Transport Control Protocol (TCP) in Optical Burst Switched Networks: Issues, Solutions, and Challenges," Communications Surveys & tutorials, IEEE , 2008.
- [8]. I. Stoica and et. al., "Chord: A scalable peer-to-peer lookup protocol for internet applications," in Proceedings of ACM SIGCOMM, 2001.
- [9]. K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in Proceeding of ACM SIGMETRICS, 2003.
- [10].I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Grid: Enabling scalable virtual organizations," Intenational Journal of High Performance Computing Applications, vol. 15, pp. 200–222, 2004.
- [11].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.
- [12].I. Chlamtac, A. Fumagalli, et. al., "CORD: Contention Resolution by Delay Lines," IEEE Journal on Selected Areas Communication, vol. 14, pp. 1014-1029, June 1996.
- [13].S. Danielsen, P. Hansen, and K. Stubkjear, "Wavelength Conversion in Optical Packet Switching," Journal of Lightwave Technology, vol. 16, no. 12, pp. 2095-2108, Dec. 1998.
- [14].V. M. Vokkarane, J. P. Jue, and S. Sitaraman, "Burst Segmentation: an Approach for Reducing Packet Loss in Optical Burst Switched Networks," Proceedings of IEEE ICC, 2002.
- [15].F. Forghieri, A. Bononi, and P. Prucnal, "Analysis and Comparison of Hotpotato and Single-buffer Deflection Routing in Very High Bit Rate Optical Mesh Networks," IEEE Transaction on Communications, vol. 43, no. 1, pp. 88-98, Jan. 1995.
- [16].W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, 1997.
- [17].L. Brakmo, S. Brakmo, and L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," Proceedings of SIGCOMM, 1994.
- [18].L. Brakmo and L. Peterson, "TCP Vegas: End-to-End Congestion Avoidance on A Global Internet," IEEE JSAC, vol. 13, no. 8, Oct. 1995, pp. 1465–80.
- [19].Basem Shihada, Qiong Zhang, Pin-Han Ho, and Jason P. Jue, "A Novel Implementation of TCPVegas for Optical Burst Switched Networks". Optical Switching and Networking Volume 7, Issue 3, July 2010.
- [20].Guray Gurel, Onur Alparslan and Ezhan Karasan, "nOBS: an ns2 based simulation tool for performance evaluation of TCP traffic in OBS net- works," Annals of Telecommunications, vol. 62, no. 5-6, 2007.
- [21].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.
- [22].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.