Photon Netw Commun (2009) 18:237–243
DOI 10.1007/s11107-009-0187-9

XCP-based transmission control mechanism for optical packet switched networks with very small optical RAM

Onur Alparslan · Shin'ichi Arakawa · Masayuki Murata

Received: 18 August 2008 / Accepted: 3 February 2009 / Published online: 27 February 2009

Abstract According to a famous rule-of-thumb, buffer size of each output link of a router should be set to bandwidth-delay product of the network, in order to achieve high utilization with TCP flows. However, ultra high speed of optical networks makes it very hard to satisfy this rule-of-thumb, especially with limited choices of buffering in the optical domain, because optical RAM is under research and it is not expected to have a large capacity, soon. In this article, we evaluate the performance of our explicit congestion control protocol-based architecture designed for very small Optical RAM-buffered optical packet switched wavelength division multiplexing networks with pacing at edge nodes in order to decrease the required buffer size at core nodes. By using a mesh topology and applying TCP traffic, we evaluate the optical buffer size requirements of this architecture and compare with a common proposal in the literature.

Keywords Small buffer · OPS · Optical RAM · TCP

1 Introduction

A well-known problem in realizing optical packet switched (OPS) networks is buffering. Recent advances in the optical networks such as dense wavelength division multiplexing (DWDM) allowed to achieve ultra high data transmission rates in optical networks. This ultra high speed of optical networks made it necessary to do some basic operations like buffering and switching in the optical domain instead of electronic domain due to high costs and limitations of electronic buffers. However, the lack of high capacity optical RAM makes it difficult to buffer enough optical packets in optical packet switched (OPS) networks. According to a rule-of-thumb [1], buffer size of a link must be $B = \mathrm{RTT} \times \mathrm{BW}$, 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. However it requires a huge buffer size in optical routers due to ultra high speed of optical links, so this buffer size is unfeasible.

Currently, the only available solution that can be used for buffering in the optical domain is using FDLs. Contended packets are switched to long fiber lines in order to be delayed. However, FDLs have important limitations. First of all, FDLs require very long fiber lines that cause signal attenuation inside the routers. 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. FDL buffering is possible with today's technology, so many researchers consider FDL buffers to resolve contentions in optical networks. On the other hand, optical RAM is under research (e.g., Takahashi et al. [2] and NICT project (phase II) [3]) and it may be available in the near future. Basic operation of optical RAM is experimentally confirmed for 40 Gbit/s 16 bit optical packets in Ref. [2]. Optical RAM solves the problems of FDLs like lack of real $O(1)$ reading operation, signal attenuation, and bulkiness. Furthermore, optical RAM is expected to have a low power consumption rate that is an important problem for electronic RAM. However, optical RAM is not expected to have a large capacity, soon. Therefore, decreasing the huge buffer requirements of OPS networks is a must in order to make use of optical buffering.

Recently, Appenzeller et al. [4] showed that when there are many TCP flows sharing the same link, a buffer sized at $B = \frac{\mathrm{RTT} \times \mathrm{BW}}{\sqrt{n}}$, where n is the number of TCP flows

O. Alparslan (✉) · S. Arakawa · M. Murata
Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
e-mail: a-onur@ist.osaka-u.ac.jp

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 flows causes a high packet drop rate in small buffered networks and limits further decreasing the buffer size. Enachescu et al. [5] proposed that $O(\log W)$ buffers are sufficient where W is the maximum congestion window size of flows when packets are sufficiently paced by replacing TCP senders with Paced TCP [6] or by using slow access links. Pacing is defined as transmitting ACK (data) packets according to a special criteria, instead of transmitting immediately upon arrival of a data (ACK) packet [6]. However, $O(\log W)$ buffer size depends on the maximum congestion window size of TCP flows that may change in time. Moreover, 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 [7] 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.

In Ref. [8], we introduced an all-optical OPS network architecture that can achieve high utilization and low packet drop ratio by using small buffering. We consider 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] intra-domain congestion control protocol for achieving high utilization and low packet drop ratio with small 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.

We selected XCP framework because it allows individual control of the utilization level of each wavelength. In our architecture, when there is traffic between an edge source–destination node pair, a rate-based XCP macroflow is created, and incoming TCP and UDP packets of this edge pair are assigned the XCP macroflow similar to TeXCP [10]. The edge nodes of OPS network apply leaky-bucket pacing to the macroflows 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 this article, we evaluate the optical RAM requirements and packet drop rates of our proposed architecture on a realistic mesh NSFNET topology with TCP traffic. We show the effect of architecture parameters and provide guidelines for selecting them. We also compare the performance of our architecture with the method of replacing TCP with paced versions that is the generally proposed solution for small buffered networks. Simulations show that our proposed architecture has better performance than Paced TCP.

The rest of the article is organized as follows. Section 2 describes the basics of XCP algorithm, and details of proposed algorithm. Section 3 describes the simulation methodology and presents the simulation results on NSFNET. Finally, we conclude and describe our future work in Sect. 4.

2 XCP control

2.1 XCP basics

XCP is a new congestion control algorithm specifically designed for high-bandwidth and large-delay networks. It makes use of explicit feedbacks received from the network. Core routers are not required to maintain per-flow state information. XCP core routers maintain a per-link control-decision timer. When a timeout occurs, core routers update their link control parameters calculated by efficiency controller (EC) and fairness controller (FC) according to link utilization, spare bandwidth, and buffer occupancy. XCP sender agent sends its traffic rate information to 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 a feedback that shows the desired traffic rate change for this XCP flow and updates the feedback header of the packet. XCP receiver simply sends back the final feedback to the XCP sender. XCP sender updates its sending rate according to this feedback.

2.1.1 Efficiency controller

EC maximizes link utilization by controlling aggregate traffic. Every router calculates a desired increase or decrease in aggregate traffic for each output port according to 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 parameter and queue control parameter, respectively, 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, and Q is the persistent queue size.

2.1.2 *Fairness controller*

FC is responsible for fairly distributing the aggregate feedback Φ to flows. When Φ is positive, FC increases the transmission rate of all flows by the same amount. When Φ is negative, FC decreases the transmission rate of each flow proportional to flow's current transmission rate. In order to achieve fairness faster, 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.

When a router receives a packet containing feedback, if its own calculated feedback is smaller than the one in the header, it updates the feedback in the header with its own feedback. Otherwise, it does not change the feedback available in the header. When an XCP source agent receives an XCP feedback, it updates its congestion window size according to the formula $\text{cwnd} = \max(\text{cwnd} + H\_\text{feedback}, s)$, where s is the packet size and H_feedback is the feedback in the ACK packet.

2.2 Rate-based paced XCP

In Ref. [8], we proposed 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 macroflow like an LSP as shown in Fig. 1, and applies leaky-bucket pacing according to rate control to the macroflow and sends to a receiver XCP agent on destination edge node. The receiver XCP agent de-multiplexes the macroflow 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). However, calculating a new feedback for and updating the header of each optical packet at ultra high speed is hard. In our proposal, like in Simplified XCP-based Core Stateless Fair Queuing [9] and TeXCP [10] proposals, each macroflow 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. It is not a problem as DWDM allows many wavelengths on a single fiber.

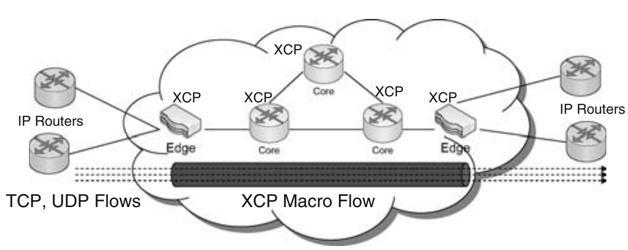

Fig. 1 Architecture

Core routers use a separate XCP control agent for each wavelength on an output link. When a probe packet of macroflow i arrives to a core router, FC of the XCP agent responsible for the wavelength that macroflow i was assigned to calculates a positive feedback p_i and a negative feedback n_i for flow i. Positive feedback is calculated by $\mathrm{p}_i = \frac{h+\max(0,\Phi)}{N}$ and negative feedback is calculated by $\mathrm{n}_i = \frac{u_i \cdot (h+\max(0,-\Phi))}{u}$, where N is the number of macroflows 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. $\text{feedback} = \mathrm{p}_i - \mathrm{n}_i$ 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 [10]. In [9], 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 OPS domain. 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 $\text{feedback} = \mathrm{p}_i - \mathrm{n}_i$ like in [9].

As explained in Ref. [8], 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 call the percentage ratio of given virtual capacity and real capacity as target utilization (TU).

We show that it is possible to use this property of XCP to limit the utilization of OPS network at a level that provides a low packet drop ratio with small buffers.

3 Evaluation

3.1 Simulation settings

Proposed network architecture and algorithms are implemented over *ns* version 2.32 [11]. As explained in detail in Ref. [8], we chose XCP's α, β, and γ parameters for edge routers as 0.2, 0.056, and 0.1, which decrease link utilization overshoots when compared with the default values in [9], respectively. These may not be the optimum values, but optimization of XCP parameters is left as a future work.

Simulator uses cut-through packet switching and buffering for data wavelengths. There is a single store-and-forward switching slow control wavelength dedicated for probe packets. Edge nodes use 0.25 s of electronic buffering that is a commonly used buffer size on the Internet. However, core routers use small only optical RAM for buffering optical packets on data wavelengths. Contention of probe packets on control wavelength is resolved by electronic RAM as O/E/O conversion is not a problem for control wavelength due to its low speed.

Figure 2 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 one or two data wavelengths depending on the simulation. All links have the same XCP TU. There are a total of 28 nodes (14 core nodes + 14 edge nodes) and 35 links (21 core links + 14 edge links) in the simulated network. The propagation delay of links between core and edge nodes is selected as 0.1 ms. XCP control period of core routers and probe packet sending interval of edge routers is selected as 50 ms by taking extra processing and queuing delays in the core routers into account. The capacity of the data and XCP control wavelengths are set to 1 Gbps and 100 Mbps, respectively.

TCP traffic is applied between edge nodes of the network. Throughout the simulation, a total of 1,586 TCP Reno flows are established according a Poisson flow arrival between randomly selected edge node pairs. Total simulation time is 40 s. Flow arrival times and their edge node pairs are the same for all simulations. We use the only data in the last 5 s of the simulations in the plots and comparisons. Data packet size is 1500 Bytes. Maximum TCP congestion window size is 20 packets. TU parameter of XCP is set to 100 and 30% of real capacity for output links of core nodes in order to show the effect of TU in our architecture. TU is always set to 100% for output links of edge nodes as they can use a high-capacity electronic buffer.

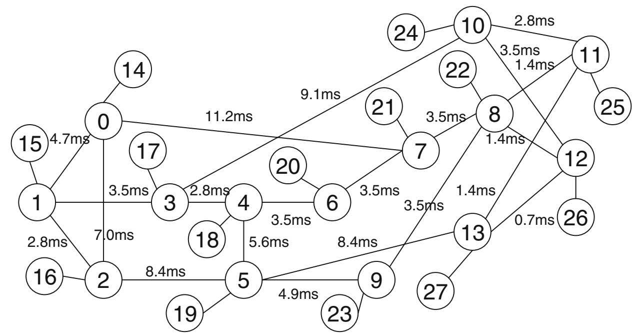

Fig. 2 NSFNET topology

We simulate the NSFNET topology with the same standard TCP traffic and compare the performance difference with and without our XCP-based architecture. Furthermore, we do comparison with the case when TCP is replaced with its paced version that is the most common proposed method in the literature for achieving high utilization in small buffered networks.

3.2 Evaluations

Figure 3 shows the average utilization of all output links of core nodes in NSFNET topology based on the link optical RAM buffer size capacity. X-axis shows the link buffer size in log scale and y-axis show the average link utilization in the range of 0 to 1 in linear scale where 1 means 100% utilization. Maximum average utilization in the network converges to around 70% as some links are under-utilized due to mesh topology and traffic matrix. In the figure, the dark blue line shows the utilization performance of our architecture when TU is set to 100% and there is a single data wavelength. We see that its performance is similar to Paced TCP proposal in the literature, which is shown by the pink line. Both of

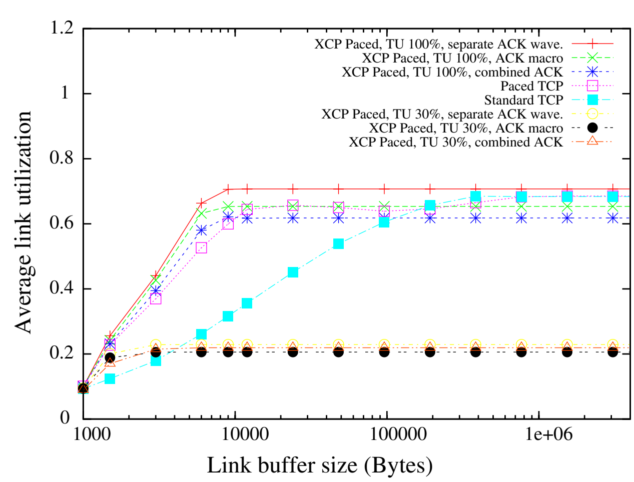

Fig. 3 Average link utilization

them achieve high utilization at a small buffer size of around 9–10 KB (around six data packet size). On the other hand, the light blue line, which shows the case when there is no XCP control or TCP pacing, hundreds of kilobytes of buffering is necessary in order to achieve the same high utilization.

A packet level tracing of the simulation result of our proposed architecture revealed that actually there is still room for improvement. We saw that the well-known ACK compression problem [12] causes some utilization inefficiency. In our architecture it is possible to solve this problem and increase utilization by simply using separate XCP macroflows for TCP ACK packets that is shown by light green line. However, using separate XCP macroflows for ACK packets on the same wavelength with macroflows for data packets may decrease the achievable utilization by XCP algorithm due to a well-known fairness problem [13] of XCP as average utilization of macroflows for ACK packets is generally less than macroflows for data packets. As a simple solution, ACK and data macroflows can be switched to different wavelengths, so ACK macroflows do not interfere with data macroflows and decrease the achievable utilization. In order to simulate it, we added one more data wavelength to the links and used it for only macroflows carrying ACK packets. The red line in Fig. 3 shows the average utilization of the wavelength for data packets after adapting these two simple solutions in our architecture. We see that it achieves the highest utilization in the simulations. Even if we do not apply these solutions for ACK compression, dark blue line shows that we still get higher average utilization than Paced TCP when the buffer size is very small like less than 9 KB. The last three lines show the performance of proposed XCP architectures in case TU is set to 30%. We see that they achieve an average utilization of around 20% at around 300 KB, which means a buffer size of only two data packets. Average utilization converges to around 20% due to traffic matrix.

Figure 4 shows the overall packet drop rate in the network at output links of core nodes. When the buffer size is very small like 1 KB, all simulated architectures have almost the same packet drop rate as packet drops mainly occur due to packet contentions irrelevant of burstiness. Light blue line shows that when there is no XCP pacing, TCP has a high packet drop rate in general unless buffer size is at least in the order of hundreds of kilobytes. Initially Paced TCP has a bit lower drop rate than normal TCP, when the buffer is more than 1 KB. However, its drop rate becomes equal to TCP when the buffer size is larger than 200 KB. On the other hand, XCP-based architectures show a much faster decrease and much lower packet drop rate than Paced TCP.

Due to multiplexing of many flows on core links like a real backbone link, there were not too much big and rapid changes at the incoming traffic rate, so setting TU to 100% did not cause so many drops from XCP utilization overshoots. TU can be selected a bit lower like 80–90% in order to make sure

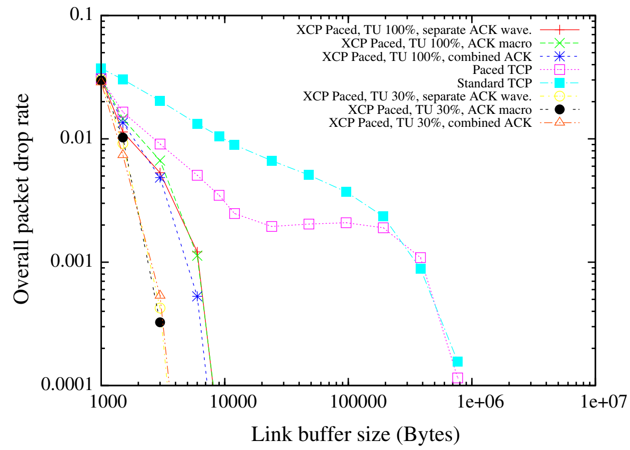

Fig. 4 Average packet drop rate

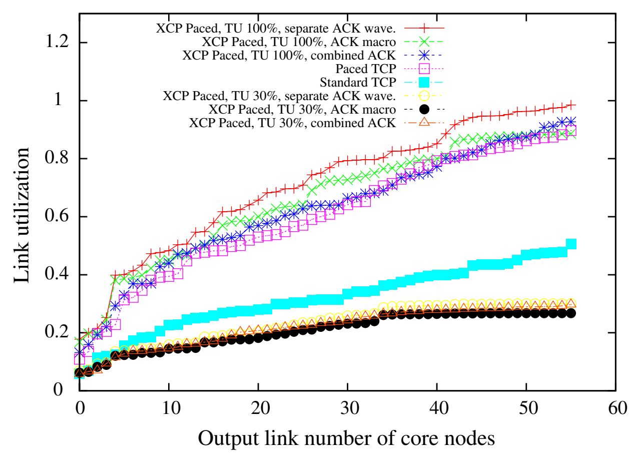

Fig. 5 Sorted utilization level of all core links

the links are always under-utilized in case there are rapid changes in the incoming traffic rate and if it is preferred to have a very low packet drop rate at the core links while achieving a high utilization. If extremely low packet drop rate is necessary and high utilization is not required, it is possible to achieve it by our architecture setting TU parameter to a low value like 30% as shown in the last three plots in Fig. 4.

Figure 5 shows the sorted utilization of data wavelength of all output links of core nodes when link buffer size is set to 9 KB. In all plots, utilizations are sorted independently from lowest to highest, so there is not a one-to-one correspondence between link numbers among plots. Some links are under-utilized and some of them are heavily congested due to the traffic matrix like in a real network. We see that setting the TU to 100% and using a separate wavelength for ACK packets gives the highest utilization on core links. Using the same wavelength for ACK packet macroflows and data packet macroflows gives the second highest utilization. If we use the same macroflows for both data and ACK packets, utilization is almost the same as Paced TCP at this simulated link buffer size of 9 KB. If we set even a smaller buffer size, Paced TCP would be penalized more and have lower utilization than our XCP-based architecture as shown in Fig. 3. Normal TCP without XCP control could achieve only half of the utilization level of the proposed solutions. The last three lines show the case when XCP's link TU is set to 30%. Separating ACK packets did not make a big difference when TU is low as all three plots are close to each other. We see that many links achieve the TU of 30% while the rest have lower utilization due to traffic matrix.

4 Conclusions

In this article, we showed the performance of our XCP-based architecture designed for OPS WDM networks with pacing at edge nodes for decreasing the optical RAM buffer requirements at core nodes. By using a mesh topology and applying TCP traffic, simulations show that even with a traffic based on normal TCP flows, our architecture gives higher utilization and much less drop rate than the proposal of replacing TCP on the Internet with paced versions, which is the generally proposed solution in the literature. Moreover, our proposal is much easier to realize than Paced TCP as it may be difficult to replace TCP stack of computers with paced version on the Internet in order to fully make use of small optical RAM buffered OPS networks in the future. Our proposal operates only at the edge/core routers of OPS domains and there is still no optical RAM-buffered OPS network deployed on the Internet. It is enough to apply our proposed XCP architecture on OPS networks when they become commercially available.

As a future work, we will evaluate the electronic buffer size requirements for pacing and XCP control at the edge nodes, and try to optimize parameters like XCP parameter set for further improving the performance. We will work on new pacing mechanisms for XCP as there may be better pacing methods than leaky-bucket pacing as shown in Ref. [14].

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

References

[1] Villamizar, C., Song, C.: High performance TCP in ANSNET. Comput. Commun. Rev. **24**(5), 45–60 (1994)
[2] Takahashi, R., Nakahara, T., Takahata, K., Takenouchi, H., Yasui, T., Kondo, N., Suzuki, H.: Photonic random access memory for 40-Gb/s 16-b burst optical packets. IEEE Photon. Technol. Lett. **16**, 1185–1187 (2004)
[3] Aoyama, T.: New generation network(NWGN) beyond NGN in Japan. http://akari-project.nict.go.jp/document/INFOCOM2007.pdf (2007)
[4] Appenzeller, G., Sommers, J., McKeown, N.: Sizing router buffers. In: Proceedings of ACM SIGCOMM, pp. 281–292 (2004)
[5] Enachescu, M., Ganjali, Y., Goel, A., McKeown, N., Roughgarden, T.: Routers with very small buffers. Part III. ACM SIGCOMM Comput. Commun. Rev. **35**, 83–90 (2005)
[6] Zhang, L., Shenker, S., Clark, D.D.: Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic. In: Proceeedings of ACM SIGCOMM, pp. 133–147 (1991)
[7] Theagarajan, G., Ravichandran, S., Sivaraman, V.: An experimental study of router buffer sizing for mixed TCP and real-time traffic. In: Proceedings of IEE ICON (2006)
[8] Alparslan, O., Arakawa, S., Murata, M.: Rate-based pacing for small buffered optical packet-switched networks. J. Opt. Netw. **6**(9), 1116–1128 (2007)
[9] Katabi, D., Handley, M., Rohrs, C.: Congestion control for high bandwidth-delay product. In: Proceedings of ACM SIGCOMM, pp. 42–49 (2002)
[10] Kandula, S., Katabi, D., Davie, B., Charny, A.: Walking the tightrope: responsive yet stable traffic engineering. In: Proceedings of ACM SIGCOMM, pp. 253–264 (2005)
[11] McCanne, S., Floyd, S.: ns Network simulator. http://www.isi.edu/nsnam/ns/ (2002)
[12] Mogul, J.C.: Observing TCP dynamics in real networks. In: Proceedings of ACM SIGCOMM, pp. 305–317 (1992)
[13] Low, S.H., Andrew, L.A., Wydrowski, B.: Understanding XCP: equilibrium and fairness. In: INFOCOM, pp. 1025–1036 (2005)
[14] Abendroth, D., Below, K., Killat, U.: The interaction between TCP and traffic shapers—clever alternatives to the leaky bucket. In: GLOBECOM (2002)

Author Biographies

Onur Alparslan received the B.S and M.S. degrees in Electrical and Electronics Engineering from Bilkent University, Turkey, and Ph.D. degree in Information Science from Osaka University, Japan, in 2002, 2005, and 2008, respectively. He is now a postdoctoral researcher at Graduate School of Information Science and Technology, Osaka University, Japan. His research interests include optical networks, simulation, and network design.

Shin'ichi Arakawa received the M.E. and D.E. degrees in Informatics and Mathematical Science from Osaka University, Japan, in 2000 and 2003, respectively. He is currently a Research Assistant at the Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of photonic networks. He is a member of IEEE and IEICE.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined IBM Japan's Tokyo Research Laboratory as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with the Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Osaka University, and since April 2004, he has been with the Graduate School of Information Science and Technology, Osaka University. He has contributed more than 450 papers to international and domestic journals and conferences. His research interests include computer communication networks and performance modeling and evaluation. He is an IEICE Fellow. He is a member of IEEE, ACM, The Internet Society, IEICE, and IPSJ.