

Using multiple per egress burstifiers for enhanced TCP performance in OBS networks

Guray Gurel · Ezhan Karasan

Received: 27 November 2007 / Accepted: 18 July 2008 / Published online: 11 August 2008
© Springer Science+Business Media, LLC 2008

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 the number of burstifiers on TCP performance for an OBS network. The goodput of TCP flows between an ingress node and an egress node traveling through an optical network is studied as the number of assembly buffers per destination varies. First, the burst-length independent losses resulting from the contention in the core OBS network using a non-void-filling burst scheduling algorithm, e.g., Horizon, are studied. Then, burst-length dependent losses arising as a result of void-filling scheduling algorithms, e.g., LAUC-VF, are studied for two different TCP flow models: FTP-type long-lived flows and variable size short-lived flows. Simulation results show that for both types of scheduling algorithms, both types of TCP flow models, and different TCP versions (Reno, Newreno and Sack), TCP goodput increases as the number of burst assemblers per egress node is increased for an OBS network employing timer-based assembly algorithm. The improvement from one burstifier to moderate number of burst assemblers is significant (15–50% depending on the burst loss probability, per-hop processing delay, and the TCP version), but the goodput difference between moderate number of buffers and per-flow aggregation is relatively small, implying that an OBS edge

switch should use moderate number of assembly buffers per destination for enhanced TCP performance without substantially increasing the hardware complexity.

Keywords Optical burst switching · Transport control protocol · Burstification

1 Introduction

Increasing demand for services with very large bandwidth requirements 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. Currently, IP routers are interconnected with virtual circuits over synchronous optical networks (SONET) through multiprotocol label switching (MPLS) [2]. However, a wavelength routed network, also called optical circuit switching (OCS), is not suitable for carrying bursty IP traffic with time-varying bandwidth demand since a whole wavelength is the smallest bandwidth unit. 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) [3], 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 [4,5].

In OBS, 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

A preliminary version of this paper was presented at the Sixth International Workshop on Optical Burst/Packet Switching (WOBS), Oct. 2006.

G. Gurel · E. Karasan (✉)
Department of Electrical and Electronics Engineering, Bilkent
University, 06800 Ankara, Turkey
e-mail: ezhan@ee.bilkent.edu.tr

G. Gurel
e-mail: guray@ee.bilkent.edu.tr

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 [6, 7].

Performance of TCP traffic in OBS networks has been recently addressed in several studies [2, 6–13]. TCP throughput degradation resulting from the additional burst assembly delay, called the *delay penalty* (DP) [6], increases as the assembly time increases [2, 6, 8, 9]. An important consequence of the 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 the *correlation gain*, which increases with the average number of packets in a burst belonging to the same TCP flow [10]. This improvement is explained by the increased time elapsing between two loss events, and it is referred to as the *delayed first loss* (DFL) gain [6]. Meanwhile, the average number of packets in a burst belonging to a particular flow depends on the access network bandwidth and the assembly timeout.

Performance improvement in OPS networks with larger optical packets is noted in [11]. It is observed that the improvement of larger burst size on throughput gets more significant as the drop probability is decreased [9]. On the other hand, increasing the burst size leads to performance deterioration as the assembly delay becomes dominant [2, 12]. It is also shown that TCP performance degrades with aggregation as a result of the synchronization between TCP flows sharing the same aggregation buffer [11, 13]. This synchronization results from simultaneous decrease of the congestion window sizes of TCP flows that have packets in a lost burst.

Another effect of the burst size on the loss performance is due to the voids formed between consecutive bursts [14]. If the burst control packets arriving to a switch have different residual offset times and a void-filling type burst scheduling algorithm such as LAUC-VF [15] is used, some bursts are scheduled into voids formed between two reservations that have been made earlier. As a result, bursts with smaller sizes can be more easily fit into these voids resulting in reduced loss probability for small-sized bursts. 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, or a non-void-filling type burst scheduling algorithm such as Horizon [5] is used.

In this paper, we focus on the effect of the 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, which is further extended to burst-

size dependent losses. We use an ns2-based [16] OBS network simulator (n-OBS) [17] for studying the performance of several TCP implementations as the number of burstifiers is changed. We show for both loss models that 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. For the burst-length dependent loss case, we show that the TCP goodput increase with per-flow aggregation is significantly larger for TCP flows having smaller residual offset times.

The organization of the paper is as follows: in Sect. 2, the ingress node model used in this study is presented. The effects of the number of burstifiers are discussed for the burst independent loss model in Sect. 3 and for the burst-length dependent loss model in Sect. 4. The conclusions of the study are presented in Sect. 5.

2 Ingress node model

The ingress node model used in this paper is shown in Fig. 1. 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 Fig. 1. When multiple destinations are considered, a burstifier queue block containing M burstifiers should be used for each egress node. Burstifiers are in the form of FIFO buffers to aggregate IP packets into optical bursts. The number of burstifiers per egress, $M \leq N$, is chosen amongst divisors of N to allow balanced mapping of TCP flows to the burstifiers. 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 each wavelength of the output WDM links. If the 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.

3 Effect of multiple assembly buffers with non-void-filling scheduling

The network topology used for studying the effects of burst assembly on the performance of OBS networks is shown in Fig. 2 for the burst-length independent loss model. The burst-length independent loss model is valid for the cases when all bursts arriving to a switch have the same residual offset times, or a non-void-filling burst scheduling algorithm is used. We simply model the core optical network as a single fiber link

Fig. 1 Ingress node model

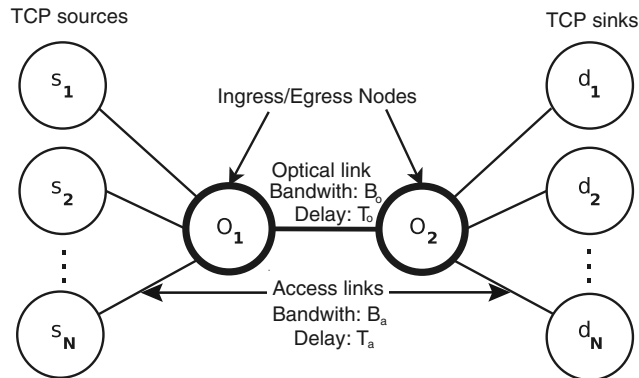
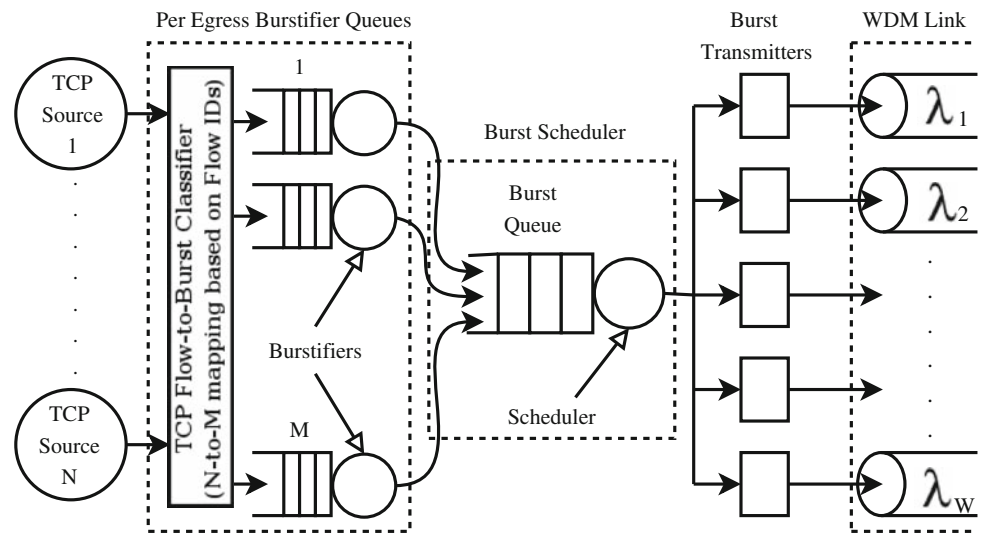


Fig. 2 Topology used for non-void-filling scheduling

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. Let B_a , T_a , B_o , and T_o denote the access link bandwidth, access link delay, optical link bandwidth, and optical link delay, respectively. We assume that each TCP source s_i employs an infinite FTP flow to the respective destination d_i , $1 \leq i \leq N$.

The parameters used in the simulations are $N = 10$, $M = 10$, $B_a = 155$ Mbps, $T_a = 1$ ms, $B_o = 1$ Gbps, and $T_o = 10$ ms. The MSS of TCP connections are set to 1,040 Bytes and the receive windows are set to 10,000 MSS.

Since the highest goodput is obtained by the timer-based algorithm, we resort to the timer-based burstification in this paper for studying the effect of the number of the burstifiers on TCP performance [6,7]. Figures 3–8 show the total goodput for $p = 0.001$ and $p = 0.01$ for TCP Reno, Newreno, and Sack, respectively, for $M = 1, 2, 5$, and 10. We

observe that increasing the number of burst assemblers significantly improves the goodput for all three TCP versions since synchronization between large numbers 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 simultaneously decrease their congestion windows. In other words, flows sharing an aggregation buffer become synchronized. In the full-aggregation case, i.e., $M = 1$, all flows $1 - N$ are synchronized and hence the optical link is underutilized [11,13]. 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 evolutions of the congestion windows of $N = 10$ TCP Reno flows are shown in Fig. 9 for $M = 1$ and $M = 10$. TCP flow synchronization effect in OBS networks is clearly observed in Fig. 9a where $M = 1$. With per-egress burstification, TCP flows are synchronized resulting in a significant drop in channel utilization after each burst loss. On the other hand, with per-flow burstification, i.e., $M = 10$, no flows are synchronized and the sum of the congestion windows is very smooth corresponding to a better bandwidth utilization as shown in Fig. 9b.

The plots also show that as the assembly time is increased, the goodput first increases, and then starts to decrease for all three TCP versions. In the region where the goodput increases with the assembly 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 the loss

Fig. 3 Total goodput with timer-based assembly for $N = 10$, $p = 0.001$, $M = 1, 2, 5, 10$ and Reno TCP

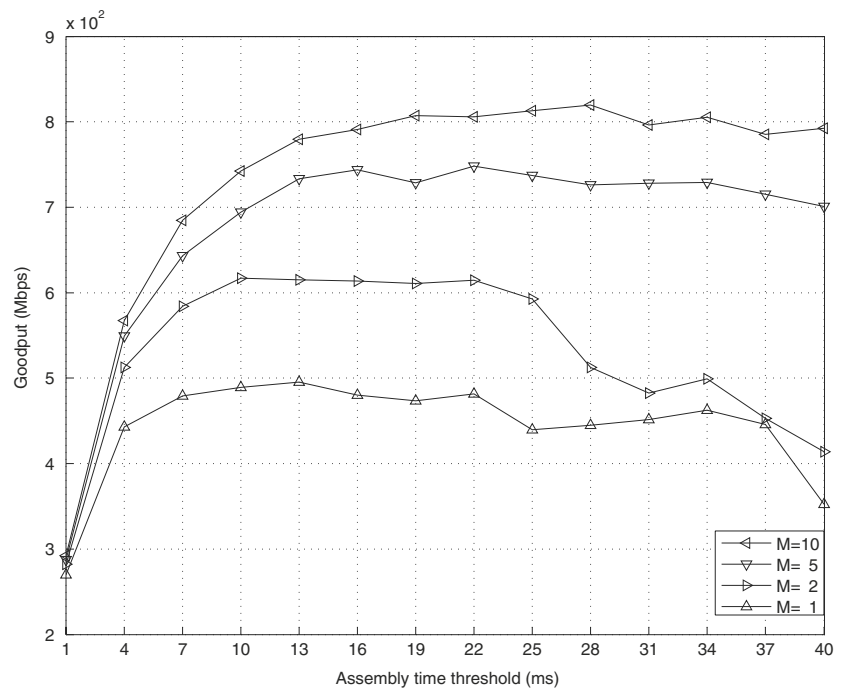
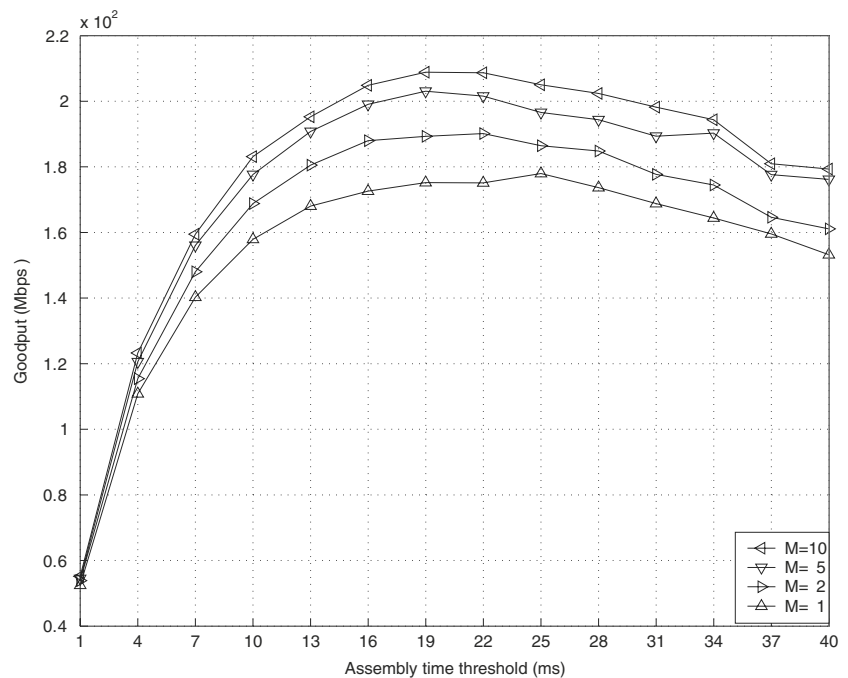


Fig. 4 Total goodput with timer-based assembly for $N = 10$, $p = 0.01$, $M = 1, 2, 5, 10$, and Reno TCP



probability p . When p is large, the congestion window cannot increase to large values due to more frequent burst losses. In this case, the burst size does not increase significantly as the timeout increases since the DFL gain does not change much with increasing timeout. 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 assembly timeout for all three TCP versions. In addition, it is observed that a relatively low number of assembly buffers may perform close to the per-flow aggregation case ($M = N$). 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 for ingress node architecture.

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

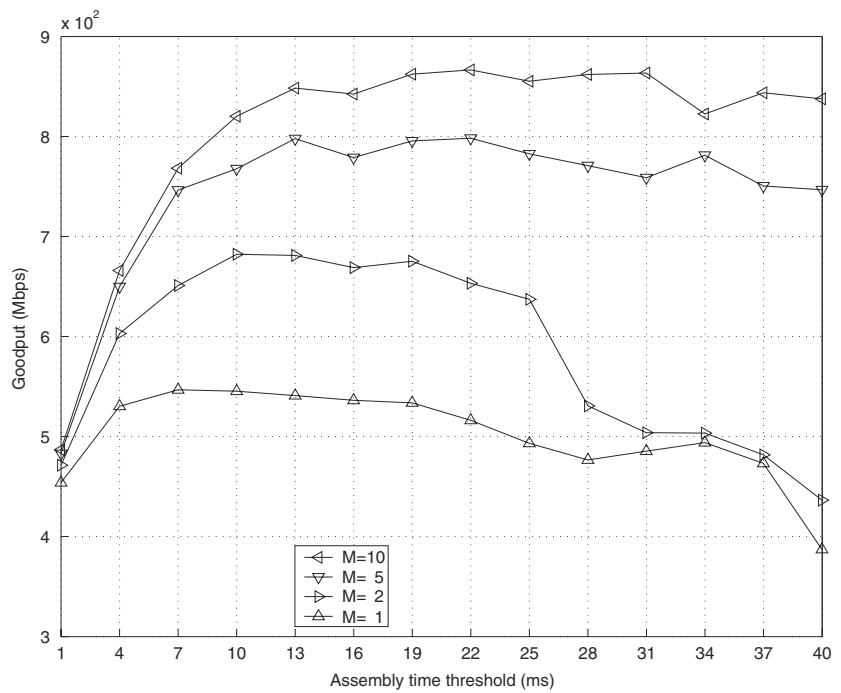
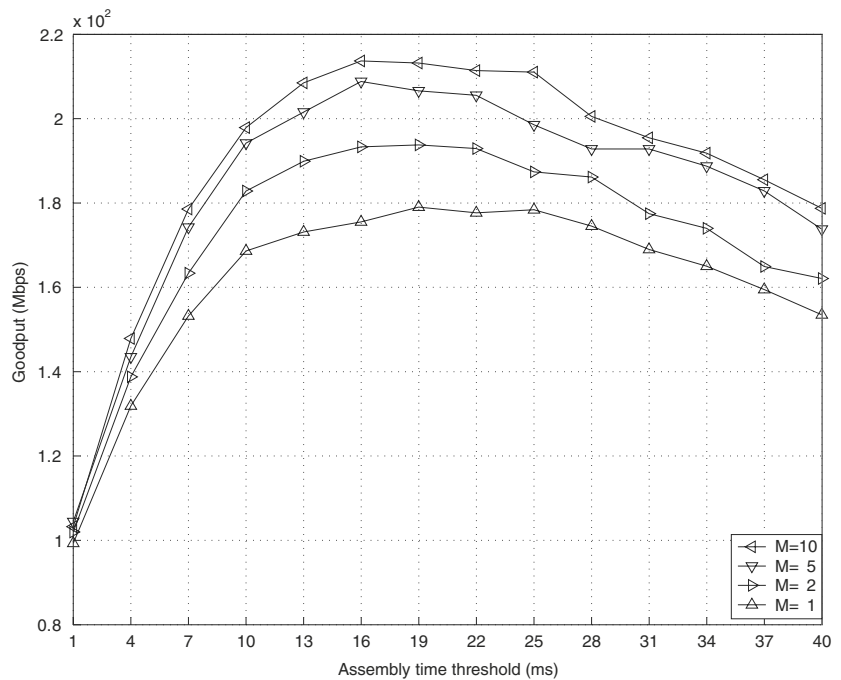


Fig. 6 Total goodput with timer-based assembly for $N = 10$, $p = 0.01$, $M = 1, 2, 5, 10$, and Newreno TCP



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, with Newreno slightly outperforming Reno.

In Table 1, the goodput enhancement of using multiple burstifiers per-egress 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 burstification for different TCP versions. The goodput enhancement is largest with Reno and smallest with Sack.

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

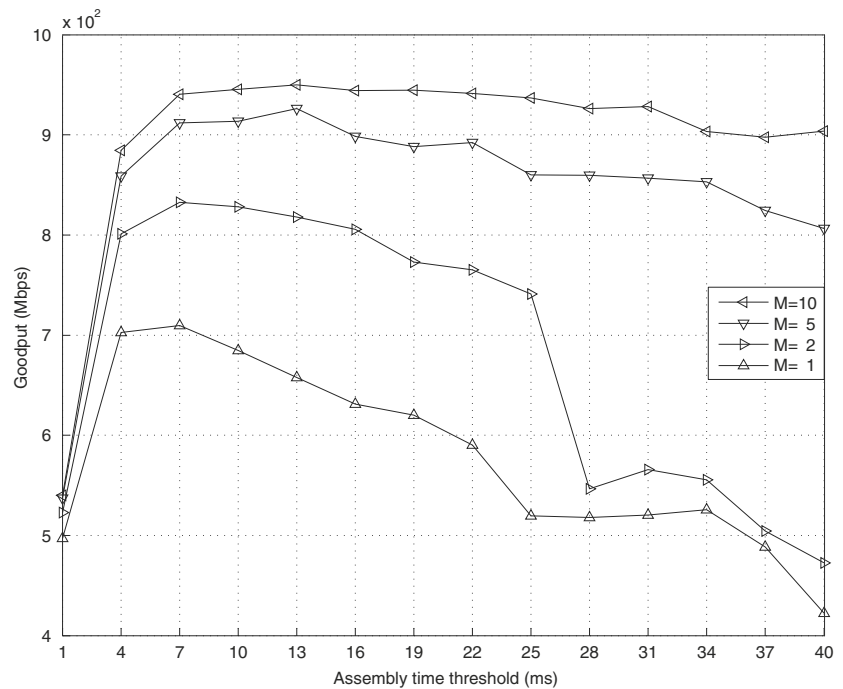
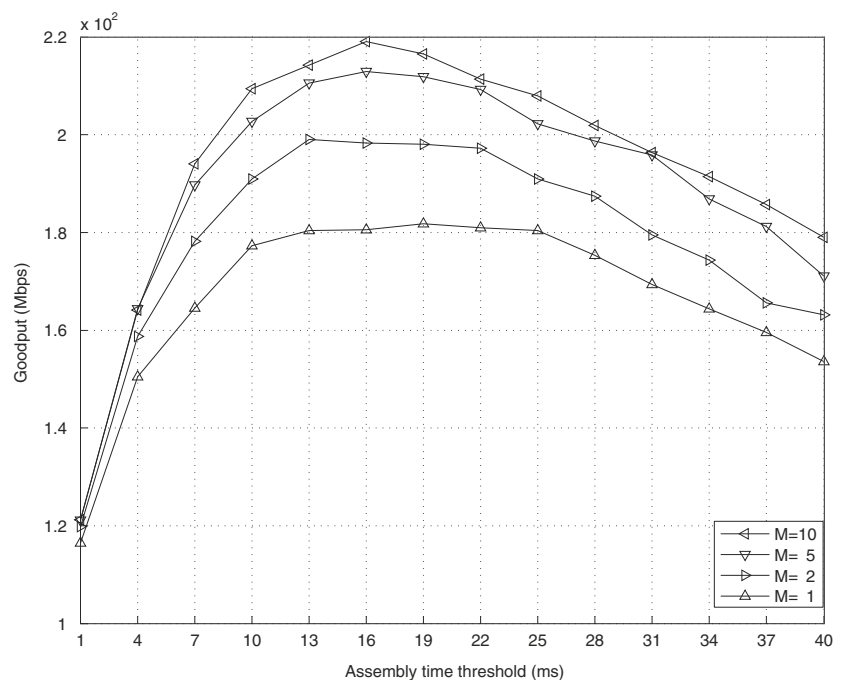


Fig. 8 Total goodput with timer-based assembly for $N = 10$, $p = 0.01$, $M = 1, 2, 5, 10$, and Sack TCP



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 particular case.

4 Effect of multiple assembly buffering with void-filling scheduling

The burst-length dependent losses naturally occur at a switch where a void-filling scheduling algorithm, e.g., LAUC-VF [15], is used and arriving bursts have different residual offset times [18, 19]. This is due to the fact that the burst length affects the probability that the burst scheduler will be successful in finding a suitable void for an incoming burst. This dependence is strongest for the flows having smaller residual

Fig. 9 Congestion window sizes, in MSS, for TCP Reno ($p = 0.01$)

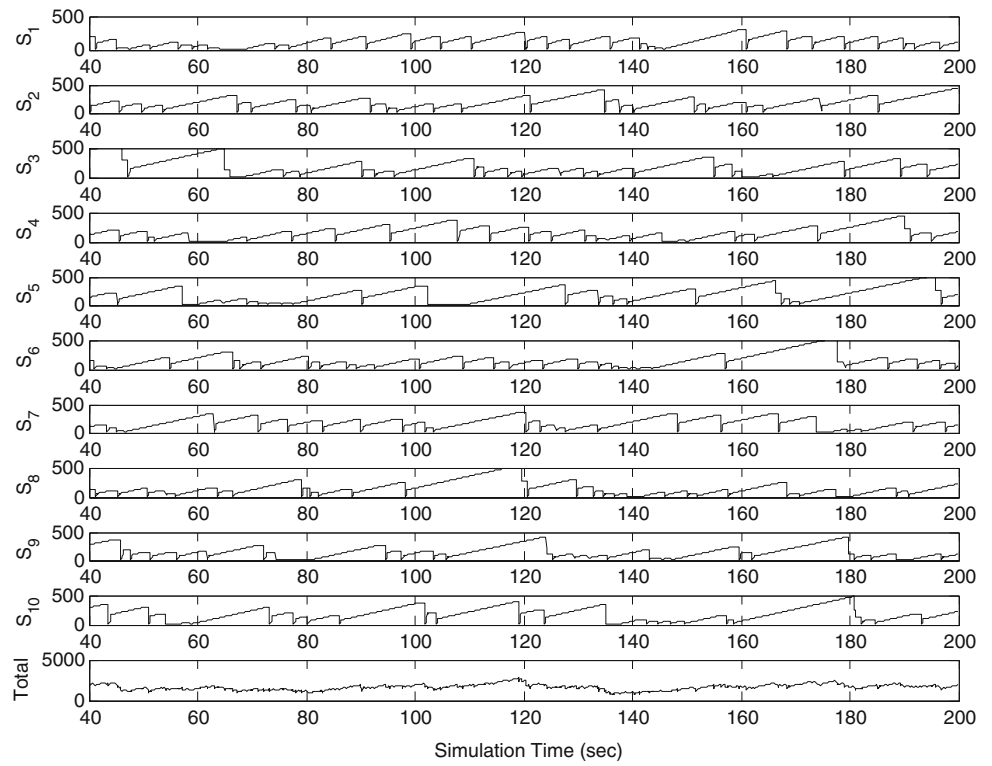
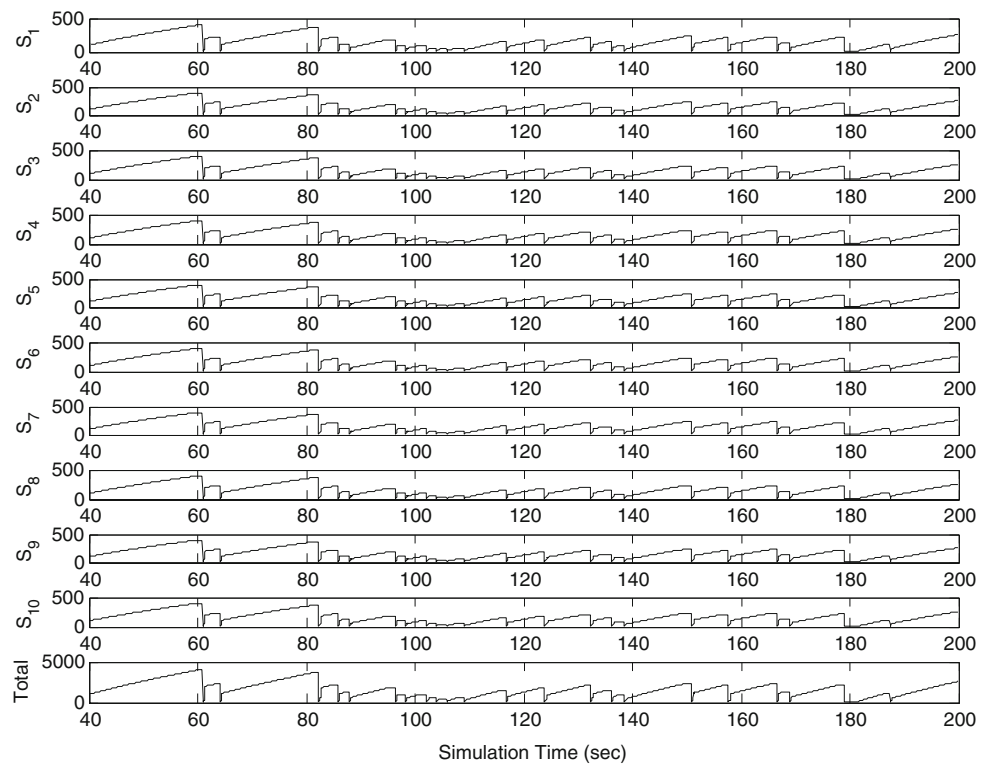


Table 1 Percentage goodput increase versus number of burstifiers for different TCP versions and loss probability

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

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.

The network topology used for studying the effects of burst length dependent losses is shown in Fig. 10. Sources S_1-S_N employ an infinite FTP flow to the respective destination D_1-D_N ($N = 20$). All optical links have $B_o = 1$ Gbps bandwidth and $T_o = 2.5$ ms propagation delay. The background

burst generator produces bursts whose sizes are exponentially distributed with mean $1/\mu$ and burst arrival process is Poisson with rate λ . All bursts are destined uniformly to the five egress nodes connected to D_1-D_{20} . Access links have $B_a = 50$ Mbps bandwidth and $T_a = 1$ ms propagation delay. LAUC-VF burst scheduling algorithm is used at the optical nodes. In the simulations in this section, we use the following parameters: $1/\lambda = 2$ ms, $1/\mu = 200$ μ s, and the nodal processing delay $\Delta = 50$ μ s (unless stated otherwise).

4.1 FTP-type long-lived TCP flows

In the first part of the simulations, we use FTP-type TCP flows where multiple concurrent long lasting FTP flows are used to generate the bursts. Figure 11 shows the loss probability for each egress node as a function of the burst length for $M = 1$ and the assembly timeout $T = 10$ ms. The statistics of the generated bursts are grouped into 10 bins according to the number of packets in the burst, which ranges from 1 to a maximum of 60 packets. It is observed that the loss probability is relatively high for the flows with smaller residual offset

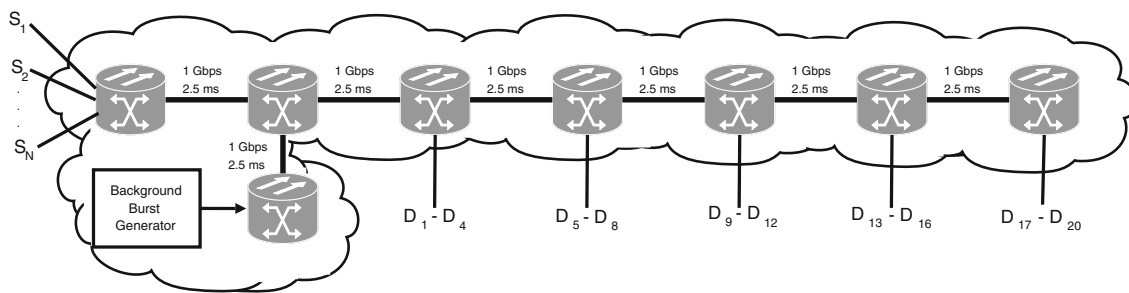


Fig. 10 Topology used for void-filling scheduling

Fig. 11 Loss probability versus burst length for different egress nodes

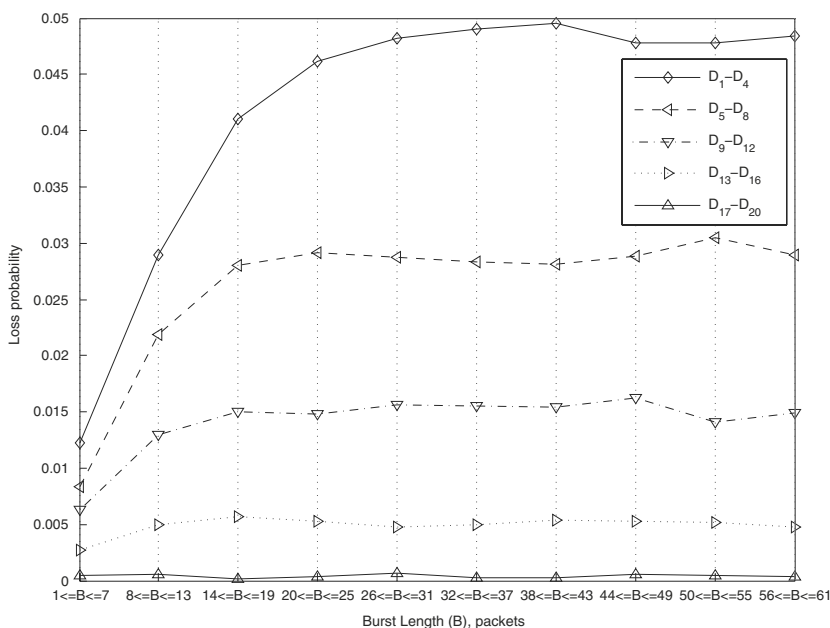


Fig. 12 Goodput and average burst size versus assembly time threshold for egress node 3

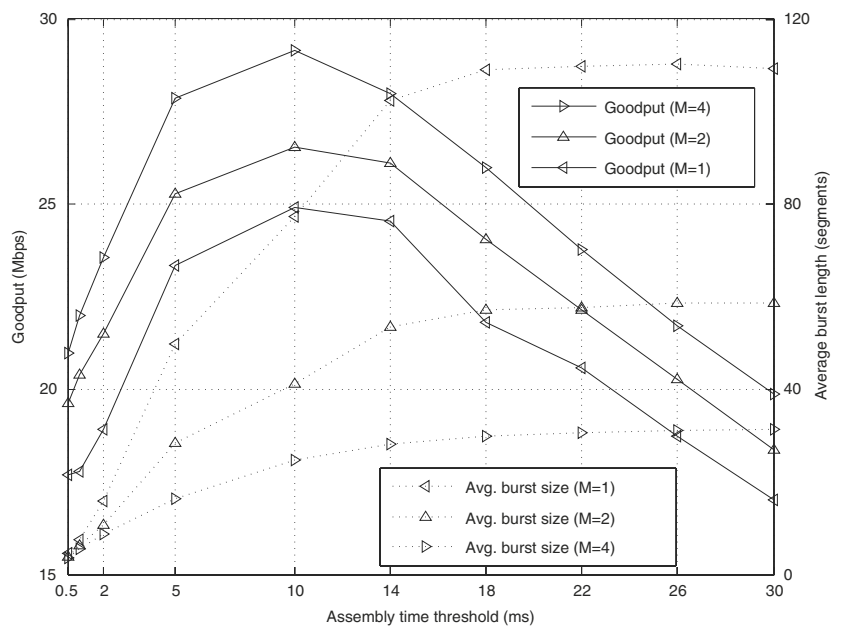
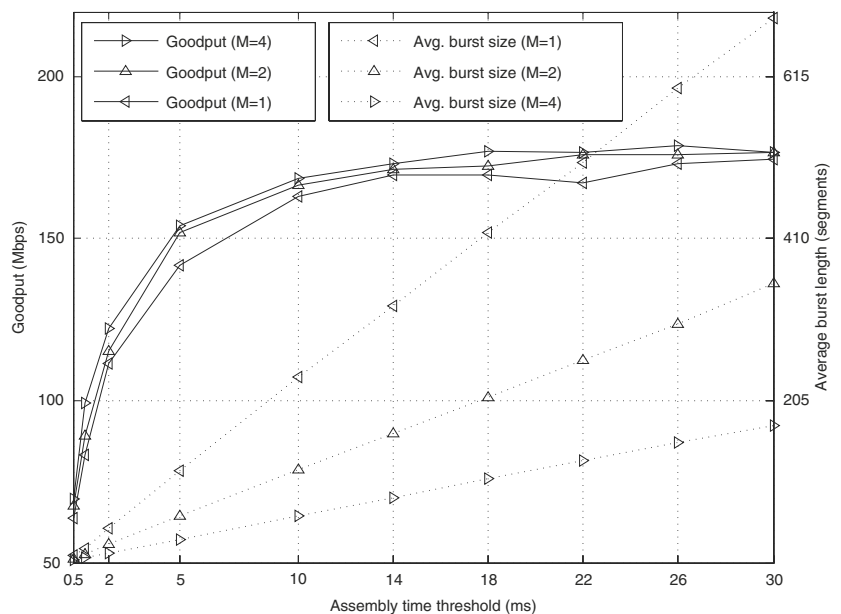


Fig. 13 Goodput and average burst size versus assembly time threshold for egress node 7



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 $D_{17}-D_{20}$, as expected.

In addition to the mechanisms mentioned in [6], 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 12 and 13 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 , 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

Table 2 Percentage goodput increase as a function of the number of burstifiers for TCP Reno

Δ (μ s)	M	Destination					
		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 3 Percentage goodput increase as a function of the number of burstifiers for TCP Sack

Δ (μ s)	M	Destination					
		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

cannot achieve very large congestion windows. The saturation of the average burst sizes coupled with the additional assembly delay causes 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 not significantly change as the burst assembly timeout is increased.

We also observe from Figs. 12 and 13 that the flows destined for D_{17} – D_{20} achieve much higher goodputs 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.

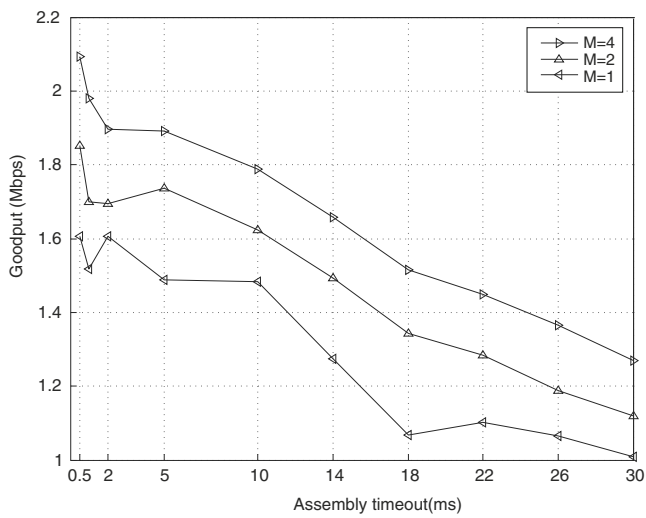
The comparison of Figs. 12 and 13 also reveal that the maximum goodput for the flows destined for D_1 – D_4 are achieved at smaller values of the burst assembly timeout compared with the flows destined for D_{17} – D_{20} . In fact, the maximum goodput is achieved before the burst size saturates for 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 shown in Fig. 11. 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 Fig. 13 since the burst losses are independent of the burst size for the flows destined to D_{17} – D_{20} .

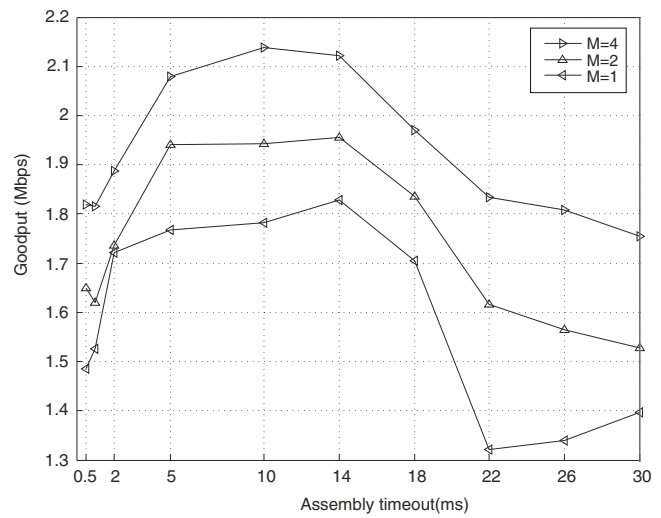
The performance improvement in the maximum average goodputs achieved by using $M = 2$ and $M = 4$ with respect to $M = 1$ for TCP Reno and TCP Sack are shown in Tables 2 and 3, respectively. The results show that the improvement in the average goodput is maximum for the flows destined for closer egress 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 with $M = 4$. 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.

4.2 Variable size short-lived TCP flows

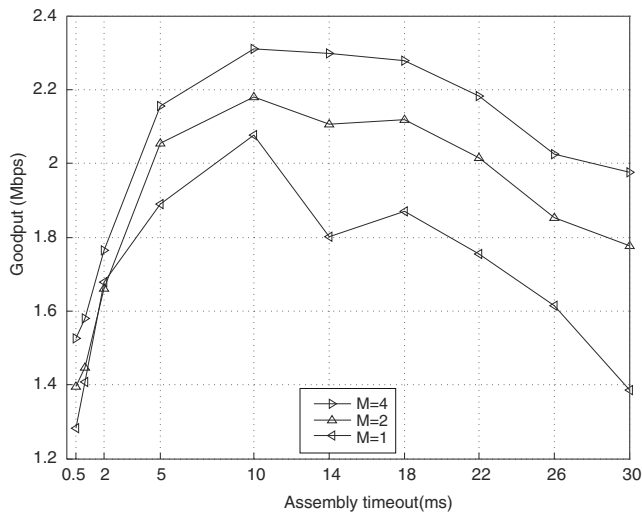
In this section, the infinite FTP flows of Sect. 4.1 are replaced by flows that mimic the Internet traffic. The heavy tail and large variance in flow sizes of typical Internet flows are modeled with Bounded Pareto distribution [20] while flows arrive according to a Poisson process with rate λ' . A Bounded Pareto distribution is denoted by tail heaviness α , minimum flow size k , and maximum flow size p . The probability



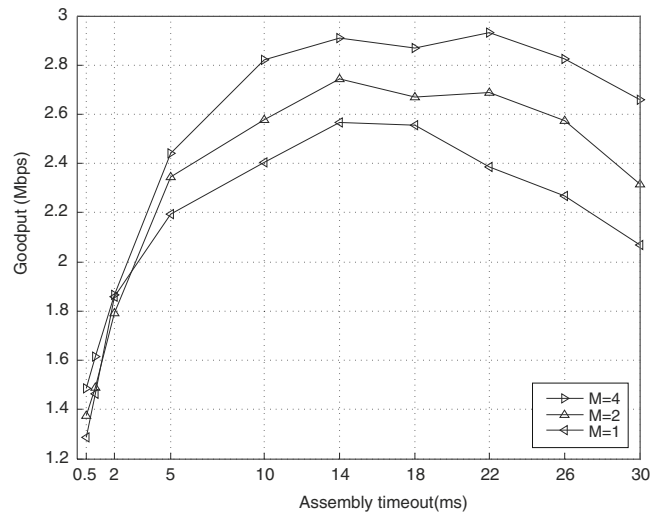
(a) $D_1 - D_4$



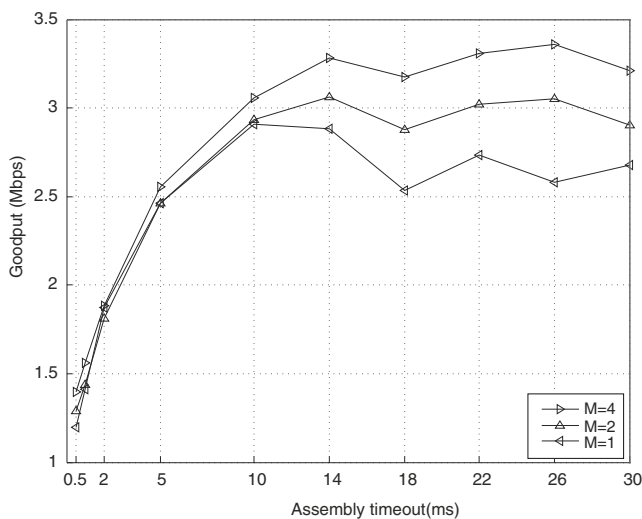
(b) $D_5 - D_8$



(c) $D_9 - D_{12}$



(d) $D_{13} - D_{16}$



(e) $D_{17} - D_{20}$

Fig. 14 Average goodput with timer-based assembly for $N = 10$, $M = 1, 2, 4$

density function $f(x)$, cumulative density function $F(x)$, and the n -th moment m_n are given as follows [20]:

$$f(x) = \frac{\alpha k^\alpha}{1 - (k/p)^\alpha} x^{-\alpha-1}, \quad k \leq x \leq p, \quad 0 \leq \alpha \leq 2$$

$$F(x) = \frac{1}{1 - (k/p)^\alpha} [1 - (k/x)^\alpha], \quad k \leq x \leq p, \\ 0 \leq \alpha \leq 2$$

$$m_n = \frac{\alpha}{(n - \alpha)(p^\alpha - k^\alpha)} (p^n k^\alpha - k^n p^\alpha)$$

Each IP router S_1 – S_{20} is assigned with a flow generator, which produces TCP Reno flows with Bounded Pareto size distribution and Poisson arrival pattern. The flows assigned to S_1 – S_{20} send their segments to the respective destination D_1 – D_{20} . For each flow generator, Bounded Pareto parameters are $\alpha = 1.2$, $k = 10$ MBytes, $p = 1$ GBytes, and flow arrival rate is $\lambda' = 0.1$ arrivals/s. TCP flow IDs are uniformly distributed in $\{0, 1, 2, 3\}$, and $M \in \{1, 2, 4\}$. In our simulations, the background burst generator is operated with $1/\mu = 200 \mu\text{s}$, $1/\lambda = 2$ ms, and the nodal processing delay is taken as $\Delta = 50 \mu\text{s}$.

The average goodput of the TCP flows is shown in Fig. 14 for each egress node. Once again, it is confirmed that increasing the number of assembly buffers improves the TCP performance. The goodputs of further egress nodes are relatively high compared to the goodputs of closer egress nodes since the drop probability is lower for bursts with higher residual offsets. For the egress nodes D_1 – D_4 , the drop probability is so high that the DFL gain cannot compensate the delay penalty as the assembly timeout is increased; therefore, the goodput constantly decreases with the burstification delay. When we look at further egress nodes, it can be seen that the effect of DFL gain becomes dominant and for the egress of D_{17} – D_{20} , the decrease in goodput for increasing assembly timeout is minimal even for large burstification delays.

The average goodput increase in percentage for all flows as a function of the number of burstifiers is given in Table 4. The results show that the improvement is most significant for the nearest egress node (more than 30% for $M = 4$), while the improvement decreases for the further egress nodes and using $M = 2$.

Table 4 Percentage goodput increase as a function of the number of burstifiers

M	Destination				
	D_1 – D_4	D_5 – D_8	D_9 – D_{12}	D_{13} – D_{16}	D_{17} – D_{20}
4	30.52	19.33	12.03	16.40	17.21
2	15.46	8.82	6.45	9.34	7.43

5 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 occurring 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 increases. This conclusion holds for different TCP versions and different burst loss models. We recommend that the edge router architecture be designed with multiple burst assemblers per egress but with less burstifiers than per-flow burstification, i.e., $1 < M < N$, in order to reduce the complexity of managing large number of buffers while achieving nearly the maximum TCP goodput.

Acknowledgements The work described in this paper was carried out with the support of the BONE-project (“Building the Future Optical Network in Europe”), a Network of Excellence funded by the European Commission through the 7th ICT-Framework Programme, and by the Scientific and Technological Research Council of Turkey (TUBITAK) under project 104E047.

References

- Listanti, M., Eramo, V., Sabella, R.: Architectural and technological issues for future optical Internet networks. *IEEE Commun. Mag.* **38**(9), 82–92 (2000)
- Yao, S., Xue, F., Mukherjee, B., Yoo, S.J.B., Dixit, S.: Electrical ingress buffering and traffic aggregation for optical packet switching and their effect on TCP-level performance in optical mesh networks. *IEEE Commun. Mag.* **40**(9), 66–72 (2002)
- Rouskas, G.N., Xu, L.: Optical packet switching. In: Sivalingam, K., Subramaniam, S. (eds.) *Emerging Optical Network Technologies: Architectures, Protocols and Performance*, pp. 111–127. Springer (2004)
- Qiao, C., Yoo, M.: Optical burst switching (OBS)—a new paradigm for an optical internet. *J. High Speed Netw.* **8**, 69–84 (1999)
- Turner, J.: Terabit burst switching. *J. High Speed Netw.* **8**, 3–16 (1999)
- Yu, X., Li, J., Cao, X., Chen, Y., Qiao, C.: Traffic statistics and performance evaluation in optical burst switched networks. *IEEE/OSA J. Lightwave Technol.* **22**(12), 2722–2738 (2004)
- Cao, X., Li, J., Chen, Y., Qiao, C.: Assembling TCP/IP packets in optical burst switched networks. *Proc. IEEE GLOBECOM* **3**, 2808–2812 (2002), Taipei, Taiwan
- Yu, X., Qiao, C., Liu, Y., Towsley, D.: Performance evaluation of TCP implementations in OBS networks. *Tech. Rep. 2003-13*, CSE Dept., SUNY, Buffalo, NY (2003)
- Gowda, S., Shenai, R.K., Sivalingam, K.M., Cankaya, H.C.: Performance evaluation of TCP over optical burst-switched (OBS) WDM networks. *Proc. IEEE ICC* **2**, 1433–1437 (2003), Anchorage, Alaska

- [10] Detti, A., Listanti, M.: Impact of segments aggregation on TCP Reno flows in optical burst switching networks. *Proc. IEEE INFOCOM* **3**, 1803–1812 (2002), New York, USA
- [11] He, J., Gary Chan, S.-H.: TCP and UDP performance for Internet over optical packet-switched networks. *Comput. Netw.* **45**(4), 505–521, (2004)
- [12] Ramantas, K., Vlachos, K., de Dios, O.G., Raffaelli, C.: TCP traffic analysis for timer-based burstifiers in OBS networks. *Proc. ONDM 176–185* (2007), Athens, Greece
- [13] Hong, D., Poppe, F., Reynier, J., Baccelli, F., Baccelli, F., Petit, G.: The impact of burstification on TCP throughput in optical burst switching networks. *Proc. ITC-18*, 89–96 (2003), Berlin, Germany
- [14] Dolzer, K., Gauger, C., Späth, J., Bodamer, S.: Evaluation of reservation mechanisms for optical burst switching. *AEU Int. J. Electron. Commun.* **55**(1), 18–26 (2001)
- [15] Xiong, Y., Vandenhoute, M., Cankaya, H.C.: Control architecture in optical burst-switched wdm networks. *IEEE J. Sel. Areas Commun.* **18**(10), 1838–1851 (2000)
- [16] “Network Simulator 2”, developed by L. Berkeley Network Laboratory and University of California Berkeley, <http://www.isi.edu/nsnam/ns>
- [17] Gurel, G., Alparslan, O., Karasan, E.: nOBS: an ns2 based simulation tool for performance evaluation of TCP traffic in OBS networks. *Ann. Telecomm.* **62**(5–6), 618–637 (2007)
- [18] Barakat, N., Sargent, E.H.: Analytical modeling of offset-induced priority in multiclass OBS networks. *IEEE Trans. Commun.* **53**(8), 1343–1352 (2005)
- [19] Kaheel, A.M., Alnuweiri, H., Gebali, F.: A new analytical model for computing blocking probability in optical burst switching networks. *IEEE J. Sel. Areas Commun.* **24**(12), 120–128 (2006)
- [20] Rai, I.A., Urvoy-Keller, G., Biersack, E.W.: Analysis of LAS scheduling for job size distributions with high variance. In: *Proceedings of ACM Sigmetrics*, pp. 218–228 (2003)

Author Biographies



Guray Gurel received B.S. and M.S. degrees in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2003 and 2006, respectively. He is currently working at Nortel Netas.



Ezhan Karasan received B.S. degree from Middle East Technical University, Ankara, Turkey, M.S. degree from Bilkent University, Ankara, Turkey, and Ph.D. degree from Rutgers University, Piscataway, New Jersey, USA, all in electrical engineering, in 1987, 1990, and 1995, respectively. During 1995–1996, he was a post-doctorate researcher at Bell Labs, Holmdel, New Jersey, USA. From 1996 to 1998, he was a Senior Technical Staff Member in the Lightwave

Networks Research Department at AT&T Labs-Research, Red Bank, New Jersey, USA. He has been with the Department of Electrical and Electronics Engineering at Bilkent University since 1998, where he is currently an associate professor. During 1995–1998, he worked in the Long Distance Architecture task of the Multiwavelength Optical Networking Project (MONET), sponsored by DARPA. Dr. Karasan is a member of the Editorial Board of *Optical Switching and Networking* journal. He is the recipient of 2004 Young Scientist Award from Turkish Scientific and Technical Research Council (TUBITAK), 2005 Young Scientist Award from Mustafa Parlar Foundation, and Career Grant from TUBITAK in 2004. Dr. Karasan received a fellowship from NATO Science Scholarship Program for overseas studies in 1991–1994. Dr. Karasan is currently the Bilkent team leader of the FP6-IST Network of Excellence (NoE) project e-Photon/One+. His current research interests are in the application of optimization and performance analysis tools for the design, engineering, and analysis of optical networks and wireless ad hoc/sensor networks. Dr. Karasan is a member of the Editorial Board of *Optical Switching and Networking* journal. He is the recipient of 2004 Young Scientist Award from Turkish Scientific and Technical Research Council (TUBITAK), 2005 Young Scientist Award from Mustafa Parlar Foundation and Career Grant from TUBITAK in 2004. Dr. Karasan received a fellowship from NATO Science Scholarship Program for overseas studies in 1991–1994. Dr. Karasan is currently the Bilkent team leader of the FP6-IST Network of Excellence (NoE) e-Photon/ONE+ and FP7-IST NoE BONE projects. His current research interests are in the application of optimization and performance analysis tools for the design, engineering and analysis of optical networks and wireless ad hoc/sensor/mesh networks.