

Improving optical burst switching networks (OBS) performance by adjusting maximum burst size and burstification time

Reza Poorzare^{1*}, Ali Poorzare², Siamak Abedidarabad²

¹ Young Researchers Club, Ardabil Branch, Islamic Azad University, Ardabil 561575675, Iran

² Ardabil Branch, Islamic Azad University, Ardabil 561575675, Iran

Corresponding Author Email: rezapoorzare@gmail.com

<https://doi.org/10.18280/rces.050101>

ABSTRACT

Received: 23 January 2018

Accepted: 19 March 2018

Keywords:

burst size, burstification time, optical burst switching, Transport Control Protocol (TCP).

Because Optical Burst Switching (OBS) networks don't have buffers it leads to problems like burst lost and burst contention when arriving at the destination. These problems downgrade OBS networks performances. When traffic of the network is not heavy burst contention can happen, but when it happens the network thinks the traffic is heavy and it decreases the rate of sending bursts when it's not necessary. Another important factor in our paper is Packet Delivery Ratio (PDR). Packet loss can happen in two ways in the network: one is because of heavy load and the other one is because of burst contention. In this paper we adjust the burst size and burst creation time to improve the performance of the OBS network. We simulate our algorithm in NS-2 then compare it with another scheme.

1. INTRODUCTION

Optical Burst Switching (OBS) networks are new field to optical area and researchers have done a lot of studies on TCP for customizing it to the new networks [1-8]. These studies are divided in to three groups: Loss-based TCP (such as TCP Reno [11] and TCP Sack [12]), delay-based TCP (such as TCP Vegas [13] and Fast TCP [9, 10]) and explicit notification-based TCP (such as XCP [14]). One of these new networks is optical burst switching (OBS) networks. OBS is a switching technique for optical networks that shows can attain dynamic and efficient bandwidth allocation for managing the internet traffic and it can deal with large amount of internet traffic [15, 16]. OBS also can be used in new application such as grid and distributed networks [17-19]. In this network packets that arrived to the edge node are aggregated and assembled into bursts and then these burst are sent through the optical network. This means we have two kinds of nodes in the network: core nodes and edge nodes. Edge nodes consist of ingress nodes and egress nodes. Ingress nodes are on sender's side and egress nodes are one receiver's side. The edge node is responsible for assembling arrived packets into burst and disassembling arrived bursts. Ingress nodes assemble arrived packets into burst and egress nodes disassemble arrived bursts into packets. Edge nodes are connected to each other via the core nodes. Core nodes are in the backbone of the network which carry the heavy traffic and is responsible for forwarding bursts through the optical network.

Because of OBS networks large optical links and bandwidth they can deal with large traffics. For handling this large traffic, OBS networks use Wavelength Division Multiplexing (WDM) technique. For better utilization and because bandwidth of OBS network's links are large, the links are divided to more than one route to carry the bursts. These links can work individually and they are not dependent to each other. We use this technique for better utilization of the bandwidth and

prevent wasting the large bandwidth.

OBS network in nature is bufferless [15], In OBS networks before sending bursts, first the edge node sends a control packet to reserve resources in the core nodes, this process is called burst reservation. Each burst uses one-way signaling resource reservation protocol and goes through the OBS network. The control packets contain information about the burst including the burst offset time and length. Offset time is the delay time between sending control packet and its corresponded burst [16].

In OBS, a burst loss can happen if a resource is being utilized for a relatively long period of time (congestion), but one of the main problems in OBS networks is burst contention. It means there is a probability that bursts can have contention in the network, and it causes burst drops, which happens when resource utilization of the network in a given period of time is low. So, when a burst loss due to random contention in OBS occurs, it might cause the TCP senders to react improperly to this burst loss and misinterpret source of this burst drop and wrongly suppose that it is due to the congestion, which is called TCP false congestion detection [22], and negatively effects the throughput of OBS network.

One of the solutions to cope with the TCP false congestion detection problem which is proposed in [17], is explicit signaling from OBS layers to TCP layers. But this approach has some drawbacks. If we generate explicit signal for every random burst contention, it increases the network overhead, and eventually, performance of the network is decreased. There are some methods to solve false congestion detection problem in OBS networks such as burst retransmission and deflection scheme at the OBS layers [18-20]. This scheme can hide some of the bursts loss events from the upper TCP layers, therefore the chance of false congestion detection problem in the network is reduced. In burst retransmission or deflection, contented bursts are retransmitted at the edge nodes or can be deflected to alternative routes, respectively. The drawback of

this scheme is if deflection routing is enabled in the network, we have to use optical buffers in the network. In [15] a threshold-based TCP Vegas is proposed. This method adjusts size of congestion window according to round trip times (RTTs) of packets received at TCP senders. If the number of RTTs that are longer than minimum RTTs exceeds the threshold, this means congestion happens in the network, otherwise there is no congestion in the network. Coordinated burst cloning and forward segment redundancy has been proposed in [15], to prevent data loss in random contention in the optical networks. In this method, redundant segments are added to each burst at the edge nodes and redundant burst segmentation (RBS) is implemented in the cores, so if a contention occurs, primarily redundant data is dropped. The drawback of this method is adding redundant to the bursts increases overhead of the network. There are some other modifications in [25-33].

In this paper we adjust maximum burst size and burstification time to obtain the best value, and by using these variables we are able to hide the effect of contentions that occur in the network. We propose that by obtaining the best maximum burst size and burstification time, we can neutralize the negative impact of burst contentions in OBS network. Extensive simulation studies shows that our method improves performance of the OBS network.

The rest of paper is organized as follows. Section 2 describes TCP Vegas as the background for the research. Section 3 presents proposed scheme. Simulation results come in section 3. In section 4 we compare our method with another scheme and finally concluding remarks are given in section 4.

2. TCP VEGAS

Due to the fact that our proposed method is for OBS networks that are running under TCP Vegas protocol, so because TCP Vegas is a widely known protocol we briefly describe Vegas as a background to the new method. For estimating available bandwidth and congestion status in the network, TCP Vegas [13, 23, 24] measures the RTT (round trip time) of each packet in the network.

In TCP Vegas congestion avoidance phase, for determining congestion status in the network, TCP Vegas compares the estimated and the measured throughput in a specific time window. First TCP Vegas computes BaseRTT as the minimum measured RTT, which is primarily determined by the propagation delay and the queuing delay.

$$Expected = \frac{cwnd}{BaseRTT} \quad Actual = \frac{cwnd}{RTT} \quad (1)$$

where, cwnd is congestion window size. TCP Vegas calculates Expected and Actual then, compares the two and adjusts next cwnd. TCP Vegas congestion avoidance mechanism goal is to maintain expected number of on-fly backlog packets.

After congestion avoidance phase we investigate slow start phase. In slow start phase TCP Vegas increases the cwnd exponentially only every other RTT, and if the cwnd reaches slow-start threshold exits the slow start phase. In order to make a valid comparison between the Expected and Actual throughput, cwnd stays fixed during the two consecutive RTTs.

In TCP Vegas retransmission phase, when a TCP receiver gets an acknowledgement (ACK), it keeps the time then by using the current time associated packet's timestamp calculates the estimated RTT. Based on following two

conditions, TCP Vegas decides whether to retransmit the packet. First, when it receives a duplicated ACK, Vegas compares current time and associated packet's timestamp and checks if its difference is greater than time out value. If it is greater than time out, the sender retransmits the packet and doesn't wait for the remaining duplicated ACKs. Second, after a retransmission if the first or the second ACK is received, Vegas again checks the elapsed time since the packet was sent to see if it's greater than time out value. If it's true Vegas retransmits the packet.

3. OBTAINING BEST MAXIMUM BURST SIZE AND BURSTIFICATION TIME

3.1 Obtaining best maximum burst size

In this section we explain how we choose the best maximum burst size. The goal of this section is to find the best maximum burst size so we can have the best throughput possible in this network. In figure 1 we show the topology that we use for our OBS network. This network uses the TCP Vegas to transfer the bursts.

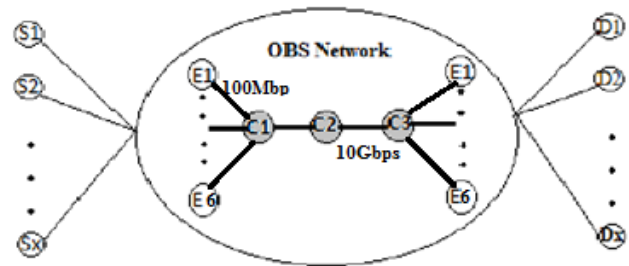


Figure 1. The network topology adopted in the simulation

As you can see in figure 1 this OBS networks consists of 12 edge nodes and 3 core nodes. The edge nodes are connected to core node with propagation delay of 10ms. The transfer rate of the links is 1Mbps and the links use 100 wavelength channels which means that each of the edge nodes are connected to the core node with a 10Gbps optical links and for transferring control packets each link has 8 wavelengths. The link that connects the core nodes to each other has 100 wavelength channels and transfer rate of each links is 100Mbps which mean the bandwidth of the connected core nodes is 10Gbps and for transferring the control packets links have 8 wavelength channels. The transfer rate of each wavelength is 1Mbps. Time out for burstification for making a burst is 1s.

In previous section we only adjusted burst size, now in this section we adjust burst creation time. In OBS network there is no buffer, due to this first we assemble the packets into a burst and then we send the burst through the network. So, we have two variables: 1) maximum burst size: As the name shows this variable determines maximum size of the burst. In assembling the packets when node reaches the maximum burst size threshold it sends the burst. For example, if the maximum burst size is 100Kb, node after assembling 100Kb packets, it turns them to a burst and sends it. 2) Burstification time: this variable determines that how long a node assembles the coming packets and then sends the burst. For example if burstification time is 5s node assembles the packets for 5 seconds and then turn them into a burst then sends the burst.

First we simulate the network only by adjusting the maximum burst size, then we only adjust burstification time and finally we combine them together and simulate the network. In each section we brought the result of the simulation and compared them.

In this section we only simulated the network by adjusting the maximum burst size. Because in this network we have contention, so if we make the size of our maximum burst size small, we send more bursts in the network and it leads to more contention in the network and more packets is dropped in the network. On the other hand if we make the size of maximum burst size large, we lose time in making each burst (because the size of each burst is large). Also, if our maximum burst size is large, in each contention we are going to lose more packets, and if our maximum burst size is small in each contention we

are going to lose lesser packets. Because of these factors, we need to choose the best maximum burst size for our network.

In our scheme for obtaining the maximum burst size we use a linear reduction equation (equation 2), in which Y_n denotes maximum burst size and we have:

$$Y_n = \frac{Y_{n-1}}{2} \quad (2)$$

In this section we only changed the maximum burst size and simulated and compared our results. We have run a lot of tests. In this way we can find the burst size with the best throughput and use it in the OBS network. Figure 2 shows the throughput of the network with different packet loss probability.

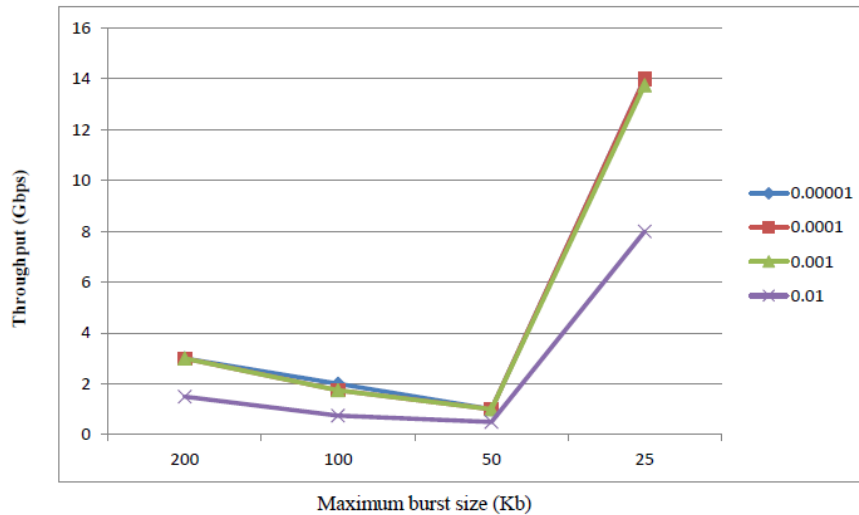


Figure 2. The effect of adjusting burst size on network’s performance with different packet loss probability

This figure demonstrates that in all the packet loss probability as we decrease the burst size the throughput of the network increases. The reason is in this scheme by decreasing the maximum burst size, we spend less time assembling packets in comparison with high maximum burst size. Also, with low maximum burst size when a contention occurs we lose less packets compare to high maximum burst size. So, low maximum burst size is appropriate for the network.

3.2 Obtaining best maximum burst size

In the previous section we only adjusted the burst size, now in this section we adjust the burstification time.

In his section we explain the method for decreasing the false contention detection and by doing that we reduce its effect on the network’s performance when TCP Vegas is deployed in the network.

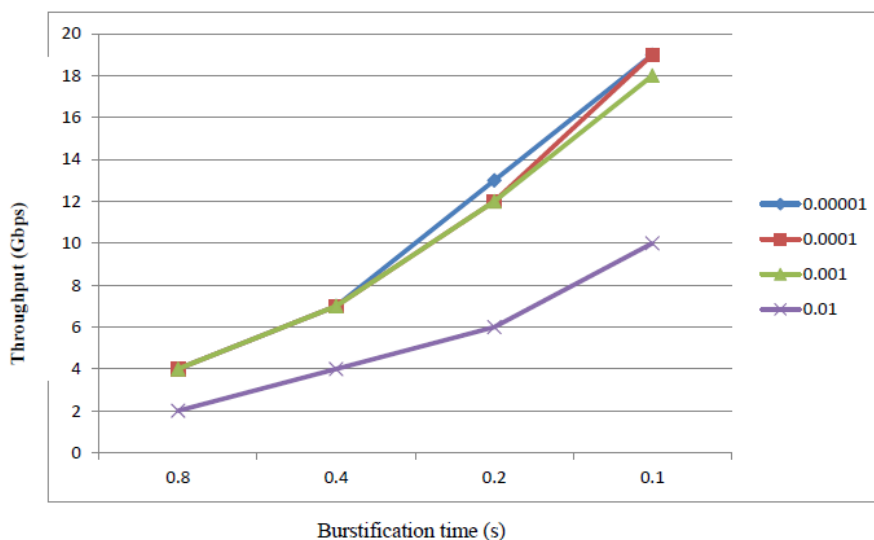


Figure 3. The effect of adjusting burstification time on network’s performance with different packet loss probability

As we explained in the previous sections, burstification time is a variable that determines how long a node should assemble the packets into a burst and when it reaches the threshold time should send the burst. Because there is contention in this network, if we set the burstification high, in every contention we lose more packets, as a result it reduces the performance. On the other hand, if we set the burstification time to be low, we send the packets more frequently and if a contention occurs we lose less packets compared to high burstification time.

In our scheme for obtaining the maximum burst size we use a linear reduction equation 3 (equation 3), in which X_n denotes maximum burst size:

$$X_n = \frac{X_{n-1}}{2} \quad (3)$$

In this section we only adjust the burstification time and simulated and compared the results. We have run a lot of tests. This way, we can find the best burstification time to obtain the best throughput and adopt it in the OBS network. Figure 3 shows the throughput of the network with different packet loss probability.

This figure demonstrates that in all the packet loss probability as we reduce the burstification time the throughput

of the network increases. The reason is in this scheme by decreasing the maximum burst size, we spend less time assembling packets and send the packets more frequently. Also, low burstification time means when a contention occurs we lose fewer packets compare to high burstification time. So, low burstification is a good choice for the network.

3.3 Obtaining best maximum burst size

In this section, we combine the two previously mentioned methods in the network and simulate the network and explain how it reduces false detection contention. This method's way of dealing with performance reduction of the network is to keep a balance between the maximum burst size and burstification time. Table 1 shows the values of maximum burst size and burstification time:

Table 1. Maximum burst size and burstification time

	1	2	3	4
Burstification time	0.1(s)	0.2(s)	0.4(s)	0.8(s)
Maximum Burst Size	200(kb)	100(kb)	50(kb)	25(kb)

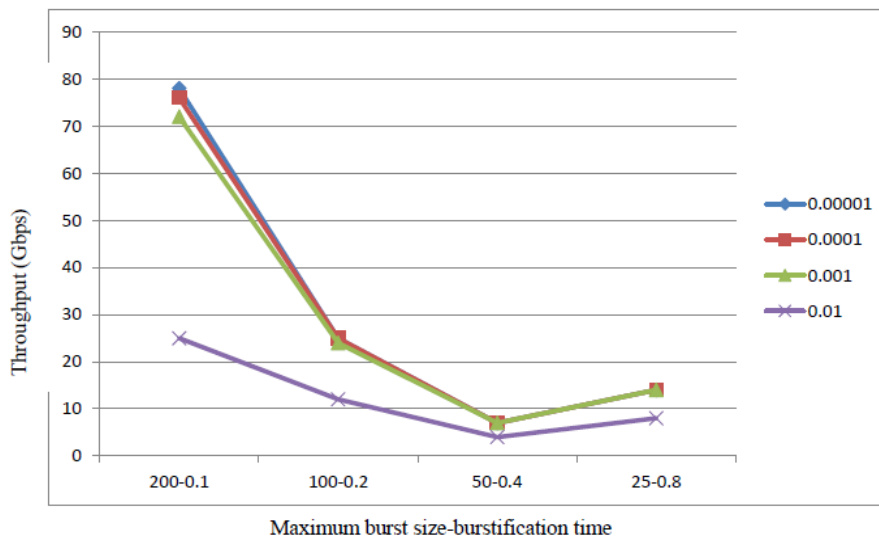


Figure 4. The effect our method on network's performance with different packet loss probability

There is a trade off in this method. If we increase maximum burst size, a reduction in burstification time can prevent the performance reduction.

We have run a lot of tests in the simulation and adjusted both the burstification time and maximum burst size and compared the results. By using this method, we can find the best burstification time and maximum burst size to obtain the best throughput and prevent network's performance reduction. Figure 4 shows the throughput of the network with different packet loss probability.

As you can see in the figure, if we balance the values of maximum burst size and burstification time we can improve the network's performance. This is because of the proper value of our variables at all the time, it's not too low and it's not too high, so when we have a contention in the network we don't lose a lot of packets and we always send a proper amount of packets.

4. COMPARING OUR SCHEME WITH ANOTHER METHOD

In this section we compare our scheme with another method. First we explain the method that we are comparing our scheme with. In the method in [21], a new implementation of TCP Vegas that adapts a mechanism based on threshold for controlling congestion of OBS network is introduced. In this paper, for controlling the congestion of the network, a base RTT (round trip time) is chosen. Then a threshold is set, if a specific number of bursts that exceed base RTT are more than the threshold, size of the window is reduced. This method is used to control the congestion of the network.

In figure 5 we compare our scheme we the method introduced in paper [21].

As you can see in the figure, our method has better throughput than compared scheme.

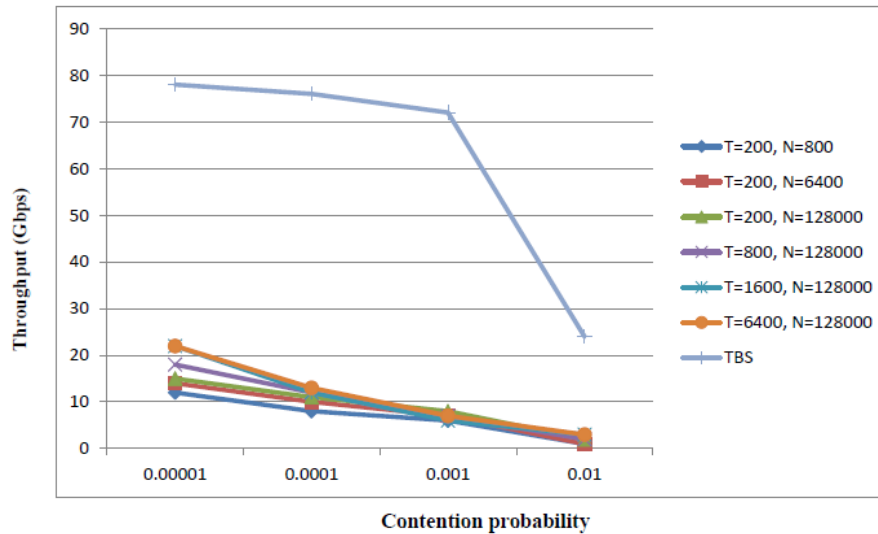


Figure 5. Comparing throughput of our method with scheme introduced in paper [21]

5. CONCLUSION

False congestion detection in OBS network is one of the major problems and it can reduce networks performance. In this paper we proposed a method based on maximum burst size and burstification time. This paper tried to make a balance between maximum burst size and burstification time in different contention probability to improve OBS network's performance and the results from NS-2 simulation proves our method.

REFERENCES

- [1] Floyd S. (2002). Quick-start for TCP and IP. Internet draft, draft-amit-quick-start-02.txt.
- [2] Jin C, Wei D, Low S. (2004). FAST TCP: motivation, architecture, algorithms, performance. In: Proceedings, IEEE Infocomm. Hong Kong, China.
- [3] Hegde S, et al. (2004). FAST TCP in high-speed networks: An experimental study. In: Proceedings. GridNets, San Jose, CA.
- [4] Xu L, Harfoush K, Rhee I. (2004). Binary increase congestion control (BIC) for fast long-distance networks. In: Proceedings, IEEE Infocomm. Hong Kong, China.
- [5] Stevens W. (1997). TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms. RFC 2001.
- [6] Mathis M, Mahdavi J, Floyd S, Romanow A. (1996). TCP selective acknowledgement options, RFC 2018.
- [7] Brakmo L, Peterson L. (1995). TCP Vegas: end-to-end congestion avoidance on a global internet, IEEE Journal on Selected Areas in Communications 13(8):1465-1480.
- [8] Katabi D, Handley M, Rohrs C. (2002). Congestion control for high bandwidth-delay product networks. In: Proceedings, ACM SIG-COMM. Pittsburgh, PA.
- [9] Jin C, Wei D, Low S. (2004). FAST TCP: Motivation, architecture, algorithms, performance, Infocom 2004. Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies 4.
- [10] Hegde S, et al. (2004). FAST TCP in high-speed networks: an experimental study. In: Proceedings. GridNets, Engineering & Applied Science, Caltech, the First International Workshop on Networks for Grid Applications.
- [11] Stevens W. (1997). TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, RFC.
- [12] Mathis M, Mahdavi J, Floyd S, Romanow A. (1996). TCP selective acknowledgement options, RFC.
- [13] Brakmo L, Peterson L. (1995). TCP Vegas: End-to-end congestion avoidance on a global internet. IEEE Journal on Selected Areas in Communication.
- [14] Katabi D, Handley M, Rohrs C. (2002). Congestion control for high bandwidth-delay product networks. ACM SIGCOMM Computer Communication, PA.
- [15] Óscar González de Dios, Ignacio de Miguel, Ramón J. Durán, Juan Carlos Aguado, Noemí Merayo, Patricia Fernández (2012). Impact of TCP synchronization on capacity dimensioning of Optical Burst Switched (OBS) links, Networks and Optical Communications (NOC).
- [16] Wang YW. (2002). Using TCP congestion control to improve the performance of Optical Switched Networks, Communications, 2003. ICC '03. IEEE International Conference 2.
- [17] Yu X, Qiao C, Liu Y. (2004). TCP implementations and false time out detection in OBS networks, Infocom.
- [18] Zhang Q, Vokkarane V, Wang Y, Jue JP. (2005). Analysis of TCP over optical burst-switched networks with burst retransmission. In: Proceedings, IEEE GLOBECOM, St. Louis, MO.
- [19] Zhang Q, Vokkarane V, Wang Y, Jue JP. (2005). Evaluation of burst retransmission in optical burst-switched networks. In: Proceedings, 2nd International Conference on Broadband Networks. Boston, MA. <https://doi.org/10.1109/ICBN.2005.1589624>.
- [20] Hsu C, Liu T, Huang N. (2002). Performance analysis of deflection routing in optical burst-switched networks, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE 1, New York, NY.
- [21] Shihada B, Zhang Q, Ho PH, Jue JP. (2010). A novel implementation of TCP vegas for optical burst switched networks. Optical Switching and Networking.
- [22] Yu X, Qiao C, Liu Y. (2004). TCP implementations and false time out detection in OBS networks. In: Proceedings, IEEE Infocomm, Hong Kong, China.

- [23] Mo J, La R, Anantharam V, Walrand J. (1999). Analysis and comparison of TCP Reno and Vegas. INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE 3.
- [24] Weigle E, Feng W. (2001). A case for TCP Vegas in high-performance computational grids. In: Proceedings, 10th IEEE International Symposium High Performance Distributed Computing, San Francisco. CA.
- [25] Shihada B, Ho PH, Hou F, Jiang XH, et al. (2006). BAIMD: A Responsive Rate Control for TCP over Optical Burst Switched (OBS) Networks, Communications, 2006. ICC '06. IEEE International Conference 6. <https://doi.org/10.1109/ICC.2006.255163>.
- [26] Peng SP, Li ZB, Wu XL, Xu AS. (2007). TCP window based dynamic assembly period in optical burst switching network, communications, 2007. ICC '07. IEEE International Conference on.
- [27] Raffaelli C, Zaffoni P. (2006). Simple analytical formulation of the TCP send rate in optical burst-switched networks, computers and communications, 2006. ISCC '06. Proceedings. 11th IEEE Symposium on.
- [28] Jayaraj A, Venkatesh T, Murthy CSR. (2008). Loss classification in optical burst switching networks using machine learning techniques: Improving the performance of TCP.
- [29] Shihada B, Zhang Q, Ho PH. (2006). Threshold-based TCP vegas over optical burst switched networks, computer communications and networks, 2006. ICCCN 2006. Proceedings. 15th International Conference.
- [30] Pleich R, Siemens AG, Munich, Germany, de Vega Rodrigo, M, Gotz J. (2005). Performance of TCP over optical burst switching networks, Optical Communication, 2005. ECOC 2005. 31st European Conference 4.
- [31] Zhang Q, Vokkarane VM, Wang YK, Jue JP. (2005). Analysis of TCP over optical burst-switched networks with burst retransmission. Global Telecommunications Conference. GLOBECOM '05. IEEE 4. <https://doi.org/10.1109/GLOCOM.2005.1578012>.
- [32] Bimal V, Venkatesh T, Murthy CSR. (2007). A markov chain model for tcp newreno over optical burst switching networks. Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE. <https://doi.org/10.1109/GLOCOM.2007.423>
- [33] Zhu L, Ansari N, Liu J. (2005). Throughput of high-speed TCP in optical burst switching networks. Communications. IEE Proceedings 152(3).