Implementation and validation of BLUE and PI queue disciplines in ns-3

Vivek Jain, Viyom Mittal, Shravya K. S., Mohit P. Tahiliani

Wireless Information Networking Group (WiNG), Department of Computer Science and Engineering NITK Surathkal, Mangalore, Karnataka 575025, India

ABSTRACT

High queuing delay arising out of the bufferbloat problem has reignited research in the area of Active Queue Management (AQM). The Internet Engineering Task Force (IETF) has created a new working group to discuss the deployment feasibility of existing and upcoming AQM algorithms (or queuing disciplines) in the Internet. Network simulations are essential to gain an accurate and deep understanding of the network algorithms before they are deployed in the Internet. ns-3 is among the most widely used network simulators, and the recent addition of Linux-like traffic control subsystem in ns-3 makes it highly suitable and reliable for studying the performance of queuing disciplines. However, the current traffic control subsystem in ns-3 has very few queuing disciplines. In an effort to provide support for more queuing disciplines and foster active research in this area, we implement two popular algorithms in ns-3: BLUE and Proportional Integral controller (PI). This paper discusses the implementation and validation of the proposed models in ns-3, and presents a detailed comparative study of both queuing disciplines based on the evaluation guidelines provided in RFC 7928.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Queue management algorithms are expected to manage router buffers by dropping packets when necessary [1]. These algorithms are classified into: Passive Queue Management (PQM) and Active Queue Management (AQM) [2]. PQM algorithms (e.g., drop-tail), as the name suggests, do not take preventive actions to manage the buffer until it is full or has reached a predefined threshold. This passive approach leads to performance issues such as [1,3]:

- Lock-Out: occurs when a single flow or a group of flows occupy the entire buffer space and leave little or no room for other flows.
- Full queue problem: reduces the burst absorbing capability of the queues because PQM algorithms drop packets only when the queue is full, and hence tend to maintain high steady-state queue length.
- Bufferbloat: refers to the problem of having large unmanaged buffers in the network that lead to high queuing delays, and severely affect the performance of time sensitive applications.

AQM algorithms, on the contrary, proactively drop packets before the queue becomes full and take preventive measures to resolve the issues discussed above. However, the complexity of deploying AQM in the Internet has been a point

* Corresponding author.
E-mail addresses: viyommittal@gmail.com (V. Mittal), tahiliani@nitk.ac.in (M.P. Tahiliani).

https://doi.org/10.1016/j.simpat.2018.01.002
1569-190X/© 2018 Elsevier B.V. All rights reserved.
of discussion at IETF because majority of the AQM algorithms require some type of manual parameter settings. It has been demonstrated that inappropriately setting the parameters of AQM algorithms can lead to diminishing results [4]. Hence, there has been an active interest in the research community to design efficient and easy-to-deploy AQM algorithms, more so nowadays due to the severe impact of queuing delay on the performance of time sensitive applications.

Some of the promising AQM algorithms designed in the past are being revisited to minimize their deployment complexity and increase their robustness against modern Internet traffic. For example, Proportional Integral controller (PI) [5] has been extended by Proportional Integral controller Enhanced (PIE) [6], and subsequently, the implementation and functionality of PIE has been further simplified by PI2 [7]. Additionally, new algorithms are being designed by combining the advantages of multiple AQM algorithms: CoDel - BLUE Alternate (COBALT) [8] is a combination of Controlled Delay (CoDel) [9] and BLUE [10].

ns-3 [11] is a widely used open-source network simulator. It provides a Linux-like traffic control subsystem which can be used to study the behavioral aspects of AQM algorithms before they are considered to be safe for deployment in the Internet. However, only a few AQM algorithms namely, Random Early Detection (RED) [12], Adaptive RED [13], CoDel and PIE are currently available in ns-3. Looking into the ongoing developments and interests of the research community, we have implemented two AQM algorithms in ns-3: BLUE and PI. Both algorithms have been implemented as queuing disciplines within the traffic control subsystem of ns-3. In this paper, we present their implementation and validation details. In addition, both algorithms have been compared by using the evaluation guidelines proposed in the RFC 7928 [14]. We believe that the addition of BLUE and PI models in ns-3 would help the researchers to study every component of these algorithms thoroughly, and extend their functionality to meet the demands of modern Internet traffic. Additionally, these models can also be leveraged to gain initial insights about the working of upcoming AQM algorithms like COBALT.

The outline of this paper is as follows: Section 2 provides a brief background and classification of AQM algorithms, ns-3 and its traffic control subsystem followed by the details of related work. Section 3 describes the implementation of proposed models in ns-3. The correctness of the proposed models is validated in Section 4. Section 5 presents a detailed comparative study of BLUE and PI algorithms. Section 6 summarizes and concludes the paper with directions for future work.

2. Background and related work

2.1. Choice of BLUE and PI algorithms

AQM algorithms employ a congestion indicator to obtain a measure of the current queue state and accordingly drop the packets. Classification of AQM algorithms is often based on the congestion indicator(s) such as: queue length (e.g., RED, PI, etc), packet loss (e.g., BLUE, etc), queue delay (e.g., CoDel, PIE, etc) and many more. Algorithms that use the same congestion indicator differ from each other in a way they correlate packet drop events with the indicator, e.g., the packet drop probability in RED is a function of the average queue length, whereas it is a function of instantaneous queue length in PI.

The motivation to implement BLUE algorithm in ns-3 is based on the following factors: BLUE algorithm is targeted to perform better than queue length based algorithms [10], and a lot of studies have shown that it indeed outperforms RED, and other variants (e.g., Adaptive RED) [15]. However, to the best of our knowledge, there does not exist an inherent model of BLUE in any network simulator. Although there is an out-of-tree ns-2 code available to simulate BLUE, it has not been updated to work with the newer versions. We believe that having an inherent model of BLUE in ns-3 would serve as a base to implement more sophisticated algorithms like Stochastic Fair BLUE (SFB)\(^1\) and COBALT. Moreover, ns-3 traffic control subsystem lacks support of packet loss based AQM algorithms like BLUE.

We chose to implement PI algorithm in ns-3 because a lot of recent research has focused on leveraging its advantages to design new AQM algorithms [7,16]. ns-2 contains an in-built model of PI. However, the development of ns-2 has stopped, and its code has not kept pace with the recent C / C++ compilers. On the contrary, ns-3 is being rapidly developed and works well with most of the latest C / C++ compilers. Moreover, an AQM Evaluation Suite has been recently developed for ns-3 [17] that allows researchers to test the functionality of AQM algorithms based on the guidelines provided in RFC 7928. Thus, having a model for PI in ns-3 would allow researchers to study the performance of base AQM algorithms before studying more recent algorithms like PI2.

At last, we note that although the performance of BLUE has been evaluated in the past against RED and different variants of RED, it is not thoroughly evaluated against other queue length based algorithms like PI. As mentioned earlier, PI algorithm significantly differs from RED even though both are based on queue length. Hence, in this paper, we compare the performance of BLUE and PI based on the evaluation guidelines provided in RFC 7928. This evaluation also serves as a secondary means to validate the implementation of BLUE and PI models in ns-3. Comparing the performance of BLUE and PI with other AQM algorithms available in ns-3 would be further interesting. However, such a comparison is out of the scope of this paper.

---

1 SFB is a BLUE based queue discipline available in the Linux kernel.
دریافت فوری
متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات