

Cost-Throughput Analysis of a Modified TCP for Optical Burst Switched Networks

Subhasis Datta
Research Scholar
National Institute of
Technology
Durgapur-713209

Avijan Dutta
Associate Professor
National Institute of
Technology
Durgapur-713209

Subhrabrata Choudhury
Associate Professor
National Institute of
Technology
Durgapur-713209

ABSTRACT

Performance of existing TCP implementations over Optical Burst Switched (OBS) networks is not satisfactory, as they suffer from false congestion detection. As the contention induced losses are more common than congestion induced losses in OBS, the TCP reduces the congestion window even when there is no congestion. This effect, in turn unnecessarily reduces the TCP throughput. So it is crucial to differentiate the contention induced loss from congestion induced loss in such networks. It has been proposed in this paper, a mechanism that makes use of short-term RTT variation and assembly times of individual TCP segments to differentiate between congestion and contention induced losses. This mechanism is a cross layer modification of a congestion control scheme called Statistical Additive Increase and Multiplicative Decrease (SAIMD). Our proposed scheme is found to be more efficient than existing one in terms of TCP throughput. The gain in throughput is primarily because of avoidance of misdirection of contention induced loss as congestion induced loss. A throughput-cost study has been conducted based on the throughput metric available from the efficiency analysis of our proposed scheme. The cost model used focuses on initial capital expenditure including switching, routing and amplification cost. An extension of ns2 called nOBS, to support OBS protocol is used for simulation to study the performance.

General Terms

Optical Networks

Keywords

OBS, Modified TCP, congestion induced loss, contention induced loss.

1. INTRODUCTION

All-Optical networks can be a promising solution for the explosive growth of the Internet traffic. Optical burst switching [1]–[3], [5], [8], among several other optical switching technologies, has gained a lot of attention recently because it combines the advantages of both circuit switching and packet switching paradigm. The features of Optical Packet Switching (OPS) and Optical Circuit Switching (OCS) have been utilized in OBS, to derive statistical multiplexing gain [3]. It is also advantageous for OBS, that no O/E/O conversion is required at every intermediate router or switch. This is elegantly done by separating control functions and data transmission. In OBS networks, the individual packets which are destined for a particular destination, is assembled into a burst. A burst is generated when a minimum burst length is reached or when the maximum assembly period is reached, whichever happens earlier. In a dual-threshold based burst assembly algorithm, both the thresholds can be updated based on traffic load. Once a burst is assembled a Burst Header Packet (BHP) is created and sent ahead of the burst to reserve the optical path. The

burst is then sent on a separate wavelength (data channels) in the optical domain. Finally the burst is disassembled at the destination and forwarded to individual receivers. Recent implementation shows that the technology is not far away from aggressive commercial deployment.

As the data burst passes through an intermediate node, the resources are released either according to the length of the burst (Just Enough Time) or after receiving an explicit notification (Just in Time). The above signaling mechanisms mitigate the need for optical buffering. Burst losses occur in OBS networks due to bufferless characteristics, where two or more control packets try to reserve a wavelength channel in overlapped time periods. The recent development of IP over OBS backbone has attracted much attention among researchers. Such adoption of TCP over OBS has resulted in new challenges where TCP could be affected by the underlying OBS bufferless characteristics.

The problem worsens because of the fact that a burst contains multiple packets from individual TCP connections that too regulated by their own flow control and congestion control mechanism. In IP packet-switched networks, packet losses are caused mainly due to buffer overflow, as a result of network congestion. Packet loss can be an effective indication of network congestion.

The existing network congestion control mechanism faces a greater challenge when OBS is deployed over TCP. Burst contention may happen due to burst character of the traffic even when the traffic load is light. In OBS, multiple consecutive packets are aggregated into a single burst. The loss or delay of a single burst affects multiple TCP packets within the TCP window. Burst delay can occur due to assembly delay, Fiber Delay Lines (FDL), burst retransmission or deflection routing. TCP cannot differentiate between congestion induced loss and contention induced loss, the congestion window is reset too frequently even when there is no congestion leading to poor throughput.

In recent past, various TCP implementations have been proposed to differentiate congestion induced loss and contention induced loss. The authors in [18] has interpreted a loss as 'congestion induced' if the short-term RTT value is higher than long-term RTT value. The short-term RTT should be higher during the state of congestion as the delays in FDL [12] buffers are higher and probability of deflection routing [11] is also higher during the state of congestion. The result obtained using the technique Statistical Additive Increase Multiplicative Decrease (SAIMD) was found satisfactory. SAIMD [18] does not consider the variation in delay resulting from variation of burst assembly time. Assembly time variance is significant under length-threshold based or dual-threshold based justifications, which are more commonly used in OBS implementation.

Here a more accurate mechanism has been proposed which can differentiate between congestion induced loss and contention induced loss even under variation of assembly time. Thus the proposed mechanism will work equally efficiently under any assembly scheme.

A protocol level modification in TCP and BHP has been proposed, that enable us to measure the assembly time at the TCP end points. Then the decision regarding the nature of the loss is taken based on both 'difference between short-term and long-term RTTs' and assembly time of the concerned burst.

The paper is organized as follows. Related works on TCP over OBS will be discussed in Section 2. Section 3 discusses our proposed solution. In section 4, results are discussed and the paper is concluded in Section 5.

2. RELATED WORKS ON TCP OVER OBS

Few studies were found investigating in recent past to mitigate the false congestion detection problem in OBS networks.

The authors in [22] investigated false Time Out (TO) detection due to random contention and proposed three approaches. The first approach is based on the estimation of number of TCP segments assembled in a burst without considering the burst assembly mechanism implemented in OBS layer. The second is based on the acknowledgement of each TCP segment by a TCP agent at the OBS edge node, though effective but violating the end-to-end TCP semantics. The last approach is based on sending negative acknowledgement to TCP senders informing about the network congestion state on event of burst dropped at one of the core nodes. The solutions provided the authors could result in high signaling overhead making it practically impossible to implement.

The authors in [24], [25] proposed retransmission of lost bursts from ingress nodes, to reduce burst loss probability. Similar solution was proposed by authors of [26] but with TCP Slack over OBS networks. Both the above two solutions require buffering of incoming bursts at the edge nodes within a time window.

In [4], the authors conducted a detailed simulation study of TCP performance over OBS network for various TCP implementations considering three access based TCP flows; speed-fast TCP, medium TCP and slow TCP. Analysis and simulation shows that when TCP is applied to an OBS network it results in a delay penalty and an associated Delayed First Loss (DFL) gain. It was shown that amount of DFL gain is always positive but the extent depends on the TCP speed and burst assembly algorithm.

The authors in [6] studied the impact of burst assembly algorithm on various TCP implementations such as TCP Reno [13], New-Reno and SACK in OBS network. It was confirmed that SACK [9] based TCP always performs better because of its efficient recovery mechanism against consecutive segment loss.

In [14], the authors discussed TCP flows in OBS network to investigate the effect of variation of bustification period and derive some general guideline dimensioning the bustification period. The result shows that accurate dimensioning of the bustification period yields negligible penalties with regards to low speed sources and significant benefits with regards to high speed sources.

The authors in [21] studied the behavior of High-speed TCP (HSTCP) in OBS networks. The study shows that small burst

assembly time and a large burst loss ratio in OBS can reduce the throughput of HSTCP significantly.

Lee et. al. [19] showed TCP throughput improvement by their proposed Drop Policy (DP) with burst retransmission where a new concept of retransmission-count for bursts is introduced to mitigate unnecessary TCP congestion control.

The authors in [17] provided a comprehensive survey on reported studies for TCP enhancements over OBS networks in order to mitigate the numerous side effects due to bufferless characteristic of burst transmission.

Venkatesh et. al. [15] proposed a new measure to differentiate between congestion and contention losses, called the number of bursts between failures (NBBF). They observe that the NBBF follows a Gaussian distribution with different parameters for contention and congestion losses.

Shihada et. al. [18] introduce a novel congestion detection scheme for high band width TCP flows over OBS networks, called Statistical Additive Increase Multiplicative Decrease(SAIMD). SAIMD maintains and analysis a number of previous RTTs at the TCP senders in order to identify whether the packet loss event is due to network congestion.

In [20] a general analytical model is established to analyze the impacts of the influential factors of both TCP and OBS layers on the performance of Grid-OBS networks. The study does not propose any modified TCP. The authors in [10] identified the ill effects of implementing TCP over a hybrid network (IP-access and OBS-core).Based on their proposition and simulation, N: 1: N split approach outperforms all other approaches.

Liu et.al. [16] focused on TCP acknowledgement (ACK) packets. Here independent ACK Assembly (IAA) scheme is presented, which assembles ACK into one specific data burst (DBACK) to avoid the disadvantage of traditional mechanism and improve TCP throughput.

It has been observed through our literature survey that differentiating between congestion induced loss and contention induced loss still remains a major issue. The proposals available so far do not work well for all types of burst assembly algorithms, as assembly-time is not utilized separately in making the decision. The information about assembly-time is not available at the TCP layer and passing information to TCP is also not discussed in any proposal.

3. PROPOSED SOLUTION

The following two subsections discusses a mechanism for measuring assembly time by TCP end points and modification in determination of congestion.

3.1 General Discussion

Statistical AIMD was proposed in [18] as a very unique mechanism for high bandwidth TCP congestion control in IP over OBS networks. The vicious effect of TCP false congestion is due to bufferless characteristics of OBS domain. When a data burst is lost, from a single or multiple TCP senders, they are notified of the segment loss through the receiving of a TD or TO. Instead of making the *cwnd* size half by TCP senders, the size of the *cwnd* is reduced by a multiplicative factor β . The factor β is dynamically determined by using the short term RTT statistics, based on collected and analyzed historical RTTs at the occurrences of any segment loss event. The merits gained by SAIMD are particularly beneficial to high bandwidth and fast TCP flows. However the schemes for differentiation of *segment loss due to congestion* and *segment loss due to contention* are effective only for timer based burst assembly

algorithms. Moreover, it doesn't apply to the most commonly used dual-threshold based assembly algorithm. As these dual-threshold based algorithm results in considerably high variance of assembly time, a congestion detection scheme must take care of the assembly time hidden within the observer RTT values of the TCP connections.

Hence, a mechanism for passing the assembly time to each TCP connection is essential.

Assuming that

t_t = time required to transmit a full packet.

t_p = propagation-time between the source and destination. This is fixed if route is fixed.

t_q = queuing delay at the FDL. This is a random variable having high variance based on the traffic conditions.

t_{ack} = time required to transmit an ACK packet. For simplicity it is assumed that $t_t = t_{ack}$.

t_{qf} = queuing delay at the FDL for forward segment.

t_{qr} = queuing delay at the FDL for ACK (reverse) segment.

But in OBS network the queuing delays are having less variance due to limited buffering capability of OBS nodes and t_p will vary as per the deflection routing scheme. But the most significant difference will occur because of the assembly time and the resulting RTT for OBS network will have the form:

$$RTT_{obs} = 2t_t + 2t_p + (t_{qf} + t_{qr}) + a_{ff} + a_{rr} \quad (1)$$

Where,

a_{ff} = Average assembly delay per forward segment.

a_{rr} = Average assembly delay per ACK segment.

It is noted that the assembly delay for every segment within a burst will not be the same and similar arguments also apply to ACK segments. In proposed scheme the assembler will make a measurement of assembly delay per segment which has been represented by a_{ff} and a_{rr} . The above discussion concludes that the knowledge of assembly time per segment is an important factor and its variance will have a very important role in variance of RTT. Hence while making decision on occurrence of congestion based on RTT variance, the role of assembly time cannot be under emphasized. The following two subsections, discusses a mechanism for measuring assembly time by TCP endpoints and modifications in determination of congestion.

3.2 Protocol Level Modification for Measuring Assembly Time at the TCP End Points

Fig 1 shows a mechanism to calculate assembly time at the TCP end points in TCP over OBS network. An extra field is added in TCP header format and also an extra header field in burst header packet (BHP) format. Both of these fields are named as assembly-time.

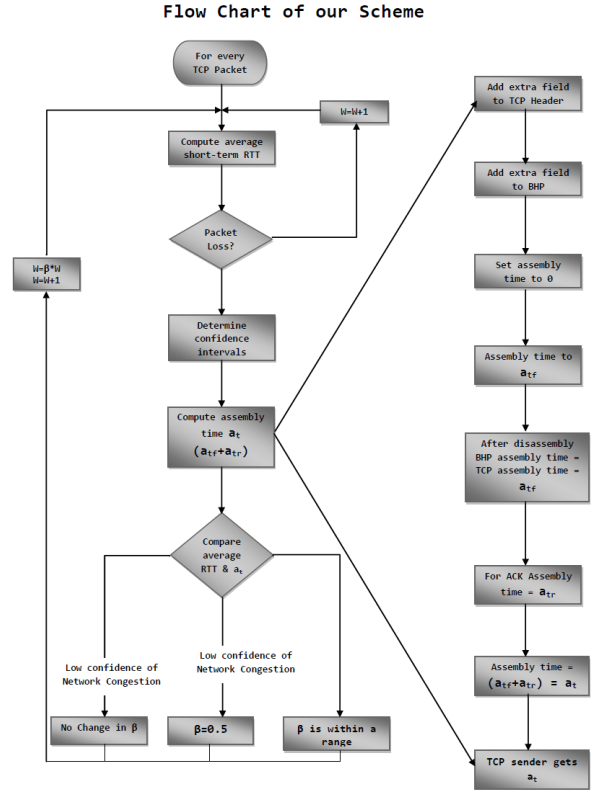
Step 1: At the sender TCP end point, the assembly-time field is set to 0.

Step2: At the sender edge node, the assembler, assembles multiple segments from multiple TCP senders into a burst and BHP packets are generated. The assembly time is recorded as a_{ff} and is called as forward assembly time, this a_{ff} is assigned to BHP header in extra field as assembly-time = a_{ff} .

Step 3: When the burst reaches the destined receiver node, the data burst is disassembled and the packets are send to their respective TCP receivers. At the disassembler, BHP header copies its assembly-time in TCP header extra field, assigning the TCP assembly-time = a_{ff} .

After disassembling the burst, TCP segments reach to corresponding receivers. TCP receiver send back TCP header of ACK segment through OBS network to TCP sender, where it contains TCP header extra field assembly-time = a_{ff} .

The ACK segment of the TCP is also to be sent within a burst. Hence the reverse assembly time is additionally added. Suppose, the assembly time generated is a_{rr} , then the BHP extra field assembly-time = a_{rr} . After ACK burst generation and corresponding BHP of ACK reaches the sender edge node disassembler, ACK burst is disassembled in the same way. At the disassembler, in ACK (TCP) header extra field assembly time, a_{rr} is copied from BHP header to ACK of that TCP segment, then assembly-time = $a_{ff} + a_{rr} = a_t$, when ACK reach the sender. Then from TCP header field, TCP sender get the total assembly-time = a_t . The flow chart in Fig. 2 shows the process to determine the assembly time.



Finally, the value of assembly time which was under emphasized in previous scheme [18] is available. Now, in next subsection it is shown how this value can be utilized in making better decisions regarding nature of loss.

3.3 Proposed Modification in Congestion Induced Loss Detection

In SAIMD [18], it has been observed that two parameters M and N were introduced to calculate average long-term RTTs, avg,rt_M and average short-term RTTs, avg,rt_N . In a TCP session, the two averages are compared to determine whether the segment loss is due to random burst contention or network congestion. Both the averages will be close to each other, if burst contention occurs. Large value of avg,rt_N can be considered as network congestion.

But here the value of a_t is utilized for making decision and the proposed Scheme has been implemented as a modified SAIMD scheme. First it is checked if *average short-term RTT* is greater than current smooth RTT with some factor multiples. If the first condition is true, then it is checked if *assembly-time* is greater than *threshold assembly-time* or not, because *assembly-time* is big factor for RTT variation. Finally, in Our Scheme, the decision regarding nature of loss is made as per the following calculation.

if (*average short-term RTT* > $k * \text{smoothed RTT}$) and (*assembly-time* > *threshold assembly-time*) then congestion occurs.

From [18], section III, subsection C, the *cwnd* adjustment scheme can be followed based on the situation. Modifying the SAIMD can cause additional overhead for maintaining the long-term RTTs. The cost is also a tradeoff with long convergence time when recovering from slow-start. The computational complexity is also constant and occurs only when there is a segment loss event.

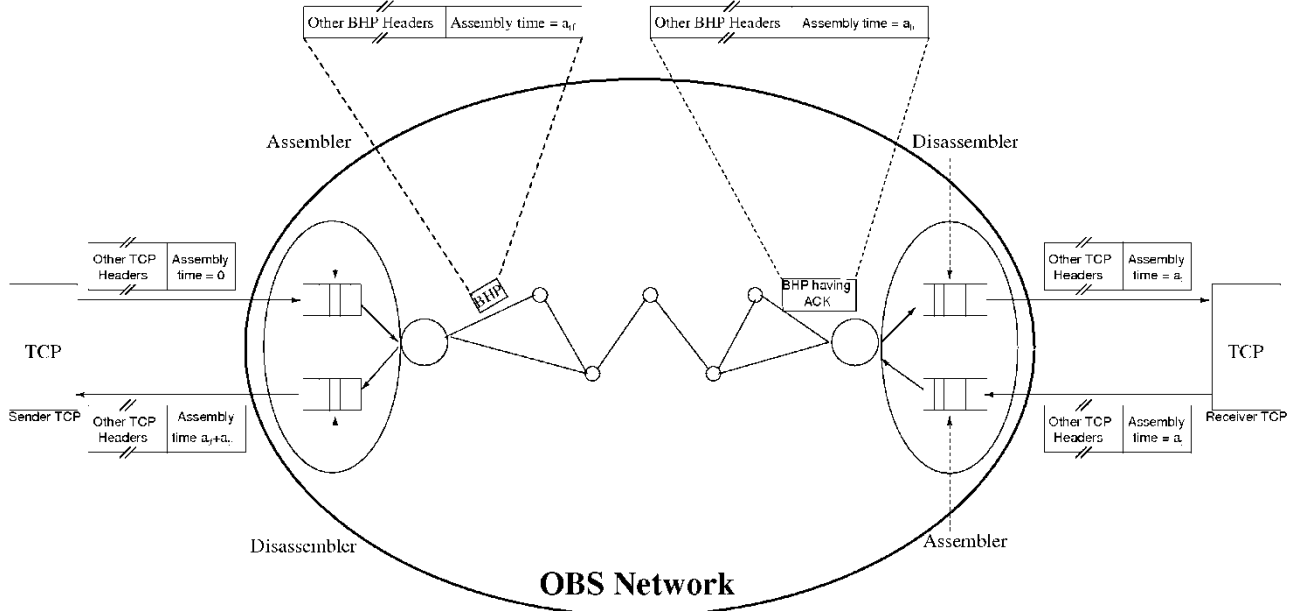


Fig.1. our scheme for measuring assembly time in TCP over OBS layer

4. RESULTS AND DISCUSSION

Throughput analysis of the proposed scheme is done in an OBS network. TCP Sack [9] has been selected for simulation scenario as it has the best throughput performance in OBS network compared to TCP Reno and New Reno [22] [23].

The proposed TCP for OBS-network is implemented on nsOBS, an extension of ns-2 for OBS network. A network of six-edge nodes and six-core nodes as shown in Fig.3 is considered. Each edge-node with 5 TCP-SACK Agents, are connected to 5 different SACK-Sinks where every sink connected at each edge-node. The TCP agents are connected with FTP traffic source and starts sending traffic simultaneously. The links have 2 control channels and 8 data channels each of which can send at 10 Gbps. Thus total data channel capacity of the links is 80 Gbps. Dual threshold-based burst assembler for bustification is used. But the assembly parameters are varied in various simulations to observe the effects of the parameters on TCP performance. The necessary

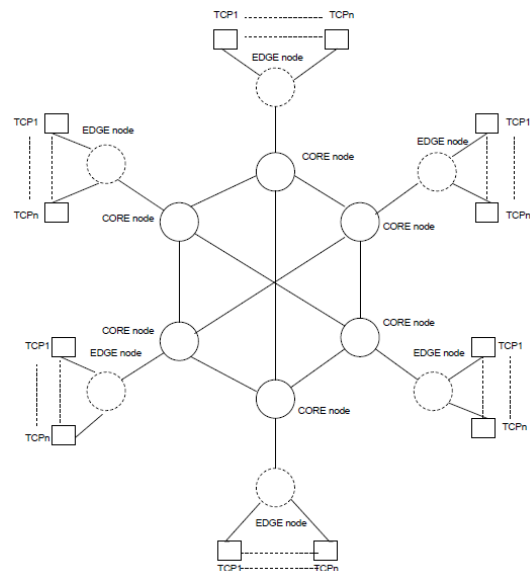


Fig. 3 OBS Topology for Simulation

changes has been made in TCP-SACK implementation, Fig. 4 shows it outperforms all other variants as it is more robust against Consecutive segment losses. In Fig.4 plots the number of bytes received with time for simple TCP, Reno TCP and SACK TCP for assembled burst of size 40KB.

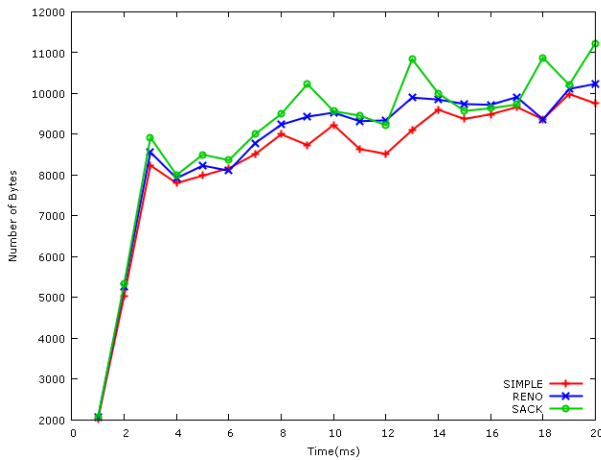


Fig. 4 Throughput Comparison of Simple TCP, Reno TCP and SACK TCP (Burst size=40,000 Bytes)

The similar plots are shown for burst length 100KB in Fig.5. It has been observed that SACK TCP outperforms all other variants as it is more robust against consecutive segment losses. Hence TCP-SACK is chosen and modified to implement our proposed scheme.

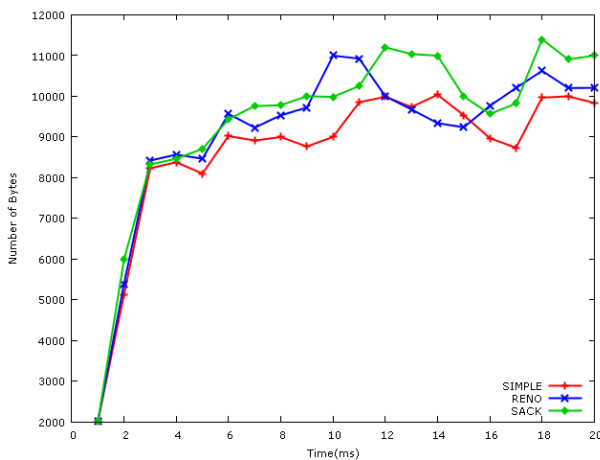


Fig. 5 Throughput Comparison of Simple TCP, Reno TCP and SACK TCP (Burst size=1, 00,000 Bytes)

In Fig.6, X Axis shows time and Y Axis shows number of bytes received by TCP. Here burst size 40,000 Bytes. Between SACK and SAIMD Scheme proposed in Shihada et. al. [18], SAIMD performs better than SACK TCP. Fig.7 is also the same comparison between SACK and SAIMD Scheme but with Burst size 1, 00,000 Bytes. SAIMD Scheme gives better performance than SACK TCP.

4.1 Throughput Comparison of SAIMD and Our Scheme in OBS Network

Fig. 8 shows SAIMD Scheme and TCP with the proposed Scheme. Observations are that the proposed TCP performs better than SAIMD with burst size = 40,000 Bytes, because the number of false congestion detection is reduced as assembly

time is accurately included in our scheme. Fig. 8, gives us the same comparison results that Our Scheme again performs better than SAIMD Scheme, with burst size=1,00,000 Bytes.

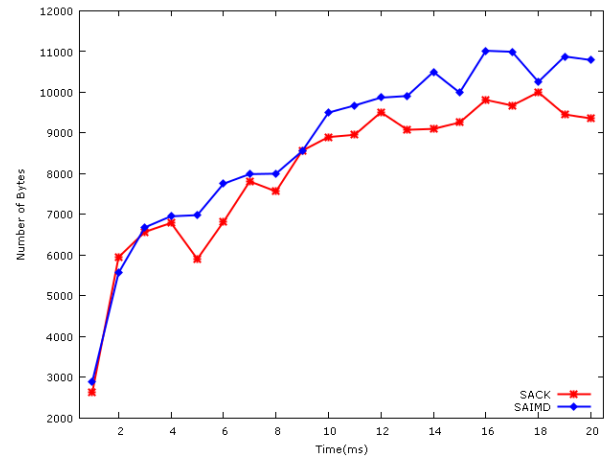


Fig. 6 Throughput Comparison of SACK and SAIMD Scheme (Burst size=40,000 Bytes)

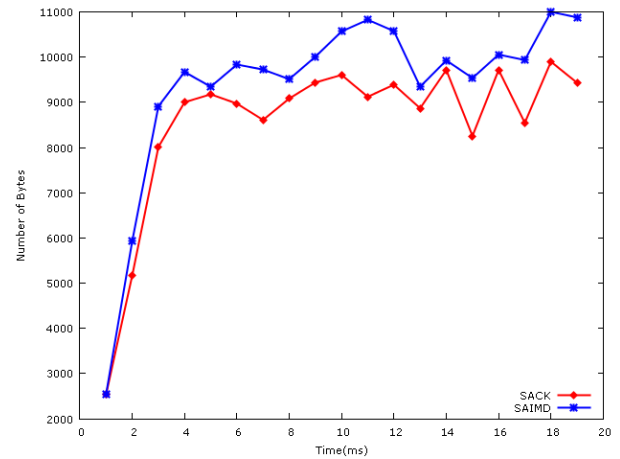


Fig. 7 Throughput Comparison of SACK and SAIMD Scheme (Burst size=1, 00,000 Bytes)

The gain in throughput improves with increase in assembled burst size, reaffirming for dual-threshold or burst-threshold based assembly mechanisms, our congestion detection scheme is more accurate.

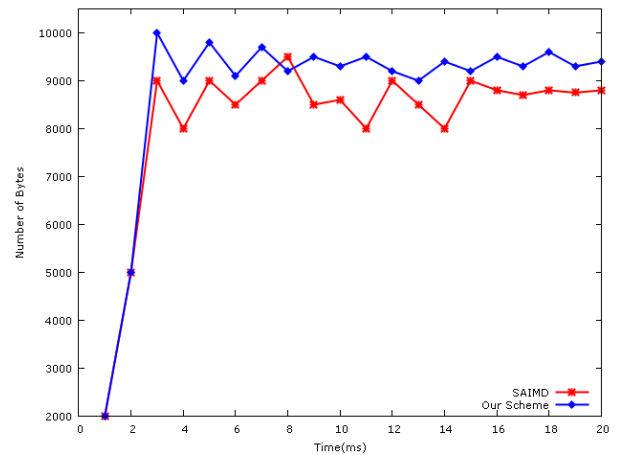


Fig. 8 Throughput Comparison of SAIMD and Our Scheme (Burst size=40,000 Bytes)

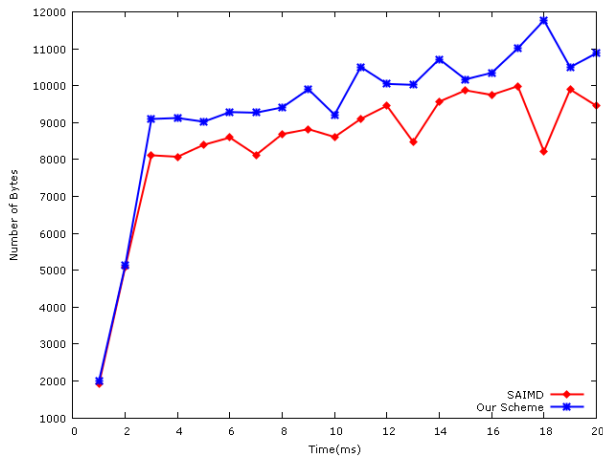


Fig. 9 Throughput Comparison of SAIMD and our Scheme (Burst size=1, 00,000 Bytes)

4.2 Throughput-Cost Comparison of SAIMD and Our Scheme in OBS Network

The throughput-cost comparison of our scheme and SAIMD can be seen in figure 10 and figure 11 with 40,000 bytes 100,000 bytes of data respectively.

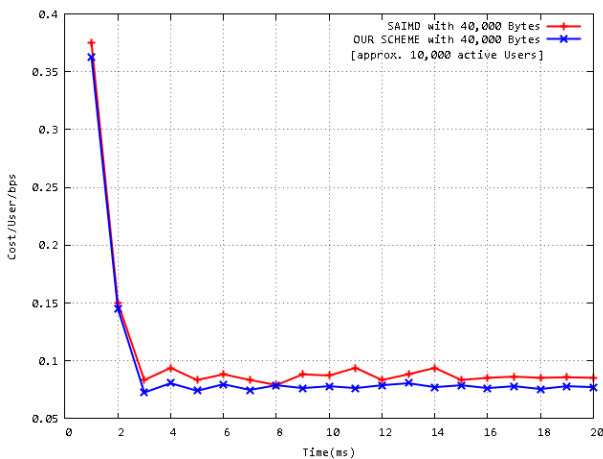


Fig. 10 Throughput-Cost Comparison of SAIMD and our Scheme (Burst size=40,000 Bytes)

Here network cost per user per bps is shown. It indicates that as the throughput increases over time, the savings in terms of cost

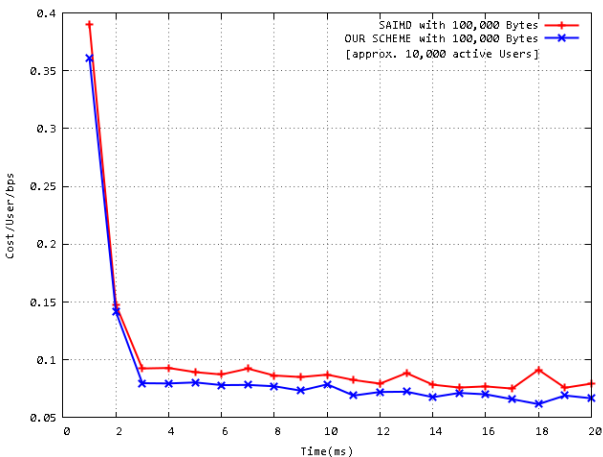


Fig. 11 Throughput-Cost Comparison of SAIMD and our Scheme (Burst size=1, 00,000 Bytes)

per bit increases. As the data rate increases, the average wavelength utilization nears to optimality. The authors in [27] It can also be concluded that our scheme is also cost-efficient when the average user data rate is high and the number of users in the network is large compared to SAIMD. The trends shown in the figure above must hold for a wide range of parameters and can support complex networks.

5. CONCLUSION

Various techniques of differentiating between congestion induced loss and contention induced loss are thoroughly studied, but none of the schemes include the impact of burst assembly time separately. Here in this work, proposed some modification in TCP and BHP structures such that the assembly delay referred by each TCP segment is known to the TCP end points, which is then utilized in making more accurate decision regarding whether a loss is congestion or contention induced. After detection, our scheme follows the cwnd adjustment scheme laid down by the authors in [18]. Simulations were conducted to validate the proposed congestion detection mechanism by comparing it with SAIMD with different network scenarios. Simulation results showed that the proposed scheme outperformed the existing SAIMD implementation. The better understanding of burst assembly and burst transmission behavior through analysis of assembly time and RTT information can give rise to more research opportunities in near future. It is observed that significant gain is achievable using our scheme. This gain is primarily because of success in avoiding misdetection of contention induced loss as congestion induced loss. To our knowledge, this is the first cross-layer technique employed in OBS network to improve the performance of TCP.

6. REFERENCES

- [1] M. Yoo, M. Jeong, and C. Qiao, "A high speed protocol for bursty traffic in optical networks," in SPIEs All-Optical Communication Systems: Architecture, Control and Protocol Issues, 3230, pp. 79-90, 1997.
- [2] M. Yoo and C. Qiao, "A new optical burst switching (OBS) protocol for supporting quality of service," in SPIE Proc., All Optical Commun. Syst.: Architecture, Control Network Issues, vol. 3531, Nov. 1998, pp. 396-405
- [3] C. Qiao and M. Yoo, "Optical burst switching(OBS)-a new paradigm for an optical Internet," Journal of High Speed Networks 8(1), pp. 69-84, 1999.
- [4] X. Chen Y. and Qiao C. Yu., "Study of traffic characteristics of assembled burst traffic in optical burst switched network.", optcomm, (149-159), 2002.
- [5] C. Qiao and M. Yoo, Optical burst switching: A New Area in Optical Networking Research, IEEE Network, pp. 16-23, May/June 2004.
- [6] X. Yu, J. Li, X. Cao, Y. Chen and C. Qiao, "Traffic statistics and performance evaluation in optical burst switched networks.", Journal of Lightwave Technology, DEC 2004.
- [7] J. Qiao C. Xu J. and Xu D Li., "Maximising throughput for optical burst switching networks". IEEE/ACM Transaction on Networking, 2007.
- [8] J. Turner, "Terabit burst switching," Journal of High Speed Networks 8(1), pp. 3-16, 1999.
- [9] J. Mahdavi, S. Flyod, and A Romanow, "Tcp selective acknowledgement options". RFC 2018, 1996.
- [10] R.R.C., Vokkarane V.M., Bikram, "TCP over optical burst switching:to split or not to split?"Journal of Lightwave Technology, 27(22):52085219, Nov 2009.
- [11] T.L. Liu and N.F. Huang C. F. Hsu. "Performance analysis of deflection routing in optical burst switching networks." IEEE INFOCOM, 2000.
- [12] I. Chlamtac et al. cord: "Contention resolution by delay lines," IEEE J. Sel. Areas Commun, 14(10141029), Jun 1996.
- [13] S. Floyd. "Quick-start for TCP and IP". Internet Draft 2002 Online,2000.
- [14] Detti, A., Listanti, M., "Impact of segments aggregation of TCP Reno flows in optical burst switching networks". In: Proceedings of IEEE INFOCOM, pp. 1803-1812 (2002).
- [15] Jayaraj, A., Venkatesh, T., Murthy, C.S.R., "Loss classification in optical burst switching networks using machine learning techniques: Improving the performance of TCP". IEEE Journal on Selected Areas in omunications 26(6), 45-54 (2008)
- [16] Lei Liu, Xiaobin Hong, Jian Wu, Jintong Lin, "Independent acknowledgement assembly and its extended mechanisms to enhance TCP performance in OBS networks", IEEE Communication Letters 13(12),986-988 (2009)
- [17] Shihada, B., Ho, P.H.: "Transport control protocol in optical burst switched networks: Issues solutions, and challenges". IEEE Communications Surveys and Tutorials Second Quarter, 70-86 (2008)
- [18] Shihada, B., Ho, P.H., Zhang, Q., "A novel congestion detection scheme in TCP over OBS networks". Journal of Lightwave Technology 27(4), 386-395 (2009)
- [19] Lee, S.K., Kim, L.Y., "Drop policy to enhance TCP performance in OBS networks". IEEE Communications Letters 10(4), 299-301 (2006)
- [20] S. Peng, Z. Li, Z. Zhang, Y. He and A. Xu, "Modeling and performance analysis of tcp over grid-obs networks.", Journal of Communication Letters, 13(7):540542, 2009.
- [21] Zhu, L., Ansari, N., Liu, J., "Throughput of high-speed TCP in optical burst switched networks". IEEE Proceedings on Communications 152(3), 349-352 (2005)
- [22] X. Yu, C. Qiao, and Y. Liu, "TCP implementation and false time out detection in OBS networks". in the proceedings of IEEE Infocom 2004.
- [23] X. Yu, C. Qiao, Y. Liu, and D Towsley, "Performance evaluation of TCP implementations in OBS networks", Technical Report,2003-13, the State University of New York at Buffalo, 2003.
- [24] Q. Zhang, V. Vokkarane, Y. Wang and J. Jue, "Analysis of TCP over optical burst switched networks with burst retransmission". in the proceedings of IEEE Globecom 2005.
- [25] Q. Zhang, V. Vokkarane, Y. Wang and J. Jue, "TCP over optical burst switched networks with burst retransmission", submitted to IEEE Journal on Selected Areas in Communication(JSAC), 2005.
- [26] 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 Communication Magazine, vol.40, no.9, pp.66-72, 2002.
- [27] G. Weichenberg, Vincent W. S. Chan, and M. M'edard, "On the throughput-cost tradeoff of multi-tiered optical network architectures", Proceedings of the Global Telecommunication Conference 2006, GLOBECOM'06. IEEE 2006.