Node Pacing for Optical Packet Switching

Onur Alparslan, Shin'ichi Arakawa, Masayuki Murata

Graduate School of Information Science and Technology, Osaka University,
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail: {a-onur,arakawa,murata}@ist.osaka-u.ac.jp

Abstract: One of the difficulties of optical packet switched networks is buffering optical packets in the network. Burstiness of Internet traffic causes high packet drop rates and low utilization in very small buffered OPS networks. In this paper, we propose a new node-based pacing algorithm and show that it can increase the utilization of very small optical RAM buffered core optical packet-switched (OPS) networks.

Keywords: small buffer, pacing, WDM, OPS, congestion control

Introduction

Buffering optical packets in the network is one of the difficulties of optical packet-switched (OPS) networks when compared with electronic packet-switched (EPS). In EPS networks, contention of packets is resolved by storing the contended packets in an electronic random access memory (RAM). Electronic RAM allows sending out the packets with *O(1)* reading operation when the output port is free. However, converting optical packets to electrical domain in order to use electronic RAM is not a feasible solution because of the processing limitations of EPS. Processing and switching must be done in the optical domain for high-speed operation.

Many researchers consider FDL buffers to resolve contentions in optical networks, because optical RAM is infeasible or has immature technology. However, optical RAM is under research, for example Takahashi et. al. [1] and NICT project (phase II) [2]. The problem is that optical RAM is not expected to have a large capacity, soon.

Recently, Enachescu et al. [3] proposed that *O(log W)* buffers are sufficient where *W* is the maximum congestion window size of flows when packets are sufficiently paced by modifying TCP senders to used Paced TCP or by using slow access links. However, *O(log W)* buffer size depends on the maximum congestion window size of TCP flows, which may change in time. Also, using slow access links is not a preferred solution when there are applications that require high-bandwidth on the network. Using Paced TCP for these applications by replacing TCP senders with paced versions can be hard.

Ref. [4] proposes applying traffic shaping at edge nodes of OPS network for minimizing traffic burstiness. It proposes a delay-based pacing algorithm that adaptively chooses packet spacing according to input traffic class for achieving bounded delay requirements. Ref. [5] proposes a RC traffic shaper for ATM networks that smoothes the traffic by adjusting the output rate based on the buffer occupancy that depends on the input traffic rate. Output rate is linearly proportional to the shaping buffer occupancy. The problem of the proposed algorithm is that it requires a large buffer for preventing cell loss. Furthermore, peak cell input rate must be known. In order to solve these problems, ref. [6] proposes Interval Filter Shaping Algorithm (IFSA) that smooths cell inter-arrival time with a low latency by a low pass filter and special scheduler.

In this paper, we propose an algorithm that can shape traffic at edge or core nodes by using the buffer occupancy information like the RC traffic shaper. However, our design solves the problems of RC traffic shaper by using a piecewise linear output transfer rate control function and making use of average input traffic rate information calculated inside the node.

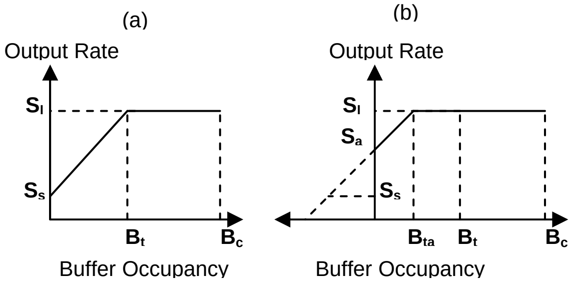

Figure 1 : Transfer functions

2. Architecture

Fig. 1(a) shows our main transfer function for calculating output link pacing rate where B_t is buffer threshold, B_c buffer capacity, S_s is initial link speed and S_l is link capacity. Pacing is applied by changing the spacing between packets. Output rate in RC traffic shaper reaches to the maximum output rate only when the buffer is full. However, in our algorithm output rate reaches to the maximum output rate after a buffer threshold is reached. This provides a safety margin for decreasing buffer overflows in case input traffic rate is higher link speed as there is still free space in the buffer. However, this may not be enough for decreasing average buffer occupancy. When this fixed transfer function is used, an average output traffic rate higher than S_l still requires a non-zero average buffer occupancy that increases packet drop probability, especially in a very small buffer network. For example, an average output traffic rate equal to link capacity requires average buffer occupancy of B_t. In order to solve this problem; we make use of both buffer occupancy and average input traffic rate for calculating output traffic (pacing) rate. We adaptively shift the x axis (buffer occupancy) of the transfer function according to average arrival rate, so that pacing uses less buffer space. Fig. 1(b) shows the adaptive transfer function where S_a is the average input traffic arrival rate and B_{ta} is the new buffer threshold after shifting the transfer function according to S_a. if S_a is smaller than S_l, S_a is taken as S_l. This adaptive transfer function allows output traffic rate being equal to average input traffic rate even when the buffer occupancy is zero, so average buffer occupancy can be decreased. In this paper, core nodes use an input buffering architecture with virtual output queuing (VOQ) scheduling as it has smaller switching fabric size and cost when compared with output buffered and combined input-out buffered switch architectures.

3. Evaluation

Proposed network architecture and algorithms are implemented over ns version 2.31 [7]. Abilene-inspired topology from ref. [8] is used in simulations. The topology has a total number of 869 nodes that consist of 75 center core nodes (C) that are not connected to edge nodes, 106 middle core nodes (M) that connected to edge nodes and 698 edge nodes (E) that IP traffic enters or exits the network topology. Non-paced TCP Reno and Paced TCP Reno are used as traffic sources. A total of 4581 TCP flows start randomly and send traffic between randomly selected edge node pairs. Total simulation duration is 20s. TCP data packet size is 1500Bytes. Propagation delay of edge and core links are 0.1ms and 1ms, respectively. All links have 1Gbps capacity. Core node links are simulated with an optical RAM input buffer size (B_c) of 50Kbits. Electronic RAM input buffer size of edge nodes is 100Mbits (100ms). B_t is always selected as half of the B_c. S_s is selected as 0.1Gbps and 0.01Gbps for core and edge nodes respectively.

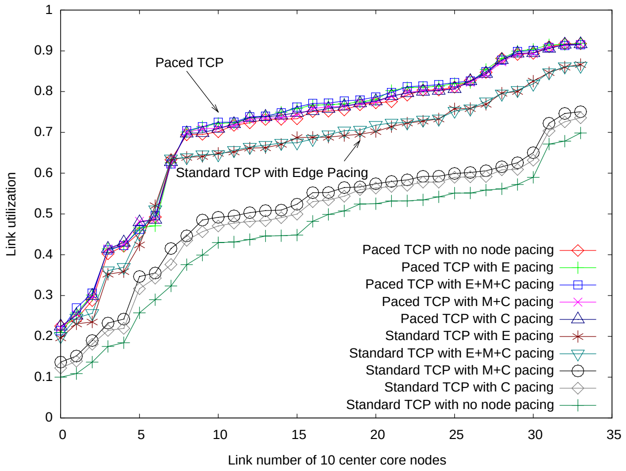

Figure 2 : Simulation Results

Different combinations of edge node pacing (E), middle core node pacing (M), center core node pacing (C) and no node pacing are simulated with Non-paced TCP and Paced TCP flows. Figure 2 shows the sorted utilization of backbone links between 10 center core nodes at the very center of the network at the end of the simulation. In all plots, utilizations are sorted independently from lowest to highest, so there is not a one to one correspondence between link numbers of the plots. When we check the simulation results in Fig. 2, we see that results of Paced TCP traffic are almost the same for different node pacing methods. The reason is that Paced TCP traffic is already smooth enough, so extra node pacing does not bring improvement when Paced TCP is used. The next group of lines show the utilization when non-paced TCP is used with E pacing. Again we see that E pacing makes the traffic smooth enough, so additional C or M pacing do not make a much difference. E pacing paces the traffic without any knowledge of RTT of flows, so its utilization is a bit lower than TCP pacing. The next group of lines is C pacing, and C+M pacing methods where the latter has a bit higher utilization. They have a lower utilization than E pacing, because there are fewer number of core nodes than edge nodes and core nodes have a much smaller buffer (2000 times smaller) than edge nodes that is not enough to fully pace the traffic. The last plot is the simulation result of no pacing that has the lowest utilization as expected. The figure shows that even when node pacing is applied to only core nodes with very small buffers, it is possible to achieve a considerable throughput increase when compared with no pacing case. If node pacing is applied to edge nodes, achievable utilization can be almost doubled at low utilized links, without requiring Paced TCP.

4. Conclusions

Our simulation results show that our edge or core based node pacing algorithm can increase the achievable utilization of very small buffered optical core links or namely throughput of TCP flows using these links. As a future work, we will further evaluate the performance of the proposed algorithm, and effect of simulation parameters.

5. Acknowledgments

This work is supported by the National Institute of Information and Communications Technology (NICT). The work of O. Alparslan was supported by Ministry of Education, Culture, Sports, Science and Technology, Japan.

6. References

[1] R. Takahashi *et. al.*, "Photonic random access memory for 40-Gb/s 16-b burst optical packets," *IEEE Photonics Technology Letters*, vol. 16, pp. 1185–1187, Apr. 2004.

[2] T. Aoyama, “New Generation Network(NWGN) Beyond NGN in Japan,” http://akari-project.nict.go.jp/document/INFOCOM2007.pdf.

[3] M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, “Part III: Routers with very small buffers,” *ACM/SIGCOMM Computer Communication Review*, vol. 35, pp. 83–90, Jul. 2005.

[4] V. Sivaraman, H. Elgindy, D. Moreland, and D. Ostry, “Packet pacing in short buffer optical packet switched networks,” in *Proceedings of IEEE INFOCOM*, 2006.

[5] T.-Y. Tung, Y.-J. Chen, and J.-F. Chang, “Design and analysis of RC traffic shaper,” *IEICE Transactions on Communications*, vol. E81-B, pp. 1–12, 1998.

[6] H. Zhu, Z. Ma, Z. Cao, and Y. Wang, “Low Latency Traffic Interval Shaping Algorithm for Traffic Access Control,” *Chinese Journal of Electronics*, vol. 11, No.2, pp. 247–251, Apr. 2002.

[7] S. McCanne and S. Floyd, “ns network simulator,” Web page:http://www.isi.edu/nsnam/ns/, Jul. 2002.

[8] L. Li, D. Alderson, W. Willinger, and J. Doyle, “A firstprinciples approach to understanding the Internet’s router-level topology,” in *Proceeedings of ACM SIGCOMM*, pp. 3–14, 2004.