Effect of Number of Burst Assemblers on TCP Performance in Optical Burst Switching Networks

Guray Gurel and Ezhan Karasan
Department of Electrical and Electronics Engineering
Bilkent University, 06800 Ankara, Turkey
Email: {guray,ezhan}@ee.bilkent.edu.tr

Abstract— **Burst assembly mechanism is one of the fundamental factors that determine the performance of an optical burst switching (OBS) network. In this paper, we investigate the influence of number of burstifiers on TCP performance for an OBS network. An ns2-based OBS network simulator is developed for simulating the optical network. The goodput of TCP flows between an ingress and an egress nodes traveling through an optical network is studied for different values of the number of assembly buffers per destination. First, the losses resulting from the congestion in the core OBS network are modeled using a burst independent Bernoulli loss model. Then, a background burst traffic is generated to create contention at a core node in order to realize a burst dependent loss model. Simulation results show that for an OBS network employing timer-based assembly algorithm, TCP goodput increases as the number of burst assemblers is increased for both types of loss models. The improvement from one burstifier to moderate number of burst assemblers is significant (15-50% depending on the burst loss probability, processing delay and the TCP version), but the goodput difference between moderate number of buffers and *per-flow* aggregation is relatively small, implying that a cost-effective OBS edge switch implementation should use moderate number of assembly buffers per destination for enhanced TCP performance.**

I. Introduction

Increasing demand for services with very large bandwidth requirements, e.g., grid networks, facilitates the deployment of optical networking technologies [1]. Using Dense Wavelength Division Multiplexing (DWDM) technology, optical networks are able to meet the huge bandwidth requirements of future Internet Protocol (IP) backbones [2]. Currently, IP routers are interconnected with virtual circuits over synchronous optical networks (SONET) through multiprotocol label switching (MPLS) [3]. However, optical circuit switching (OCS) is not suitable for carrying bursty IP traffic with time-varying bandwidth demand. In addition, delays during connection establishment and release increase the latency especially for services with small holding times. An alternative to OCS is optical packet switching (OPS), which can adapt to changing traffic demands and requires no reservation, but the optical buffering and signal processing technologies have not matured enough for possible deployment of OPS in core networks in the near future. Optical burst switching (OBS) is proposed as a short-term feasible solution that can combine the strengths and avoid the shortcomings of OCS and OPS [4]. Fig. 1 shows a typical OBS network.

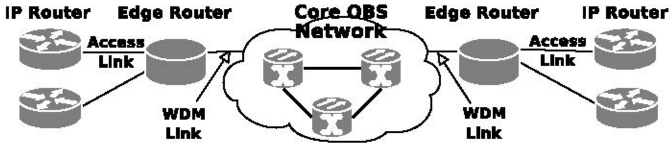

Fig. 1. An OBS network

In OBS, the IP packets reaching the edge router are aggregated into *bursts* before being transmitted in the optical core network. The assembly algorithm at the edge router keeps track of the size of the burst and the delay experienced by the first packet in the burst. A timer-based assembly algorithm creates a burst when the delay for the first packet reaches a timeout while a size-based algorithm creates a burst when the size of the burst reaches a threshold. A size/timer-based hybrid burstifier creates a burst when either of the size or time thresholds is reached. As far as TCP throughput is concerned, size-based burstification performs the worst, size/timer-based performs better and timer-based performs the best [5], [6]. The throughput degradation that results from the additional burst assembly delay, called *delay penalty* (DP) [5], increases as assembly time is increased [3], [5], [7], [8]. An important consequence of burst assembly is the combined loss or combined successful delivery of consecutive packets in a burst belonging to the same TCP flow. The improvement in TCP rate as a result of this correlation is called *correlation benefit*, and the correlation gain increases with the average number of packets in a burst belonging to the same TCP flow [9]. This improvement is explained by the increased time between two loss events, and it is referred to as the *delayed first loss* (DFL) gain [5]. The average number of the packets in a burst belonging to a particular flow depends on the access network bandwidth and assembly timeout [5].

Performance improvement in OPS networks with larger optical packets is noted in [10]. It is seen that the improvement of larger burst size on throughput gets more significant as drop probability is decreased [8]. On the other hand, increasing the burst size leads to performance deterioration as assembly delay becomes dominant [3]. It is also shown that the TCP performance degrades with aggregation as a result of the synchronization between TCP flows sharing an aggregation buffer [11]. This synchronization results from simultaneous decrease of congestion window sizes of TCP flows that have

1-4244-0425-8/06/$20.00 ©2006 IEEE

packets in a lost burst.

Another effect of burst size on the loss performance is due to the voids formed between consecutive bursts [12]. If the burst control packets arriving to a switch have different residual offset times, some bursts are scheduled into voids formed between two reservations that have been made earlier. As a result of this, bursts with smaller sizes can be fit into these voids more easily resulting in a smaller loss probability for small-sized bursts. This burst length dependent losses do not occur if all bursts arriving at a switch have the same residual offset times, e.g., when they are all destined for the same egress node.

TCP flows are classified as *slow* when only one of their packets are found in a given burst, *fast* when their whole congestion window is found in the burst and *medium* otherwise [9].

In this paper, we focus on the effect of number of assembly buffers on TCP throughput. We consider two loss models. First, we study the case when the burst losses are burst-size independent. Then, we consider the case of burst-size dependent losses. We use an ns2-based [13] OBS network simulator (n-OBS) [14] for studying the performance of several TCP implementations as the number of burstifiers is changed. We show for both loss models that the TCP goodput increases significantly as the number of assembly buffers per destination is increased since the effect of flow synchronization is reduced. This improvement saturates as the number of burst assemblers is increased further, e.g., when per-flow aggregation is used. A cost-effective ingress node implementation should use moderate number of buffers per destination for enhanced TCP performance. For the burst-length dependent loss case, we show that the TCP goodput increase is larger with per-flow aggregation for TCP flows having smaller residual offset times.

The organization of the paper is as follows: in section II, the network model used in the simulations is presented. The effects of the number of burstifiers are discussed for the burst independent loss model in Section III and for the burst-length dependent loss model in Section IV. The conclusions of the study are presented in Section V.

II. Network Model

Figure 2 shows the model used for the ingress node. The burstifier queues shown are kept per-egress, and there is a group of M assembly buffers generating bursts destined for the same egress node. For simplicity, the burstifier queues for a single egress are shown in Figure 2. When multiple destinations are possible, the burstifier queue block containing M burstifiers should be placed for each egress node. The number of burstifiers, $M \leq N$, is chosen amongst divisors of N to allow simple mapping. Burstifiers use FIFO buffers to aggregate packets. When a burst is generated by any burstifier, the burst is sent to the nodal burst scheduler. The scheduler keeps track of the schedule on the wavelengths of the output WDM links. If scheduler is able to find a suitable interval on an available wavelength over the first link of the route for this burst, the burst waits in the electronic burst queue until the reservation interval. The burst queue is necessary in order to avoid contention between bursts coming from different burstifiers.

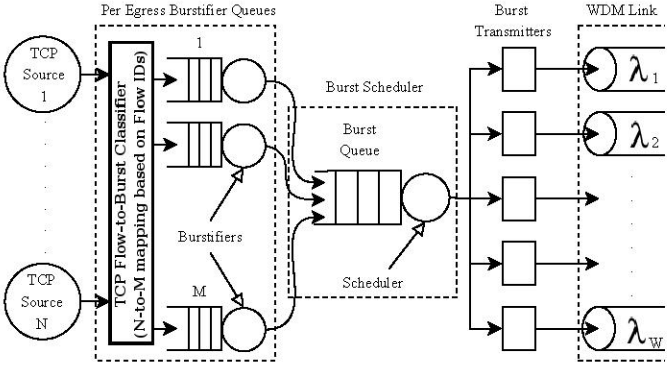

Fig. 2. Ingress node model

The network topology used for studying the effects of burst assembly on the performance of OBS networks is shown in Figure 3 for the burst independent loss model. Burst length independent loss model is valid for the cases when all bursts arriving to a switch have the same residual offset times or the nodal processing delay is negligible with respect to the burst size. In this case, the core optical network is modeled as a single fiber with Bernoulli distributed drop probability p in $O_1 \rightarrow O_2$ direction to account for losses due to contentions in the core network. The optical link in $O_2 \rightarrow O_1$ direction and access links are lossless. On the reverse path, ACK packets do not experience any drops or assembly delays. B_a, T_{pa}, B_o and T_{po} denote the access link bandwidth, access link delay, optical link bandwidth and delay, respectively. Each source node s_i employs an infinite FTP flow to the respective destination $d_i, 1 \leq i \leq N$.

III. Effect of Assembly Buffering for Burst Independent Losses

In order to study how size/timer-based algorithm affects the goodputs of TCP flows, it is evaluated using different values for the assembly timeout and burst size threshold. Given access bandwidth, B_a, the maximum burst size, B_M, depends on the assembly time, T_B, with the following relation [9]:

$$\frac{B_M}{B_a} \leq T_B \qquad (1)$$

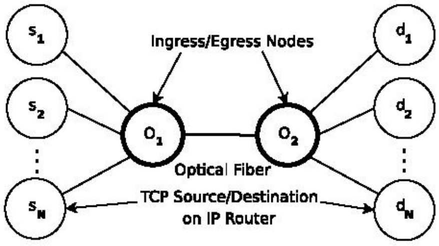

Fig. 3. Topology used in simulations

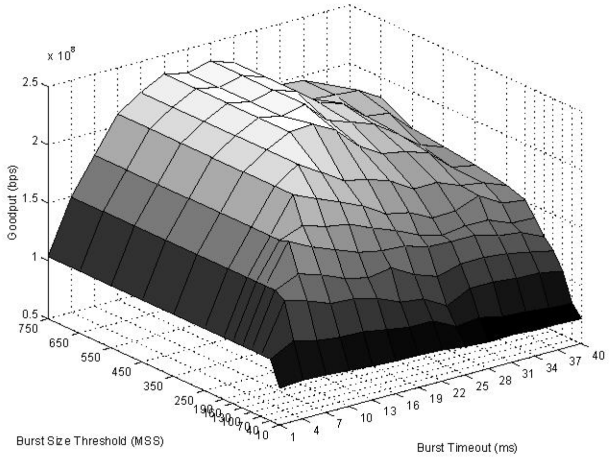

Fig. 4. Total goodput for $N = 10$, $M = 10$ and $p = 0.01$ using TCP Newreno

Similarly, equality in (1) yields the minimum time, T_B, needed to assemble a burst of size B_M. In other words, for those set of parameters where assembly time threshold is smaller than this minimum time, the size/timer-based burstifier operates as timer-based burstifier.

Figure 4 shows the performance of size/timer-based assembly algorithm over a range of burst size thresholds and assembly time thresholds (timeouts). The parameters used are $N = 10$, $M = 10$, $p = 0.01$, $B_a = 155Mbps$, $T_{pa} = 1msec$, $B_o = 1Gbps$ and $T_{po} = 10msec$. The MSS of TCP sources are set to 1040 Bytes and the receive windows are set to 10000 MSS. The employed TCP version is Newreno.

For a fixed timeout, it is observed that the goodput increases as the size threshold is increased until the maximum achievable burst size corresponding to the timeout is reached. Increasing the burst size threshold further has no effect on goodput since the assembly algorithm acts as timer-based for larger size thresholds.

For a fixed burst size threshold, the goodput increases as the burstification timeout is increased for small timeouts. This is the region where DFL gain dominates the effects of DP and increasing timeouts yield better goodputs. However, once the minimum assembly time corresponding to current size threshold is exceeded, DFL gain stays constant while the effects of DP begin to dominate and consequently the goodput decreases.

The largest burst size threshold in Figure 4 is larger than the maximum achievable burst size on the given timeout range. Therefore, for the largest burst size threshold, the hybrid burstifier mechanism acts as a timer-based algorithm for all timeouts. On the other hand, since size/timer-based algorithm reduces to size-based algorithm for infinite timeout, the performance at the largest timeout reflects the performance of the size-based algorithm.

Since the highest goodput is obtained by the timer-based algorithm, we resort to the timer-based burstification in the rest of the paper for studying the effect of the number of the burstifiers on TCP performance. Figures 5-10 show the performance of the timer-based algoritm for $p = 0.001$ and $p = 0.01$ for TCP Reno, Newreno and Sack. The goodput values for $M =$ 1,2,5 and 10 are plotted together for comparison. We observe that increasing the number of burst assemblers significantly improves the goodput for all three TCP versions since synchronization between large number of TCP flows is avoided as the number of burstifiers is increased. When a burst is lost in the optical core, all the sources that have packets in that burst decrease their congestion windows. In other words, flows sharing an aggregation buffer becomes synchronized. In the full-aggregation case, i.e., $M = 1$, all flows $1 - N$ are synchronized and hence the optical link is underutilized. When the degree of synchronization is reduced by increasing the number of burstifiers, the congestion windows of flows belonging to different burst assemblers tend to balance each other and the link is better utilized.

The plots also show that as the assembly time is increased, goodput first increases, then starts to decrease for all three TCP versions. In the region where goodput increases with timeout, the delay penalty is small and DFL gain is dominant, therefore increasing the burst size increases the goodput. On the other hand, the improvement provided by DFL gain saturates after some timeout value and the delay penalty begins to dominate, which causes the goodput to deteriorate.

Another important observation is that the rate of decrease in goodput as the timeout is increased depends on loss probability p. When p is large, the congestion window cannot increase to large values due to more frequent burst losses. In this case, the increase in the timeout does not increase the burst size significantly and the increase in DFL gain with increasing timeout is not significant. As a result, the goodput decreases more rapidly with increasing timeout due to the delay penalty. On the other hand, larger bursts are generated as the timeout is increased when p is small, and the DFL gain increases with the timeout. This partially compensates the effect of the delay penalty, and the goodput does not degrade much with the increasing throughput for all three TCP versions. In addition, it is observed that a relatively low number of buffers may perform close to the per-flow aggregation case. Since the cost of additional burstifiers can be compromised by the improvement in goodput, employing moderate number of buffers with respect to the number of flows constitutes a cost-effective solution.

Although all three TCP versions exhibit similar characteristics as the timeout and the number of burstifiers are changed, TCP Sack achieves the highest goodput among the three TCP versions. Sack outperforms the other two versions since it quickly retransmits the lost segments with selective acknowledgements. Reno and Newreno have very close performances, however Newreno slightly outperforms Reno.

In Table I, the goodput enhancement of using multiple burstifiers with respect to the single burstifier case, i.e., per destination burstification, is shown for different TCP versions, number of TCP flows and loss probability. For $N = 10$ and $p = 0.001$, the goodput with per-flow burstification increases 33-65% compared to the case with per-destination burstifica-

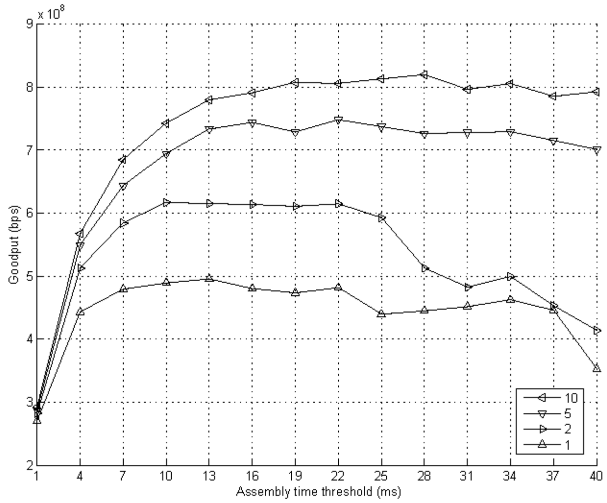

Fig. 5. Total goodput with timer-based assembly for $N = 10$, $p = 0.001$, $M = 1, 2, 5, 10$ and Reno TCP

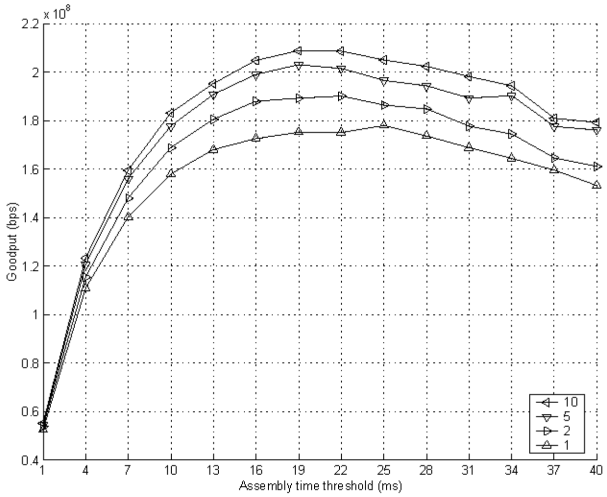

Fig. 6. Total goodput with timer-based assembly for $N = 10$, $p = 0.01$, $M = 1, 2, 5, 10$ and Reno TCP

tion for different TCP versions. The goodput enhancement is largest with Reno and smallest with Sack. We also observe that the goodput achieved with $M = 5$ is very close to the per-flow burstification case. For $N = 10$ and $p = 0.01$, the goodput enhancement with per-flow burstification with respect to per-destination burstification is about 15-20%. Similarly, the goodput achieved with $M = 5$ is very close to the per-flow burstification case. The burstification architecture at the edge router should be designed taking into account both the goodput enhancement and additional management complexity of using multiple burstifiers, and $M = 5$ seems to provide a nice compromise for this case.

IV. Burst Length Dependent Losses

The burst-length dependent losses naturally occur at a switch where arriving bursts have different residual offset times. This dependence is strongest for the flows having

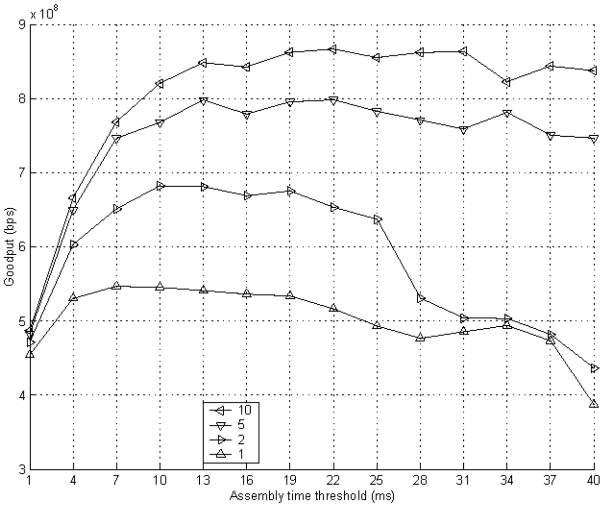

Fig. 7. Total goodput with timer-based assembly for $N = 10$, $p = 0.001$, $M = 1, 2, 5, 10$ and Newreno TCP

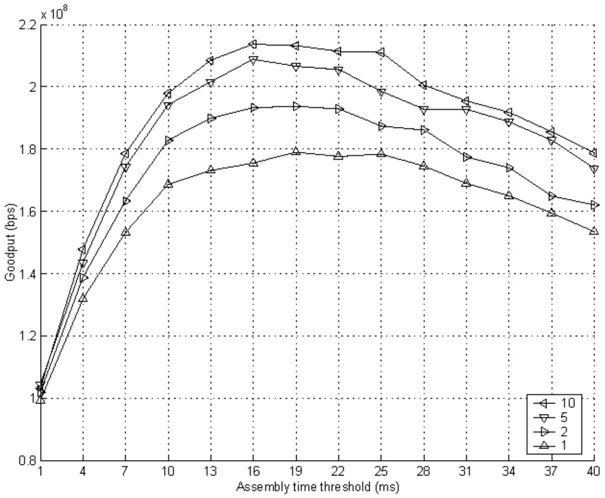

Fig. 8. Total goodput with timer-based assembly for $N = 10$, $p = 0.01$, $M = 1, 2, 5, 10$ and Newreno TCP

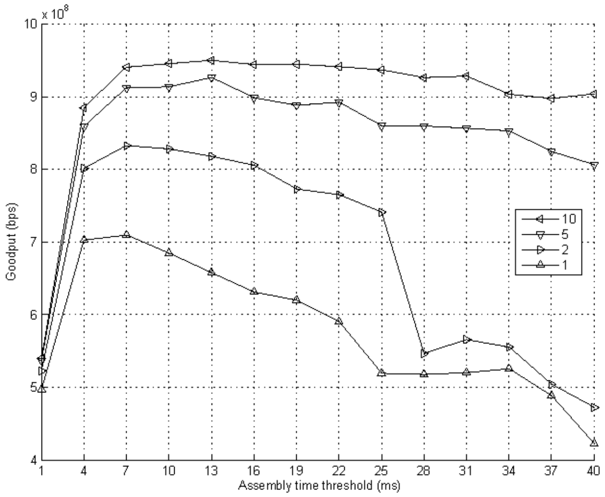

Fig. 9. Total goodput with timer-based assembly for $N = 10$, $p = 0.001$, $M = 1, 2, 5, 10$ and Sack TCP

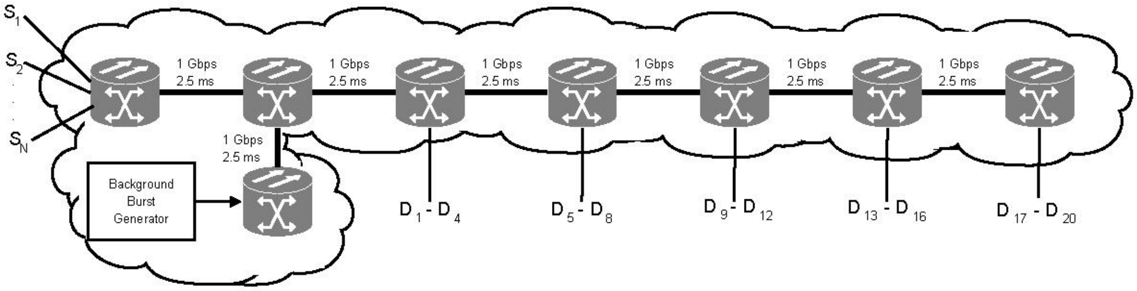

Fig. 11. Topology used in simulations

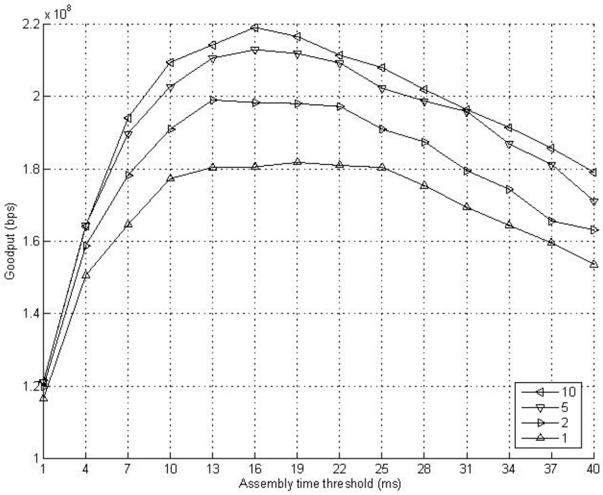

Fig. 10. Total goodput with timer-based assembly for $N = 10$, $p = 0.01$, $M = 1, 2, 5, 10$ and Sack TCP

TABLE I

PERCENTAGE GOODPUT INCREASE VERSUS NUMBER OF BURSTIFIERS FOR DIFFERENT TCP VERSIONS AND LOSS PROBABILITY

p	M	RENO	NEWRENO	SACK
0.001	2	24.55	24.77	17.31
	5	51.00	45.99	30.50
	10	65.40	58.48	33.84
0.01	2	6.85	8.22	9.48
	5	14.10	16.63	17.16
	10	15.20	19.36	20.52

smaller residual offset times, and the dependence becomes weaker for flows having larger residual offset times. For the flow with the largest residual offset time, the burst losses occur independent of their sizes.

Figure 11 shows the network topology used for studying the effects of burst length dependent losses. Sources $S_1 - S_N$ employ an infinite FTP flow to the respective destination $D_1 - D_N$ (N=20). Optical links have 1 Gbps bandwidth and 2.5 msec propagation delay. The background burst generator produces bursts whose sizes are exponentially distributed with $1/\mu$ and burst arrivals are Poisson with rate λ. All bursts are destined uniformly to the five egress nodes connected to $D_1 - D_{20}$. Access links have 50 Mbps bandwidth and 1 msec propagation delay.

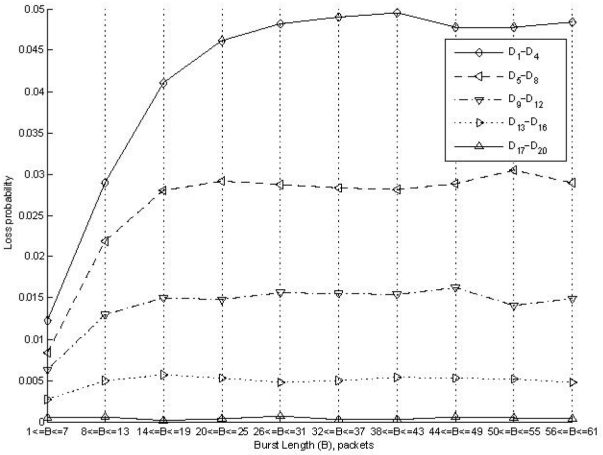

Fig. 12. Loss probability vs. burst length for different egress nodes

Figure 12 shows the loss probability for each egress node as a function of the burst length with the parameters $1/\mu = 200\mu$ sec, $1/\lambda = 2msec$, $M = 1$, the nodal processing delay $\Delta = 50\mu sec$ and the assembly timeout $T = 10msec$. The statistics of the generated bursts is grouped into 10 bins according to the number of packets in the burst, which ranges from 1 to a maximum value of 60 packets. It can be seen that the loss probability is relatively high for the flows with smaller residual offset times, as expected. Moreover, the loss probability increases as the burst size increases. The impact of void filling mechanism in the core router scheduler becomes important for those bursts that are closer to their destinations because they need to fit in the voids created beforehand by the bursts that have larger residual offset times. Consequently, the dependence of the loss probability on the burst size is strongest for the bursts destined to $D_1 - D_4$. Such a correlation is not observed for the bursts destined to $D_{17} - D_{20}$.

In addition to the mechanisms mentioned in [5] such as DP, the loss penalty and correlation gain, this observation brings forward another critical factor in analysis of TCP performance in OBS networks. The significance of the burst length dependent losses depends on the residual offset time, per-hop processing delay (Δ) and the burst transmission time.

Figures 13 and 14 plot the goodput and the average burst size as a function of the burst assembly timeout for the nearest and farthest egress nodes, respectively, and for different values of the number of burstification buffers, M, using the parameters $1/\mu = 200\mu$ sec, $1/\lambda = 2msec$, $\Delta = 50\mu sec$, when TCP Reno is used. We observe that for both destinations the average goodputs increase with the number of burstifiers. It is also observed that the average burst size increases linearly with the assembly timeout for flows destined to $D_{17} - D_{20}$. On the other hand, the average burst size first increases and then saturates for the flows destined to $D_1 - D_4$. This is due to the fact that the TCP flows destined to $D_1 - D_4$ experience much more frequent burst losses and consequently they do not achieve very large congestion windows. The saturation of the average burst sizes coupled with the additional assembly delay cause the drop in the average goodput for flows destined for $D_1 - D_4$ as the assembly timeout increases. On the other hand, the TCP flows destined for $D_{17} - D_{20}$ can achieve very large congestion windows and the resulting burst sizes increase with the assembly timeout. The correlation benefit achieved by having longer bursts is partially compensated by the delay penalty, and the average TCP goodput does significantly change as the burst assembly timeout is increased.

We observe from Figures 13 and 14 that the flows destined for $D_{17} - D_{20}$ achieve much higher goodput compared with the flows destined for $D_1 - D_4$. Although the flows destined for $D_{17} - D_{20}$ experience larger delays, their much smaller loss probability results in higher goodput.

The comparison of Figures 13 and 14 also reveal that the maximum goodput for the flows destined for $D_1 - D_4$ are achieved at smaller values of the burst assembly timeout compared with the flows destined for $D_{17} - D_{20}$. In fact, the maximum goodput is achieved before the burst size saturates for the flows destined for $D_1 - D_4$. This is due to the fact that the loss probability increases significantly as the burst size increases for the flows destined for $D_1 - D_4$ as it was shown in Figure 12. Although the correlation gain is increasing with the burst size, the burst length dependent nature of the burst losses causes the average goodput to start decreasing before the average burst size reaches its maximum. A similar behavior is not observed in Figure 14 since the burst losses is independent of the burst size for the flows destined for $D_{17} - D_{20}$.

The performance improvement in the maximum average goodputs achieved by using $M = 2$ and $M = 4$ with respect to the case of $M = 1$ for TCP Reno and TCP Sack are shown in Tables II and III, respectively. The results show that the improvement in the average goodput is maximum for the flows destined for closer egress nodes, and the average goodput improvement generally increases with the increasing nodal processing delay Δ. The improvements are in the range of 17-35% for the closest nodes and the average goodput improvement over all destinations is 6-37% for TCP Reno and $M = 4$. For the case of $M = 2$, the average goodput increases are in the range of 3-10% compared to $M = 1$. The performance improvements for TCP Sack are slightly larger compared to TCP Reno.

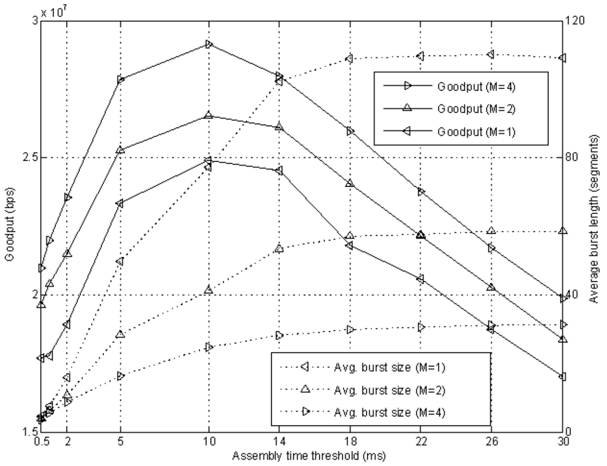

Fig. 13. Goodput and average burst size vs assembly time threshold for egress node 3

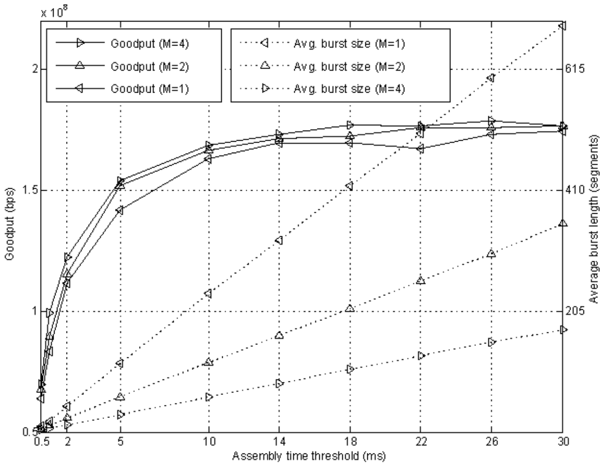

Fig. 14. Goodput and average burst size vs assembly time threshold for egress node 7

V. Conclusion

The performance of TCP over OBS networks is studied in this paper in terms of the number of burstifiers used at the edge routers. Increasing the number of burst assemblers per destination reduces the negative effects of synchronization between TCP flows occuring as a result of lost bursts containing packets belonging to multiple TCP flows. We show that TCP goodput is increased significantly when edge routers with multiple burstifiers per destination are used, and the goodput increases as the number of burstifiers increase. This conclusion holds for different TCP versions and different burst loss models. We argue that the edge router architecture can be designed with less number of burst assemblers than the per-destination burstification in order to reduce the complexity of managing large number of buffers while achieving nearly the maximum goodput.

TABLE II

PERCENTAGE GOODPUT INCREASE AS A FUNCTION OF THE NUMBER OF BURSTIFIERS FOR TCP RENO

		Destination					
Δ (μsec)	M	1-4	5-8	9-12	13-16	17-20	Avg.
50	4	16.91	8.15	6.86	2.91	2.43	6.22
50	2	6.47	4.22	4.19	3.28	1.36	3.87
100	4	34.82	26.83	8.61	6.21	1.89	6.91
100	2	13.91	7.78	2.85	4.91	0.82	2.69
200	4	26.78	35.79	31.73	6.70	15.52	23.15
200	2	13.86	14.86	12.36	4.31	6.01	6.70
500	4	26.49	27.83	31.22	34.97	15.95	36.92
500	2	13.36	10.94	14.53	16.76	3.27	10.24

TABLE III

PERCENTAGE GOODPUT INCREASE AS A FUNCTION OF THE NUMBER OF BURSTIFIERS FOR TCP SACK

		Destination					
Δ (μsec)	M	1-4	5-8	9-12	13-16	17-20	Avg.
50	4	39.41	8.47	8.79	5.43	0.38	4.91
50	2	19.72	4.76	3.73	3.15	0.04	3.03
100	4	48.81	54.93	13.05	10.35	0.62	6.33
100	2	26.21	25.25	6.09	8.68	0.46	2.72
200	4	44.79	57.58	45.30	6.91	0.46	24.45
200	2	25.43	25.01	26.07	4.74	0.00	4.35
500	4	47.83	38.83	48.91	54.20	1.29	37.88
500	2	24.76	17.81	25.86	25.44	0.73	8.07

ACKNOWLEDGMENT

This work is partially supported by FP6 IST e-Photon/ONe NoE project and by the Scienctific and Technological Research Council of Turkey (TÜBİTAK) under project 104E047.

REFERENCES

[1] M. Listanti, V. Eramo, R. Sabella, *Architectural and technological issues for future optical Internet networks*, IEEE Communications Magazine, Vol. 38, No. 9, September 2000, pp. 82-92.

[2] J. Turner, *Terabit burst switching*, J. High Speed Networks, Vol. 8, pp. 3-16, 1999.

[3] S. Yao, F. Xue, B. Mukherjee, S. J. B. Yoo, S. Dixit, *Electrical ingress buffering and traffic aggregation for optical packet switching and their effect on TCP-level performance in optical mesh networks*, IEEE Communications Magazine 40 (9) (2002) 66-72.

[4] C. Qiao and M. Yoo, *Optical burst switching (OBS) - A new paradigm for an optical internet*, J. High Speed Networks, Vol. 8, pp. 69-84, 1999.

[5] X. Yu, J. Li, X. Cao, Y. Chen and C. Qiao, *Traffic statistics and performance evaluation in optical burst switched networks*, IEEE/OSA Journal of Lightwave Technology, Vol. 22(12), pp. 2722- 2738, Dec. 2004.

[6] X. Cao, J. Li, Y. Chen, C. Qiao, *Assembling TCP/IP packets in optical burst switched networks*, Proc. IEEE GLOBECOM'02, November 2002, pp. 2808-2812.

[7] X. Yu, C. Qiao, Y. Liu, and D. Towsley, *Performance evaluation of TCP implementations in OBS networks*, Tech. Rep. 2003-13, CSE Dept., SUNY, Buffalo, NY, 2003.

[8] S. Gowda, R.K. Shenai, K.M. Sivalingam, H.C. Cankaya, *Performance evaluation of TCP over optical burst-switched (OBS) WDM networks*, Proc. IEEE ICC'03, May 2003, pp. 1433-1437.

[9] A. Detti, M. Listanti, *Impact of segments aggregation on TCP Reno flows in optical burst switching networks*, Proc. IEEE INFOCOM'02, 2002, pp. 1803-1812.

[10] Jingyi He, S.-H. Gary Chan, *TCP and UDP performance for Internet over optical packet-switched networks*, Computer Networks 45(4): 505-521 (2004).

[11] D. Hong, France, F. Poppe, J. Reynier, F. Baccelli, G. Petit, *The impact of burstification on TCP throughput in optical burst switching networks*, ITC-18, Sep. 2003, Berlin.

[12] K. Dolzer, C. Gauger, J. Späth, and S. Bodamer, *Evaluation of reservation mechanisms for optical burst switching*, AEÜ International Journal of Electronics and Communications, vol. 55(1), pp.18-26, January 2001.

[13] *Network Simulator 2*, developed by L. Berkeley Network Laboratory and University of California Berkeley, http://www.isi.edu/nsnam/ns

[14] G. Gurel, O. Alparslan and E. Karasan, *nOBS : an ns2 based simulation tool for TCP performance evaluation in OBS networks*, Proc. Simulation Tools for Research and Education in Optical Networks, Oct. 2005, France.