Optical Rate-based Paced XCP for Small Buffered Optical Packet Switching Networks

Onur Alparslan
Graduate School of Information Science and Technology, Osaka University,
1-3, Yamadagaoka, Suita, Osaka 560-0871, Japan
Email: a-onur@ist.osaka-u.ac.jp

Shin'ichi Arakawa
Graduate School of Economics, Osaka University,
1-7 Machikaneyama, Toyonaka, Osaka 560-0043, Japan
Email: arakawa@econ.osaka-u.ac.jp

Masayuki Murata
Graduate School of Information Science and Technology, Osaka University,
1-3, Yamadagaoka, Suita, Osaka 560-0871, Japan
Email: murata@ist.osaka-u.ac.jp

***Abstract*— One of the difficulties of 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 solution that can be used for buffering in the optical domain is using long fiber lines called Fiber Delay Lines (FDL). FDLs provide a small and fixed amount of buffering. Burstiness of Internet traffic and short-term over-utilizations cause high packet drop rates in small and fixed buffered OPS networks.**

In this paper, we propose an architecture using a XCP-based congestion control algorithm specially designed for OPS WDM networks with pacing at edge nodes for minimizing the buffer requirements at core nodes. We show how the FDL requirements change with the number of flows on a wavelength, FDL granularity and wavelength utilization by using a dumbbell topology. We show the requirements and parameter settings for stable operation and high utilization.

I. 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 a 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. They are expensive and there can be a limited number of FDLs in a router due to space considerations, so they can provide a small amount of buffering. Second, FDLs provide only a fixed amount of delay.

Having a very small buffering capacity and lack of variable delay buffering brings some important performance problems to optical packet switched (OPS) networks. According to a rule-of-thumb, an output link of a router needs a buffer sized at $B = RTT \times BW$, 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. Recently, Appenzeller et al [1] showed that when there are many TCP flows sharing the same link, 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, 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 very small amount of buffering capacity. Further decreasing the buffer requirements is necessary. However, bursty nature of TCP causes a high packet drop rate in small buffered networks and limits further decreasing the buffer size. Recently, [2] proposed that $O(\log W)$ buffers are sufficient where W is the maximum congestion window size of each flow when pacing [3] is applied to all TCP flows and the link is under-utilized. However, this proposal requires setting a maximum congestion window size to all flows, so there is a limit on per flow rate. If there are not enough number of flows, limiting the rate of all flows may cause a high level of under-utilization. Also, replacing all TCP agents on the Internet is hard to realize. Ref. [4] shows that TCP pacing may introduce new problems like TCP congestion window synchronization of many TCP flows. As an alternative, Ref. [2] proposes that if the access links are much slower than core links, a natural spacing between packets occur and this spacing allows small buffering without applying pacing to TCP agents. However, [2] states that using much slower access links limits the rate of end-to-end transmission, so this is not a preferred solution when super computers communicate. It is 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 replacing all sender or receiver agents

of all computers using the network.

Shaping the traffic at the edge nodes of an OPS network is much more applicable and cost efficient than shaping the traffic at the clients, because there is no need to replace the agents of all clients, so it is a possible and general solution. Also, applying pacing to an aggregated macro flow between a source-destination edge node pair of OPS network instead of applying pacing at the clients to macro flow's individual flows is more effective on minimizing burstiness in the OPS network, because even when the individual flows are properly paced, the macro flow between a source-destination edge node pair may end up behaving bursty. Also the individual paced flows may become burstier before arriving to the OPS network due to disturbances on packet spacing by big buffers of other networks that the packets will possibly propagate through.

[5] and [6] propose applying traffic shaping at edge nodes of OPS network for minimizing traffic burstiness. [5] evenly spaces all packets of a macro flow and shows that low packet drop ratio is possible with low FDL requirements by applying a proper spacing. However, [5] does not propose a method for choosing the optimum space between packets. [6] proposes a delay-based pacing algorithm that adaptively chooses optimum packet spacing according to input traffic for achieving bounded end-to-end delays, instead of evenly spacing by a leaky-bucket algorithm. However, [5] and [6] does not consider the problem of short-term or long-term over-utilization of small buffered OPS links.

XCP [7] 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 [7] as a window-based reliable congestion and transmission control algorithm. TeXCP, which is a traffic engineering protocol based on XCP framework with an unreliable rate based congestion control instead of reliable window-based transmission and congestion control of XCP, was proposed in [8]. TeXCP allows traffic engineering by applying load balancing with multi-path routing. Original XCP [7] is not suitable for small buffered networks because of too bursty behavior. Thus, pacing is necessary. Also, routers must read and update the feedbacks carried in all data packet headers by calculating a per packet feedback. However, calculating a feedback for each packet and updating the header of each optical packet at ultra high speed of optical links is hard. Also, parameter sets used in original XCP and TeXCP are not suitable for small buffered networks.

In this paper, we propose using a XCP framework-based intra-domain congestion control protocol specially designed for slotted WDM OPS networks for achieving high utilization and low packet drop ratio with small FDL buffers. XCP framework is selected because XCP framework allows individual control of the utilization level of each wavelength. Selecting target wavelength utilization less than actual wavelength capacity in XCP control algorithm can prevent queue buildups. Also, XCP allows limiting the utilization of wavelengths at a level that is stable for a selected FDL granularity as we will show in this paper. In our architecture, each edge source-destination node pair creates a rate-based XCP macro flow, and assigns the TCP and UDP flows to the XCP macro flow as shown in Fig. 1 and forwards the packets according to the XCP macro flow rate like in XCP-based Core Stateless Fair Queuing [7] and TeXCP. We apply token-based leaky-bucket pacing to the macro flows at edge nodes by using the rate information provided by XCP for minimizing the burstiness in the OPS network. Therefore, there is no need to update the TCP and UDP agents on clients.

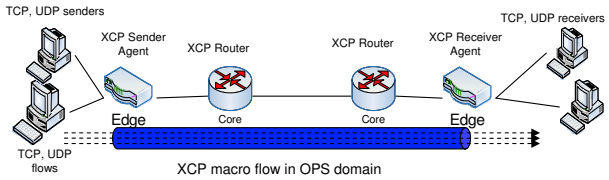

Fig. 1. XCP macro flows

[5] and [6] use fixed length optical packets with a size equal to OPS slot length by assembling incoming IP packets. In this paper, variable sized IP packets using variable number of slots enter OPS network without any assembling.

We next show the FDL buffer requirements by using the proposed optical rate-based paced XCP algorithm. We show how the FDL requirements change with the number of flows on a wavelength, FDL granularity and wavelength utilization by using a dumbbell topology. Also we show the requirements for stable operation and high utilization.

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

II. Architecture

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 network. It decouples the utilization control from fairness control. Core routers calculate flow-specific feedbacks by using the information provided by the flows and send the feedbacks to the XCP sender agents. Core routers do not require maintaining per-flow state information

XCP core routers maintain a per-link control-decision timer. When a timeout occurs, core router updates its control decisions calculated by Efficiency Controller and Fairness Controller.

1) Efficiency Controller (EC): Efficiency Controller 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 $\Phi = \alpha \cdot S - \beta \cdot Q/d$. 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.

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

B. XCP Variants

XCP was first proposed in [7] 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. XCP-based Core Stateless Fair Queuing algorithm creates a XCP macro flow, assigns the TCP and UDP flows to the XCP macro flow and forwards the TCP and UDP packets inside the XCP macro flow according to the XCP macro flow rate. [7] states that the algorithm can be further simplified by using special probe packets for receiving the feedbacks for macro flow rate calculation instead of attaching congestion header to forwarded packets.

TeXCP [8] is a traffic engineering protocol using a rate-based XCP congestion control allowing traffic engineering by applying load balancing with multi-path routing. TeXCP creates a macro flow and assigns TCP and UDP flows to this macro flow and forwards the packets according to XCP flow rate similar to simplified XCP-based Core Stateless Fair Queuing.

Another XCP-based algorithm is WXCP [9], which is a flow control protocol for wireless multi-hop networks. WXCP uses different congestion metrics special for wireless networks and applies pacing.

C. FDL Architecture and Voids

FDL architecture used in this paper is a single stage equidistant FDL set with B delay lines. Switch and FDL architecture [10] is shown in Fig. 2. There is no void-filling, because void-filling algorithms that prevent packet-reordering are generally complex. Complex algorithms are not preferred because of the electronic processing limitations due to high speed of OPS switches.

In the FDL architecture, length of 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.

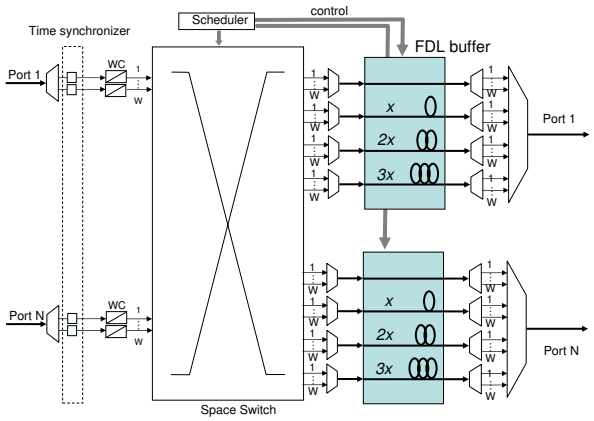

Fig. 2. Switch and FDL architecture

Using FDLs and a slot-based architecture causes voids, which decrease the effective throughput of output links. Voids in the architecture can be classified into two groups.

1) Voids in Slots: Voids in slots occur when the packet size is not equal to a multiple of slot size. For example, if the slot size is 52Bytes and if a 53Bytes packet arrives, the packet will be carried in two slots with a total length of 104Bytes. There will be a 51Bytes void in the second slot. The throughput decrease due to voids in slots becomes most effective when average size of arriving packets is much less than slot size. For example, if the slot size is 1500Bytes and only 40Bytes packets are carried in the network, 97.3% of each slot is wasted due to voids.

2) Void Slots in FDLs Between Packets: When FDL granularity is larger than a single slot, unused void slots may occur in FDLs, because FDLs can provide only fixed delays and may delay a packet more than the required delay when FDL set cannot provide the required delay amount.

D. Optical Rate-based Paced XCP

We propose Optical Rate-based Paced XCP as an intra-domain traffic shaping and congestion control protocol in an OPS network domain. In this architecture, XCP sender agent on an edge node multiplexes incoming flows and creates a macro flow as shown in Fig. 1, and applies pacing with rate control to the macro flow and sends to a receiver XCP agent on destination edge node. The receiver XCP agent de-multiplexes the macro flow and forwards the packets of individual flows to their destinations.

In the original XCP [7], feedbacks are carried in the header of data packets. 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 ultra high speed is hard. In our proposal, simplified XCP-based Core Stateless Fair Queuing

[7] and TeXCP [8], 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 channel and data channels. Using a separate single wavelength with low transmission rate for probe packets allows applying electronic conversion for updating feedback in packet headers and buffering the probe packets in electronic RAM in case of a contention.

Core routers use a different XCP control agent for each wavelength on an output link. When a probe packet of macro flow i arrives to a core router, 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 $p_i = \frac{h+max(0,\Phi)}{N}$ 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. $feedback = p_i - n_i$ gives the required change in the flow rate as a feedback. When a core router receives a probe packet, router calculates and compares its own feedback with the feedback available in the probe packet. If core router's own 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. Core routers can estimate the number N by counting the number of probe packets received in the last control interval or use the number of LSPs if GMPLS is available [8]. In [7], the control interval is calculated as the average RTT of flows using the link. In TeXCP and our architecture, control interval is the maximum RTT in the network. TeXCP uses a simplified version of XCP's fairness controller algorithm without the bandwidth shuffling algorithm of XCP, but our algorithm uses the bandwidth shuffling algorithm. 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 [7].

As explained in Sec. II-A, Φ is calculated for a wavelength by using the equation $\Phi = \alpha \cdot S - \beta \cdot Q/d$ 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 XCP algorithm for calculating S. Giving a false capacity value less than actual wavelength capacity causes under-utilization. XCP algorithm converges to the given virtual capacity. We use this property of XCP to operate OPS network at a maximum utilization level that guarantees stable operation.

Pacing is implemented by using a token-based leaky bucket algorithm. In this algorithm, a packet is sent to the output link, if there is a token in the token buffer. After packet is sent, a token is removed from the token buffer. If there is no token in the token buffer, packet must wait until a new token arrives. Arrival time of next token to the token buffer is calculated by dividing the size of the sent packet with the current assigned rate of XCP macro flow. Changing the token buffer size affects the burstiness of the flow. Limiting token buffer size to 1 token gives the least bursty output traffic, so token buffer size is selected as 1 token in the simulations.

Even though voids in slots decrease the effective throughput, voids in slots do not cause a stability problem for optical XCP, because size of void in a slot is constant throughout the network, and XCP is adapted to use size of slots occupied by the packets instead of the real size of packets as a metric. For example, if the slot size is 52Bytes and if a 53Bytes packet is sent to the network by the edge XCP sender agent, the edge XCP sender agent assumes that it sent a 104Bytes (2 slots) packet and does the pacing of the next packet accordingly. Also when a XCP core router forwards this 53Bytes packet, XCP agent on the core router assumes that router is forwarding a 104Bytes packet and updates its estimation variable of input traffic rate accordingly.

Unlike the voids in slots, the void slots in FDLs between packets can cause stability problems in XCP, because the number of void slots caused by a packet is not predictable until the packet arrives to a node. The number of void slots caused by a packet changes at each node. Also, void slots in FDLs are served by the routers as if they are not empty, so void slots increase the 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. In this case, it can be possible to guarantee a stable operation of XCP by carefully selecting the target utilization. In the worst case, all packets entering an FDL occupy minimum number of slots and each packet causes the maximum possible void slot number inside the FDL, which equals to $Granularity - 1$ slots. Therefore, maximum achievable stable throughput is approximately,

$$\frac{MinDataSlot}{MinDataSlot + VoidSlot}, \tag{1}$$

where $MinDataSlot$ is the number of slots occupied by the smallest possible packet size and $VoidSlot$ is the maximum single possible void slot size. Plugging $VoidSlot = Granularity - 1$ gives

$$\frac{MinDataSlot}{MinDataSlot + Granularity - 1} \tag{2}$$

Setting the target utilization in optical XCP routers to smaller than this value protects router from load overshoots. 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 above.

When there are multiple wavelengths on links and wavelength converters on routers, routers can distribute the macro flows to wavelengths uniformly and therefore decrease the number of macro flows per wavelength for further decreasing buffer requirements. Furthermore, it is possible to create multiple macro flows between source-destination edge pairs for UDP and TCP flows with different QoS requirements. When there are multiple wavelengths, a macro flow can be

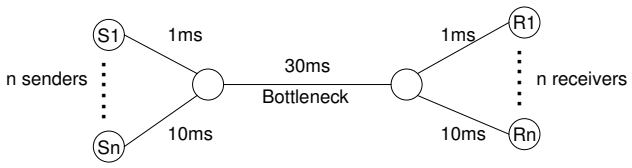

Fig. 3. Dumbbell topology

assigned to a wavelength according to QoS requirements. Also XCP framework allows differential bandwidth allocation among macro flows.

III. Evaluation

A. Simulation Settings

Proposed protocol and slotted WDM OPS architecture is implemented over *ns* version 2.28 [11]. A dumbbell topology as seen in Fig. 3 is used for computer simulations. XCP sender agents on edge nodes always have data to send. Simulator uses cut-through packet switching for data wavelengths. There is a single slow control wavelength dedicated for probe packets. Control wavelength uses store-and-forward switching. The number of edge nodes n ranges from 2 to 100. XCP agents start sending data randomly in the first 10s and continue until the simulation ends. Total simulation duration is 40s. [12] shows that most common small packets on Internet2 are in the range of 40Bytes to 52Bytes, so slot size is selected as 52Bytes. Probe packet size is also 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. O/E/O conversion is not a problem for control wavelength due to its low speed. Simulated packet size distributions are

- All packets are 1 slot size (52 Bytes)
- All packets are 29 slot size (1508 Bytes)
- All packets are 12 slot size (624 Bytes)

Dumbbell topology is simulated with FDL granularities ranging between 1 to 29 slots and target utilizations between 10% and 90% and macro flow numbers (n) of 2, 3, 4, 5, 10, 50, 100. Each source node sends data to only one corresponding destination edge node. Simulation results give the maximum number of fiber delay lines used through the simulation. Also the total fiber length of the single stage FDL set is evaluated by using the equation

$$\sum_{k=1}^{B} k \cdot x, \tag{3}$$

where B is the maximum number of fiber delay lines used throughout the simulation and x is the FDL granularity.

In this paper, we show the results of single data wavelength on links. As the packet size distribution gets smaller, it is necessary to simulate at a lower wavelength capacity because of the simulation time considerations. Therefore, wavelength capacity is normalized to packet size distribution for transferring roughly the same number of packets in the simulations. The capacity of the data wavelength is set to 5Gbps, 2069Mbps and 172Mbps when all packets are 29, 12 and 1 slot size, respectively. The capacity of the control wavelength is 100Mbps. Minimum RTT is 64ms, maximum RTT is 100ms and average RTT is 82ms in the network. XCP control period of core routers and probe packet sending interval of edge routers is 100ms.

XCP parameter α=0.4 used in [7] sometimes causes utilization overshoots and oscillations. Therefore, we selected a more conservative α=0.2, which gives a slower but more stable link utilization and decreases utilization overshoots. When α parameter is decreased, it is also necessary to decrease γ parameter responsible for bandwidth shuffling. Otherwise, too much under-utilization may occur in some links in case these links carry flows that are bandwidth throttled in other bottleneck links as explained in [7]. Therefore, γ=0.05 is used instead of γ=0.1 in [7]. β must be selected according to the formula $\beta = \alpha^2\sqrt{2}$ as proved in [7], so β=0.056 is straightforward.

B. Simulation Results

Figures 4 and 5 show the maximum buffer size utilized in the simulations. In all subplots, x-axis shows the number of macro flows on the bottleneck link in log scale. In the top subplots, y-axis shows the maximum number of delay lines used in the simulations in log scale. In the bottom subplots y-axis shows the total fiber length used in the simulations calculated by Equation 3 in terms of slot number in log scale. In both figures, (a) and (d) show the case when all packets are 1 slot size, (b) and (e) show the case when all packets are 12 slot size, (c) and (f) show the case when all packets are 29 slot size. The missing points in subplots are the simulations that became unstable because of void slots in FDLs due to high FDL granularity and therefore required a too big buffer size due to queue buildups.

Fig. 4 shows the simulation results when target utilization is 30%. Fig. 4(a) shows that XCP is unstable for 30% target utilization when FDL granularity is 4, 12 or 29 slots, because void slots between packets inside FDLs prevent achieving a throughput equal to the input traffic rate and thus cause queue-buildup as effective load on router exceeds 100%. Delay line requirements are similar for stable simulations, but there is a difference between total FDL length requirements in Fig. 4.(d).

When all packets have a size of 624Bytes, Fig. 4(b) shows that simulations were stable for all FDL granularities. Only FDL granularity of 29 slots showed some oscillations in buffer requirements pointing to that it is close to the unstable region. This is an expected result, because when the size of packets is bigger, the ratio between the size of void slots and data slots inside FDLs becomes smaller. However, both the maximum number of fiber delay lines used in the simulation in Fig. 4(b) and the total fiber length in Fig. 4(e) are much higher than the case where packets were 1 slot size. Again this is an expected result, because solving the contention of

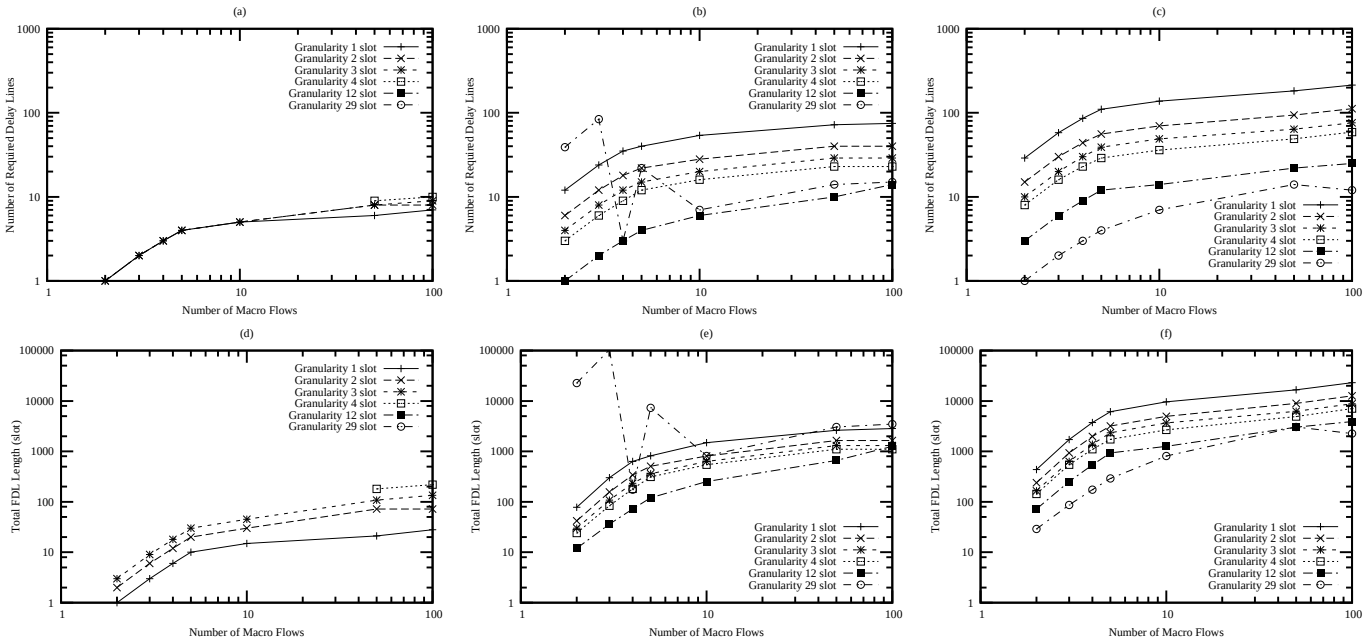

Fig. 4. Number of required fiber lines when target utilization is 30% of wavelength capacity and packet size is (a) 52Bytes, (b) 624Bytes, (c) 1508Bytes. Total required FDL length when target utilization is 30% and packet size is (d) 52Bytes, (e) 624Bytes, (f) 1508Bytes.

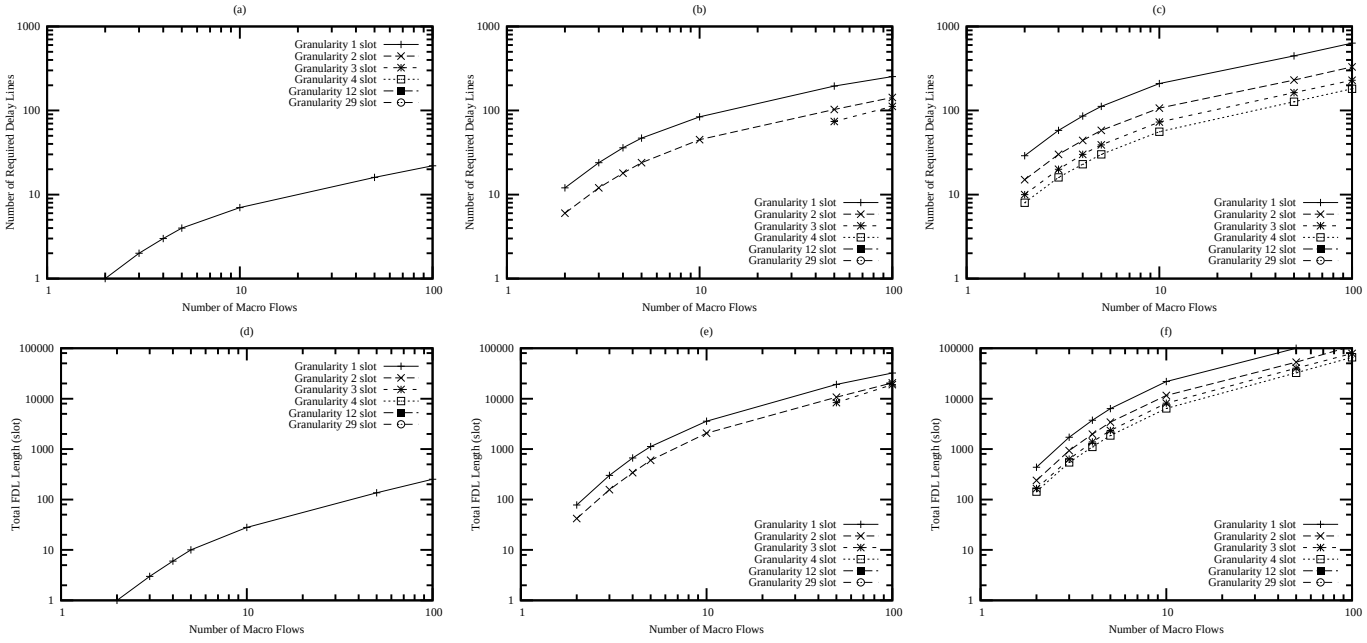

Fig. 5. Number of required fiber lines when target utilization is 90% of wavelength capacity and packet size is (a) 52Bytes, (b) 624Bytes, (c) 1508Bytes. Total required FDL length when target utilization is 90% and packet size is (d) 52Bytes, (e) 624Bytes, (f) 1508Bytes.

bigger packets requires bigger buffers. For example, when two packets with 1 slot size arrive at the same time and contend, an FDL line with a size of 1 slot is enough. However, when two packets with 12 slots size arrive at the same time and contend, an FDL line with a size of 12 slots is necessary for solving the contention.

When all packets have a size of 1508Bytes, all simulated FDL granularities are stable, but both the maximum number of required fiber delay lines shown in Fig. 4(c) and the total fiber length in Fig. 4(f) are higher than Fig. 4(b) and Fig. 4(e), respectively.

Fig. 5 shows the simulation results of when target utilization is increased to 90%. When Fig. 4 is compared with Fig. 5, we see that less number of FDL granularities are stable at

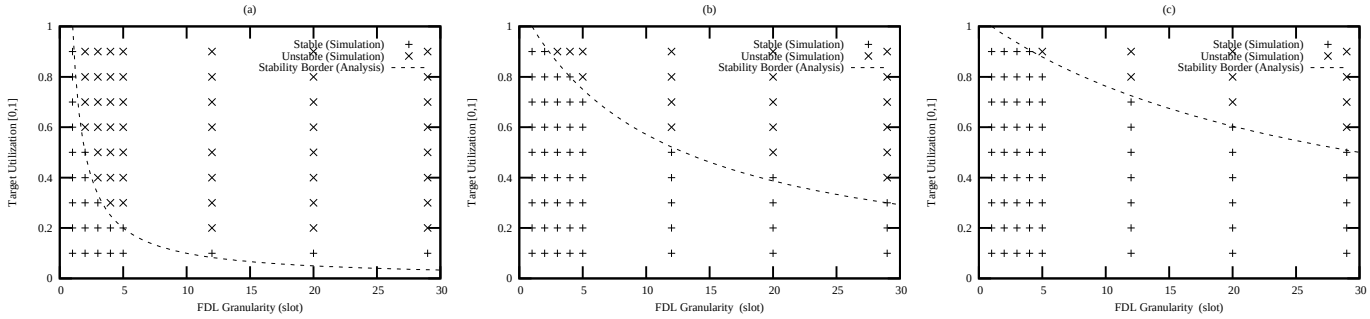

Fig. 6. Stability when packet size is (a) 52Bytes, (b) 624Bytes, (c) 1508Bytes.

this target utilization due to void slots. For example, when all packets have a size of 1 slot, only the granularity of 1 slot, which causes no void slots, is stable. Also the buffer requirements of stable simulations are much higher due to increased probability of contention.

Fig. 6 shows the stability results of simulations and the safety region estimated by stability border equation when there are two macro flows. In the subplots, "+" shows the simulations that were stable with small buffer requirements and "x" shows the simulations that are unstable with queue buildup. Dotted line is the plot of equation 2. Area below this line is the safety region. In all subplots, x-axis shows the FDL granularity in terms of slots and y-axis shows the target utilization used in the simulation in the range [0-1]. The target utilization of 1 means 100% utilization. Fig. 6(a) shows the stability results when all packets are 1 slot size. When all packets are 1 slot size, most of the simulations were unstable due to voids in FDLs. High utilization can be achieved only when FDL granularity is low. Fig. 6(b) shows the stability when all packets are 12 slots size and Fig. 6(c) shows the stability when all packets are 29 slots size. All three cases show that the achievable utilization ratio increases as the packet size increases. This is due to decrease of the ratio of the void slots in FDLs as the size of packets increase, as explained above. When we check the stability region, we see that the simulation results closely follow the stability border line calculated by the equation. Only a few simulations were stable at the outside of the safety region specified by the stability border line. None of the simulations were unstable in the safety region.

In general, we see that small packets determine the stable target utilization and FDL granularity. In other words, size of packets in terms of slots determine the buffer requirements. An OPS network must be stable and have a low packet loss rate for a wide range of packet size distributions. We cannot guarantee that packet size distribution will not change in time. Therefore, the maximum target utilization for a FDL granularity must be selected carefully according to the worst case scenario, which is the case when all packets are 1 slot size in the simulations. Among the stable FDL granularities, the granularity with smallest buffer requirement can be selected. Simulations show that a stable high target utilization like 90% can be achieved only when FDL granularity is 1. Operating the wavelengths at 30% maximum target utilization allows using FDL granularity of 1, 2 or 3 slots. However, FDL granularity of 3 slots is close to the safety region border, so it is better to choose FDL granularity of 1 or 2 slots. Simulations with 29 slots packets in Fig. 4(c) show that FDL granularity of 2 requires less fiber delay lines, so FDL granularity of 2 looks like the best choice.

C. Comparison with Paced TCP

We show transient utilization of bottleneck link if reliable window-based transmission and congestion control of TCP is used instead of unreliable rate-based XCP. We use TCP Reno by applying pacing. There are 20 macro flows on the bottleneck link. Data wavelength speed is 500Mbps. FDL granularity is 1. Packet size is 1508Bytes. TCP ACK packet size is 52Bytes, because time-stamps option is used for more precise estimation of RTT for pacing. Target utilization of XCP is 90%. There is no limit on congestion window size of TCP. There is a single-way traffic. For a fair comparison, first we find the maximum number of delay lines used by XCP. The maximum number of delay lines is found as 250. Then, we simulate Paced TCP with this buffer size.

Fig. 7 shows the transient utilization of bottleneck link. X-axis shows the time and y-axis shows the utilization in the range [0-1]. XCP converges to its target utilization in a very short time and keeps a stable utilization. Paced TCP Reno gives a lower utilization on the average with a saw-tooth behavior due to high synchronization and packet losses. Actually, effective throughput is even lower because Paced TCP must add extra congestion and transmission header to each transmitted packet. If incoming packets are small like 40Bytes, large amount of utilization will be wasted by extra congestion header. Paced TCP must send ACK packets for reliable transmission and ACK packets use bandwidth, so effective utilization gets even lower if there is two-way traffic. Also, reliable window-based control may increase the jitter and RTT of flows. Optical rate-based paced XCP does not have any of these problems.

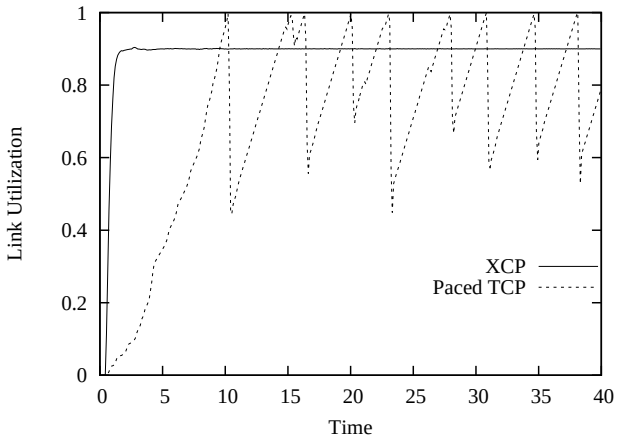

Fig. 7. Transient bottleneck wavelength utilization of Paced TCP Reno and Optical Rate-based Paced XCP

IV. Conclusions and Future Work

In this paper, we proposed an architecture using a XCP-based congestion control algorithm specially designed for OPS WDM networks with pacing at edge nodes for minimizing the buffer requirements. We evaluated how the FDL requirements change with the number of flows on a wavelength, FDL granularity and wavelength utilization by using a dumbbell topology.

We showed that big packets and small packets have different FDL requirements. Small packets require low granularity for stability, but big packets require high granularity for decreasing the number of required FDL lines. We showed how to select target utilization and FDL granularity for stable operation

When a meshed topology is used, buffers in core routers may cause disturbances on packet spacing of a macro flow, make the macro flow burstier and therefore increase the buffer requirements. Currently, we are evaluating the the performance of proposed OPS architecture on a meshed network with realistic packet size distributions for showing the effect of multiple-hop paths, wavelength converters, multiple wavelengths and packet size distribution on the FDL requirements.

Acknowledgement

O. Alparslan is supported by Ministry of Education, Culture, Sports, Science and Technology, Japan.

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, July 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] A. Aggarwal, S. Savage, and T. Anderson, "Understanding the performance of TCP pacing," in *Proceedings of IEEE INFOCOM*, pp. 1157–1165, Mar. 2000.

[5] V. Sivaraman, D. Moreland, and D. Ostry, "Ingress traffic conditioning in slotted optical packet switched networks," in *Proceedings of ATNAC*, Dec. 2004.

[6] V. Sivaraman, D. Moreland, and D. Ostry, "A novel delay-bounded traffic conditioner for optical edge switches," in *Proceedings of HPSR*, May 2005.

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

[8] 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.

[9] Y. Su and T. Gross, "WXCP: Explicit congestion control for wireless multi-hop networks," in *Proceedings of the 13th International Workshop on Quality of Service (IWQoS)*, June 2005.

[10] T. Yamaguchi, K. Baba, M. Murata and K. Kitayama, "Scheduling algorithm with consideration to void space reduction in photonic packet switch," *IEICE Transactions on Communications*, Vol. E86-B, pp. 2310–2318, August 2003.

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

[12] S. Shalunov and B. Teitelbaum, "Bulk TCP use and performance on Internet2," 2002.