TCP and UDP Performance for Internet over Optical Packet-Switched Networks

Jingyi He[†] S.-H. Gary Chan[‡]
[†]Department of Electrical and Electronic Engineering
[‡]Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong

Abstract— **Optical packet-switched (OPS) network is a strong candidate for the future Optical Internet. In this paper, we study packet aggregation and deflection routing as employed in OPS networks on the performance of upper layer Internet protocols represented by TCP and UDP. Regarding packet aggregation schemes, we study no aggregation, mixed-flow aggregation, and per-flow aggregation. Our results show that for both TCP and UDP, mixed-flow aggregation achieves the highest throughput, and per-flow aggregation significantly decreases the fairness at large aggregation intervals. Both aggregation schemes increase UDP delay jitter. Regarding deflection routing, we show that deflection routing significantly improves TCP throughput in spite of the out-of-order packet delivery. However, the congestion of the deflection path significantly affects the improvement that can be achieved. UDP throughput does not suffer from out-of-order packet delivery, and hence its throughput improvement by deflection routing can be even more prominent. The larger the deflection cost (delay difference between the deflection path and the shortest path) is, the larger UDP delay jitter results. Deflection cost, however, does not affect the throughput and fairness very much for both TCP and UDP.**

I. Introduction

Driven by the ever-increasing demand for bandwidth, the core of the Internet has been evolving from an electronic network to an optical one (i.e., the so-called Optical Internet). Optical packet-switched (OPS) network is a strong candidate for this Optical Internet. Compared with wavelength-routed network, which is deemed as another candidate and offers bandwidth granularity only at the wavelength level, OPS network offers bandwidth granularity at the packet level, resulting in a much higher bandwidth efficiency. Moreover, because of its packet-switched nature, OPS network can have richer routing functionalities and greater flexibility in supporting diverse services [1]. In this paper, we will focus on delivering Internet traffic over an OPS core network.

In an OPS-based Internet, end users are still attached to electronic networks. Their traffic is aggregated before being forwarded to the OPS backbone. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are currently the predominant transport protocols used at the end users. These packets will eventually be forwarded in the optical core. In this paper, we explore the impact of mechanisms as used in the OPS network (namely, routing and packet handling) on the performance of the Internet protocol suite. The performance of interest are throughput (goodput), fairness, and, in case of UDP, delay jitter.

In particular, we investigate the following two mechanisms as employed in OPS networks which potentially impact the performance of TCP and UDP:

- Packet aggregation — In OPS networks, IP packets destined to the same destination can be aggregated to form a larger optical packet. This is necessary especially when the optical switches are not fast enough to process individual IP packets. Clearly, packet aggregation implies waiting delay for some packets. Moreover, an optical packet may contain multiple packets from the same flow. When it is dropped or mis-routed in the OPS network, the throughput of the flow may be adversely affected. In this regard, how packet aggregation affects the TCP and UDP performance is an important issue. We study in this paper three aggregation schemes: no aggregation, aggregating packets from heterogeneous flows into an optical packet (mixed-flow aggregation), and aggregating packets from the same flow (per-flow aggregation).
- Deflection routing — In packet-switched networks, contention occurs when two or more packets are to be forwarded to the same output port. In electronic networks, such contention is resolved in a store-and-forward manner by temporarily buffering the contending packets in random access memory (RAM). However, since there is no effective optical RAM, contention in OPS networks can usually be resolved by deflection routing, in which the packets losing the contention are temporarily misrouted, or "deflected", to other output ports.[1] Clearly, deflection routing avoids packet being lost by routing them to longer paths. Deflection routing also leads to out-of-order delivery of packets. For a protocol which has reassembly deadline and congestion control mechanism such as TCP, this will adversely affect its performance.

In this paper, we study the impact of the aforementioned schemes as used in unslotted irregular OPS networks on TCP and UDP performance. Our results show that:

1) For both TCP and UDP, mixed-flow aggregation achieves the highest throughput, and per-flow aggregation significantly decreases the fairness at large aggregation intervals. Both aggregation schemes increases UDP delay jitter.
2) Deflection routing significantly improves TCP throughput in spite of the out-of-order packet delivery. However, the congestion of the deflection path significantly affects the improvement that can be achieved. UDP throughput does not suffer from out-of-order packet delivery, and hence its throughput improvement by deflection routing can be more prominent.
3) Deflection cost (delay difference between the deflection path and the shortest path) does not affect the throughput and fairness very much for both TCP and UDP, while a larger one

This work was supported, in part, by the Areas of Excellence (AoE) Scheme on Information Technology funded by the University Grant Council in Hong Kong (AoE/E-01/99), and by the Competitive Earmarked Research Grant of the Hong Kong Research Grant Council (HKUST6014/01E & HKUST6199/02E).

[1] Another way to resolve contention is wavelength conversion. However, since it is much more expensive, we would not consider it in this study.

0-7803-7802-4/03/$17.00 © 2003 IEEE

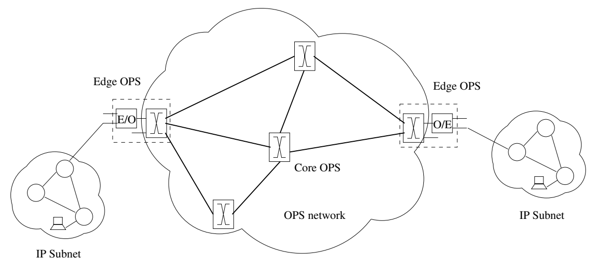


Fig. 1. An optical packet-switched (OPS) network used to transport traffic between IP subnetworks.

results in larger UDP delay jitter.

We briefly discuss previous work as follows. Deflection routing has long been proposed and studied. However, it is mainly investigated in *slotted* optical network of *regular* topology such as the Manhattan Street Network (MS-Net) and the ShuffleNet [2], [3]. We believe that in the future Optical Internet, unslotted network with irregular mesh topology is more likely [4]. There are several pieces of work that study deflection routing in irregular OPS networks [5], [6]. However, these studies have not considered the effect of out-of-order packet delivery by assuming that the receiver has an infinite reassembly time and buffer. Though the effect of finite reassembly buffer has been discussed in [7], the study assumes a very simple flow control model markedly different from TCP, where duplicated acknowledgement, fast retransmit, and fast recovery are involved. We have studied the impact of deflection routing on TCP performance in an unslotted irregular network in [8]. However, the paper has not considered the special characteristics of OPS networks such as the lack of optical RAM and the use of packet aggregation. Packet aggregation was previously studied mainly in the context of web traffic in the Internet [9], [10].

The rest of this paper is organized as follows. We present in Section II the system description of the OPS network under study and our simulation model. In Section III, we present some illustrative simulation results of the impact of packet aggregation on TCP and UDP performance. In Section IV, we present the effect of deflection routing. We conclude in Section V.

II. System Description and Simulation Model

A. System Description

We consider that an OPS network is used to transport the traffic between IP subnetworks, as shown in Fig. 1. The OPS network consists of edge OPSes and core OPSes inter-connected by optical links. The core OPSes are responsible for route selection and label swapping for the optical packets. The edge OPSes interface with IP subnetworks, and are responsible for packet classification, aggregation, and segregation, in addition to the switching functionalities.

An IP packet encounters two edge OPSes on its path to the destination: an ingress OPS where it enters the OPS network and an egress OPS where it leaves the network. At the ingress OPS, the IP packet may firstly be buffered and then aggregated with other packets to form a larger packet. These aggregated IP packets are transmitted into the OPS network after an addition of an optical packet header and electronic-optical (E/O) conversion. When the optical packet reaches the egress OPS, it is segregated through optical-electronic (O/E) conversion into individual IP packets, which are then transmitted to their respective destinations.

In terms of packet aggregation, we consider a scheme in which an aggregate packet is formed when a fixed aggregation interval expires, i.e., *fixed-interval aggregation*. In practice, we may also want to limit the size of an aggregate packet. In that case, an aggregation interval expires when the size of the aggregate packet reaches a predetermined threshold. There can be two ways to aggregate packets:

- *Mixed-flow aggregation*: In this scheme, IP packets from different flows may be aggregated into an optical packet as long as they are destined to the same egress OPS. Therefore, at an ingress OPS, only one aggregation timer is needed for each egress OPS.
- *Per-flow aggregation*: In this scheme, an optical packet can only consist of IP packets from the *same* flow. For each active flow, an aggregation timer is needed, which starts from the first packet of the flow at the ingress OPS. This scheme makes the manipulation of individual flows possible. However, it is clearly more complex: IP packets are to be further differentiated according to their flow identification and per-flow queueing is needed at the ingress OPS.

We next discuss the condition under which deflection occurs. Note that a newly-formed optical packet at the ingress OPS may not be able to be sent out to its preferred outgoing port right away, because the port is occupied by another optical packet in transmission. For example, if the bandwidth of the outgoing link is B, the size of an earlier optical packet is S, and the interval between the formation time of the new optical packet and that of the earlier packet is τ, the new packet will not be able to be forwarded to the same outgoing port if $\tau < S/B$. In this case, deflection routing can be used to resolve the contention. At the core OPSes, without optical buffer, deflection happens when $\tau' < S/B$, where τ' is the inter-arrival time between two optical packets contending for the same output link.

B. Simulation Model

The simulation model we use is shown in Fig. 2. We focus on a set of flows which are to be transported by an OPS network, and enter at the same ingress switch r_0 and leave at the same egress switch r_1. The flows are sourced at nodes s_i $(i = 1, 2, \ldots, n)$ and destined to nodes d_i $(i = 1, 2, \ldots, n)$. We consider that for each $s_i - d_i$ pair there are m flows. The link connecting s_i with r_0 and that connecting d_i with r_1 have bandwidth B_i and delay t_i. All the links inside the OPS network have bandwidth B_0. The delay of the shortest path between r_0 and r_1 is assumed to be t_0 (a constant). The optical packets deflected away from the shortest path will be subjected to a longer (random) delay of $t_0 + t_c$, where t_c is a positive random variable indicating the deflection cost. At the ingress OPS r_0, the IP packets are aggregated with an aggregation interval Δ, and size limit $B_0\Delta$.

After an aggregate packet enters the OPS network, it may be deflected at different points and for different number of times, and some of them may even be dropped because of the ongoing background traffic on their deflection paths. In order to capture this situation properly for a general network, we introduce in our model a parameter p, which is defined as the probability that an optical packet deflected at its ingress OPS is dropped on the deflection path. Clearly, p depends on the traffic load on the deflection path: p is small if there is little background traffic and large if there is heavy traffic on the deflection path. Hence we call p the *congestion factor* of the deflection path. For simplicity, we assume that the packets forwarded to the shortest path are not dropped.

In the reverse direction, r_1 acts as the ingress OPS and r_0 the egress OPS. The traffic in this direction (e.g., TCP ACKs) is treated in the same manner as the traffic in the forward direction.

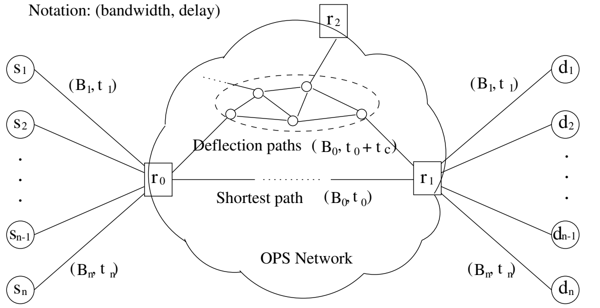


Fig. 2. The simulation model we use.

TABLE I
BASELINE PARAMETERS USED IN THE SIMULATION.

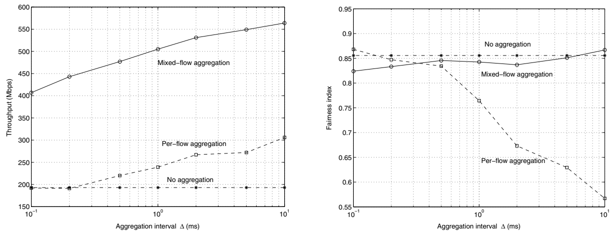
Symbol	Definition	Default value
n	Number of source and destination nodes	10
m	Number of flows for each $s_i - d_i$ pair	50
p	Congestion factor of the deflection paths	0.2
B_0	Bandwidth of the links in the OPS network	622 Mbps
B_i	Bandwidth of the links outside the OPS network	155 Mbps
L	Offered UDP load	800 Mbps
t_0	Delay of the shortest path between r_0 and r_1	10 ms
t_i	Delay of the links outside the OPS network	$2i$ ms
t_v	Interval of the delay changes	$[1, 100]$ ms
Δ	Aggregation interval	1 ms
μ	Mean of the deflection cost	5 ms
σ	Standard deviation of the deflection cost	4 ms
S_0	IP Packet size	1500 bytes

We use the *ns*-2 network simulator to do our simulation [11].[2] In the simulation, the aggregation interval Δ varies from 0.1 ms to 10 ms, as most of the up-to-date all-optical switches have a switching speed in the order of milliseconds. We let t_c be a truncated Gaussian random variable with mean μ and standard deviation σ, i.e., $t_c = \max(0, N(\mu, \sigma))$. Since the background traffic on the deflection paths may vary with time, we let the delay of the deflection path $t_0 + t_c$ change every t_v time, where t_v is uniformly distributed over a certain time interval, for example, $[1, 100]$ ms. In the simulation, all the flows have the same packet size of $S_0 = 1500$ bytes. The starting times of the flows are uniformly distributed in a time interval (e.g., $[0.1, 0.5]$ s) to avoid global synchronization. For TCP flows, we use TCP-Reno and bulk data transfer (e.g., FTP) traffic. For UDP, the packets of each flow arrive according to a Poisson process, with an overall traffic load of L. The default (baseline) parameters used in our simulation are summarized in Table I.

We are mainly interested in the throughput, fairness among the flows, and packet delay jitters (of UDP flows). In terms of fairness, we use the Jain's fairness index defined as [12]:

$$f = \frac{\left(\sum_{i=1}^{N} x_i\right)^2}{N \sum_{i=1}^{N} x_i^2}, \tag{1}$$

where N is the total number of flows and x_i is the throughput of flow i, $i = 1, \ldots, N$. The value of Jain's fairness index is always between zero and one, i.e., $f \in (0, 1]$. A larger value means better fairness, with $f = 1$ meaning perfect fairness. In terms of delay jitter of a UDP flow, we use the standard deviation of packet delay of that flow. We define the overall delay jitter as the average jitter over all flows.



(a) Throughput (b) Fairness index

Fig. 3. TCP performance versus Δ with different aggregation schemes.

III. EFFECT OF PACKET AGGREGATION

In this section, we present the illustrative results of TCP and UDP performance over the OPS network, with different packet aggregation schemes employed: no aggregation, mixed-flow aggregation, and per-flow aggregation. As aggregation is performed every fixed interval, we study the impact of using different interval values.

A. TCP Performance

We show in Fig. 3 the TCP performance with respect to the aggregation interval Δ. The case without aggregation (a flat line in both of the figures) is also shown for comparison. We first look at the overall TCP throughput as shown in Fig. 3(a). Mixed-flow aggregation achieves much higher TCP throughput than per-flow aggregation and no aggregation. Per-flow aggregation achieves higher throughput than no aggregation only at large aggregation intervals. For both mixed-flow aggregation and per-flow aggregation, TCP throughput increases with Δ.

TCP throughput is affected by packet aggregation in the following ways. First, mixed-flow aggregation reduces the contention at the ingress OPS by regulating the incoming traffic, resulting in higher throughput. Without packet aggregation, the contention at the ingress OPS is significant because of the bursty nature of TCP traffic, and remarkable packet losses result. Mixed-flow aggregation aggregates the packets and forwards them to the optical switch every fixed interval. Therefore, the traffic to be switched becomes much more regular, and packet losses are significantly reduced. Per-flow aggregation, however, does not help reduce the contention, because the aggregation timers of the flows are not synchronized, and hence aggregated packets from different flows are forwarded to the optical switch still in a random manner. Even worse is that with larger (aggregate) packet size an idle period is likely larger and hence the bandwidth is less efficiently used. We will show how this effect decreases the throughput in UDP case later. For the TCP case here, however, this effect is not very severe, because the traffic load is relatively low (as indicated by the throughput) and hence the aggregate packet size is not large. Actually, we observe that per-flow aggregation achieves even higher throughput than no aggregation at large aggregation intervals. The major reasons are less TCP retransmissions and unfair share of bandwidth among the flows, as will be explained in the following.

Packet aggregation reduces TCP retransmissions when deflection routing is used, hence improving TCP throughput. With deflection routing, a deflected TCP packet will arrive late at its destination and possibly trigger TCP fast retransmit (if more than three packets of larger sequence numbers have arrived through the shortest path). Given the burstiness of TCP traffic, a mixed-flow aggregate packet

[2] Note that *ns*-2 is for simulating store-and-forward networks, while in an OPS network without optical buffer, optical packets are transmitted in a cut-through manner without any queueing delay at intermediate nodes. Modifications are hence needed to account for this characteristic.

is likely to consist of packets from only a certain number of flows, with multiple packets for each flow. As the multiple packets from the same flow are in order among themselves, when an aggregate packet is deflected, the block of in-order packets which will arrive late in general does not cause more TCP retransmissions than a single deflected TCP packet. This is also true for per-flow aggregation. With a larger Δ, deflection occurs less frequently, resulting in fewer retransmissions. Therefore, the TCP congestion window can be kept at a larger value on average, as well as there are fewer duplicated packets wasting the bandwidth. The overall effect is larger TCP throughput for larger Δ. Of course, Δ could not be too large either, because that would result in too large delay for some packets, which may break the TCP pipelining transmission.

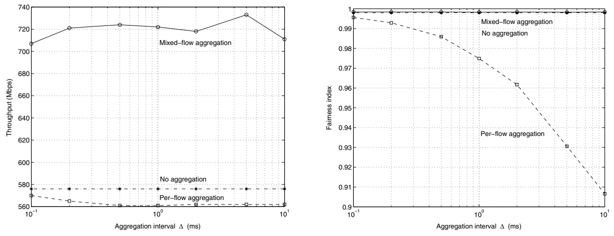
We next look at the TCP fairness as shown in Fig. 3(b). In general, packet aggregation degrades the TCP fairness. This is because packet aggregation does not pack packets evenly from all the flows, given the burstiness of TCP traffic. With (aggregate) packet losses and deflections, the unfairness among the flows may therefore be worsened. With mixed-flow aggregation, TCP fairness index slightly increases with Δ. This is reasonable because with a larger Δ an aggregate packet consists of packets from more flows. When it is deflected or dropped, the effect is more evenly distributed among all the flows. For per-flow aggregation, however, TCP fairness index decreases when Δ is large. This is because large Δ leads to large aggregate packets. As the TCP packets in an aggregate packet are all from the same flow, a loss of such an aggregate packet easily leads to TCP timeout. Therefore, some flows are likely severely starved while giving bandwidth to others, resulting in even worse fairness. In such cases, however, the overall TCP throughput can be even larger, because a small number of flows occupying most of the bandwidth means that these flows suffer few packet losses and retransmissions, and hence most of the bandwidth accounts for the throughput (goodput). This has been demonstrated in Fig. 3(a).

B. UDP Performance

We show in Fig. 4 UDP throughput and fairness versus Δ, with an overall traffic load $L = 800$ Mbps. Regarding throughput, like the TCP case, mixed-flow aggregation achieves higher values than per-flow aggregation and no aggregation, because it reduces contention and hence packet losses. On the other hand, unlike TCP, per-flow aggregation achieves lower throughput than no aggregation. The major reason is the inefficient use of bandwidth because of the large aggregate packet size (given the high traffic load here), as we have discussed earlier. We also observe that UDP throughput does not show much dependence on Δ. The reason is that UDP does not go through back-off mechanism as in TCP for out-of-order packet delivery, hence the effect of larger Δ in reducing retransmissions for TCP does not apply here.

Regarding fairness, both mixed-flow aggregation and no aggregation achieve almost perfect fairness. Per-flow aggregation, however, achieves fairness degrading as Δ increases. This is because all the flows share the same Δ, and hence a flow once suffers from a deflection loss of its optical packets would more likely suffer from such losses again later. As a result, some flows suffer from packet loss more often than others, leading to an unfair share of bandwidth. As Δ increases, the fairness decreases because the loss from one interval to another becomes more correlated.

Packet delay jitter is another important performance metric for UDP traffic. We show in Fig. 5 the average delay jitter over all the UDP flows versus Δ. Packet aggregation introduces aggregation delay which increases the delay jitter. The larger Δ is, the larger delay jitter results. With the given load $L = 800$ Mbps, the packet arrival rate of each flow is one packet per 7.5 ms. For per-flow aggregation, when the aggregation interval Δ is small, in most cases an aggregate packet will consists of only one IP packet. In this case, nearly all the deflected packets are subjected to the same aggregation delay (statistically). Therefore, the delay jitter resulting from aggregation delay is small. When Δ is larger, the probability that an aggregate packet consists of multiple IP packets is larger. In such case, the IP packets in an aggregate packet suffer much different aggregation delays, resulting a fast increase in the overall delay jitter. For mixed-flow aggregation, with the given traffic load, an aggregate packet is generally formed and forwarded when its size threshold is reached before the aggregation interval expires. In this case, the randomness of aggregation delay is larger than that of per-flow aggregation when Δ is small. We hence observe that mixed-flow aggregation results in larger delay jitter than per-flow aggregation when Δ is not very large. On the other hand, when Δ is large, this mechanism results in significant reduction in aggregation delay compared with the fixed aggregation interval. Therefore, mixed-flow aggregation can achieve a smaller delay jitter than per-flow aggregation when Δ is large.



(a) Throughput (b) Fairness index

Fig. 4. UDP performance versus Δ with different aggregation schemes.

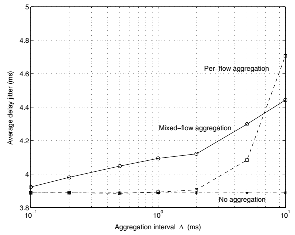
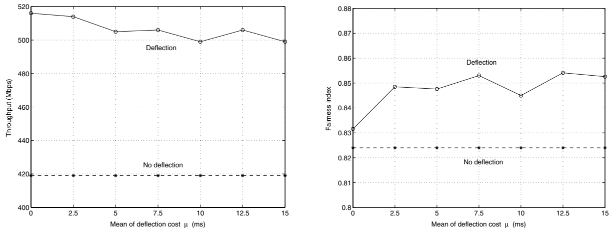


Fig. 5. UDP packet delay jitter versus Δ with different aggregation schemes.

IV. Effect of Deflection Routing

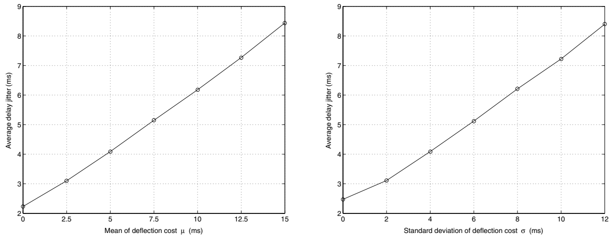
In this section, we study the effect of deflection routing on TCP and UDP performance. Specifically, we study the influence of deflection cost (mean μ and standard deviation σ) and the congestion factor p, with mixed-flow aggregation and $\Delta = 1$ ms.

We show in Fig. 6 TCP throughput and fairness index versus the mean deflection cost μ. The case without deflection (the flat line) is also shown for comparison. We can see that deflection routing significantly improves TCP throughput ($\sim 20\%$), although it results in out-of-order packet delivery. The reason is that packet loss is more harmful to TCP than out-of-order packet delivery. Moreover, deflection routing also improves TCP fairness. As can be seen, however, the throughput and fairness index do not show much dependency on μ. In an OPS network without optical buffer, there is no queueing delay at intermediate nodes, and hence a deflected aggregate packet will arrive (at the egress OPS) later than its following one along the shortest path as long as its deflection cost is larger than the aggregation interval.



(a) Throughput (b) Fairness index

Fig. 6. TCP performance versus the mean of deflection cost μ.



(a) Delay jitter vs. μ (b) Delay jitter vs. σ

Fig. 7. UDP delay jitter versus μ and σ.

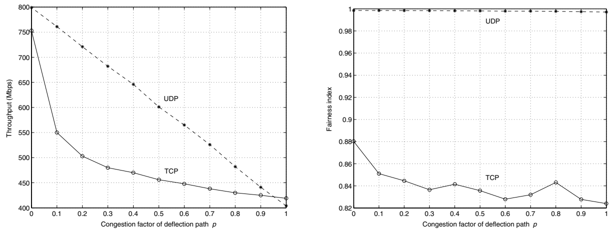
This is in general true in our study, given $\Delta = 1$ ms and $\sigma = 4$ ms. Therefore, in terms of triggering TCP fast retransmit, larger μ does not make much difference. We observe a slightly decreasing trend of throughput, which is mainly because larger μ results in larger average round-trip time (RTT). Our simulation results also show that the standard deviation of the deflection cost σ does not affect TCP throughput and fairness either.

With or without deflection, UDP can always achieve very good fairness. Regarding throughput, as UDP does not have any back-off mechanism in response to out-of-order packet delivery, deflected packets directly account for throughput improvement, and the improvement can be even more prominent than TCP. For the same reason, UDP throughput is independent of μ and σ. However, UDP delay jitter performance is affected by out-of-order packet delivery. Our simulation results indicate that the average delay jitter increases linearly with both μ and σ, as shown in Fig. 7.

We finally show in Fig. 8 the effect of p on TCP and UDP performance. As expected, larger p results in more packet losses and hence lower throughput for both TCP and UDP. For UDP, the throughput decreases linearly with respect to p, because the amount of packet losses is proportional to p. For TCP, in contrast, the throughput drops sharply as p increases from 0 to 0.1, and then keeps on decreasing more mildly. This actually means that packet loss is very harmful to TCP throughput, and the loss of deflected packet significantly decreases the throughput improvement by deflection routing. Regarding fairness, UDP can achieve almost perfect fairness irrespective of p. For TCP, the fairness index shows an obvious drop when p increases from 0 to 0.1, similar to the throughput performance. The increase of p beyond that does not show much effect on the fairness index. Our results also show that larger p leads to smaller UDP delay jitter, because fewer deflected packets are received.

V. Conclusions

Optical packet-switched (OPS) networks will likely carry Internet traffic in the future. In this paper, we have examined the performance of Internet protocols (TCP and UDP) over OPS networks. In particular, we have studied the impact of packet aggregation and deflection routing as used in the underlying OPS networks on TCP/UDP performance. The performance metrics we are interested in are throughput, fairness, and delay jitter. Regarding packet aggregation schemes, we have studied no aggregation, mixed-flow aggregation and per-flow aggregation. Our results show that for both TCP and UDP, mixed-flow aggregation achieves the highest throughput, and per-flow aggregation significantly decreases the fairness at large aggregation intervals. Both aggregation schemes increases UDP delay jitter. Regarding deflection routing, we show that deflection routing significantly improves TCP throughput in spite of the out-of-order packet delivery. However, the congestion of the deflection path significantly affects the improvement that can be achieved. UDP throughput does not suffer from out-of-order packet delivery, and hence its throughput improvement by deflection routing can be even more prominent. Deflection cost does not affect the throughput and fairness very much for both TCP and UDP, while larger deflection cost results in larger UDP delay jitter.



(a) Throughput (b) Fairness index

Fig. 8. TCP and UDP performance versus p.

References

[1] M. J. O'Mahony, D. Simeonidou, D. K. Hunter, and A. Tzanakaki, "The application of optical packet switching in future communication networks," *IEEE Communications Magazine*, vol. 39, no. 3, pp. 128–135, Mar. 2001.

[2] A. K. Choudhury, "A comparative study of architectures for deflection routing," in *Proc. IEEE GLOBECOM'92*, pp. 1911–1920, Dec. 1992.

[3] S.-H. G. Chan and H. Kobayashi, "Packet scheduling algorithms and performance of a buffered shufflenet with deflection routing," *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 4, pp. 490–501, Apr. 2000.

[4] T. Chich, J. Cohen, and P. Fraigniaud, "Unslotted deflection routing: a practical and efficient protocol for multihop optical networks," *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 47–59, Feb. 2001.

[5] S. Yao, B. Mukherjee, S. J. B. Yoo, and S. Dixit, "All-optical packet-switched networks: A study of contention-resolution schemes in an irregular mesh network with variable-sized packets," in *Proc. OptiComm 2000*, pp. 235–246, Nov. 2000.

[6] J. P. Jue, "An algorithm for loopless deflection in photonic packet-switched networks," in *Proc. IEEE ICC'2002*, pp. 2776 –2780, Apr. 2002.

[7] A. K. Choudhury and N. F. Maxemchuk, "Effect of a finite reassembly buffer on the performance of deflection routing," in *Proc. IEEE ICC'91*, pp. 1637–1646, Jun. 1991.

[8] J. He and S.-H. G. Chan, "TCP performance with deflection routing in the Internet," in *Proc. IEEE ICON'2002*, pp. 383–388, Aug. 2002.

[9] B. R. Badrinath and P. Sudame, "Gathercast: The design and implementation of a programmable aggregation mechanism for the Internet," in *Proc. Ninth International Conference on Computer Communications and Networks*, pp. 206–213, 2000.

[10] H. Tounsi, L. Toutain, and F. Kamoun, "Small packets aggregation in an IP domain," in *Proc. Sixth IEEE Symposium on Computers and Communications*, pp. 708–713, 2001.

[11] "http://www.isi.edu/nsnam/ns."

[12] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," in *DEC Research Report TR-301*, Sept. 1984.