Journal of Network and Computer Applications 70 (2016) 183–193

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Towards fair and low latency next generation high speed networks: AFCD queuing

Lin Xue [a], Suman Kumar [b], Cheng Cui [a], Praveenkumar Kondikoppa [a], Chui-Hui Chiu [a], Seung-Jong Park [a,*]

[a] School of Electrical Engineering and Computer Science, Center for Computation and Technology, Louisiana State University, Baton Rouge, LA, USA
[b] Department of Computer Science, Troy University, Troy, AL, USA

ARTICLE INFO

Article history:
Received 28 May 2015
Received in revised form
25 March 2016
Accepted 31 March 2016
Available online 22 April 2016

Keywords:
Active queue management
Approximate fairness
Controlled delay
High speed networks

ABSTRACT

In packet switched high-speed networks, heterogeneous nature of TCP flows, a relatively newer characteristics of IP networks, and high burstiness have made it difficult to achieve low queuing delay and fair allocation of bandwidth among flows. Existing queue management (QM) schemes were designed to achieve either one or the other or both simultaneously and have been fairly successful at meeting either fairness or low queuing delay but not both at the same time. In this paper, unlike previous research efforts, the two requirements, fairness and low queuing delay are decoupled and addressed separately. We propose Approximated-Fair and Controlled-Delay (AFCD) queuing for next generation high speed networks that aims to meet following design goals: approximated fairness, controlled low queuing delay, high link utilization and simple implementation. The design of AFCD utilizes a novel synergistic approach by forming an alliance between approximated fair queuing and controlled delay queuing. It uses very small amount of state information in sending rate estimation of flows and makes drop decision based on a target delay of individual flow. Through experimental evaluation in a 10 Gbps high speed networking environment, we show AFCD meets our design goals by maintaining approximated fair share of bandwidth among flows and ensuring a controlled very low queuing delay with a comparable link utilization. AFCD is locally stable for small target delay in a high speed networking environment.

1. Introduction

To meet the demand of high speed networks, several TCP variants have been proposed. Extensive performance evaluations of these protocols convinced network users to adopt several of these seemingly promising proposals. Furthermore, open sourcing of these protocols make it flexible for user to choose their protocols. Therefore, current computer network is ruled by heterogeneous TCP variants (Yang et al., 2011) such as TCP-SACK, HighSpeed TCP (HSTCP) (Floyd et al., 2003), CUBIC TCP (CUBIC) (Ha et al., 2008), etc. TCP flows are dominant in Internet2 traffic (Internet2, 2011) accounting for around 85% flows while the rest being UDP flows with most of the network usages being long-lived bulk data transfer. One of the key design goals of the proposed TCP variants is fair bandwidth sharing with other competing TCP flows. However, recent study (Xue et al., 2012a) discloses severe inter-protocol unfairness among heterogeneous long-lived TCP flows in high speed networks where faster TCP flows consume most of the network bandwidth, whereas slow ones starve.

Packet delay is an equally key high speed network performance measure with others being throughput and fairness. Although high speed networks do not have Bufferbloat problem (Gettys and Nichols, 2012), queuing delay in high speed networks needs careful consideration. For example, in a typical setting of a 10 Gbps router (Cisco, Buffers, Queues, and Thresholds on the Catalyst 6500 Ethernet Modules, 2007) in high speed networks, the output buffer size of routers could be up to 89 MB, which may create large queuing delay in high speed networks. The large queuing delay may create bad user experience in live concert (Internet2, 2009), video streaming, etc., over high speed networks like Internet2. Also, popular applications, such as Online shopping, Voice over IP, HDTV, banking, and gaming, require not only high throughput but also low delay. In fact, importance of packet delay is growing with the emergence of a new class of high performance computing applications such as algorithmic trading and various data center applications with soft real time deadlines that demand extremely low end-to-end latency (ranging from microsecond to orders of milliseconds). In fact, packet queuing delay has been an critical issue in emerging data center networks (Alizadeh et al., 2012;

---

[*] Corresponding author.

E-mail addresses: xuelin@cct.lsu.edu (L. Xue), sumank@cct.lsu.edu (S. Kumar), ccui@cct.lsu.edu (C. Cui), pkondi1@tigers.lsu.edu (P. Kondikoppa), cchiu1@lsu.edu (C.-H. Chiu), sjpark@cct.lsu.edu (S.-J. Park).

http://dx.doi.org/10.1016/j.jnca.2016.03.021
1084-8045/

Perry et al., 2015) and distributed large-scale networks (Ousterhout et al., 2013). It is clear that the large latency may result in poor performance of the networks. Therefore, predictable and controllable queuing delay is highly desired in next generation high speed networks.

Performance of heterogeneous TCP flows depends on router parameters (Tang et al., 2010), but high bandwidth, high latency, and bursty nature of high speed networks make it a challenging task to design queue management (QM) schemes that ensure minimal latency while maintaining fair share of bandwidth among heterogeneous flows. There have been considerable efforts to address these challenges through various QM proposals (Shreedhar and Varghese, 1996; Stoica et al., 2003; Pan et al., 2003; Nichols and Jacobson, 2012). The QM scheme in Shreedhar and Varghese (1996) uses stateful information of the flows to classify the incoming packets, and puts the classified packets into different queues. Therefore, it is not suitable for large scale networks. The QM scheme in Stoica et al. (2003) is stateless in core layer, but it needs stateful information in edge layer. In Pan et al. (2003), the authors proposed a stateless active queue management (AQM) solution, and it achieves approximate fairness for heterogeneous flows. Although the fairness problem has been addressed in these QM schemes, controllable and predictable queuing delay has been overlooked. Recently proposed controlled delay (CoDel) AQM scheme (Nichols and Jacobson, 2012) focuses on providing extremely low queuing delay, and CoDel works well in a wide range of scenarios. While CoDel provides extremely low queuing delay, fairness issue is at bay. Only very recently that there is an attempt to bring fair queuing into CoDel and a variation of CoDel has been implemented in Linux kernel (Duzamet, 2012) that provides fairness by classifying different flows into different CoDel queues. However, the history of research on classification based QM suggests that any approach requiring huge amount of state information is not practical for large scale networks.

With above key observations, we are highly motivated to address fairness and queuing delay issues in high speed networks. In this paper, we propose an AQM scheme: Approximated-Fair and Controlled-Delay (AFCD) queuing that satisfies the requirement of high speed networks by providing approximated fairness and controllable low queuing delay. AFCD is a synergy of fair AQM design and controlled delay AQM design. The key idea of our novel AFCD queuing is to form an alliance between fairness based queuing and controlled delay based queuing to find out the dynamics between fairness and latency. Proposed AQM uses a relatively small amount of state information to provide approximated fairness while ensuring very low queuing delay for high speed networks. AFCD achieves comparable throughput as other popular AQM schemes as well. Extensive evaluation of AFCD shows that AFCD performs well in a 10 Gbps high speed networking testbed CRON (CRON, 2011). Through analysis, it is also shown to be locally stable in high speed networking environment for small target delay. The main contributions of this paper fall into following two aspects: (1) we proposed a new algorithm AFCD. AFCD calculates a global target delay and an individual target delay for each connection based on current network dynamics (e.g., flow rate estimation, queuing time, etc.). Therefore, AFCD controls the delay and fairness among long-lived flows. (2) AFCD achieves for the first time both fairness and low-latency while maintaining a high-throughput and small-state-information in high speed networks.

The rest of the paper is organized as following. In Section 2, we present background of this work and related works in this area of research. Section 3 is the design of AFCD. Section 4 presents evaluation of AFCD in a real high speed networking environment. Section 5 presents a local stability condition using fluid model equations for small target delay and paper is concluded in Section 6.

2. Background and related work

Drop-tail (DT) is the most used QM in commercial router nowadays. Packets are served in the order of first in first out. When a DT queue is full, packets are simply dropped at the tail. However, several studies have shown the limitations that imposed by DT. AQM scheme is suggested to eliminate those limitations. The authors in Iannaccone et al. (2001) compare AQM with DT and show that AQM has a minor impact on the aggregate performance metrics and AQM is sensitive to traffic characteristics that may compromise their operational deployment.

Random early detection (RED) (Floyd and Jacobson, 1993) is an AQM scheme. Packets are dropped early and randomly before the queue is full. RED has two thresholds: *min* threshold and *max* threshold. When the exponentially average queue size is smaller than the *min* threshold, no packet is dropped. When the exponentially average queue size is bigger than the *max* threshold, all packets are dropped. When the exponentially average queue size is between *min* threshold and *max* threshold, packets are marked/dropped based on a probability which is increased linearly with the queue size. Tuning of RED parameters had been very challenging for network operators since level of congestion and the parameter settings affect the average queue size and the throughput is also sensitive to the traffic load and to RED parameters. Various modifications have been suggested to improve RED, such as DRED (Aweya et al.) with multiple packet drop precedence to allow differentiating traffic based on priority, Gentle RED (Rosolen et al., 1999) with smooth dropping functions and many more. HERED (Abbasov and Korukolu, 2009) is proposed to replace the hazardous rate estimation of packet dropping function in RED. Adaptive RED (ARED) (Floyd et al., 2001) solves the parameter tuning problem by dynamically adjusting RED's maximum drop probability by observing the average queue length to decide whether to make packet drop more or less aggressive. The recently proposed the Proportional Integral controller Enhanced (PIE) AQM (Pan et al., 2013) randomly drops a packet on congestion. It is similar to ARED. However, like CoDel (Nichols and Jacobson, 2012) (CoDel is described later in this section), it uses queuing latency instead of queue length. It uses the dynamics of latency to determine the congestion level. Also, recent study (Martin et al., 2014) tries to address fairness issues in docsis-based cable networks, but the latency issue is not mentioned.

Besides, totally new AQM schemes are also proposed in different studies such as BLUE (Chang Feng) which uses packet loss and link idle events, instead of queue length, to manage congestion. Stochastic fair blue (SFB) (Feng et al., 2001) is an AQM for enforcing fairness among a large number of flows. SFB uses a bloom-filter to identify the non-responsive flows. The bloom-filter hashes the incoming packets to a hash value which stands for the flows. SFB maintains a mark/drop probability *pm* for each of the flows and a *qlen* which is the number of queued packets belonging to the flow. SFB has two thresholds: *max* is the maximum length of *qlen* and *target* is the desired length of *qlen*. If a flow's *qlen* is larger than *max*, packets are dropped. If a flow's *qlen* is smaller than *max*, packets are marked/dropped randomly with probability *pm*, meanwhile *pm* is adjusted to keep *qlen* between 0 and *target*. If *pm* of a flow reaches 1, the flow is identified as non-responsive flow, and therefore SFB enters rate-limit function to rate-limit the non-responsive flows.

Also, there have been a surge of congestion control algorithms that relies on single-router feedback. Explicit congestion control systems such as XCP (Katabi et al., 2002), EMKC (Zhang et al., 2004), VCP (Xia et al., 2005), RCP (Dukkipati and McKeown, 2006) may offer certain benefits over traditional models of additive packet loss. Still there are problems which cannot be alleviated even by these explicit congestion control mechanisms. In real time