

SABlue: 一种带加速因子的自适应 AQM 算法

陈伟杰^{①②} 王万良^{*①②} 蒋一波^① 郑建炜^①

^①(浙江工业大学计算机科学与技术学院 杭州 310023)

^②(浙江工业大学信息工程学院 杭州 310023)

摘要: 该文在分析 SBlue 算法的基础上,提出了一种带加速因子的自适应 AQM 算法 SABlue(Self-tune Accelerate Blue)。算法将瞬时队长作为早期拥塞检测参量,根据队列负载因子控制丢包步长,实现丢包概率幅度的自适应调整,最终将路由队列长度稳定在目标区域内。为了提高网络突变跨度较大情况时算法的响应速度,在队列警戒区域内引入了加速因子。实验表明, SABlue 面对突变流和不同 RTT 的网络场景,队列收敛时间短,丢包率小,且具有较好的鲁棒性,算法综合性能优于其他 AQM 算法。

关键词: 网络拥塞控制; 主动队列管理; Blue 算法; SABlue

中图分类号: TP393.07

文献标识码: A

文章编号: 1009-5896(2011)02-0479-05

DOI: 10.3724/SP.J.1146.2010.00210

SABlue: A Self-tune AQM Algorithm with Acceleration Factor

Chen Wei-jie^{①②} Wang Wan-liang^{①②} Jiang Yi-bo^① Zheng Jian-wei^①

^①(College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

^②(College of Information Engineering, Zhejiang University of Technology, Hangzhou 310023, China)

Abstract: In this paper, a self-tune AQM (Active Queue Management) algorithm with acceleration factor is presented by analyzing Blue algorithm and its variants, which is called SABlue (Self-tune Accelerate Blue). In order to make the queue length kept in the aim area, this algorithm adopt instantaneous queue length as the parameter of incipient congestion detection and calculate the step size of packet drop probability by using load factor. Furthermore, for the sake of response speed, the acceleration factor is led into alert area when the network traffic is changed suddenly. The experiments demonstrate that SABlue algorithm is more robust, carrying lower packet loss and shorter convergence time in the situation of dynamic traffic and RTT variation. The comprehensive performance of SABlue is more excellent than other AQM algorithms.

Key words: Network congestion control; Active Queue Management (AQM); Blue algorithm; SABlue

1 引言

拥塞控制机制在维护网络稳定和保障服务质量上起着至关重要的作用。自 1998 年 IETF 建议路由采用 AQM(Active Queue Management)并将 RED 提名为候选算法以来, AQM 成为了拥塞控制研究热点^[1]。 AQM 主要思想是路由实时采样网络拥塞信号感知网络拥塞程度,并通过特定的队列丢包策略,动态调整路由丢包概率;源端根据丢包/ECN (Explicit Congestion Notification)反馈信号动态调整发送速率,以缓解网络拥塞或提高链路利用率^[2]。

近年来,为解决 RED 参数配置复杂等难题,国

内外学者提出了各类 AQM 算法^[3]。文献[4]利用缓冲区溢出和链路空闲事件估计网络拥塞状况,提出了 Blue 算法,该算法使用相对小的缓冲区,保障了较小的端到端时延。文献[5]在 Blue 基础上,利用负载因子和队列控制函数,提出了 Yellow 算法。针对 Blue 算法在处理突变流时鲁棒性差,导致丢包率高或链路利用率低等缺陷,文献[6]提出一种基于平均队长反馈思想的丢包概率调整算法 SBlue;文献[7]给出了一种参数自适应的 Blue 改进方法,但上述改进的 Blue 算法都是将平均队长作为早期拥塞检测参量,面对网络的突变依然存在队列控制性能差、响应速度慢等缺点。

本文首先分析了 SBlue 存在的缺陷,在此基础上提出了一种带加速因子的自适应 Blue 算法 SABlue(Self-tune Accelerate Blue)。算法有两个特点:(1)将瞬时队长作为早期拥塞检测参量,根据负

2010-03-09 收到, 2010-07-28 改回

国家自然科学基金(60573123)和教育部高校博士点基金(20060337002)资助课题

*通信作者: 王万良 wwl@zjut.edu.cn

载因子自适应调整丢包概率步长,使队列稳定在目标区域;(2)针对网络突变跨度较大的情况,当队长严重偏离目标区域时,本文引入加速因子,来提高算法响应速度和稳定性。实验表明,与其他算法相比,SABlue面对各种动态网络场景,具有较好的鲁棒性。

2 SBlue 算法缺陷分析

SBlue是Blue变种算法中对队列控制能力相对较好的一种,具体算法参见文献[6]。该算法将平均队长作为早期拥塞检测参量,同时引入反馈思想将平均队长维持在特定的目标区域 $[\min_{th} \max_{th}]$ 内,改善了Blue算法在队列控制上的性能。

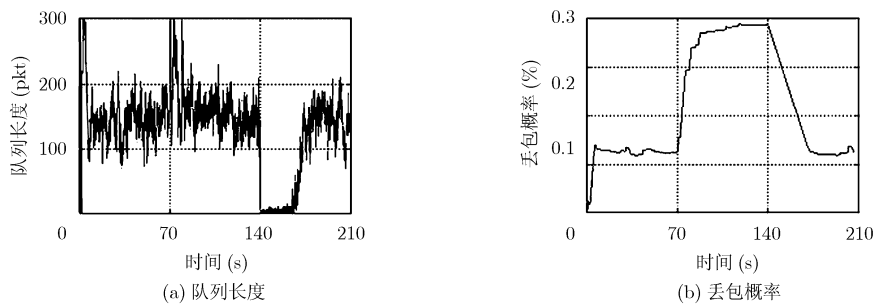


图1 突变流下的SBlue性能

3 SABlue 算法

3.1 SABlue 原理

本文在分析SBlue缺陷的基础上,提出了一种带加速因子的自适应Blue算法SABlue。该算法采用瞬时队长 $qlen$ 作为丢包概率步长调整的依据,依据 $qlen$ 与目标区域偏离程度,赋予相应的调整步长,并根据路由拥塞程度将缓冲分为5个区域(如图2)。

算法的目标是要将队长稳定在区域III内。当队长偏离目标区域较小时,通过微调丢包概率来调节队长,如网络轻载区域II和网络重载区域IV;当队长严重偏离目标区域时,如缓冲空闲警戒区域I和缓冲溢出警戒区域V,本文引入加速因子 α 来加大

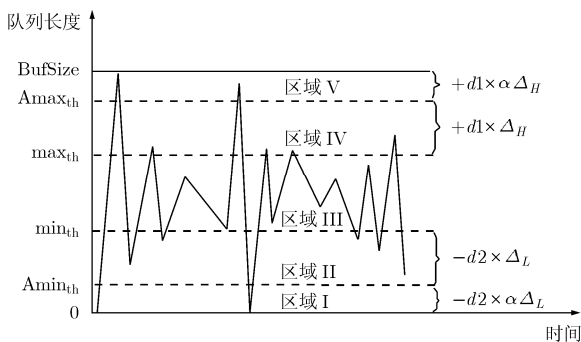


图2 负载分段区域

下面将通过一组实验来论述SBlue面对突变流时存在的缺陷。实验基于NS2^[8]网络仿真平台,仿真拓扑和SBlue参数配置见第4节。SBlue算法链路利用率为98.1771%,丢包率为0.3768%。从图1(a)可得,在大部分时间,队长都能稳定在 $[\min_{th} \max_{th}]$ 区域。但从图1(b)可得,SBlue在应对突减流问题上,虽然丢包率能达到稳态值,但历时较长。这是由于采用指数加权滑动平均方法计算得到的平均队长引入了大惯性环节,使得算法面对突变流响应速度慢。同时,平均队长并不能反映队列稳定情况,特别是当瞬时队长周期性震荡时,平均队长却是稳定值。综合上述,平均队长并不适用来衡量链路拥塞程度。

丢包概率步长调整,使队长尽快重新进入并稳定在目标区域。 $Amin_{th}$ 和 $Amax_{th}$ 分别为 0.1 BufSize 和 0.9 BufSize 。SABlue根据上述5个区域特点给出了相应的丢包策略函数,描述如下:

$$P_{new} = P_{old} \pm \phi \quad (1)$$

$$\Delta_L = \frac{1}{1 - \frac{\min_{th} - qlen}{BufSize}} \quad (2)$$

$$\Delta_H = 1 + \frac{qlen - \max_{th}}{BufSize} \quad (3)$$

式(1)为丢包概率调整函数,其中 ϕ 为调整步长,当 $0 \leq qlen \leq \min_{th}$ 时取负号;当 $\max_{th} \leq qlen \leq BufSize$ 时取正号。式(2)和式(3)为本文设计的两个负载因子 Δ_L 和 Δ_H ,分别表征 $qlen$ 偏离 \min_{th} 和 \max_{th} 程度的量,用来度量网络拥塞程度。式(4)给出了各区域 ϕ 的调整方法,可知 ϕ 的取值不仅取决于 $d1$ 和 $d2$,而且与负载因子 Δ_L 和 Δ_H 以及加速因子 α 有关。结合图2分析式(4),当 $qlen \leq \min_{th}$ 时,丢包概率 P 表现出激进,容易导致链路空闲,SABlue通过负载因子 Δ_L 计算丢包调整步长 ϕ ,来控制丢包概率 P 减小的幅度, Δ_L 越大则 ϕ 越大,丢包概率 P 减小的幅度越大;同样,当 $qlen \geq \max_{th}$ 时,丢包概率 P 表现出保守,容易导致缓冲溢出,SABlue通过负载因子 Δ_H 控制丢包概率 P 增大的幅度。

$$\phi = \begin{cases} d2 \times \alpha \Delta_L, & 0 \leq qlen \leq A \min_{th} \\ d2 \times \Delta_L, & A \min_{th} < qlen \leq \min_{th} \\ 0, & \min_{th} < qlen < \max_{th} \\ d1 \times \Delta_H, & \max_{th} \leq qlen < A \max_{th} \\ d1 \times \alpha \Delta_H, & A \max_{th} \leq qlen \leq BufSize \end{cases} \quad (4)$$

当调整步长 ϕ 为 $d1 \times \Delta_H$ 和 $d2 \times \Delta_L$ 时, 能适应网络变化较小的情况; 但当网络突变跨度较大时, $qlen$ 远远偏离目标区域, 仅依靠负载因子 Δ_L 和 Δ_H 已不能满足快速响应的需求。故算法在区域 I 和区域 V 内引入加速因子 α 来加大 ϕ 。 α 的取值将影响算法的性能, 本文通过大量的实验, 研究了参数 α 的配置, 最后得出当 $\alpha \in [0.14 \ 0.37]$ 范围时, 算法综合性能最优。

3.2 SABlue 算法实现及分析

SABlue 通过负载因子 Δ_L 和 Δ_H 以及加速因子 α , 计算丢包调整步长 ϕ , 来控制丢包概率 P 调整幅度, 使队长快速稳定在区域 III 内, SABlue 算法继承了 Blue 算法的架构, 伪代码如下(其中, $last_time$ 为上次更新的时间, f_{time} 为更新间隔时间; $d1$ 和 $d2$ 决定了当 $qlen$ 偏离目标区域, P 增加/减小的基准量):

Upon packet arriving event:

If ($now - last_time$) > f_{time}

If ($qlen \geq \max_{th}$)

$$\Delta_H = 1 + \frac{qlen - \max_{th}}{BufSize}$$

$$\Delta_H = \left(\Delta_H \geq 1 + \frac{A \max_{th} - \max_{th}}{BufSize} \right) ? \alpha \Delta_H : \Delta_H$$

$$P = P + d1 \times \Delta_H$$

Else If ($qlen \leq \min_{th}$)

$$\Delta_L = \frac{1}{1 - \frac{\min_{th} - qlen}{BufSize}}$$

$$\Delta_L = \left(\Delta_L \geq \frac{1}{1 - \frac{\min_{th} - A \min_{th}}{BufSize}} \right) ? \alpha \Delta_L : \Delta_L$$

$$P = P - d2 \times \Delta_L$$

End If

$last_time = now$

End If

重复上小节实验, SABlue 参数配置见第 4 节。算法链路利用率为 98.9136%, 算法丢包率为 0.2512%。由图 3 可知, 算法响应网络突发能力都得到了提升。

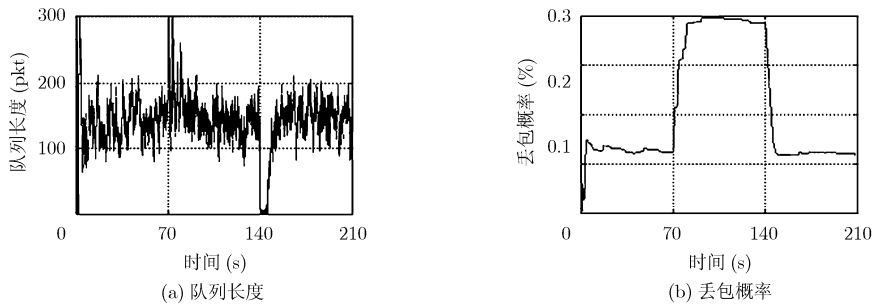


图 3 突变流下的 SABlue 性能

4 仿真与分析

为了进一步验证 SABlue 算法的有效性和可行性, 本文在 NS2^[8]仿真平台上进行了一系列的对比仿真实验。AQM 算法参数配置如下: ARED^[9]的上下限队列阈值分别为 70 pkt 和 210 pkt, 自适应更新时间 0.1 s; PI^[10]的目标队长 $Q_{ref} = 140$; SBlue^[6]的上下限队列阈值分别为 100 pkt 和 200 pkt, 更新时间 $f_{time} = 0.01$ s; SABlue 的加速因子 $\alpha = 2$, 其余参数与 SBlue 一样。仿真拓扑如图 4, 节点 $n1$ 和 $n2$ 间存在一条瓶颈链路, 缓冲大小为 300 pkt, 链路配置详情见图 4 说明。瓶颈链路采用 AQM 算法, TCP 设置为允许 ECN 模式, 分组大小为 500 B。

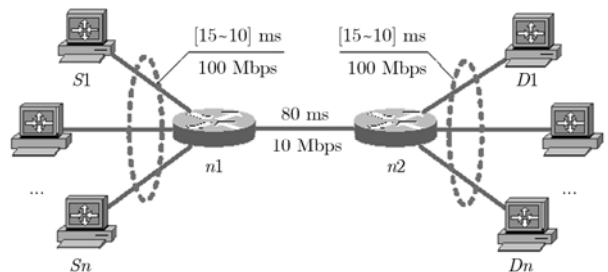


图 4 仿真拓扑图

4.1 不同 TCP 连接数下的算法性能比较

在实际网络环境中, 每一时刻的数据流数 N 都可能在变化, 因此一个优秀的 AQM 算法必须能很好处理 N 不断变换的情况。本组实验目的是验证

SABlue 在不同 TCP 连接数 N 下的性能。实验分别对 $N = 100 \times i (i = 1, 2, \dots, 8)$ 下的网络场景进行重复仿真。

图 5 描述了 N 变化情况下, 各算法的平均队长、队列标准偏差、链路利用率与丢包率性能。ARED 在 N 较小时有较好的队列控制能力, 但随着 N 的增大, 特别是当 $N > 300$ 时平均队长迅猛增长; PI 在队列控制性能和链路利用率上表现的很优秀, 随 N 的增加平均队长呈减少趋势以及偏差保持在较低值; SABlue 随着 N 的增大, 队列控制性能较好, 略逊于 PI 算法, 但比 SBlue 和 ARED 要好。从图 5(c), 5(d) 可知, SABlue 算法在链路利用率上略低于 PI 算法, 但丢包率远远小于 PI 算法和 ARED 算法。由上分析可得, ARED 面对 N 变化情况, 算法不稳定; PI 算法在 N 变化时, 性能稳定但丢包率大; SBlue 和 SABlue 在丢包率性能上同样优秀, 但 SABlue 算法的链路利用率要高于 SBlue, 而且平均队列长度和队长稳定性也要优于 SBlue。所以, 本文提出的 SABlue 算法在面对网络连接数 N 变化时, 总体性能最优。

4.2 不同 RTT 时延下的算法性能比较

RTT (Round-Trip Time) 是最重要网络参数之一, 理论和实践证明此参数变化将大大影响 AQM 算法性能。通过下面的实验仿真研究了 SABlue 面对不同 RTT 情况下的性能。实验分别对 RTT =

$[50 \times i] \text{ ms } (i = 1, 2, \dots, 7), N = 300$ 的网络场景下进行重复的仿真。

由图 6(c), 6(d) 可知 4 种算法链路利用率和丢包率都随着 RTT 增加而明显的下降, 由于每个反馈丢包/ECN 信号控制量与 RTT 成正比, 因此当 RTT 较大时容易导致 AQM 算法不稳定, 导致链路利用率下降。图 6(a), 6(b) 显示 SABlue 和 PI 算法平均队长和标准偏差受 RTT 的变化影响不大, ARED 受 RTT 队列变化最大, 可见 ARED 敏感于 RTT。虽然 SABlue 与 SBlue 链路利用率和丢包率变化相近, 但是从队列管理性能上讲 SABlue 优于 SBlue。实验可得, SABlue 面对 RTT 的变化有较好的鲁棒性。

5 结束语

自适应 AQM 算法是网络拥塞控制领域一个热门话题。为了克服多数 AQM 算法面对突变流响应慢的缺陷, 本文基于 Blue 算法提出了一种带加速因子的自适应 AQM 算法(SABlue), 算法将瞬时队列长度作为早期拥塞检测参量, 通过负载因子和加速因子控制丢包概率调整的幅度。通过实验仿真和分析, SABlue 能够适应变化的网络场景, 与 ARED, PI, SBlue 算法相比, 具有更快的瞬时响应速度和更好的队列控制性能。本文的后续工作是研究算法在处理混合数据流时, 特别是非响应流, 算法的鲁棒性、稳定性等问题。

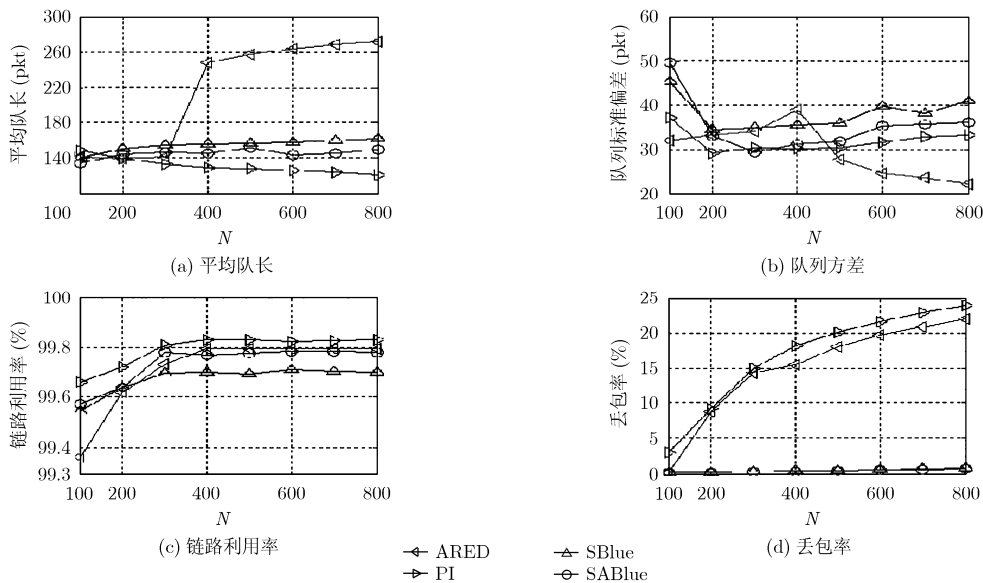


图 5 不同 TCP 流数下的算法性能比较

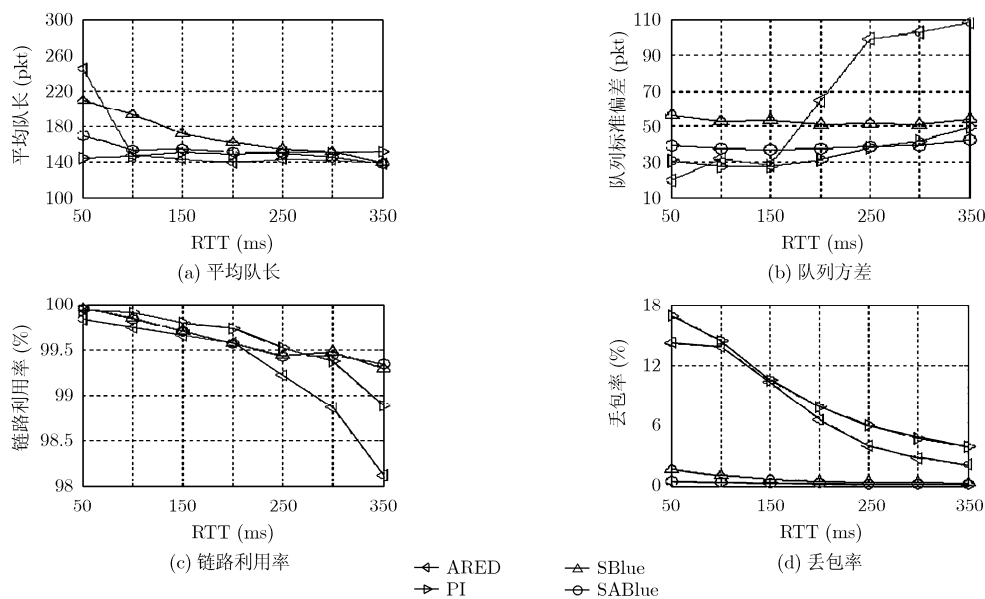


图6 不同 RTT 下的算法性能比较

参考文献

- [1] Zhang W, Tan L, and Peng G. Dynamic queue level control of TCP/RED systems in AQM routers[J]. *Computers & Electrical Engineering*, 2009, 35(1): 59-70.
 - [2] 王万良, 蒋一波. 网络控制与调度方法及其应用[M]. 北京: 科学出版社, 2009: 15-21.
Wang W and Jiang Y. Network Control and Scheduling[M]. Beijing: Science Press, 2009: 15-21.
 - [3] Thiruchelvi G and Raja J. A survey on active queue management mechanisms[J]. *International Journal of Computer Science and Network Security*, 2008, 8(12): 130-145.
 - [4] Feng W, Shin K G, and Kandlur D D, et al. The blue active queue management algorithms[J]. *IEEE/ACM Transactions on Networking*, 2002, 10(4): 513-528.
 - [5] Long C, Zhao B, and Guan X, et al. The yellow active queue management algorithm[J]. *Computer Networks*, 2005, 47(4): 525-550.
 - [6] 吴春明, 姜明. SBlue: 一种增强 Blue 稳定性的主动式队列管理算法[J]. *通信学报*, 2005, 26(3): 68-74.
Wu C and Jiang M. SBlue: stabilized Blue[J]. *Journal on Communications*, 2005, 26(3): 68-74.
 - [7] 刘伟彦, 孙雁飞, 张顺颐等. 一种参数自适应的主动队列管理算法—自适应 BLUE[J]. *电子与信息学报*, 2009, 31(2): 442-446.
Liu W, Sun Y, and Zhang S, et al. An adaptive active queue management algorithm self-adaptive BLUE[J]. *Journal of Electronics & Information Technology*, 2009, 31(2): 442-446.
 - [8] The Network Simulator(NS2). <http://www.isi.edu/nsnam/ns/>, 2009.
 - [9] Floyd S. Adaptive RED: an algorithm for increasing the robustness of RED active queue management. <http://www.icir.org/floyd/papers/adaptiveRed.pdf>, 2001.
 - [10] Hollot C V, Misra V, and Towsley D, et al. Analysis and design of controllers for AQM routers supporting TCP flows[J]. *IEEE Transaction on Automatic Control*, 2002, 47(6): 945-959.
- 陈伟杰: 男, 1985年生, 博士生, 研究方向为网络优化、拥塞控制、Qos路由。
王万良: 男, 1957年生, 博士生导师, 教授, 研究方向为计算机网络控制及其调度。
蒋一波: 男, 1982年生, 讲师, 研究方向为网络优化、智能计算。