Computer Communications 70 (2015) 54–67

Contents lists available at ScienceDirect

Computer Communications

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

Fast fairness convergence through fair rate estimation in Variable-structure congestion Control Protocol

Huixiang Zhang [a,*], Hairui Zhou [b], Chunlei Chen [a], Guanzhong Dai [a]

[a] School of Automation, Northwestern Polytechnical University, Xi'an 710072, China
[b] 28th Institute, China Electronics Technology Group Corporation, Nanjing 210007, China

ARTICLE INFO

Article history:
Received 28 April 2014
Revised 24 June 2015
Accepted 12 August 2015
Available online 18 August 2015

Keywords:
High-speed networks
Congestion control
Variable-structure congestion Control Protocol
Fair rate estimation
Fairness convergence

ABSTRACT

Traditional Transmission Control Protocol (TCP) faces significant limitations such as unclear congestion implication, low utilization in high-speed networks, unstable throughput and limited fairness. In order to overcome such limitations, numerous research works have been done in effective congestion control algorithms. Among these, Variable-structure congestion Control Protocol (VCP) leverages only the existing two explicit congestion notification bits for network congestion feedback, yet achieves high utilization, low persistent queue length, negligible packet loss rate and reasonable fairness. However, VCP converges to fairness relatively slowly due to its large multiplicative decrease parameter. To address this problem, we propose an end-host based method for fast fairness convergence, namely VCP-FFC (VCP with fast fairness convergence). In each Additive Increase and Multiplicative Decrease (AIMD) epoch, VCP-FFC estimates a fair rate in end-hosts, and adjusts the congestion window according to the fair rate to achieve fast fairness convergence. We evaluate the performance of VCP-FFC over a wide range of network scenarios using NS2. Simulation Results show that VCP-FFC not only maintains the good properties of VCP, but also significantly accelerates fairness convergence. Dynamic analysis shows that VCP-FFC is asymptotically stable and converges to a unique stationary point, i.e. the fair rate share.

1. Introduction

1.1. Congestion control with explicit feedback

The traditional Transmission Control Protocol (TCP) has exposed the following significant limitations when deployed in high-speed networks: (1) TCP relies on implicit feedback such as timeouts, duplicate acknowledgments, and round-trip measurements in end-hosts. But these implicit congestion signals may be affected by several factors, such as network congestion, link noise and equipment malfunctions [1]. (2) TCP employs an Additive Increase and Multiplicative Decrease scheme, which is too conservative when increasing the congestion window and overly aggressive when decreasing the window upon a congestion signal. As a result, TCP fails to fully utilize the network capacity in high-speed networks [2]. (3) TCP is unfair for the connections with long round-trip delays, because TCP throughput is inversely proportional to the round-trip time [3,4].

In order to overcome such limitations, an effective solution is to use the explicit feedback mechanism. The simplest form of explicit feedback is to deploy Active Queue Management (AQM) with Explicit Congestion Notification (ECN) [5] support. AQM can set a congestion experienced bit in the packet header instead of dropping the packet. TCP+AQM/ECN proposals reduce packet loss rate and queue sizes, but may become unstable when delay or link capacity increases [6].

Recent efforts in the research of congestion-control mainly involve explicit rate feedback. A typical example is eXplicit Congestion Control (XCP) [7]. XCP introduces a 20-byte congestion header. With this header, the sender can transmit its current congestion window size and the estimated round-trip time to the routers along the end-to-end path. The routers can estimate the fair rate for each flow and send it back to the sender using the congestion header. XCP achieves high utilization, negligible packet loss rate, low persistent queue length and fair bandwidth allocation. However, explicit schemes like XCP are hard to deploy in today's Internet due to the lack of extra multi-byte in the IP header.

To solve this problem, Xia proposed an explicit load factor based scheme (Variable-structure congestion Control Protocol, VCP) [8]. VCP only leverages the two existing explicit congestion notification bits for network congestion feedback, yet achieves comparable

* Corresponding author. Tel.: +8615929966410.
E-mail addresses: zhanghuixiang@nwpu.edu.cn, zhanghuixiang@gmail.com (H. Zhang), zhou_hairui@126.com (H. Zhou), thunder_flash@yahoo.cn (C. Chen), daigz@nwpu.edu.cn (G. Dai).

http://dx.doi.org/10.1016/j.comcom.2015.08.008
0140-3664/

performance to XCP. The main algorithms of VCP are developed in end-hosts, and the algorithms of VCP routers are much simpler than those of XCP. These make VCP a promising candidate for the next generation Internet.

1.2. Slow fairness convergence of VCP

A major limitation of VCP is that it converges significantly more slowly to a fair allocation than XCP. VCP utilizes Additive Increase and Multiplicative Decrease (AIMD) to provide fairness amongst the competing flows. Let $\beta(\beta < 1)$ denotes the Multiplicative Decrease (MD) parameter. When receiving a congestion signal, the sender performs the MD algorithm which reduces the congestion window with β. As we know, β controls how fast the AIMD-based protocol converges to fairness. In order to guarantee that the system always operates in the AIMD mode after entering the steady state, VCP sets a large MD parameter of 0.875 (i.e., $\beta = 0.875$). That is the main reason why VCP converges to fairness slow.

Recent research on explicit load factor feedback tries to decrease the MD parameter to improve the fairness convergence speed. MLCP [9] argues that using multiple levels of MD allows the protocol to adjust its convergence properties according to the dynamic behaviors of the network. MLCP adopts an 8-level MD scheme, and sets the value of β as a function of the load factor. When the load factor exceeds 120%, a small value of $\beta = 0.5$ is set to improve the convergence rate and responsiveness to congestion; when the load factor exceeds 100%, a high value of $\beta = 0.875$ is set to smooth rate variations.

Binary Marking Congestion Control (BMCC) protocol [10] argues that a high resolution estimate of the computed load factor is needed to achieve efficient and fair bandwidth allocation with high convergence speeds. BMCC employs a packet marking scheme called Adaptive Deterministic Packet Marking (ADPM) [11] to obtain congestion estimates with up to 16-bit resolution using the two existing explicit congestion notification bits. The MD parameter is set to $\beta_{max} = 0.875$ when the load factor exceeds 100%, and β is decreased linearly with the load factor until $\beta_{min} = 0.65$.

These methods either require additional bits in the IP header to encode the load factor (like MLCP, which uses 4-bit to encode multilevel load factor), or increase algorithmic complexity (like BMCC, which uses ADPM to obtain high resolution feedback information and thus causes more computation in both routers and end-hosts). Extra bits and computation cost make it difficult to deploy these methods in real networks. Moreover, end-hosts simply adjust the MD parameter according to the load factor, which cannot significantly accelerate the fairness convergence. The convergence time is also related to window adjustment policies. For example, MLCP decreases β from 0.875 to 0.5 based on MD levels when MD policy is applied, but MLCP uses Inversely-proportional Increase (II) policy in steady state. Thus, its fairness convergence is not always faster than VCP (as shown in Fig. 6).

1.3. Our proposal of fair rate estimation

Variable-structure congestion Control Protocol (VCP) has two typical features. Firstly, VCP controls the sending rate by manipulating a congestion window, and utilizes Additive Increase and Multiplicative Decrease (AIMD) to provide fairness amongst the competing flows. In a steady state, VCP's congestion window resembles a saw-tooth shape like TCP. For the senders, a saw-tooth cycle (or called AIMD epoch [8]) starts after a MD response and ends at the reception of the next event of overload (i.e. the response is again MD). Hence, a saw-tooth cycle includes exactly one multiplicative decrease phase and involves a number of steps through additive increase. For flows of the same round-trip time, VCP is able to estimate the fair rate in end-hosts by utilizing the concept of explicit fair-share calculation [12,13].

Secondly, VCP scales the congestion control parameters used by the end-hosts according to their round-trip times (RTTs). In this way, the fair rate allocation is ensured for flows with heterogeneous RTTs. Hence, if the fair rate can be estimated in end-hosts for flows with the same RTT, it is also feasible for the flows with heterogeneous RTTs. End-hosts can adjust their congestion windows directly according to the estimated fair rate, and accelerate the fairness convergence.

Based on the above idea, a new protocol called VCP-FFC is proposed. VCP-FFC tries to accelerate the fairness convergence by estimating the fair rate in end-hosts. VCP-FFC only uses two explicit congestion notification bits to encode the load factor and inherits the router algorithm of VCP, which means no extra bits are required in IP header and no complicated computation will be introduced in routers. Thus, VCP-FFC can converge quickly to fairness, and achieves high utilization, negligible packet loss rate, low persistent queue length and reasonable fairness.

For a better understanding of VCP-FFC, we analyze its dynamic behavior of fairness convergence. We will prove that the model asymptotically achieves global stability, which means there is a unique equilibrium rate in a VCP-FFC network and all senders' rate tends to reach the equilibrium rate asymptotically. In addition, we also show that the essence of VCP-FFC is equivalent to decreasing the MD parameter.

The remainder of the paper is organized as follows. In Section 2, we briefly discuss the VCP protocol and the causes for its slow fairness convergence. In Section 3, we provide a detailed description of VCP-FFC, and analyze its dynamic properties. In Section 4, we evaluate the performance of VCP-FFC through extensive simulations and compare it with VCP, XCP and MLCP, respectively. We provide an overview of related work in Section 5. Finally, we conclude the paper and present future works in Section 7.

2. Analysis of VCP protocol

2.1. Overview of VCP protocol

During every time interval t_ρ, VCP routers estimate a load factor ρ for each of its output links:

$$\rho = \frac{\lambda + \kappa\tilde{q}}{\gamma C t_\rho} \tag{1}$$

Because 75–90% of flows have round-trip time (RTT) less than 200 ms, VCP set $t_\rho = 200$ ms, where λ is the amount of input traffic during the period t_ρ; $\tilde{q}$ is the persistent queue length during the period t_ρ; κ controls how fast the persistent queue drains and is set to 0.5; γ is the target utilization and is set to 0.98 and C is the link capacity.

The level of congestion is classified into three regions based on the load factor ρ. If $0 \le \rho < 80\%$, it is defined as low-load region; if $80\% \le \rho < 100\%$, it is defined as high-load region; if $\rho \ge 100\%$, it is defined as overload region. The region is encoded into the 2-bit explicit congestion notification field in the IP header of each data packet. Then the encoded load region is sent back by the receiver to the sender via ACK packets.

Based on the encoded load region, the sender performs one of the following actions: Multiplicative Increase (MI) in the low-load region, Additive Increase (AI) in the high-load region, and Multiplicative Decrease (MD) in the overload region. By performing MI in the low-load region, VCP flows can exponentially ramp up their bandwidth to improve network utilization quickly. The MI response function is shown in Eq. (2), where $\varepsilon = 0.0625$ [8].

$$MI: cwnd(t + rtt) = cwnd(t) \times (1 + \varepsilon) \tag{2}$$

Once high utilization is attained, AIMD provides long-term fairness amongst the competing flows. The AI and MD response functions are