Computer Communications 000 (2016) 1–9

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Adaptive low-priority congestion control for high bandwidth-delay product and wireless networks☆

Xianliang Jiang[a,b], Guang Jin[a,*]

[a] Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo, China
[b] College of Computer Science and Technology, Zhejiang University, Hangzhou, China

ARTICLE INFO

Article history:
Received 18 May 2016
Revised 7 October 2016
Accepted 18 November 2016
Available online xxx

Keywords:
High bandwidth-delay product
Wireless network
Congestion control
Low priority
Congestion level

ABSTRACT

The low-priority service is an exciting and attractive choice for networking applications (e.g. automatic update, backup, peer-to-peer file share) which create traffic that is considered less urgent than that of others and become a renewed interest at the Internet Engineering Task Force (IETF). A low-priority protocol, which provides the low-priority service, can exploit the residual bandwidth of the bottleneck link and achieve the high throughput, low-latency data delivery in traditional networks. However, due to the conservative and inappropriate congestion control mechanisms, the existing low-priority protocols (e.g. LEDBAT) cannot effectively utilize the residual bandwidth of the bottleneck link in high bandwidth-delay product (HBDP) and wireless networks. In this paper, we propose an adaptive Congestion level-based Low-Priority congestion Control (CLPC) protocol to improve the efficiency of low-priority protocols and maintain the low-priority features. Specifically, the CLPC sender adopts an one-way path delay to estimate the congestion level and adjust the aggressiveness of congestion control mechanisms. Different from other low-priority protocols, the CLPC protocol is more aggressive when the bottleneck link of HBDP and wireless networks has residual bandwidth. This makes a faster convergence of link utilization in HBDP networks. Combining the random loss detection, CLPC can achieve the high throughput in wireless networks. The extensive simulations in NS-2 show that CLPC can improve the transmission performance significantly as compared to other low-priority protocols in HBDP and wireless networks. Furthermore, we implement the CLPC protocol in the Linux kernel (3.13) and setup a testbed to measure it. The results also indicate the feasibility and effectiveness of CLPC.

1. Introduction

The Transmission Control Protocol (TCP), which adopts the Additive Increase Multiplicative Decrease (AIMD) mechanism [1], is an usual choice of the bulk data transfer over high-speed links. This may be driven by the requirements of the end-to-end reliable data delivery. Because of the built-in congestion control mechanisms, picking TCP as an end-to-end protocol means employing a given way of sharing network resources (buffers, bandwidth, $\cdots$) among competing flows. Under ideal conditions, long-lived TCP flows sharing a bottleneck link at a best-effort manner tend to obtain the fair share of bottleneck bandwidth. However, the fair share is not necessarily always the best objective for different applications. For example, TCP-based unattended applications, e.g. automatic update, may be less urgent than others and can tolerate longer flow-completion time than interactive web applications. In addition, a TCP-based bulk data transfer for unattended applications may saturate the bottleneck buffer and cause a large queueing delay that affect the performance of coexisting interactive applications.

To tackle aforementioned issues, the low-priority protocol (LPP), which achieves the low-priority service (LPS) by inferring and reacting to the occurrence of the network congestion on a network path earlier than the standard TCP, was proposed. Note that the bandwidth reservation can also ensure the performance of interactive applications [2]. In [3], Ros et al. regards the low-priority as a service which results in a smaller bandwidth and delay impact on the standard TCP than standard TCP itself when both low-priority and standard TCP flows share a bottleneck link. It also means that LPS allows latency-sensitive flows to occupy more available bandwidth when they need it. Actually, delay-based protocols, e.g. Ve-

---

☆ Manuscript received May xx, 2016. This work was supported in part by the National Natural Science Foundation of China (61601252); Natural Science Foundation of Zhejiang Province (LY12F02013) of China; Ningbo Municipal Technology Innovation Team (2011B81002) of China.

* Corresponding author.
E-mail addresses: jiangxianliang@nbu.edu.cn (X. Jiang), jinguang@nbu.edu.cn (G. Jin).

http://dx.doi.org/10.1016/j.comcom.2016.11.007
0140-3664/© 2016 Elsevier B.V. All rights reserved.

Please cite this article as: X. Jiang, G. Jin, Adaptive low-priority congestion control for high bandwidth-delay product and wireless networks, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.11.007

gas [4] and FAST [5], can detect and handle the incipient congestion earlier than loss-based protocols, e.g. Reno [1] and CUBIC [6]. Hence, delay-based protocols have the LPS characteristics when they coexist with loss-based protocols. In implementation, the LPS characteristics can be achieved through a mechanism that resembles TCP but exhibits a more cautious behavior, e.g. increasing the congestion window by less than one packet per round-trip time (RTT) in congestion avoidance. However, due to the conservative increasing and aggressive decreasing of the congestion window, LPP can not effectively utilize the residual bandwidth of the bottleneck link(s) in high bandwidth-delay product (high-BDP, named HBDP) and wireless networks. Therein, the BDP, which can be calculated by *bandwidth* × *delay*, means that the “network pipe” has a large capacity (e.g. in satellite network, wide area network). The HBDP indicates that TCP has a large congestion window. Furthermore, the wireless means that the complex transmission channel would cause the high packet loss rate. In short, the study of LPP is still a challenge in HBDP and wireless networks.

To understand why LPP performs poorly in HBDP and wireless networks, one should has insights into the operation of LPP. Note that LPP inherits main features of standard TCP while exhibits the low-priority characteristics. LPP regards the timeout or triple duplicate acknowledgements (ACKs) [3] events as an indication of packets loss and reduces the size of the sender's congestion window. This mechanism usually works well in traditional (not HBDP) networks. Unfortunately, due to the existence of the random packet loss caused by the high link error probability, fading, and interference *etc.*, existing LPPs may reduce the congestion window unnecessarily and result in a poor performance in wireless networks. Furthermore, existing LPP congestion avoidance algorithms adopt the similar AIMD mechanism like standard TCP. This makes LPPs cannot utilize the network bandwidth efficiently, especially in HBDP networks. The LPP sender increases the congestion window (*cwnd*) by one or less than the maximum segment size (MSS) every RTT when the bottleneck link has residual bandwidths, which is inferred from the one-way path delay. Otherwise, the LPP sender reduces *cwnd* by half when triple duplicate ACKs are received or reduces *cwnd* to one (2 or 4 probably [7]) when the retransmission timeout occurs. In HBDP networks, LPP requires a large window to fully utilize the network capacity. According to above analyses, one can see that LPPs need a long time to fully utilize the bottleneck bandwidth after that *cwnd* is reduced by half or to one.

In this paper, we propose an adaptive **C**ongestion level-based **L**ow-**P**riority congestion **C**ontrol (**CLPC**) protocol to resolve above issues in HBDP and wireless networks. The idea is inspired by LEDBAT [8], CLTCP [9] and TCP-FIT [10]. Therein, LEDBAT has been introduced into BitTorrent and standardized by the IETF recently. CLTCP is our previously proposed algorithm which uses the bit stream of explicit congestion notification (ECN) to measure the extent of network congestion and adaptively adjust the TCP congestion control mechanism. TCP-FIT uses *N* virtual Reno sessions, which can be adjusted according to the end-to-end delay, to simulate a single Reno session. Different from other state-of-the-art LPP schemes, CLPC can occupy the residual bottleneck bandwidth rapidly in HBDP and wireless networks and keep the LPS gracefully. We conduct extensive experiments in the packet-level simulator and results show that CLPC outperforms other LPP algorithms in HBDP and wireless networks. We also implement the CLPC protocol in end-hosts with Linux kernel version 3.13 and construct a testbed to verify the feasibility of CLPC.

The rest of the paper is organized as follows. In Section 2, the related work is reviewed. We describe the key idea and components of CLPC in Section 3. In Section 4, we evaluate the effectiveness of CLPC using packet-level simulations. And at last, the conclusion is concluded in Section 6.

2. Related work

A variety of congestion control algorithms, which are used in different TCPs for different purposes, have been proposed to achieve the high link utilization in HBDP and wireless networks. They have their own merits and shortcomings respectively. Generally, these protocols can be classified into two categories: *router-assisted* and *end-to-end*. Therein, the *router-assisted* congestion control algorithms, like XCP [11], RCP [12], BMCC [13], perform well in HBDP and wireless networks but confront the challenges of incremental deployment and backward compatibility with legacy TCPs. Accordingly, the *end-to-end* congestion control algorithms are more attractive since they do not require any special support from routers in the core networks. Actually, the *end-to-end* congestion control [14] relies on the packet loss or delay to detect the network congestion. And related works can be further divided into loss-based congestion control (LCC), delay-based congestion control (DCC), and the synergy of both LCC and DCC.

The LCC algorithms, which are widely adopted in the current Internet, perform the congestion control reactively by considering the packet loss. To improve the performance of LCC algorithms in HBDP networks, most TCP variants, e.g. HSTCP [15], CUBIC [6], Agile-SD [16], modify the additive increase and multiplicative decrease factors of the TCP's congestion control to achieve the high link utilization and throughput rapidly. However, the aggressiveness in adjusting *cwnd* intensifies the oscillation of the TCP throughput. Existing researches also indicated that most TCP variants cannot obtain the fair share of bottleneck bandwidth when competing with the standard TCP. How to achieve the high throughput and maintain the TCP-friendly is a dilemma. Wang et al. proposed a virtual parallel TCP [10] performing gracefully in both HBDP and wireless networks. To achieve near-optimal throughputs while preserving TCP-friendly and fairness, Mittal et al. [17] proposed a recursively cautious congestion control algorithm coupling the standard TCP with LPP.

The DCC algorithms, which are more efficient in the stable networks, assume that the increasing of packets' RTT indicates the coming of the network congestion and attempt to proactively adjust *cwnd* based on the variation of packets' RTT. In [23], the first DCC scheme was proposed in an interconnected and heterogeneous computer network. The author believes that the optimal *cwnd* is related to the gradient of the *delay-window* curve. Thereafter, Vegas [4], which detects the network congestion by observing the changes of the sending rate, was proposed to improve the performance of TCP Reno. To adapt to HBDP networks, FAST [5], which adopts the minimum RTT to detect the network congestion, was proposed to grab the network bandwidth rapidly. To solve the increasing queue backlog when the number of flows increases, Tan et al. proposed an enhanced FAST, which can achieve the (α, n)-proportional fairness [18,19], based on the virtual link price. In [20], Ge et al. analyzed the impacts of two-way FAST flows. Jung et al. proposed ACP [21], which combines the estimation of the bottleneck queue size and a measure of fair sharing, to achieve the high utilization, fair sharing of the bottleneck bandwidth, and fast convergence. To keep the low packet latency while delivering the bulk data, Mittal et al. [22] proposed to use RTT gradients to adjust the sending rate. Actually, the inaccurate RTT measurement would make DCC algorithms unstable. In addition, DCC algorithms also suffer from significantly low throughput if the competing flows are LCC ones, e.g. Reno.

Considering the advantages of the LCC and DCC algorithms, several researchers focused on the synthetical algorithms of both categories. In [24], Tan et al. proposed the compound TCP by adding a scalable delay-based component into the standard TCP. Xu et al. [25] regarded the queuing delay as the primary congestion indicator and the packet loss as the second congestion indicator,

Please cite this article as: X. Jiang, G. Jin, Adaptive low-priority congestion control for high bandwidth-delay product and wireless networks, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.11.007