Rate-based pacing for small buffered optical packet-switched networks

Onur Alparslan,* Shin'ichi Arakawa, and Masayuki Murata

Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 560-0871, Japan
**Corresponding author: a-onur@ist.osaka-u.ac.jp*

Received May 31, 2007; revised June 30, 2007; accepted July 6, 2007; published August 16, 2007 (Doc. ID 83587)

One of the difficulties of optical packet-switched (OPS) networks is buffering optical packets in the network. O(1) reading operation is not possible in the optical domain, because there is no equivalent optical RAM available for storing packets. Currently, the only available solution that can be used for buffering in the optical domain is using long fiber lines called fiber delay lines (FDL). However, FDLs have important limitations and may cause high packet drop rates due to the burstiness of Internet traffic. We propose an architecture using an explicit congestion control protocol (XCP) based utilization control algorithm designed for OPS wavelength-division-multiplexing (WDM) networks with pacing at the edge nodes for decreasing the buffer requirements at core nodes. We evaluate the FDL requirements on a meshed network with multiple-hop paths and show how FDL requirements change with slot size, utilization, FDL granularity, scheduling, and packet size distribution.

OCIS codes: 060.1810, 060.4250, 060.4510.

1. Introduction

Optical packet-switched (OPS) networks have some major differences and limitations when compared with electronic packet-switched (EPS) networks. One of the difficulties of OPS networks is buffering optical packets in the network. In EPS networks, contention is resolved by storing the contended packets in random access memory (RAM) and sending out the packets with O(1) reading operation when the output port is free. However, the operation is not possible in the optical domain, because there is no equivalent optical RAM available for storing packets. Converting packets from optical domain to electronic domain in order to use electronic RAM is not a feasible solution because of the processing limitations of EPS. Current electronic devices are not fast enough to process the data at the ultra high speed of optical networks. Therefore, processing and switching in the optical domain is necessary.

Currently, the only solution that can be used for buffering in the optical domain is using long fiber lines called fiber delay lines (FDL). Contended packets are switched to FDLs in order to be delayed. However, FDLs have important limitations. First of all, FDLs require very long fiber lines, which cause signal attenuation inside the routers. There can be only a limited number of FDLs in a router due to space considerations, so they can provide only a small amount of buffering. Second, FDLs provide only a fixed amount of delay.

Having a very small buffering capacity and a lack of variable delay buffering brings some important performance problems to OPS networks. According to a rule-of-thumb [1], an output link of a router needs a buffer sized at $B=\mathrm{RTT}\times BW$, where RTT is the average round-trip time of flows and BW is the bandwidth of the output link, in order to achieve high utilization with TCP flows. Recently, Appenzeller *et al.* [2] showed that when there are many TCP flows sharing the same link, a buffer sized at $B=(RTT\times BW)/\sqrt{n}$, where n is the number of TCP flows passing through the link, is enough for achieving high utilization. However, a significant decrease in buffer requirements is possible only when there are many flows on the link. This buffer requirement is still high for high-speed OPS routers with a very small amount of buffering capacity. Further decreasing the buffer requirements is necessary. However, the bursty nature of TCP flows causes a high packet drop rate in small buffered networks and limits further decreasing the buffer size.

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 [4] 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. Furthermore, this proposal was based on the assumption that most of the IP traffic is from TCP flows. A recent paper [5] shows that even small quantities of bursty real-time traffic can interact with well-behaved TCP traffic and increase the buffer requirements.

It may be better to design a general architecture for OPS networks 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 an OPS network is a more effective way to minimize the burstiness of traffic entering the OPS domain.

Reference [6] proposes applying traffic shaping at the edge nodes of an 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. Reference [7] proposes traffic shaping at the edge nodes by using a modified form of renegotiated service with rate prediction to reduce contention in the core. Edge nodes use packet scheduling to rearrange possible contended slots before entering the core, thus reducing core optical buffering. Proposed architecture needs information about relevant network scheduling.

In [8], we introduced an all-optical OPS network architecture that can achieve high utilization and a low packet drop ratio by using FDL-based small buffering. We considered an OPS domain where packets enter and exit the OPS domain through edge nodes. We proposed using an explicit congestion control protocol (XCP) based [9] intradomain congestion control protocol for achieving high utilization and a low packet drop ratio with small FDL buffers. XCP [9] is a new congestion control algorithm using a control theory framework. XCP was specifically designed for high-bandwidth and large-delay networks. XCP was first proposed in [9] as a window-based reliable congestion and transmission control algorithm. XCP framework is selected because XCP framework allows the individual control of the utilization level of each wavelength. We showed that selecting a target wavelength utilization less than the actual wavelength capacity in an XCP control algorithm can allow operating at a utilization level that can give a low packet drop rate for a selected FDL granularity as shown in [8]. 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, similar to TeXCP [10]. The edge nodes of an OPS network apply leaky-bucket pacing to the macro flows by using the rate information provided by XCP for minimizing the burstiness. As a result, there is no need to modify the TCP and UDP agents of computers or limit the speed of access links for decreasing burstiness.

In [8], we introduced some preliminary buffer requirement results for only a very low packet drop rate on a simple dumbbell topology. In this paper, we evaluate the

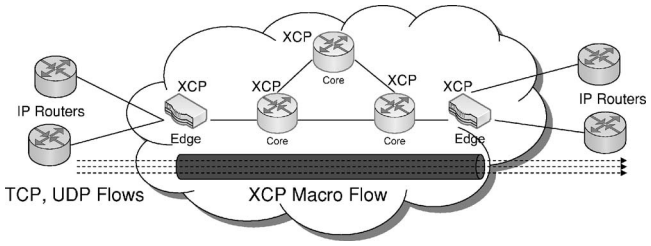

Fig. 1. XCP macro flows.

FDL requirements and packet drop rates on a realistic mesh NSFNET topology with multiple-hop paths. We show how FDL requirements change with utilization, FDL granularity, scheduling, and packet size distribution.

The rest of the paper is organized as follows: Section 2 describes the FDL architecture used in our architecture and the effects of voids and slot size. Section 3 describes the basics of the XCP algorithm, variants of the XCP, and details of the proposed algorithm. Section 4 describes the simulation methodology and presents the simulation results on NSFNET. Finally, we conclude in Section 5.

2. FDL Architecture, Slots Size, and Voids

The FDL architecture used in this paper is a single stage equidistant FDL set with B delay lines. Switch and FDL architecture are shown in Fig. 2 [11].

In the FDL architecture, the length of the delay lines will be given in terms of slot number. FDL length distribution increases linearly $(x, 2x, 3x, 4x, \ldots)$ where x is FDL granularity. The number of required FDLs (denoted by B) will be evaluated for different FDL granularities. Minimizing the number of delay lines in a switch is necessary because the size and the cost of the switching fabric increases as the number of delay lines increase. The size of the switching fabric is the main cost factor of routers. Increasing the granularity can decrease the number of required FDL lines and therefore decrease the cost of the switch. However, selecting a too high granularity may increase the packet loss rate as we will show in the simulations.

We are using the slotted variable-length packet approach (SVLP) [12] for adapting asynchronous, variable-length packets coming from the electrical domain to the synchronous and slotted OPS network. Variable-sized IP packets divided into a variable number of slots enter the OPS network as a train of slots without any burst assembling. OPS routers switch the slot train of a packet as a whole and sequentially. Using a slotted architecture decreases the packet drop rate, but requires slot synchronization before switching.

In this approach, using FDLs and a slot-based architecture may cause voids, which decreases the effective throughput of output links. The size of voids depends on the slot size, so slot size is an important design parameter that affects the effective utilization and buffer requirements. Voids in the architecture can be classified into two groups.

2.A. Voids in Slots

Voids in slots occur when the packet size is not equal to a multiple of the slot size. In this case, padding is applied to the last slot. Padding decreases the utilization efficiency depending on the packet size distribution and the slot size. For example, if the

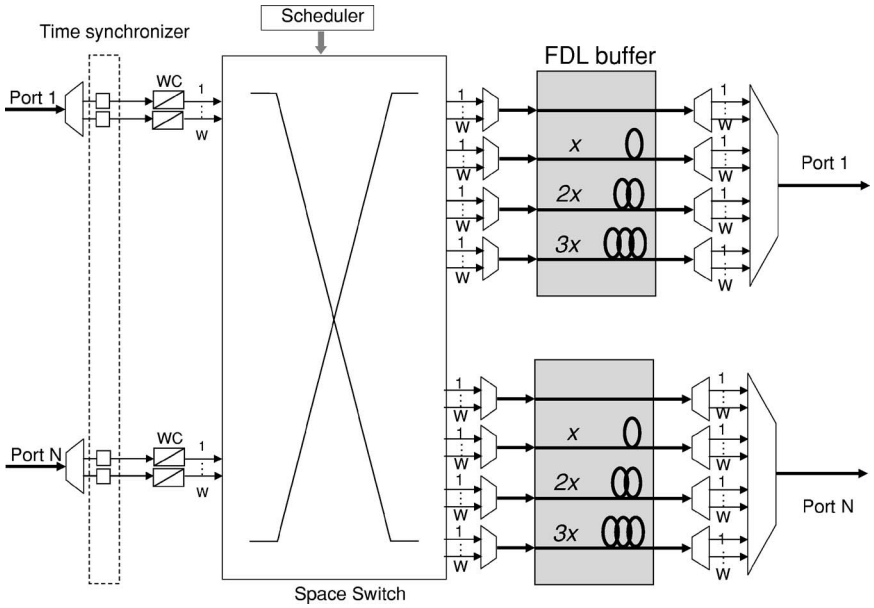

Fig. 2. Switch and FDL architecture.

slot size is 500 bytes and if a 501 byte packet arrives, the packet will be carried in two slots with a total length of 1000 bytes. There will be a 409 byte void due to padding in the second slot. Using a small slot size can increase the efficiency in general due to less padding. However, an efficiency increase may not be high at a too small slot size since guard bands and slot headers start becoming the main source of the decrease in efficiency. Furthermore, buffer requirements may increase as the switching operation becomes similar to an unslotted network.

If we select a large slot size, effective throughput is low when the average size of arriving packets is small. For example, if the slot size is 1500 bytes and a 40 byte packet is carried in the slot, 97.3% of the slot is wasted due to padding (void).

Selecting the right MTU for the OPS link layer is important. Selecting an MTU less than the slot payload size does not make sense, because a single packet can never fully utilize the slot capacity. We do not consider burst assembling, so assembling multiple packets in order to fill in voids in a slot is not a choice. The MTU can be selected as any value equal to or bigger than the slot capacity as SVLP-based adaptation divides variable-sized IP packets into a variable number of slots transparently. On the other hand, the extreme case of using only a single slot for carrying a whole packet without dividing them into slots by selecting the slot payload size and the MTU of the OPS link layer equal to each other can greatly simplify the FDL design process, as FDL granularity can be directly selected as one slot size independent of the packet size distribution. Also, scheduling algorithms become simpler because there is no need to take into account the length of a train of slots as all packets are inside a single slot [12] and there is no need for void-filling scheduling when FDL granularity is one slot size.

2.B. Void Slots in FDLs Between Packets

When the slot size is small, buffer requirements can be decreased by increasing the FDL granularity. However, if the FDL granularity is larger than a single slot, unused void slots may occur in FDLs. FDL sets with granularity larger than a single slot can provide only a limited set of required delays. For example, an FDL set with a granularity of two slots can delay the packets by 2, 4, 6, 8, … slots, but cannot delay the packets by 1, 3, 5, 7, … slots. When the required delay is not supported by the FDL set, the FDL set may delay a packet more than the required delay. In this case, extra delaying of the packets causes unused void slots in FDLs and output. Using a void-filling scheduling algorithm can fill in some of the void slots. However, void-filling algorithms increase the scheduler complexity. A simple output buffering architecture without void filling in output buffering was used in [8]. Furthermore, void-filling algorithms may cause packet reordering, so they must be carefully applied. We use a void-filling algorithm that prevents packet reordering among input–output ports of the switch. The algorithm keeps a list of the departure times of the last buffered packets destined from each input port to each output port. Also, it keeps a list of the voids in the buffer reservation table of each output port. When there is a packet to be buffered, in order to prevent reordering we find the first void in the buffer reservation table that starts after the departure of the last buffered packet using the same input–output pair. Then starting from this void we search for the first void that the packet fits into by using the FDL set. If there is no suitable void, the packet is scheduled after the last packet in the buffer. We show and compare the simulation results both with and without void-filling scheduling.

Voids are shown as an example in Fig. 3. FDL granularity is selected as four slots. $m+1$st packet contends with mth packet, so $m+1$st packet must be delayed by five slots for solving the contention. However, an FDL set with a granularity of four slots can delay the packets by only 4, 8, 12, … slots, so the packet is delayed by eight slots instead of five. Therefore, three void slots occur between mth and $m+1$st packets. Furthermore, the size of these two packets is not equal to a multiple of the slot size, so voids occur inside their last slots.

3. XCP Control

3.A. 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 net-

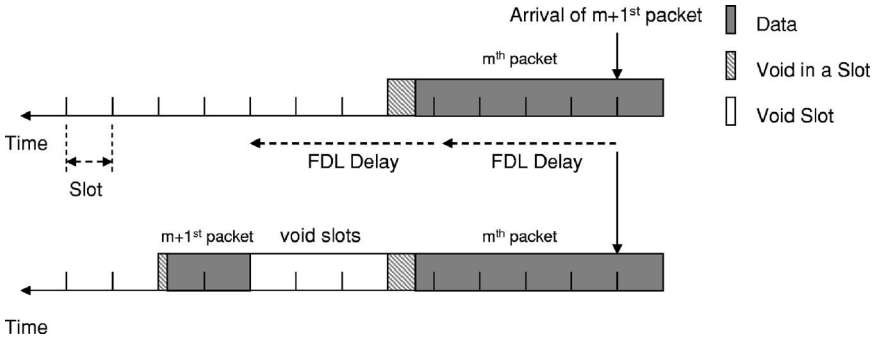

Fig. 3. Voids.

work. It decouples the utilization control from the fairness control. Core routers are not required to maintain per-flow state information.

The working principle of XCP is as follows: XCP core routers maintain a per-link control-decision timer. When a timeout occurs, core routers update their link control parameters calculated by the efficiency controller (EC) and the fairness controller (FC) according to link utilization, spare bandwidth, and buffer occupancy. An XCP sender agent sends its traffic rate to an XCP receiver in the header of data packets or a probe packet. On the way to the XCP receiver, XCP core routers read this information and calculate feedback that shows the desired traffic rate change for this XCP flow and updates the feedback header of the packet. The XCP receiver simply sends back the final feedback to the XCP sender. The XCP sender updates its sending rate according to this feedback.

3.A.1. EC

The EC is responsible for maximizing link utilization by controlling aggregate traffic. Every router calculates a desired increase or decrease in aggregate traffic for each output port by using the equation $\Phi = \alpha S - \beta Q/d$. In this equation, Φ is the total amount of desired change in input traffic; α and β are spare bandwidth control parameter and queue control parameter, 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.

3.A.2. FC

After calculating the aggregate feedback Φ, the FC is responsible for fairly distributing this feedback to flows. The FC uses an additive increase multiplicative decrease (AIMD)-based control for distributing the feedback. It means that when Φ is positive, the FC increases the transmission rate of all flows by the same amount. When Φ is negative, the FC decreases the transmission rate of each flow proportional to the flow's current transmission rate. However, when Φ is small, the convergence to fairness may take a long time. Furthermore, if Φ is zero, XCP stops converging. To prevent this problem, bandwidth shuffling, which redistributes a small amount of traffic among flows, is used. This shuffled traffic is calculated by $h = \max(0, \gamma u - |\Phi|)$, where γ is the shuffling parameter and u is the aggregate input traffic rate in the last control interval.

3.B. XCP Variants

XCP was first proposed in [9] as a window-based reliable congestion and transmission control protocol. The same paper also proposes a XCP-based core stateless fair queuing as a gradual deployment method. The XCP-based core stateless fair queuing algorithm creates an XCP macro flow, and assigns the TCP and UDP flows to the XCP macro flow. The edge nodes forward the TCP and UDP packets inside the XCP macro flow according to the XCP macro flow rate. Reference [9] states that the algorithm can be further simplified by using special probe packets for receiving the feedback for the macro flow rate calculation instead of attaching a congestion header to forwarded packets.

TeXCP [10] is a traffic engineering protocol using a rate-based XCP congestion control allowing traffic engineering by applying load balancing with multipath routing.

TeXCP creates a macro flow and assigns TCP and UDP flows to this macro flow and forwards the packets according to the XCP flow rate similar to simplified XCP-based core stateless fair queuing.

3.C. Rate-Based Paced XCP

In [8], we proposed rate-based Paced XCP as an intradomain traffic shaping and congestion control protocol in an OPS network domain. In this architecture, the XCP sender agent on an edge node multiplexes incoming flows and creates a macro flow such as the label switched path (LSP) shown in Fig. 1, and applies leaky-bucket pacing according to the rate control of the macro flow and sends it to a receiver XCP agent on the destination edge node. The receiver XCP agent demultiplexes the macro flow and forwards the packets of individual flows to their destinations.

In the original XCP [9], feedbacks are carried in the header of data packets (per-packet feedback). In this case, core routers must read and update the feedback in the packet header by calculating a new feedback. However, calculating a new feedback for and updating the header of each optical packet at ultrahigh speed is hard. In our proposal, using simplified XCP-based core stateless fair queuing [9] and TeXCP [10], each macro flow sends its feedback in a separate probe packet once in every control period, instead of writing feedback to packet headers, so there is no need for calculating a per-packet feedback. Probe packets are carried on a separate single control wavelength, which means that we are separating the control and data channels. Using a separate single wavelength with a low transmission rate for probe packets allows the applying of an electronic conversion for updating feedback in packet headers and buffering the probe packets in electronic RAM in case of contention.

Core routers use a separate XCP control agent for each wavelength on an output link. When a probe packet of macro flow i arrives at a core router, the FC of the XCP agent responsible for the wavelength that macro flow i was assigned to calculates a positive feedback p_i and a negative feedback n_i for flow i. Positive feedback is calculated by $p_i=[h+\max(0,\Phi)]/N$ and negative feedback is calculated by $n_i=u_i(h+\max(0,-\Phi))/u$, 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. $feedback=p_i-n_i$ gives the required change in the flow rate as a feedback. When a core router receives a probe packet, the router calculates and compares its own feedback with the feedback available in the probe packet. If a core router's own feedback is smaller than the one in the probe packet, the core router replaces the feedback in the probe packet with its own feedback. Otherwise, the core router does not change the feedback. Core routers can estimate the number N by counting the number of probe packets received in the last control interval or the number of LSPs if generalized multiprotocol label switching (GMPLS) is available [10]. In [9], the control interval is calculated as the average RTT of flows using the link. In TeXCP and our architecture, the control interval is the maximum RTT in the network. TeXCP uses a simplified version of XCP's FC algorithm without the bandwidth shuffling algorithm of XCP, but our algorithm uses the bandwidth shuffling algorithm of XCP. In TeXCP, core routers send both p_i and n_i feedback by probe packets to sender agents, but in our algorithm core routers send only $feedback=p_i-n_i$ like in [9].

As explained in Subsection 2.A.1, Φ is calculated for a wavelength by using the equation $\Phi=\alpha S-\beta Q/d$ where S is the spare bandwidth that is the difference between the wavelength capacity and input traffic on this wavelength in the last control interval. Therefore, wavelength capacity must be explicitly given to the XCP algorithm for calculating S. Giving a false capacity value less than the actual wavelength capacity causes underutilization. The XCP algorithm converges to the given virtual capacity. We use this property of XCP to limit the utilization of the OPS network at a level that provides a low packet drop ratio with the available FDL set.

4. Evaluation

4.A. Simulation Settings

Proposed protocol and slotted WDM OPS architecture is implemented over *ns* version 2.28 [13]. It is assumed that there is backlogged traffic at the edge buffers, so each edge node sends traffic to all other edge nodes at the maximum possible rate controlled by XCP. We chose XCPs α, β, and γ parameters for edge routers as 0.2, 0.056,

and 0, respectively. The α parameter controls the utilization convergence speed. Higher α values allow a faster change in utilization, but also cause higher utilization overshoots. We chose the α parameter as 0.2, which gives a slower but more stable link utilization and decreases utilization overshoots when compared with the value 0.4 selected in [9]. When the α parameter is decreased, it is also necessary to decrease the γ parameter responsible for bandwidth shuffling. Otherwise, too much underutilization may occur in some links in case these links carry flows that are bandwidth throttled in other bottleneck links as explained in [9]. Therefore, $\gamma=0.05$ is used instead of $\gamma=0.1$ in [9]. FDL architecture makes it hard to evaluate and provide a buffer occupancy value to the XCP algorithm. Furthermore, our aim is to have a small buffered network and the effect of the queue parameter in XCP calculations is low as the persistent queue size is small (usually zero unless there is an overload) due to a small buffered network, so the β parameter is set to zero in the core routers. These may not be the optimum values, but the optimization of XCP parameters is left as a future work.

The simulator uses cut-through packet switching for data wavelengths. There is a single slow control wavelength dedicated to probe packets. The control wavelength uses store-and-forward switching. XCP agents start sending data randomly in the first 10 s and continue until the simulation ends. Total simulation duration is 40 s.

Edge nodes use electronic buffering, but core routers use only FDLs for buffering optical data packets. The contention of probe packets on a control wavelength is resolved by electronic RAM. O/E/O conversion is not a problem for control wavelength due to its low speed.

Figure 4 shows the simulated NSFNET topology. The nodes numbered from 0 to 13 are the core nodes and the rest are the edge nodes connected to the core nodes. All links (including edge and core links) have a single data wavelength with the same capacity. All links have the same XCP target utilization. There are a total of 28 nodes (14 core nodes+14 edge nodes) and 35 links (21 core links+14 edge links). The propagation delay of the links between the core and edge nodes is selected as 0.1 ms. All links (including edge and core links) apply optical packet switching. Each edge node sends traffic to all other edge nodes at the maximum possible rate, so there are multiple bottleneck links. The XCP control period of core routers and the probe packet sending interval of edge routers is selected as 50 ms by considering the extra processing and queuing delays in the core routers. The target utilization is set to 90% for the output links of the edge nodes in order to prevent buffer buildups in the buffer between the link and Paced XCP sources. The capacity of the data wavelength is set to 1 Gbit/s. The capacity of the XCP control wavelength is 100 Mbit/s.

In all simulations, we evaluate the aggregate packet drop rate inside the OPS core network. We compare the packet drop rate of the proposed pacing architecture with the packet drop rate of Poisson traffic arrival. For the traffic matrix of Poisson traffic, we use the traffic matrix that proposed XCP-based architecture converges to when there is a low packet drop rate with enough buffering and an FDL granularity of 1. TCP traffic is well known to be burstier than Poisson distribution, and the aggregation of many TCP flows does not converge to a Poisson stream [14]. Burstiness increases the buffer requirements. Reference [3] decreases the buffer requirements by making the traffic Poisson-like by modifying TCP senders to use Paced TCP or by using slow access links. In this paper, we show that our proposed architecture can achieve packet drop rates lower than Poisson traffic.

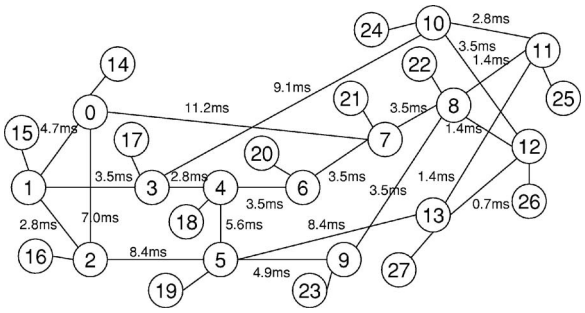

Fig. 4. NSFNET topology.

4.B. Evaluations

4.B.1. Slot Size is Equal to MTU

We first compare the performance of the proposed architecture and Poisson traffic when slot size is equal to MTU, which is selected as 1500 bytes in the simulations. It means that each packet is carried inside only a single slot. Figure 5 shows the aggregate packet drop rate of the proposed architecture and Poisson traffic (called Paced and Poisson in the figures, respectively) at 90%, 60%, and 30% target link utilization, which is the utilization due to optical packets including data payload and wasted void padding inside them. The x axis shows the limit of the number of delay lines per output in linear scale and the y axis is the aggregate packet drop rate in the core in a log scale. Average, minimum, and maximum drop rate results of ten simulations, which have different flow starting times, are plotted. We see that the deviation increases as the drop rate decreases, but the overall tendency is the same. Therefore, in the following figures a single simulation is done for each parameter set. FDL granularity is one slot in all plots. Higher granularities do not bring significant improvement, so they are not plotted. As seen in Fig. 5, the improvement of the proposed architecture increases as the link utilization increases. We see that the buffer requirement of the proposed paced architecture at a low packet drop rate such as 10^{-6} is approximately eight times lower than Poisson traffic arrival. Even at 30% utilization, we see that the proposed architecture can get much lower packet drop rates with the same buffering even though Poisson traffic has low buffer requirements at this utilization. For example, when there are three delay lines per output link, the packet drop rate of the proposed architecture is approximately ten times lower. The improvement becomes bigger as the number of delay lines increases. This is an expected result as [15] theoretically shows that multiplexed periodic streams such as the Paced traffic in our architecture give a lower packet drop rate than the Poisson traffic as they are less burstier [14] than Poisson traffic. Also, [15] shows that the drop rate comes closer to Poisson as the number of periodic streams increase. Therefore, we can expect to have a lower drop rate by applying pacing to macro flows at the edge nodes as in our architecture than individually pacing TCP flows. When there is no buffering, both Paced and Poisson traffic has the same packet drop rate as seen in Fig. 5. This is an expected result, because the packet drop rate depends only on the packet contention probability that is independent of burstiness.

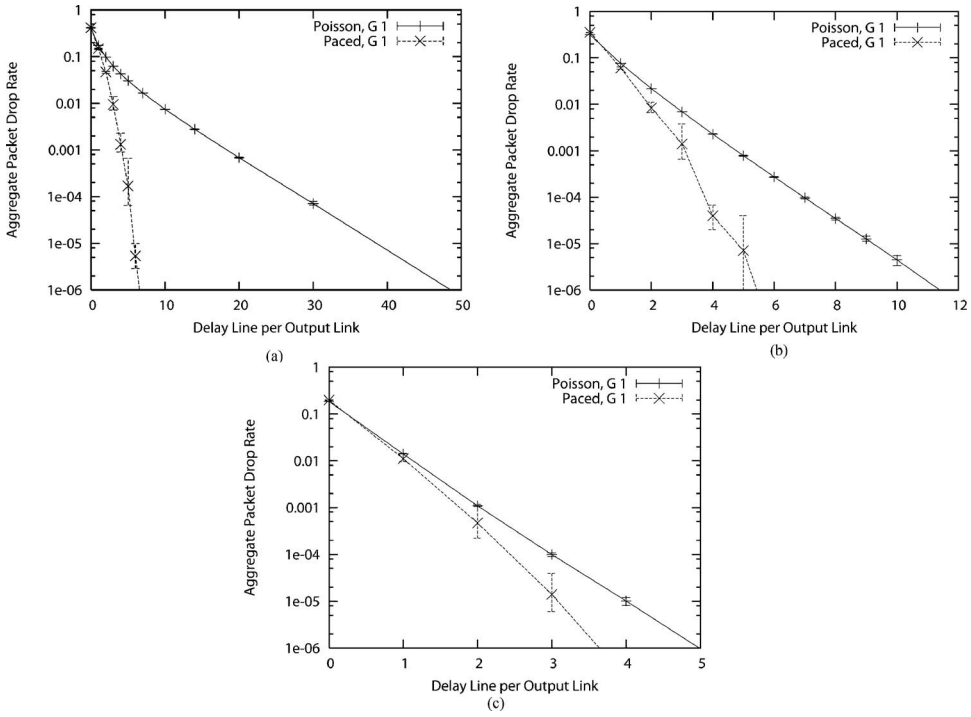

Fig. 5. Aggregate packet drop rate with limited number of FDLs per link when slot size is 1500 bytes for (a) 90%, (b) 60%, and (c) 30% utilization.

When slot size is equal to MTU, effective throughput may be low when the average size of arriving packets is small. On the other hand, it can greatly simplify the FDL design process, as FDL granularity can be directly selected as one slot size independent of packet size distribution as higher granularities do not bring significant improvement and the scheduling algorithm becomes simpler as explained in Subsection 2.A.

4.B.2. Slot Size is 500 Bytes

When slot size is lower than MTU, packet size distribution must be taken into account as big packets may use multiple slots. In this section we compare the performance of the proposed architecture and Poisson traffic when the slot size is tentatively selected as 500 bytes, which is one third of the MTU.

It is hard to do a direct comparison of different slot sizes, because guard bands and slot headers start becoming the main source of decrease in efficiency as we decrease the slot size or increase the link speed. The size of the guard bands depends on the type of switching hardware and link speed.

Reference [16] shows that the size of packets in Internet2 traffic is mainly composed of very small and big packets and there is an approximately 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 less than or equal to one slot (500 byte) size;
- All packets are three slots (1500 byte) size;
- 60% of packets are less than or equal to one slot (500 byte) size, 40% of packets are three slots (1500 byte) size (realistic traffic).

Figure 6 shows the aggregate packet drop rate for different packet size distributions, FDL granularities for the proposed architecture, and Poisson traffic when target utilization is 90%. In all subplots, the x axis shows the limit of the number of delay lines per output and the y 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. Figures 6(a)–6(c) show the drop rates when packet size distribution is 1500 bytes, all less than or equal to 500 bytes and realistic distribution, respectively. Note that effective utilization and throughput is different in the simulation of different packet size distributions at the

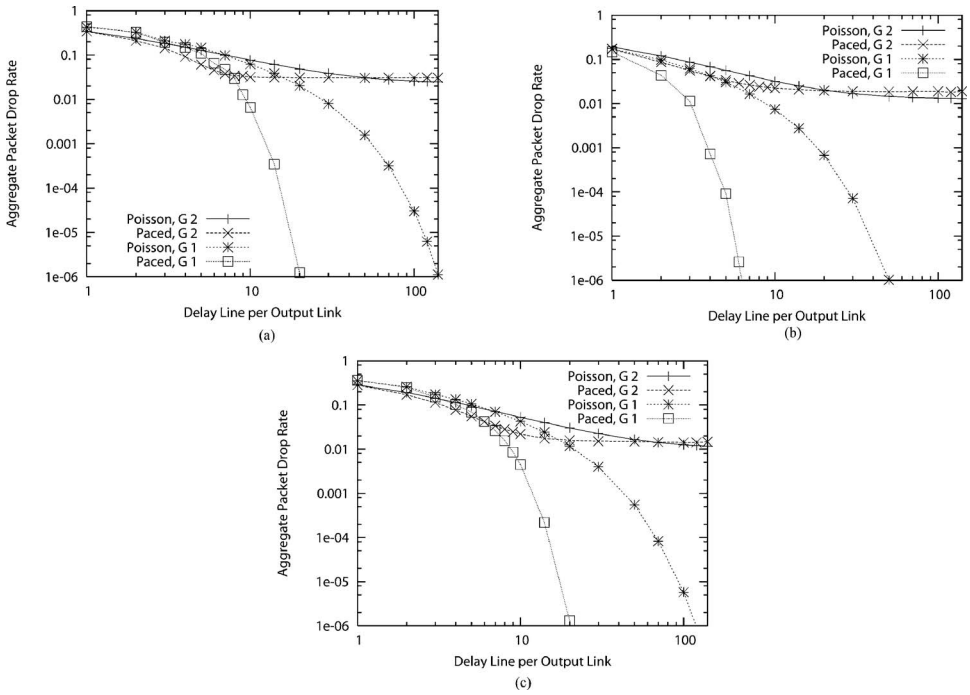

Fig. 6. Aggregate packet drop rate with limited number of FDLs per link for 90% utilization when packet size distribution is (a) all packets are 1500 bytes, (b) all packets are less than or equal to 500 bytes and (c) realistic distribution.

same target link utilization due to padding inside the optical packets. When we check these subplots, we see that packet size distribution has a big impact on the FDL requirements. In all subplots, FDL granularity of one slot gives a fast decrease in the drop rate, but the FDL granularity of two slot tends to stay constant after a decrease at the beginning because of the the void slots in FDLs due to high granularity. Voids increase the effective load to higher than link capacity, so a low packet drop rate cannot be achieved with a 90% utilization with small buffers at a high FDL granularity. In all subplots, we see that the buffer requirement of the proposed paced architecture at a low packet drop rate such as 10^{-6} is approximately eight times lower than Poisson traffic arrival such as what is shown in Fig. 5(a). Actually Fig. 6(b) is almost the same as Fig. 5(a), because Fig. 6(b) is merely a case where slot size and packet size distribution are down-scaled to one third of Fig. 5(a). Realistic packet size distribution in Fig. 6(b) shows that mainly big packets determine the buffer requirements, so its buffer requirements are almost the same as the case where all packets are 1500 bytes in size in Fig. 6(c) and approximately three times higher than the case where all packets are less than or equal to one slot (500 bytes) in size in Fig. 6(b).

Figure 7 shows the aggregate packet drop rate when target utilization is 30%. Figures 7(a), 7(c), and 7(e) and Figs. 7(b), 7(d), and 7(f) show the drop rates with the proposed pacing architecture and Poisson traffic, respectively. When we check these sub-

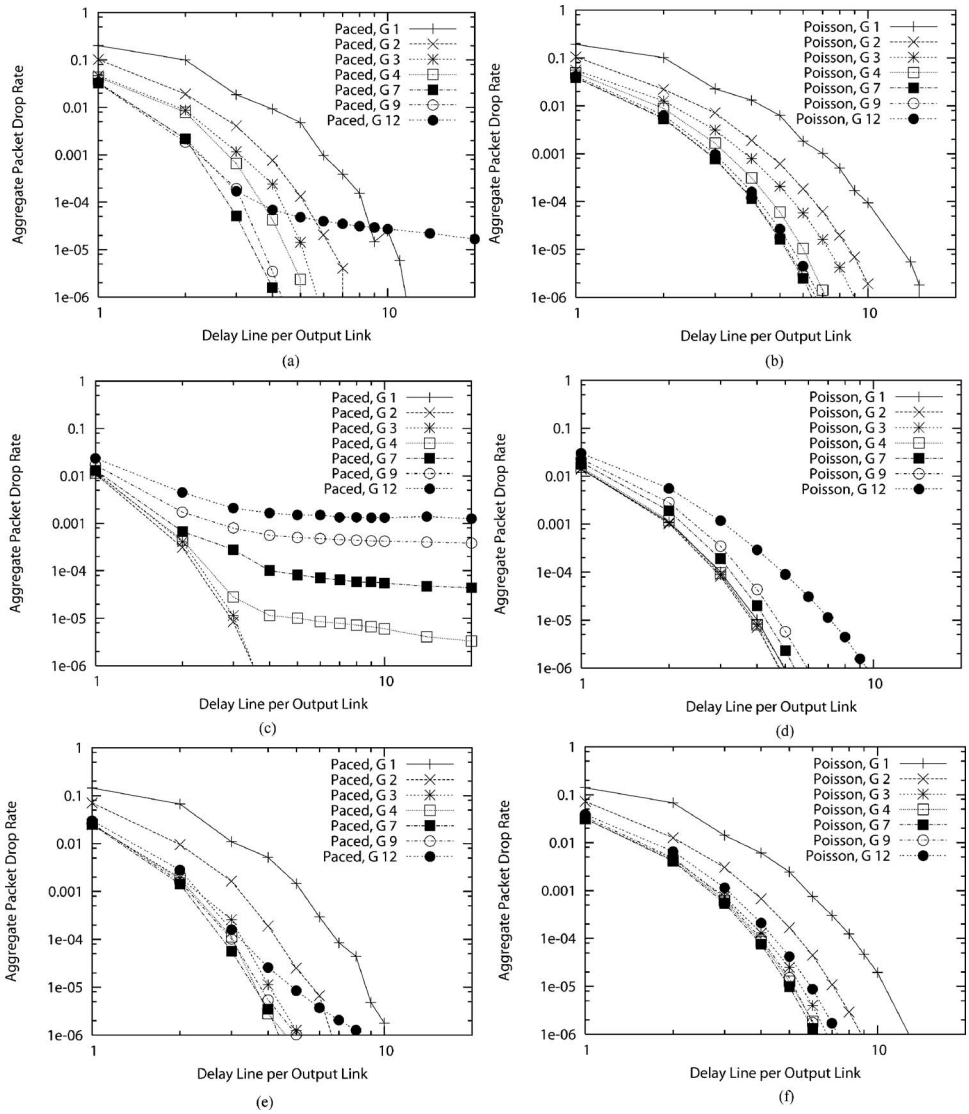

Fig. 7. Aggregate packet drop rate with limited number of FDLs per link for 30% utilization when packet size distribution in all packets is 1500 bytes with (a) Paced and (b) Poisson traffic, all packets are less than or equal to 500 bytes with (c) Paced and (d) Poisson traffic, realistic distribution with (e) Paced and (f) Poisson traffic.

plots, we see that packet size distribution has a big impact on the optimum FDL granularity. Figures 7(a) and 7(e) show that the proposed architecture has an optimum FDL granularity of seven slots for realistic packet size distribution and when all packets are 1500 bytes. On the other hand, when packet size distribution means that all packets are less than or equal to 500 bytes, the optimum granularities are one, two, and three slots as seen in Fig. 7(c). A granularity of seven slots has a much higher packet drop rate than when there are multiple delay lines per output link. In general, small FDL granularities tend to give a sharp decrease in the drop rate as the number of delay lines increase, but high FDL granularities tend to stay almost constant after a decrease at the beginning in the proposed architecture. This constant drop rate in Figs. 7(a) and 7(c) is mainly because of a load overshoot due to the void slots in FDLs because of high granularity. When we compare these high granularities for Paced and Poisson traffic, we see that Paced architecture gives higher drop rates mainly because of the synchronized packet contentions due to pacing. Otherwise, Paced architecture has lower packet drop rates at the same FDL granularities unless granularity is high. Realistic packet size distribution of the Paced architecture at a granularity of 12 slots in Fig. 7(e) shows that contention synchronization is lower when there is a mixed packet size distribution using a different number of slots.

Void slots in FDLs are served by the routers as if they are not empty, so void slots increase the effective load. When there are void slots in FDLs, using the size of slots occupied by the packets as a metric does not give a reliable measure of congestion. It can be possible to prevent overloading by carefully selecting the target utilization. In the worst case, all packets entering an FDL occupy a minimum number of slots and synchronized packet arrivals cause the maximum number of possible void slots, which is denoted by V, inside the FDL. Therefore, in a single stage equidistant FDL set, the lowest efficiency in the worst case is approximately

$$\frac{M}{M+V}, \tag{1}$$

where M is the number of slots occupied by the smallest possible packet size and V is the maximum number of void slots that may occur upon arrival of a packet. V equals $x-1$ slots, where x denotes the FDL granularity, as there must be an occupied slot using the link and causing contention. Plugging $V=x-1$ gives

$$\frac{M}{M+x-1}. \tag{2}$$

Setting the target utilization in optical XCP routers to a value smaller than the result of this equation can protect output wavelengths from load overshoots and protects from drops due to void slots. It is better to apply a safety margin for possible rate oscillations and use a target utilization a little lower than the value calculated by using the equation here. When we use the formula for all packets that are one slot size and three slot size for a 30% utilization, we get the limit FDL granularities three slots and seven slots, respectively. When we check the simulation results in Figs. 7(a) and 7(b), we see that all simulated FDL granularities bigger than these granularities have a higher drop rate than these granularities.

4.B.3. Scheduling

Using a void-filling scheduling algorithm can greatly decrease the packet drops due to void slots. In order to show the effect of void filling, NSFNET is simulated with realistic packet size distribution at 60% target utilization and eight delay lines per output link. Slot size is 500 bytes. Figure 8 shows the packet drop rates of the proposed architecture and Poisson traffic with and without the void-filling scheduling algorithm with packet reordering prevention. The x axis shows the FDL granularity in linear scale and the terms of the slots and the y axis is the aggregate packet drop rate in the core in log scale. We see that when granularity is 1, the void-filling algorithm gives the same performance as the non-void-filling algorithm, as expected because there are no void slots. As we increase the granularity, we see that the void-filling algorithms start giving much lower drop rates. Also the proposed architecture gives much lower drop rates than Poisson traffic with the same scheduling. After an optimum granularity, drop rates start increasing due to void slots. As we further increase the granular-

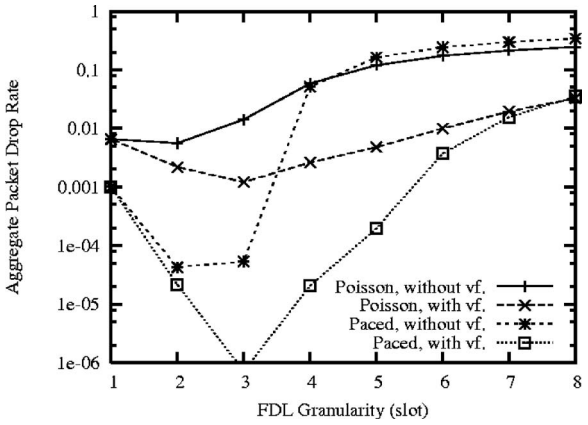

Fig. 8. Effect of FDL granularity and void-filling scheduling on aggregate packet drop rate.

ity, proposed architecture and Poisson traffic start giving a similar packet drop rate, because load overshoot due to void slots becomes the main reason for the packet drops. Optimum granularity is two slots for non-void-filling algorithms, while it is three slots for void-filling scheduling. In general, we see that applying void-filling scheduling and using the proposed architecture together can greatly decrease the drop rate.

5. Conclusions

In this paper, we proposed an architecture designed for OPS WDM networks with pacing at the edge nodes for minimizing the buffer requirements. We evaluated the packet drop rates with extensive simulations on a meshed network with multiple-hop paths and showed how FDL requirements change with slot size, FDL granularity, scheduling, and packet size distribution. Simulation results with meshed networks showed that our architecture can provide a packet loss ratio lower than Poisson traffic in core OPS networks with small FDL buffers. We showed that large and small packets have different FDL requirements. Small packets require low granularity for a low packet drop rate, but large packets require high granularity for decreasing the number of required FDL lines. Therefore, the selection of the slot size and MTU of the architecture has a strong impact on the buffer requirements. Selecting a big slot size equal to the MTU may decrease the efficiency due to padding for small packets, but it has low FDL requirements and can greatly simplify the design process and allow much simpler scheduling. In the case of using a slot size smaller than the MTU, we showed that void-filling scheduling can greatly decrease the buffer requirements.

As for future work, we will evaluate the buffer requirements and maximum achievable utilization when realistic TCP traffic is applied to the architecture and electronic buffer size requirements for pacing at the edge. We evaluated packet drop rate in this paper, but achieving link utilization with TCP flows is an important metric for ISPs. We are exploring more advanced FDL and switch models, and optimum parameter settings for further decreasing FDL requirements.

Acknowledgments

This work was supported in part by the National Institute 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.

References

1. C. Villamizar and C. Song, "High performance TCP in ANSNET," Comput. Commun. Rev. **24**, 45–60 (1994).
2. G. Appenzeller, J. Sommers, and N. McKeown, "Sizing router buffers," in *Proceedings of ACM SIGCOMM* (Association for Computing Machinery, 2004), pp. 281–292.
3. M. Enachescu, Y. Ganjali, A. Goel, N. McKeown, and T. Roughgarden, "Part III: routers with very small buffers," Comput. Commun. Rev. **35**, 83–90 (2005).

4. 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 1991* (Association for Computing Machinery, 1991), pp. 131–147.
5. G. Theagarajan, S. Ravichandran, and V. Sivaraman, "An experimental study of router buffer sizing for mixed TCP and real-time traffic," presented at the 14th IEEE International Conference on Networks (ICoN), Singapore, 13–15 Sept. 2006.
6. V. Sivaraman, H. Elgindy, D. Moreland, and D. Ostry, "Packet pacing in short buffer optical packet switched networks," in *Proceedings of IEEE INFOCOM* (IEEE, 2006).
7. Z. Lu, D. K. Hunter, and I. D. Henning, "Contention reduction in core optical packet switches through electronic traffic smoothing and scheduling at the network edge," J. Lightwave Technol. **24**, 4828–4837 (2006).
8. O. Alparslan, S. Arakawa, and M. Murata, "Optical rate-based paced XCP for small buffered optical packet switching networks," in *Proceedings of the Fourth International Workshop on Protocols for Fast Long-Distance Networks (PFLDnet)* (National Institute of Information and Communications Technology, 2006), pp. 117–124.
9. D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proceedings of ACM SIGCOMM* (Association for Computing Machinery, 2002), pp. 89–102.
10. S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the tightrope: Responsive yet stable traffic engineering," in *Proceedings of ACM SIGCOMM 2005* (Association for Computing Machinery, 2005), pp. 253–264.
11. T. Yamaguchi, K. Baba, M. Murata, and K. Kitayama, "Scheduling algorithm with consideration to void space reduction in photonic packet switch," IEICE Trans. Commun. **E86-B**, 2310–2318 (2003).
12. D. Careglio, J. Sole-Pareta, and S. Spadaro, "Optical slot size dimensioning in IP/MPLS over OPS networks," in *Proceedings of 7th IEEE International Conference on Telecommunications (ConTEL2003)*, Vol. 2 (IEEE, 2003), pp. 759–764.
13. S. McCanne and S. Floyd, "ns network simulator," http://www.isi.edu/nsnam/ns/, Jul. 2002.
14. H. Jiang and C. Dovrolis, "Why is the Internet traffic bursty in short time scales?," in *Proceedings of SIGMETRICS 2005*, (Association for Computing Machinery, 2005), pp. 241–252.
15. J. Roberts and J. Virtamo, "The superposition of periodic cell arrival streams in an ATM multiplexer," IEEE Trans. Commun. **39**, 298–303 (1991).
16. S. Shalunov and B. Teitelbaum, "TCP use and performance on Internet2," http://ben.teitelbaum.us/internet2/papers/i2tcp-imeas2001.pdf.