

一种参数自适应的主动队列管理算法—自适应BLUE

刘伟彦^{①②} 孙雁飞^① 张顺颐^① 刘斌^③

^①(南京邮电大学网络技术研究中心 南京 210003)

^②(江苏广播电视大学信息工程系 南京 210017)

^③(江苏广播电视大学传媒艺术系 南京 210036)

摘要: BLUE算法是一种典型的主动队列管理 (Active Queue Management, AQM) 算法, 研究表明BLUE算法优于RED算法。BLUE算法使用丢包事件和链路空闲事件控制网络拥塞。但由于BLUE算法在参数设置方面存在不足, 尤其是当TCP连接数突然剧烈变动时, 容易导致队列溢出或空闲的频繁发生。该文引进参数自适应机制, 提出了自适应BLUE算法, 并借鉴了RED算法的早期拥塞检测机制。NS仿真实验表明该算法能有效保持队列长度的稳定, 减少队列溢出或空闲现象的发生, 在提高链路利用率的同时降低丢包率。

关键词: 拥塞控制; 自适应; 主动队列管理; BLUE算法

中图分类号: TP393.07

文献标识码: A

文章编号: 1009-5896(2009)02-0462-05

An Adaptive Active Queue Management Algorithm—Self-Adaptive BLUE

Liu Wei-yan^{①②} Sun Yan-fei^① Zhang Shun-yi^① Liu Bin^③

^①(Institute of Information Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

^②(Department of Information Engineering, Jiangsu Radio and TV University, Nanjing 210017, China)

^③(Department of Media and Art, Jiangsu Radio and TV University, Nanjing 210036, China)

Abstract: As a classical Active Queue Management (AQM) algorithm, compared with RED (Random Early Detection), BLUE has many advantages. BLUE uses packet loss and link idle events to manage congestion. However, there are still some insufficiencies in parameter setting for BLUE. Especially when TCP connections changed dramatically will lead to queue overflow and underflow. Based on the study of BLUE, a novel self-adaptive BLUE is proposed. NS simulation results show that the algorithm can effectively stabilize the queue occupation independent of the number of active TCP connections and mitigate the queue overflow and underflow, it can improve link utilization and decrease packet loss rate at the same time.

Key words: Congestion control; Self-adaptive; Active Queue Management (AQM); BLUE algorithm

1 引言

网络的稳定性、可靠性对Internet是非常重要的。当一个数据包在到达目的地之前被丢弃, 那么它在传输过程中所消耗的网络资源全都浪费了, 严重的情况下还会导致拥塞崩溃^[1,2]。

目前, Internet 中普遍采用的是基于窗口的端到端的拥塞控制^[3], 路由器采用先来先服务 FIFS(First In First Serve, FIFS)的调度算法和尾部丢弃(drop-tail)策略处理网络拥塞, 这种端到端的拥塞控制机制对网络的稳定起到了关键作用。

但是随着网络的发展, 各种音频、视频多媒体应用的不断丰富及无线接入的普及, 逐渐暴露出 TCP 拥塞控制算法的不足^[4]。这种基于窗口的端到端的拥塞控制机制不提供任

何服务质量保证, 只是力求最好地传输数据, 因此会导致较高的传输延时和分组丢失率。drop-tail 队列管理机制在维持高的带宽利用率和低的队列延时间存在着矛盾, 为了解决这一矛盾, 人们提出了主动队列管理机制(Active Queue Management, AQM)。

AQM是IETF推出的基于FIFO调度策略的队列管理机制, 它使得路由器能够控制在什么时候丢多少包, 从而有效地管理队列长度。

随机早期检测RED(Random Early Detection)算法是最早提出的AQM算法, 基本思想是路由器通过监控队列的平均长度来探测拥塞, 一旦发现拥塞逼近, 就随机地选择源端来通知拥塞, 使源端在队列溢出之前减小拥塞窗口, 降低发送数据速度, 从而缓解网络拥塞。

但由于RED算法的性能敏感于设计参数和网络状况, 在特定的网络状况下会导致多个TCP的同步, 造成队列震荡、吞吐量降低和时延抖动加剧^[5,6]。自RED算法被首次提出后

2007-07-31 收到, 2008-06-02 改回

国家 863 计划项目(2003AA121560), 江苏省自然科学基金(BK2007603)和 2007 年江苏省高等学校大学生实践创新项目资助课题

它的参数设置问题一直没有得到彻底解决,因此RED算法没有在Internet中得到广泛的应用。

基于上述原因, Feng等人提出了一种新的主动队列管理算法BLUE^[5]。

2 BLUE算法

2.1 BLUE算法的基本原理

BLUE算法使用丢包事件和链路空闲事件来管理拥塞,算法维持一个丢包概率 P_d 。由于队列溢出而导致连续丢包时,算法就增加 P_d ; 相反,如果由于链路空闲而出现空队列时,算法就减小 P_d 。因此BLUE算法能够有效地控制发送拥塞通知信息的速度。BLUE算法实现如下:

```

Upon packet loss(or  $Q_{len} > L$ ) event:
if( $(now-last\_update) > freeze\_time$ )
{ $P_d = P_d + d_{in}$ 
last\_update = now}
Upon link idle event:
if( $(now-last\_update) > freeze\_time$ )
{ $P_d = P_d - d_{de}$ 
last\_update = now}

```

其中 last_update 为 P_d 最新更新时间; freeze_time 为连续两次更新 P_d 的最小时间间隔, freeze_time 的设定有效避免了由于 P_d 的改变过于频繁而引起队列的剧烈波动。 d_{in} 为队列溢出时 P_d 的增量; d_{de} 为队列空闲时 P_d 的减量,通常 d_{in} 比 d_{de} 大。这样赋予丢包事件更大的比重,使BLUE算法能够对流量的迅速增加很快做出反应。

与RED算法相比, BLUE算法用相对较小的缓冲区就能够完成拥塞控制。

2.2 BLUE算法存在的问题

BLUE算法依据丢包事件和链路空闲事件进行拥塞控制,当TCP流量相对稳定时BLUE算法能很好地控制队列长度^[7,8]。但当TCP流量发生显著变化时算法不能及时调整 P_d ,导致队列长度的增减都非常迅速^[7,9,10],可以从下面的对比仿真实验中看出。

仿真实验采用Berkeley开发的网络仿真软件NS-2^[11]进行,实验使用的网络拓扑如图1所示。

网络有5个源子网、5个目的子网和两台路由器 R_1, R_2 组成。 R_1, R_2 间形成一条瓶颈链路,实验中TCP流输入模型为FTP模型。参数设置如下:初始丢包率 $P_m = 0.05$, freeze_time = 50ms, $d_{in} = 0.04$, $d_{de} = 0.004$, 缓冲区 $B = 300$ Packets, 实验时间 $t = 100$ s。

首先考虑稳定环境下TCP连接数不多时BLUE算法的性能。图2是BLUE算法在FTP连接数为200时平均队列长度和丢包概率的变化情况。从图2可以看出, TCP连接数相对稳定时空队列或队列溢出现象很少出现,丢包概率也基本平稳, BLUE算法性能良好。

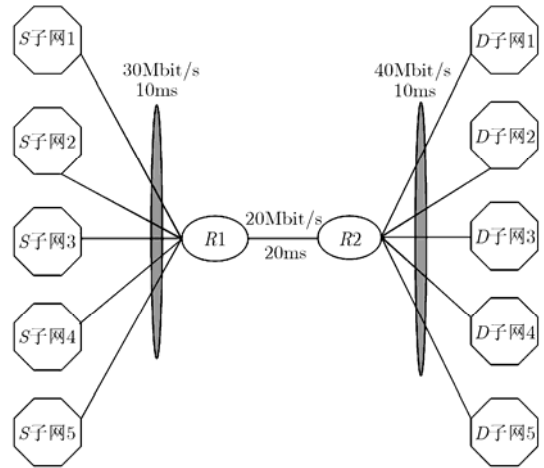


图1 网络拓扑结构

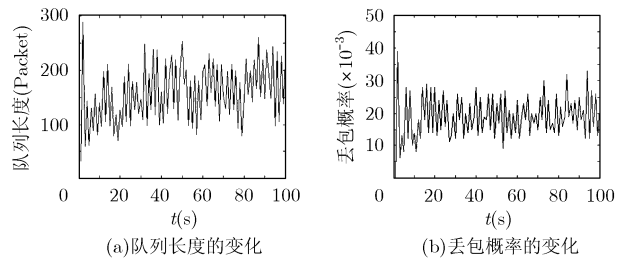


图2 TCP连接数稳定

图3是相同数目的TCP连接数,只在30-60s之间连接数减少一半的平均队列长度、丢包概率的变化。

从图3可知,当TCP连接数变化时队列空闲和溢出的出现较多,且两种现象交替出现,队列长度震荡幅度较大。原因是由BLUE参数设置缺乏引自适应性,不能在流量变化的早期做出反应,当队列溢出时, BLUE算法才加大丢包概率,连续丢包而导致 TCP陷入超时,严重时还会降低链路利用率。

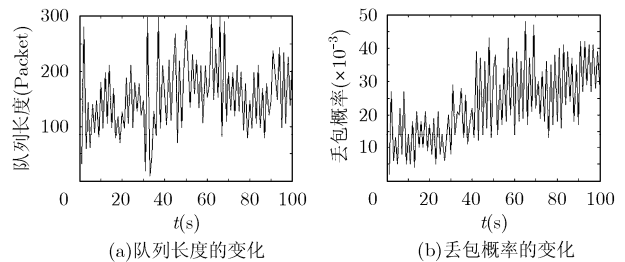


图3 TCP连接数变化时的情况

由实验知, BLUE算法在连接数剧烈变动时队列增减迅速,导致队列空闲和溢出频繁发生,主要有以下两方面的原因。

一方面,由BLUE算法调丢包率的特点所致。据文献^[5],参数 d_{in} 通常比 d_{de} 高一个数量级,因此算法对队列溢出采取的措施要比对空队列严厉的多,当网络中连接数量突然大幅

度减少时,它很难及时大幅度地减少丢包率以避免空队列的持续出现。另一方面,由于BLUE算法调整拥塞通知是通过常量 d_m , d_{de} 完成的,而根据文献[2]Floyd的研究, P_d 和连接数量 N 之间存在非线性关系,因此当TCP连接数 N 很大时, P_d 的改变量也需要相应加大才行。

以上分析说明BLUE算法通过常量 d_m , d_{de} 调节 P_d 是不合理的,这也正是BLUE算法有待改进的地方。

3 基于参数自适应的BLUE算法

本文的自适应BLUE算法针对BLUE算法在参数设置方面存在的不足提出的改进算法。

3.1 算法基本原理

算法的基本思想是引进参数自适应机制,目标是保持队列长度的稳定性,避免队列溢出和空闲现象的频繁发生,同时提供一定的公平保障。自适应BLUE算法除了利用丢包事件和链路空闲事件控制拥塞外,还通过指数加权平均算法计算平均队列长度,根据平均队列长度的变化检测早期的拥塞。因此该算法能有效地屏蔽由突发流量产生的短时间队列长度的增减而产生的抖动,在一定程度上消除了一般AQM算法对突发性流的歧视。

3.2 算法结构描述

算法设定了两个控制阈值 Min_th , Max_th ,根据阈值把网络划分为3种状态。当队列长度的变化率超过一定的控制阈值才进行较大的参数调整,避免了对正常队列的波动进行不必要的调整。算法有效解决了现实网络中固有的由于用户需求的突发性和不确定性引起的频繁波动。

当链路空闲且队列长度小于 Min_th 时,网络处于轻载区,丢包概率 $P_d = 0$ 。

当链路空闲且队列长度大于 Max_th 时,

$$P_d = P_d + \beta d_{de}, \quad \beta > 1 \quad (1)$$

β 为链路空闲时 P_d 的调接系数。尽管队列长度大,但链路空闲,网络仍处于轻载区,以小概率丢包,在一定程度上消除了对突发性流的歧视。

当队列溢出丢包且队列长度大于 Max_th 时,

$$P_d = P_d + \alpha \left(\frac{avg}{Max_th} \right)^2 d_m, \quad \alpha > 1 \quad (2)$$

α 为队列溢出时 P_d 的调接系数, $\alpha > \beta$ 。网络处于重载区,迅速增大 P_d ,及时缓解网络拥塞。

当队列溢出丢包且队列长度在 Min_th , Max_th 之间时,

$$P_d = P_d + \alpha d_m, \quad \alpha > 1 \quad (3)$$

网络状态稳定,队列长度适中,采取较为缓和的丢包策略。

当队列溢出丢包且队列长度小于 Min_th 时,

$$P_d = P_d - \alpha d_m, \quad \alpha > 1 \quad (4)$$

网络状态稳定,队列长度偏小,减小丢包率,充分利用网络

资源,减少不必要的丢包。本文算法自适应地调节丢包概率防止空队列或队列溢出的出现,有效避免了对正常的队列波动调节而加剧队列的波动。

3.3 算法实现

算法借鉴了RED算法的早期拥塞检测机制,根据平均队列长度的变化自适应地调节 P_d ,只有当平均队列长度变化率达到一定程度时才进行比较大的参数调整。算法实现如下:

```

For packet arriving
Calculate the average queue size avg
avg = (1 - wq) × avg + wq × q
Upon packet loss event and avg > Max_th
if ((now - last_update) > freeze_time)
Pd = Pd + α ( avg / Max_th )2 dm    (α > 1)
last_update = now

```

其中 avg 为平均队列长度; q 为当前队列长度; w_q 为当前队列长度的加权系数,反应了 avg 对 q 的敏感程度,一般取值范围为 $0 < w_q < 1$ 。

4 算法仿真及性能分析

为了检验本文算法的有效性,进行了仿真实验。为了与前面的BLUE算法比较,实验平台及网络拓扑、TCP流的输入模型,参数设置等都与前面2.2节相同。

4.1 TCP连接数量稳定时队列长度与丢包率的变化

算法参数设置: $Min_th = 80$, $Max_th = 240$, $w_q = 0.002$, $\alpha = 3$, $\beta = 2$,缓冲区 $B = 500$ packet, $t = 100$ s,其余参数设置同BLUE算法。本文实验均采用了显示拥塞通告(Explicit Congestion Notification, ECN)机制。

实验1 TCP连接数量(TCP连接数=200)稳定时的队列长度和丢包率的变化,实验结果如图4所示。与图2相比,两种算法在TCP连接数量稳定时队列长度都比较稳定,但本文算法的丢包率小于BLUE算法。原因是本文算法能根据平均队列长度自适应地调节丢包概率 P_d ,避免了队列溢出和队列空闲状态的出现。

4.2 TCP连接数量发生突变时队列长度与丢包率的变化

实验2 TCP连接数量改变时队列长度及丢包率的变化

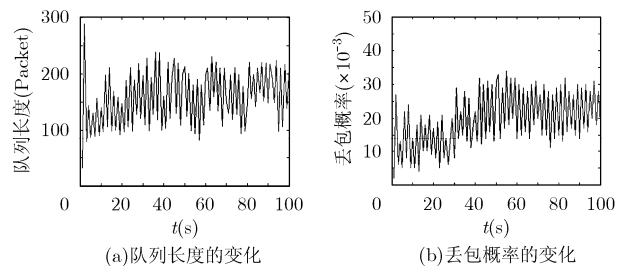


图4 TCP连接数稳定

化。实验条件同实验1, TCP连接数在30-60s之间突然减少一半, 其余时间段为200, 实验结果如图5所示。

从图5可以看到, 即使在TCP数发生突变的情况下本文算法仍能有效地保持队列的稳定。第30sTCP流突然减半时以及第60sTCP流增倍时, 丢包概率都能迅速做出反应, 队列长度比较稳定。原因是本文算法通过提前检测机制, 自适应地调整丢包概率。而同样条件下的BLUE算法, 队列长度振荡幅度大, 且队列溢出和空闲现象较多。原因是BLUE算法在发生丢包事件后才增加丢包概率, 反应滞后。

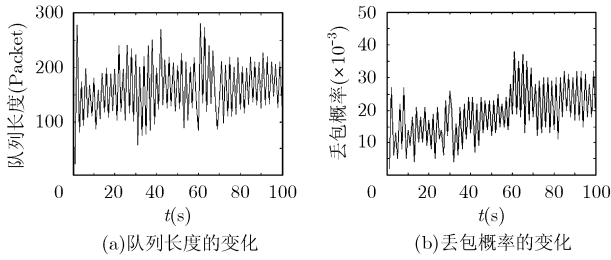


图5 TCP连接数有变化的情况

由对比实验可以看出, 本文算法在TCP连接数量突然发生改变时能减少空队列和队列溢出现象的发生, 减少了丢包率, 能有效保持队列长度的稳定性。

4.3 两种算法在不同缓存条件下丢包率、链路利用率、队列时延的变化比较

实验3 检验了TCP连接数量突然发生改变时, BLUE算法和本文算法在不同缓存空间条件下链路利用率、丢包率和队列时延的比较。

参数设置及TCP连接数量的变化同实验2, 实验结果见图6, 图7, 图8。

由图6, 图7, 图8可以看出, 总体上两种算法的丢包率都随着缓存空间的增大而呈现减少的趋势, 只在TCP连接数

变化瞬间增大; 而队列延时则正相反, 都随着缓存空间的增大而增大; 链路利用率只在TCP连接数量突然改变时有波动, 很快随着缓存空间的增大而增加并逐渐趋于平稳。

从实验结果可以看出, 在相同的实验条件下本文算法的丢包率、队列延时都小于BLUE算法, 而链路利用率比BLUE算法高。原因是本文算法采用参数自适应调节机制, 当平均队列长度大于理想范围时就及时增大 P_q 的幅度, 能及时减少队列长度而不是等待队列溢出时才做出滞后的调节反应, 因而能迅速阻止队列长度的继续快速增加, 从整体上减少了丢包, 降低了丢包概率。相反, 当平均队列长度小于理想范围时就及时减少 P_q 的值, 从而迅速恢复队列长度, 有效地提高了链路的利用率。

队列延时方面, 本文算法也比 BLUE 算法小一些, 但优势不是特别的明显, 原因是本文算法的主要目标是保持队列的稳定。

由以上分析得出, 在相同的条件下本文算法的各种性能均比 BLUE 算法有所提高, 尤其是 TCP 连接数量剧烈变动时性能比较好, 能提高队列的稳定性、减少丢包率。

5 结束语

本文通过分析BLUE算法的不足之处, 针对该算法在TCP连接数量剧烈变动时导致参数设置不能有效控制网络拥塞而导致的队列不稳定问题, 引入参数自适应机制, 同时借鉴了RED算法的早期拥塞检测机制, 提出了基于参数自适应的主动队列管理算法。算法根据丢包事件和链路空闲事件来管理拥塞控制, 并通过检测平均队列长度的变化来检测早期的拥塞, 在不同状态下根据网络的实际状况自适应调整丢包概率, 充分利用了网络资源, 又保证了一些成间歇性变化的流的公平性, 减少了无谓的丢包。从NS仿真实验结果可以看出本文算法可以有效地、稳定地提高BLUE 算法的性能。

参考文献

- [1] Jacobson V. Congestion avoidance and control. *ACM Computer Communication Review*, 1988, 18(4): 314-329.
- [2] Floyd S and Fall K. Promoting the use of end-to-end congestion control in the internet. *IEEE/ACM Trans on Networking*, 1999, 7(4): 458-472.
- [3] Floyd S and Jacobson V. Random early detection gateway for congestion avoidance. *IEEE/ACM Trans on Networking*, 1993, 1(4): 397-413.
- [4] 林闯, 单志广, 任丰原. 计算机网络的服务质量(QoS). 北京: 清华大学出版社, 2004: 73-102.
Li Chuang, Shan Zhi-guang, and Ren Feng-yuan. *QoS of Computer Networks*. Beijing: Tsinghua University Press, 2004: 73-102.
- [5] Feng W, Shin K, and Kandlur D, et al. The blue active queue management algorithms. *IEEE/ACM Trans Networking*, 2002, 10(4): 513-528.

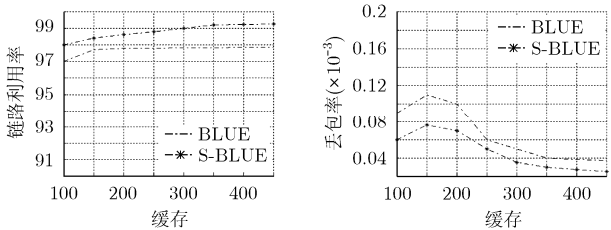


图6 缓存大小与链路利用率关系

图7 缓存大小与丢包率关系

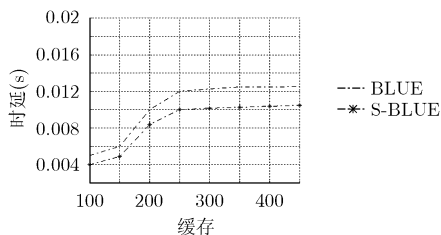


图8 缓存大小与队列时延的关系

- [6] Floyd S. A report on some recent developments in TCP congestion control. *IEEE Communication Magazine*, 2001, 39(4): 84-90.
- [7] Liu Wei-yan, Zhang Shun-yi, Zhang Mu, and Liu Tao. A parameter adaptive-BLUE based on DT in IP network. ISDA' 2006, Ji'nan, China, 2006, (1): 1086-1091.
- [8] 吴春明, 姜明. SBlue: 一种增强稳定性的主动队列管理算法[J]. 通信学报, 2005, 26(3): 68-74.
Wu Chun-ming and Jiang Ming. SBlue: stabilized Blue. *Journal on Communications*, 2005, 26(3): 68-74.
- [9] 杨云, 徐佳, 王秋平等. 一种精确度加强的主动队列管理算法 PEBLUE[J]. 小型微型计算机系统, 2006, 27(4): 592-595.
Yang Yun, Xu Jia, and Wang Qiu-ping. Precision enhanced active queue management algorithm PEBLUE. *Journal of Chinese Computer Systems*, 2006, 27(4): 592-595.
- [10] 张顺亮, 叶澄清, 李方敏. 一种基于速率的BLUE改进方法[J]. 计算机研究与发展, 2004, (4): 660-666.
Zhang Shun-liang, Ye Cheng-qing, and Li Fang-min. A rate-based method to enhance BLUE[J]. *Journal of Computer Research and Development*, 2004, (4): 660-666.
- [11] Network simulationor-NS2[EB/OL]. [Http://www.mash.cs.berkeley.edu/ns](http://www.mash.cs.berkeley.edu/ns).
- 刘伟彦: 女, 1973年生, 讲师, 博士, 研究方向为计算机通信与IP技术、多媒体通信技术.
- 孙雁飞: 男, 1976年生, 博士, 讲师, 研究方向为新一代网络监测、管理与控制.
- 张顺颐: 男, 1944年生, 南京邮电大学副校长, 教授, 博士生导师, 研究方向为计算机通信网与IP技术.
- 刘 斌: 男, 工程师, 研究方向为计算机网络.