Available online at www.sciencedirect.com
ScienceDirect
Int. J. Electron. Commun. (AEÜ) 63 (2009) 294–303
International Journal of Electronics and Communications
www.elsevier.de/aeue

Improving Reno and New-Reno's performances over OBS networks without SACK

Yan Zhang*, Lemin Li, Sheng Wang

Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks, University of Electronic Science and Technology of China, Chengdu 610054, China

Received 13 August 2007; accepted 3 February 2008

Abstract

In TCP over optical burst switching (OBS) networks, consecutive multiple packet losses are common since an optical burst usually contains a number of consecutive packets from the same TCP sender. It has been proved that over OBS networks Reno and New-Reno achieve lower throughput performances than that of SACK, which can address the inefficiency of Reno and New-Reno in dealing with consecutive multiple packet losses. However, SACK adopts complex mechanisms not only at the sender's but also at the receiver's protocol stack, and thus has a higher difficulty in deployment.

In this paper we propose B-Reno, a new TCP implementation designed for TCP over OBS networks. Using some simple modifications to New-Reno only at the sender's protocol stack, B-Reno can overcome the inefficiencies of Reno and New-Reno in dealing with consecutive multiple packet losses and thus improve their throughputs over OBS networks. Moreover, B-Reno can also achieve performance similar with that of SACK over OBS networks while avoiding SACK's difficulty in deployment due to complex mechanisms at both the sender's and the receiver's protocol stack.

Keywords: TCP; OBS; Throughput

1. Introduction

With the rapid progress of the Internet's traffic amount and with the evolution and maturation of the optical technology, IP over wavelength division multiplexing (WDM) networking seems to be a promising architecture to support the expected huge Bandwidth demand of next generation networks (NGN). However, due to the absence of optical processing and buffering capability, optical packet switching (OPS) is still immature to provide effective queuing and control logic, so it can be considered only as a medium to long-term solution. In the same time another promising approach is optical burst switching (OBS), which can efficiently utilize the huge capacity of Bandwidth provided by optical communications and while avoiding the need for optical buffering [1,2].

Being regarded as a promising technology to support the future IP over WDM networking, OBS has received a lot of attention [3]. Since TCP is the dominating protocol for data transmission in Internet today and will likely remain so in NGN, recently the performance of TCP over OBS networks has been investigated [4–7].

The works in [4–6] mainly focus on the impact of burst assembly on TCP performance over OBS networks. Although the burstification process increases the value of round trip time (RTT) and thus negatively affects the TCP performance, the correlated successful transmission of packets in a burst can increase the congestion window faster than in traditional networks and thus result in positive effects on TCP performance [4,5]. Moreover, we can use adaptive burst

* Corresponding author. Fax: +86 28 8320113.
E-mail address: zhangy@uestc.edu.cn (Y. Zhang).

1434-8411/$ - see front matter © 2008 Elsevier GmbH. All rights reserved.
doi:10.1016/j.aeue.2008.02.002

assembly algorithms instead of static burst assembly algorithms to improve TCP performance [6]. On the other hand, the work in [7] studies TCP's false timeout (FTO) over OBS networks, which can be seen as a remarkable impact of burst losses on TCP performance. FTO can be introduced as follows. Due to the bufferless switching at OBS core nodes, a burst is often dropped because of contention rather than congestion. So, the burst loss in OBS networks should be more or less random, even at low traffic loads. However, in a TCP over OBS network, the random burst loss due to contention may be falsely interpreted as congestion by the TCP layer. For example, if a burst containing all of the packets/segments[1] in a TCP sending window is lost due to contention, the TCP sender will be forced to timeout (TO). Since such a TO is not a real TO that indicates network congestion, it is called FTO [7]. FTO can unnecessarily force TCP sender to perform Slow Start, and thus result in significant performance degradation.

To alleviate the performance degradation caused by FTO in TCP over OBS networks, the work in [7] proposes B-TCP, a TCP implementation using some schemes that can detect FTOs and react with Fast Retransmission when a FTO is detected. However, these schemes require either that the TCP sender can estimate the maximal number of packets contained in a burst, or that the OBS nodes can provide the information of the packets contained in a burst to the TCP sender. On the other hand, the works in [8–10] adopt the method of burst retransmission in OBS network to reduce the randomness of burst losses due to contention, and thus can decrease the probability of FTO and improve TCP performance. However, the method of burst retransmission also needs special support from the OBS nodes.

According to [4–10], the cause of FTO is the randomness of burst losses due to the bufferless nature of the OBS core nodes, and FTOs are most likely to occur for fast TCP flows. However, we find that these conclusions are not so accurate for Reno, the most popular TCP implementation. As mentioned in [7], in a TCP over OBS network, a burst usually contains a number of consecutive packets from a TCP sender. So, once a burst is lost, consecutive multiple packets contained in this burst are consequently lost for the TCP sender. According to the simulation results in [11], even a small number of consecutive multiple packet losses can force Reno's sender to timeout. Specifically, two consecutive packet losses can force Reno to timeout when the congestion window size is less than 10 packets, and at most three consecutive packet losses can trigger a TO no matter what the congestion window size is. Since most of flows, besides fast flows, can have an arrival rate that is able to let more than two packets be assembled in one burst [4], we believe that Reno suffers from FTO in TCP over OBS networks more severely than the former expectation that only fast TCP flows are mostly like to encounter FTO. So we deem that there should be two causes of FTO for Reno in TCP over OBS networks. One is the randomness of burst losses, the other is the inefficiency of Reno in dealing with consecutive multiple packet losses. Compared with the work in [7] that manage to detect FTOs and the works in [8–10] that manage to reduce the randomness of burst losses, our work in this paper is to make Reno more efficient in dealing with consecutive multiple packet losses and thus improve its performance.

In this paper, we propose B-Reno (Burst Reno), a new TCP implementation designed specially for TCP over OBS networks. B-Reno is based on New-Reno, another TCP implementation whose performance over OBS networks has been investigated in [7,12]. New-Reno can achieve an even worse performance than that of Reno over OBS networks. This is because when the number of packets contained in a lost burst is relatively large the retransmission phases of New-Reno can be significantly prolonged, and during these periods only one lost packet can be recovered in each retransmission round and no new packet can be sent. Compared with New-Reno, B-Reno uses some aggressive schemes that can retransmit more than one lost packets and send more new packets in each retransmission round during the phase of fast recovery, and thus can achieve a better performance over OBS networks.

As proved in [7,12], SACK, another TCP implementation, can achieve a better performance over OBS networks than those of Reno and New-Reno. By adding the information about missing packets in a receiver's window to the ACKs so that a sender can retransmit multiple lost packets in one retransmission round, SACK can address the inefficiency of Reno and New-Reno in dealing with multiple packet losses. However, SACK adopts complex mechanisms not only at the sender's but also at the receiver's protocol stack, which means a higher difficulty in deployment. Compared with SACK, B-Reno only needs some simple modifications to New-Reno at the sender's protocol stack and does not need any special support from the receiver. We prove that over OBS networks B-Reno can achieve a performance similar with that of SACK.

The rest of the paper is organized as follows. In Section 2 we provide some background information about OBS networks and the three TCP implementations including Reno, New-Reno and SACK. Details about B-Reno are presented in Section 3. Then in Section 4 we present the results of comprehensive simulations, through which we compare the performance of B-Reno with the above mentioned three TCP implementations over OBS networks. Finally, in Section 5 we give out the conclusions of this paper.

2. Backgrounds

There are three main burst assembly algorithms in OBS networks including fixed assembly period (FAP), AAP and MBLMAP [6]. In this paper we consider the burst assembly

[1] We will use the terms packet and segment interchangeably in this paper as in [5].