Switch Architectures For Small-buffered Optical Packet Switched Networks

Onur Alparslan*, Shin'ichi Arakawa, Masayuki Murata
Graduate School of Information Science and Technology,
Osaka University, 1-3, Yamadaoka, Suita, Osaka 560-0871, Japan
{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. Currently, one solution that can be used for buffering in the optical domain is using long fiber lines called Fiber Delay Lines (FDL). However, FDLs provide only a small and fixed amount of delay. Thus, burstiness of Internet traffic and over-utilizations cause high packet drop rates in small and fixed delayed OPS networks.

Recently, we proposed a new network architecture using a XCP-based congestion control algorithm for OPS WDM networks with pacing at edge nodes for minimizing the buffer requirements at core nodes. In this paper, we investigate input and output optical switch architectures for minimizing the size of optical switching fabric with the proposed network architecture. We show the number of FDLs and switch size requirements of architectures depending on FDL granularity and packet size distribution.

1 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, there is no equivalent optical RAM available for O(1) reading operation. Converting optical packets to electrical domain in order to use electronic RAM is not a feasible solution because of the processing limitations of EPS. Therefore, processing and switching must be done in the optical domain for high-speed operation. Currently, one solution that can be used for buffering in the optical domain is using long fiber lines called Fiber Delay Lines (FDL). However, FDLs require long fiber lines, which cause signal attenuation. Furthermore, FDLs provide only a fixed amount of delay and there can be a limited number of FDLs in a router due to space, power and cost considerations.

According to a rule-of-thumb, buffer size of each output link of a router must be $B = RTT \times BW$, where RTT is the average round trip time of flows and BW is the bandwidth of output link, in order to achieve high utilization with TCP flows. Appenzeller et al. [1] showed that a buffer sized at $B = \frac{RTT \times BW}{\sqrt{n}}$, where n is the number of TCP flows passing through the link, is enough for achieving high utilization. However, this buffer requirement is still high for high speed OPS routers with very small amount of buffering capacity.

Recently, Ref. [2] proposed that $O(\log W)$ buffers are sufficient where W is the maximum congestion window size of flows when TCP flows are paced [3] and the link is under-utilized. The buffer size depends on the maximum congestion window size TCP flows. Ref. [2] proposes pacing by using Paced TCP or using access links much slower than OPS core links. Replacing TCP senders with paced versions can be hard. Also using slow access links is not a preferred solution when there are applications that require high-bandwidth on the network. Therefore, it may be better to design a general architecture for OPS network that

- can achieve high utilization in a small buffered OPS network independent of the number of TCP or UDP flows,
- does not require limiting the speed of access links,
- does not require replacing sender or receiver agents of computers using the network.

Applying pacing to the input traffic at the edge nodes of an OPS network can be a good choice for achieving these goals. Even if TCP pacing is applied at the clients, the aggregated traffic arriving to the OPS network may end up behaving bursty. Therefore, pacing at the edge of OPS network is more effective on

*This work was supported in part by National Intitute of Information and Communications Technology of Japan (NICT). The work of O. Alparslan was supported by Ministry of Education, Culture, Sports, Science and Technology, Japan.

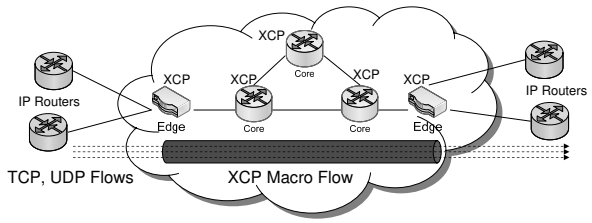


Figure 1. XCP macro flows

minimizing burstiness of traffic entering the OPS domain.

In Ref. [4], we introduced an all-optical OPS network architecture that can achieve high utilization and low packet drop ratio by using FDL-based small buffering. In our architecture, we consider an OPS domain where packets enter and exit the OPS domain through edge nodes. We proposed using a XCP-based [5] intra-domain congestion control protocol for achieving high utilization and low packet drop ratio with small FDL buffers. We showed that XCP can be used for controlling and limiting the utilization level of each wavelength. Selecting a target wavelength utilization less than actual wavelength capacity in XCP control algorithm can prevent queue buildups and allow operating at a utilization level that can give a low packet drop rate for a selected FDL granularity as shown in [4]. In our architecture, if there is traffic between an edge source-destination node pair, a rate-based XCP macro flow is created, and incoming TCP and UDP packets of this edge pair are assigned the XCP macro flow as shown in Fig. 1. The edge nodes apply leaky-bucket pacing to the macro flows by using the rate information provided by XCP for minimizing the burstiness. Variable sized IP packets using variable number of slots enter OPS network without any assembling.

Switching fabric size is an important cost factor in routers. Many switching fabric architectures like MEMS, optomechanical, electrooptic, thermooptic, liquid-crystal based switches are proposed for optical switching [6]. However, the number of switching elements in the fabric increases together with the overall cost as the number of ports of the switch increases. Also increasing the switch size introduces high crosstalk and insertion losses in many proposed switching fabric architectures. These losses require optical amplification that further increases the overall cost as explained in [6]. In [4], a simple output FDL buffered switch was used as switching architecture. In this paper, we investigate and compare input and output buffered optical switch architectures for minimizing the size of optical switching fabric of core nodes while achieving higher throughput with small buffers. For this purpose we apply the proposed FDL-based small-buffered network architecture. We show how the FDL requirements of different switch architectures change with FDL granularity and packet size distribution by using a star topology.

The rest of the paper is organized as follows. Section 2 describes the basics of XCP algorithm, and switch and FDL architectures proposals, and details of proposed algorithm. Section 3 describes the simulation methodology and presents the simulation results on star topology. Finally, we conclude in Section 4.

2 Architecture

2.1 XCP Basics

XCP is a new congestion control algorithm specifically designed for high-bandwidth and large-delay networks. XCP makes use of explicit feedbacks received from the network. Core routers are not required to maintain per-flow state information. Each XCP core router updates its control decisions calculated by an Efficiency Controller and a Fairness Controller when timeout of a per-link control-decision timer occurs.

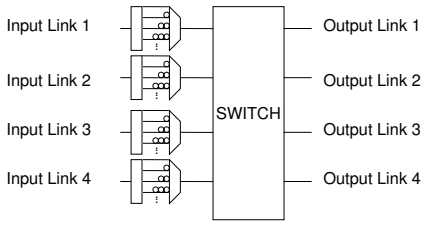
Efficiency Controller (EC) controls the input aggregate traffic in order to maximize link utilization. A desired increase or decrease in aggregate traffic for each output port is calculated by using the equation $\Phi = \alpha \cdot S - \beta \cdot Q/d$, where Φ is the total amount of desired change in input traffic, α and β are spare bandwidth control and queue control parameters, respectively and d is the control decision interval. S is the spare bandwidth that is the difference between the link capacity and input traffic in the last control interval. Q is the persistent queue size.

After calculating the aggregate feedback Φ, Fairness Controller (FC) fairly distributes this feedback to flows according to an AIMD-based control. However, convergence to fairness may take a long time when Φ is small. Bandwidth shuffling, which redistributes a small amount of traffic among flows, is used in order to solve this problem. Amount of shuffled traffic is calculated by $h = max(0, \gamma \cdot u - |\Phi|)$, where γ is the shuffling parameter and u is the rate of aggregate input traffic in the last control interval.

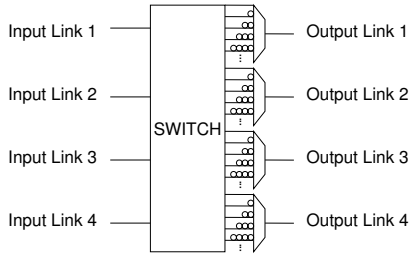
2.2 Switch, Scheduler and FDL Architectures

In [4], we evaluated the FDL requirements of an output buffered switching architecture where FDL lines are connected to the output ports of the switch as shown in Fig. 2(b) for a single wavelength. If there are many fiber delay lines per output link, such a switch requires many output ports and therefore a big switching fabric However, switching fabric size is usually one of the biggest factors determining overall router cost, so in this paper we try to decrease the size of the switching fabric.

In [4], output buffering without void filling was used as the buffering architecture and scheduling al-



(a) Input Buffering



(b) Output Buffering

Figure 2. Switch and FDL architectures

gorithm. In this paper we evaluate the switch size and buffer requirements of a different architecture called input buffering with virtual output queuing (VOQ) scheduling shown in Fig. 2(a) for a single wavelength. We also evaluate the buffer requirements of void filling scheduling version of output buffering for comparision. Speedup is 1 in all switches.

Buffers are implemented as a single stage equidistant fiber delay lines like in [4]. FDL length distribution increases linearly (x, $2x$, $3x$, $4x \ldots$) where x is FDL granularity. The number of required FDLs (denoted by B) is evaluated for different FDL granularities. When output buffering is used, required switch size for a single wavelength is $N \times BN$, where N is the number of links assuming the number of output and input links is the same, as seen in Fig. 2(a). On the other hand, input buffering decreases the main switch size to $N \times N$ independent of the number of delay lines. Each input link requires a $1 \times N$ small switch in front of its FDL set. Therefore, input buffering can be implemented by dividing the switching fabric into smaller switches instead of a single and large main switch. This may bring a drastic decrease in switching fabric cost especially if B is high. However, a well-known problem of input buffering is head-of-line blocking, which limits the achievable utilization. We apply virtual output queuing (VOQ) scheduling for minimizing this problem.

A FDL set provides only a limited set of required delays, unless granularity is a single slot. When the required delay is not supported by the FDL set, packets may end up to be delayed more than the required delay. Extra delaying the packets causes unused void slots, which decrease the achievable throughput of output links. Void filling scheduling algorithms decrease the number of such unused slots and decrease the FDL requirements. However, void-filling algorithms increase the scheduler complexity, so a simple output buffering architecture without in output buffering was used in [4]. In this paper, we evaluate a void filling version of output buffering architecture for a more fair comparison with input buffering architecture where void filling is necessary for VOQ. Void filling algorithms may cause packet reordering, so they must be carefully applied. We prevent packet reordering among the packets that will be switched through the same input-output link pairs in both input buffering and void filling version of output buffering.

2.3 Rate-based Paced XCP

In [4], we proposed Optical Rate-based Paced XCP as an intra-domain traffic shaping and congestion control protocol, which is similar to TeXCP [7] in electronic networks, in an OPS network domain. In this architecture, XCP sender agent on an ingress edge node multiplexes incoming flows destined to the same egress edge node and creates a macro flow as shown in Fig. 1, and applies pacing with rate control to the macro flow according to XCP rate calculation.

XCP feedbacks of OPS edge nodes are carried in separate probe packets that XCP sender agents send only once in every control period. There is no feedback information carried in header of data packets, so there is no need for calculating a per-packet feedback in core routers unlike in original XCP [5]. We are separating the control channel and data channels. Probe packets are carried on a separate single control wavelength that is slow enough for carrying only probe packets. Low transmission rate of control wavelength allows applying electronic conversion for updating the probe feedback and buffering the probe packets in electronic RAM in case of a contention.

When a probe packet of macro flow i arrives to a core router, the XCP agent responsible for controlling the wavelength of macro flow i calculates a positive feedback p_i and a negative feedback n_i for macro flow i. Positive feedback is calculated by $p_i = \frac{h+max(0,\Phi)}{N}$ and negative feedback is calculated by $n_i = \frac{u_i \cdot (h+max(0,-\Phi))}{u}$, where N is the number of macro flows on this wavelength, u_i is the traffic rate of flow i estimated and sent by the XCP sender in the probe packet and h is the shuffled bandwidth. N can be estimated by counting the number of probe packets received in the last control interval. Another possible method is using the number of LSPs if GMPLS is available [7]. Control interval is the maximum RTT in the network. Control interval can be selected a bit longer than the maximum RTT for in order to compensate for processing and buffering delays of control packets. Feedback, which is the required change in the flow rate, is calculated by $feedback = p_i - n_i$. If this feedback is smaller than the one in the probe packet, core router replaces the feedback in the probe packet with its own feedback. Otherwise, core router does not change the feedback in the probe packet.

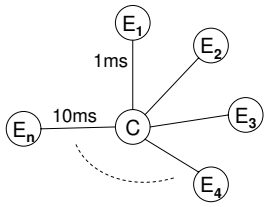


Figure 3. Star topology

As explained in Sec. 2.1.1, wavelength capacity must be explicitly given to XCP algorithm for calculating S. If we give a false virtual capacity value less than actual wavelength capacity, XCP algorithm converges to the given virtual capacity and causes underutilization. It is possible to make use of this property of XCP for operating OPS network at a limited utilization level that provides low packet drop ratio. We call the ratio of given virtual capacity and actual capacity as target utilization.

3 Evaluation

3.1 Simulation Settings

Proposed network architecture and buffering models are implemented over *ns* version 2.28 [8]. XCP agents start sending data randomly in the first 10s and continue until the simulation ends. It is assumed that there is a backlogged traffic at edge buffers, so each edge node sends traffic to all other edge nodes at the maximum possible rate controlled by XCP. We chose XCP's α, β and γ parameters for edge routers as 0.2, 0.056 and 0.1, respectively, as explained and used in [4]. However, input buffering architecture implemented by FDLs makes it hard to provide buffer occupancy data to XCP algorithm. Furthermore, our aim is to have a small buffered network and effect of queue parameter is low as persistent queue size is usually small with such a small buffered network, so β parameter is set to zero in the core routers. [9] shows that this parameter set is stable. Total simulation duration is 40s.

Slot size is selected as 52Bytes, because Ref. [10] shows that most common small packets on Internet2 are in the range of 40Bytes to 52Bytes. The selection of optimum slot size is left as a future work. Probe packet size is selected as equal to the slot size. FDLs are used for resolving contention of data packets. Contention of probe packets on control wavelength is resolved by electronic RAM as explained in section 2.3. Ref. [10] shows that size of packets in Internet2 traffic is mainly composed of very small and big packets and there is around 3:2 ratio between these two, so this packet size distribution is used in the simulations as a realistic packet size distribution. Simulated packet size distributions are

- All packets are 1 slot (52 Bytes) size
- All packets are 29 slots (1508 Bytes) size
- 60% of packets are 1 slot (52 Bytes) size , 40% of packets are 29 slots (1508 Bytes) size (realistic traffic)

The star topology shown in Fig. 3 is used for computer simulations. There is a single core node for switching the packets. Star topology is simulated when there are 12 edge nodes in the network. Each source node sends data to all other edge nodes, so each link carries 11 macro flows (LSPs) in each direction. Simulated FDL granularities range between 1 to 100 slots. Target utilization parameter of XCP is set to 30% for output links of core node as Ref. [1] states that network operators usually run backbone links at loads of 10%-30%. Target utilization is set to 90% for output links of edge nodes as they can use electronic RAM for buffering.

There is a single data wavelength on links. Propagation delay of links range between 1ms and 10ms in the network. XCP control period of core routers and probe packet sending interval of edge routers is 40ms. The capacity of the data wavelength is set to 1Gbps when packets are 29 slots size and realistic packet size distribution. When all packets are 1 slots size, wavelength capacity is set to 100Mbps due to simulation time constraints. The capacity of the XCP control wavelength is 100Mbps.

Figure 4 shows the aggregate packet drop rate in the simulations. In all subplots, y-axis shows the limit of the number of delay lines per link and x-axis is the aggregate packet drop rate in the core, both in log scale. *G* lines in the figure show the applied FDL set granularity. When we compare the simulation results of input buffering and output buffering in Fig. 4, we see that the delay line requirements for the same packet drop rate are close, especially for the high granularities when packets are big and all granularities when all packets are 52Bytes. FDL requirements of input buffering is a bit higher.

In the graph, we see that granularities between 1-50 slots in big size packet simulations and granularities between 1-3 slots in simulation of 52Bytes packets show a sharp decrease in drop rate as the number of delay lines increase. However, if we increase the granularity, the drop rate decreases first and then becomes almost constant or decreases with a lower rate, because void slots in FDLs and output due to high FDL granularity causes synchronized packet drops and limits the achievable utilization as explained in [4].

Packet size is distribution in the network is an important factor on the selection FDL granularity and achievable utilization. If we want a network to have very low packet drop rate, it is necessary to select the FDL granularity according to the worst case scenario that is the case of all packets in the network have the minimum possible size, which is taken as 1

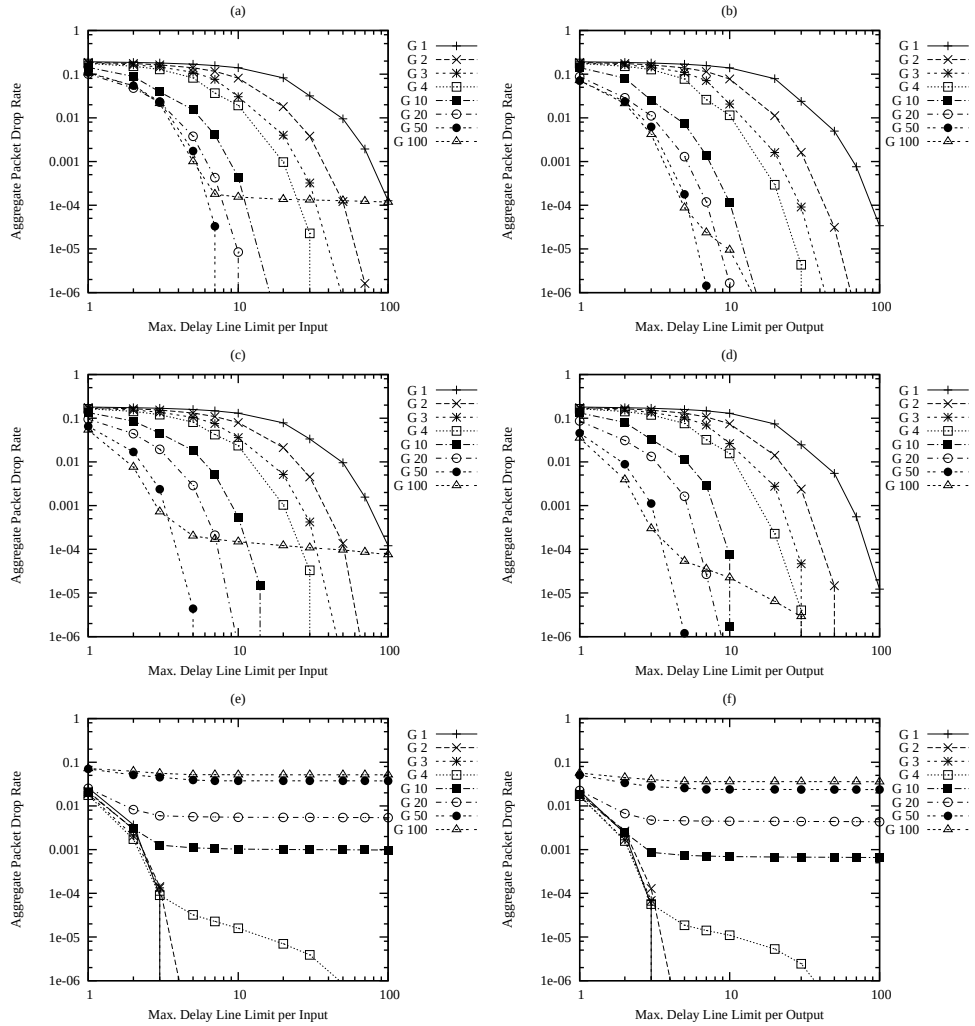


Figure 4. Aggregate packet drop rate with limited number of FDLs per link when packet size distribution is realistic packet size distribution with input buffering (a) and output buffering (b), all 1508Bytes with input buffering (c) and output buffering (d), all 52Bytes with input buffering (e) and output buffering (f)

slot size in the simulations this paper. After selecting an FDL granularity that can achieve the target utilization and required packet drop ratio with small packets, the number of delay lines of the switch can be evaluated and selected according to the FDL requirements in simulation of a traffic composed of big packets and required packet drop ratio. For example, Fig. 4(e,f) show that FDL granularities of 1, 2 and 3 slots can achieve very low drop rate when all packets are 1 slot size. When we check the FDL requirements of these granularities with simulation of big packets in Fig. 4(a,b,c,d), we see that it is possible to get low packet drop rate with around 30-40 delay lines per link with granularity of 3 slots. On the other hand, around 1% drop rate may be enough for internet traffic. In such a case, using as low as 4-5 delay lines per link with granularity of 20 slots looks enough for all simulated packet size distributions.

4 Conclusions

In this paper, we investigated some optical switch architectures for minimizing the size of optical switching fabric with the proposed network architecture based on pacing the traffic. We compared the buffering architectures input buffering with VOQ, and output buffering with void filling. We evaluated the packet drop rates depending on FDL granularity and packet size distribution.

We showed that input buffering requires comparable number of delay lines as output buffering architectures at 30% utilization, which is typical for backbone links of network operators, with pacing. Input buffering can be implemented by dividing the switching fabric into smaller switches instead of a single and large main switch, so input switching may decrease costs when the cost of a large and single switch is higher. The drawback of input buffering is that its scheduling

algorithm is more complex than scheduler of output buffering, but processing power requirements of input buffering may be decreased with some optimizations. As a future work, we will evaluate the performance of switches with more realistic traffic with TCP and UDP flows on big mesh topologies and also compare other possible switch architectures.

References

[1] G. Appenzeller, N. McKeown, J. Sommers, and P. Barford, "Recent results on sizing router buffers," in *Proceedings of the Network Systems Design Conference*, Oct. 2004

[2] 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.

[3] L. Zhang, S. Shenker, and D. Clark, "Observations on the dynamics of a congestion control algorithm: The effects of two-way traffic," in *Proceeedings of ACM SIGCOMM*, pp. 133–147, Sept. 1991.

[4] O. Alparslan, S. Arakawa, and M. Murata, "Optical rate-based paced XCP for small buffered optical packet switching networks," in *Proceedings of PFLDnet*, pp. 117–124, Feb. 2006.

[5] D. Katabi, M. Handley, and C. Rohrs, "Internet congestion control for future high bandwidth-delay product environments," in *Proceedings of ACM SIGCOMM*, Aug. 2002.

[6] G.I. Papadimitriou, C. Papazoglou, and A.S. Pomportsis, "Optical switching: Switch fabrics, techniques, and architectures," *Journal of Lightwave Technology*, vol. 21, no. 2, pp. 384–405, 2003.

[7] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in *Proceedings of ACM SIGCOMM 2005*, Aug. 2005.

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

[9] H. Balakrishnan, N. Dukkipati, N. McKeown, and C. J. Tomlin, "Stability Analysis of Explicit Congestion Control Protocols," Stanford University Department of Aeronautics and Astronautics Report: SUDAAR 776, Sep. 2005.

[10] S. Shalunov and B. Teitelbaum, "Bulk TCP use and performance on Internet2," http://people.internet2.edu/ ben/ papers/i2tcp-imeas2001.pdf, Aug. 2001.