

# 基于采样的 gCHOKe 主动队列管理算法的研究

张 萌,周井泉,唐上进

(南京邮电大学 电子科学与工程学院,江苏 南京 210003)

**摘要:**主动队列管理(Active Queue Management, AQM)算法是网络拥塞控制中非常重要的研究领域之一,文中主要研究了主动队列管理算法的公平性。gCHOKe 算法是一种近似公平的主动队列管理算法,一旦发生 gCHOKe 击中则继续抽取分组进行比较直至无击中,但它仍然存在带宽公平性不足的问题,因此提高非适应性流的击中有效性以及惩罚非适应性流的力度是提高算法公平性的关键因素。文中在 gCHOKe 算法的基础上,提出了一种基于采样的主动队列管理算法 sg-CHOKe (Sampling based gCHOKe),在进行 gCHOKe 击中处理时首先采样多个分组与已到达分组进行比较,并采用新的策略机制丢包,来提高算法的公平性。仿真实验结果表明,sgCHOKe 在维护网络带宽公平分配上性能优于 CHOKe, gCHOKe 算法,具有更高的击中有效性与带宽公平性,并且拥有更高的网络公平性指数。

**关键词:**gCHOKe; sgCHOKe; 主动队列管理; 带宽公平性

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2015)09-0098-04

doi: 10.3969/j.issn.1673-629X.2015.09.021

## Research on Improved gCHOKe Algorithm Based on Sampling

ZHANG Meng, ZHOU Jing-quan, TANG Shang-jin

(College of Electronic Science and Engineering, Nanjing University of Posts and Telecommunications,  
Nanjing 210003, China)

**Abstract:** Active Queue Management (AQM) algorithm is one of very important research field of network congestion control. Focus on the fairness of the active queue management algorithm in this paper. gCHOKe is an algorithm of fair approximation of active queue management. When gCHOKe hit happened, the algorithm extracts another packet and judge whether gCHOKe hit happening or not. The gCHOKe hit, whose effectiveness and the force of punishing non-response flows are the key factors to enhance the fairness algorithm, is lack of fairness. Based on gCHOKe algorithm, sgCHOKe (Sampling based gCHOKe) is proposed in this paper, through analyzing the low effectiveness of non-response flows hit, which samples several packets from queue to compare with the arrived packet and employ a new mechanism of packet-loss. The simulation results based on NS-2 show that compared with CHOKe and gCHOKe, sgCHOKe can maintain the fairness of different bandwidths better, achieving better effectiveness of non-response flows hit, with high index of network fairness.

**Key words:** gCHOKe; sgCHOKe; active queue management; fairness of bandwidth

## 0 引言

随着互联网规模越来越庞大,网络拥塞已经成为一个迫切需要解决的问题。而主动队列管理算法(Active Queue Management, AQM)正是网络拥塞控制研究的热点之一。它是 IETF 推出的基于 FIFO 调度策略的队列管理机制,它能有效地在队列溢出之前检测出早期的拥塞,并且向源端发出拥塞指示<sup>[1-3]</sup>。比较经典的主动队列管理算法有 RED 算法、CHOKe 算

法<sup>[4]</sup>、ARED 算法<sup>[5]</sup>、BLUE 算法<sup>[6]</sup>。RED 算法是 Floyd 等提出的最早的 AQM 算法,也是比较成熟的 AQM 算法,但存在参数设置敏感和队列长度不稳定的问题;CHOKe 算法是 RED 算法基础上的改进,它加入了击中机制,增加了算法的公平性;ARED 算法是自适应的 RED 算法,同样是基于 RED 算法的改进;BLUE 算法记录过去的丢包率和链路利用状态,自适应调节丢包率,但仍然有参数设置敏感的问题。而公平性对

收稿日期: 2014-11-24

修回日期: 2015-03-04

网络出版时间: 2015-08-26

基金项目: 江苏省普通高校研究生科研创新基金(CXLX12\_0417)

作者简介: 张 萌(1990-),男,硕士研究生,研究方向为网络拥塞控制;周井泉,博士,教授,硕士生导师,研究方向为通信网络的信息管理和控制。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150826.1603.074.html>

于主动队列管理算法越来越重要。因为大多数 AQM 算法对各个流的丢弃是同等对待的,没有考虑到适应性流(TCP 友好流)和非适应性流(一些 UDP 流和不响应拥塞的 TCP 流)的区别对待,这样可能会导致带宽被大量非适应性流侵占。

而 gCHOKe (Geometric CHOKe) 算法<sup>[7]</sup>是在 CHOKe 基础上公平性的改进,它仍然每次只抽取一个分组进行击中比较,但是如果击中则再抽取一个分组进行击中比较直到不匹配为止,这样能有效地惩罚非适应性流,保护 TCP 流,有效地提高了算法的公平性,但是它的击中有效性仍存在一定问题。

文中提出了 sgCHOKe (Sampling based gCHOKe) 算法,随机采样多个分组与已到达分组进行击中比较,并按新的丢弃策略丢包,使得改进算法具有更高的击中准确性。仿真结果表明,在相同网络环境下,sg-CHOKe 算法在带宽公平性上优于 gCHOKe 算法,并且拥有更高的公平性指数。

## 1 相关工作

改进算法在 gCHOKe 算法的基础上加入了新的采样机制,按新的丢包策略丢包。

### 1.1 gCHOKe 算法

CHOKe 算法在 RED 算法的基础上引入了击中机制,一旦拥塞发生,先从缓冲队列中随机取一个分组与已到达分组做匹配,如果两者匹配,即击中,丢弃这两个分组,如果两者不匹配,则继续进入 RED 处理。gCHOKe 在 CHOKe 的击中机制上进行了改进,它加大了对非适应性流的惩罚力度,当拥塞发生时它从队列中随机抽取一个分组与已到达分组作比较,若属于同一个流则击中,则继续抽取一个分组进行比较,直到它们不属于同一个流或者达到预先设定的比较次数为止,最后丢弃所有击中的分组,其余未击中的分组继续按 RED 算法来处理。由此可以看出,当预先设定的比较次数为 1 时,gCHOKe 等价于 CHOKe,可以说 CHOKe 算法属于 gCHOKe 算法的一个特例。

当队列发生拥塞时,gCHOKe 算法丢弃分组的概率为:

$$P_{\text{gCHOKe}} = \frac{2h_0 - h_0^2}{1 - h_0} \quad (1)$$

其中,  $h_0$  表示非适应性流的匹配概率。

可以从该公式看出,gCHOKe 在公平性方面仍存在一定问题。首先由于 gCHOKe 识别非响应流属于近似识别<sup>[7]</sup>,gCHOKe 击中是随机的,有可能会击中适应性分组,即坏击中,所以当 gCHOKe 误击中适应性流时,很有可能会造成对响应流的误击,适应流带宽会不断减小,反而造成相反的效果。其次,gCHOKe 每次只

抽取一个分组进行比较,当有大量非适应性流进入缓冲队列时,惩罚力度依旧很低,因此带宽公平性<sup>[8-10]</sup>并没有显著提高。

### 1.2 采样机制

sgCHOKe 算法在 gCHOKe 算法基础上加入了采样机制。由于在当网络中流数量很多的情况下,gCHOKe 算法每次仍然只取一个进行击中比较,因此 CHOKe 击中有效性并不高。采样机制在拥塞发生后从队列中采样  $m$  个包与已到达包做比较,其中  $m$  的取值是按照最小阈值和最大阈值之间划分的区间来判断的,将该区间划分为  $k$  个等分区间,若平均队长位于区间  $i$ ,则  $m = i^{[11-12]}$ 。通过这种采样机制,经过多个样本的比较能够有效地提高非适应性流的识别率,从而能有效地提高非适应性流的击中有效性,加大惩罚力度。

### 1.3 新的丢包策略

当发生拥塞,即平均队列长度在最大阈值 MAXth 和最小阈值 MINth 之间时,先采样  $m$  个包进行击中比较,如果 CHOKe 击中数大于  $m/2$ ,那么将这  $m$  个分组全部丢弃,用  $m/2$  作为临界点是考虑到高速的非适应流通常会在  $m$  中占一定比例(1/3 以上),这样可以有效防止丢弃脆弱流,提高丢包准确度<sup>[8-9]</sup>;如果小于  $m/2$ ,只要存在击中,仍然按 gCHOKe 的策略进行匹配,即再从队列中取一个包与已到达包做比较,直到与已到达的包不匹配,将抽取到的未击中的包放回队列并丢弃所有击中的包。采用新的丢包策略能很好地提高 CHOKe 击中的概率并抑制非适应性的 UDP 流。

## 2 基于采样的 gCHOKe 改进算法

基于对 gCHOKe 算法存在的非适应性流击中有效性不足的问题,文中提出了一种基于采样的 gCHOKe 改进算法 sgCHOKe。算法框图如图 1 所示。

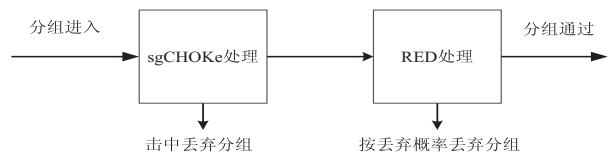


图 1 sgCHOKe 算法框图

当分组进入队列后,先进行 sgCHOKe 处理丢弃非适应性流后,再进入 RED 进行处理。具体的 sgCHOKe 算法流程如图 2 所示。

具体步骤如下:

步骤 1: 分组到达后,首先判断分组平均队列长度所在区间,若平均队列长度小于 MINth,则进入 RED 处理,若平均队列长度大于 MAXth,则丢弃到达分组。当平均队列长度在 MINth 和 MAXth 之间时发生拥塞,进入 sgCHOKe 处理阶段。

步骤 2: 当进入 sgCHOKe 处理后,先从缓冲队列中

采样  $m$  个包,与一个已到达分组进行击中比较。其中  $m$  是按照最小阈值和最大阈值划分的,根据平均队列长度位于哪个区间来确定  $m$  的值。

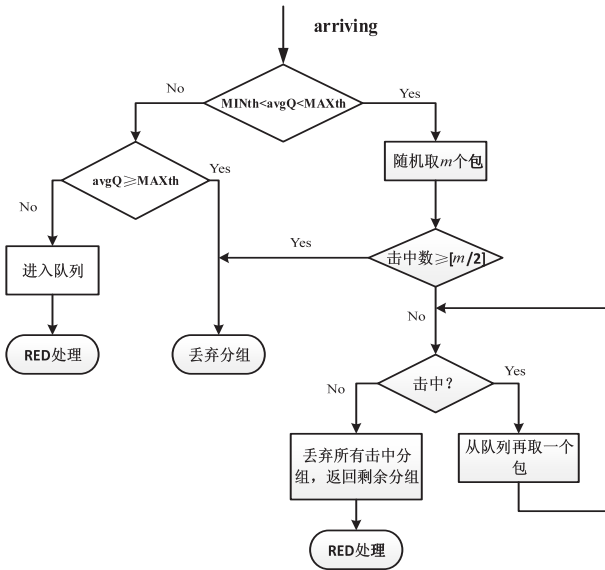


图 2 sgCHOCe 算法工作流程图

步骤 3: 当 sgCHOCe 击中的个数大于  $m/2$  时, 丢弃这  $m$  个分组。说明此时非适应性流很多。而当 sgCHOCe 击中个数小于  $m/2$  且 sgCHOCe 发生击中时, 则根据 gCHOCe 的击中机制继续从队列中取一个分组再进行击中比较, 直到不再发生击中, 然后丢弃所有击中的分组, 返回剩余的分组并进入 RED 处理。

### 3 仿真实验分析

实验的主要目的是验证 sgCHOCe 算法的带宽公平性, 在 NS2<sup>[13]</sup> 仿真软件下进行实验仿真, 对 CHOCe, gCHOCe, sgCHOCe 三种算法进行 TCP、UDP 带宽的比较以及公平性指数的衡量。

实验网络拓扑如图 3 所示。

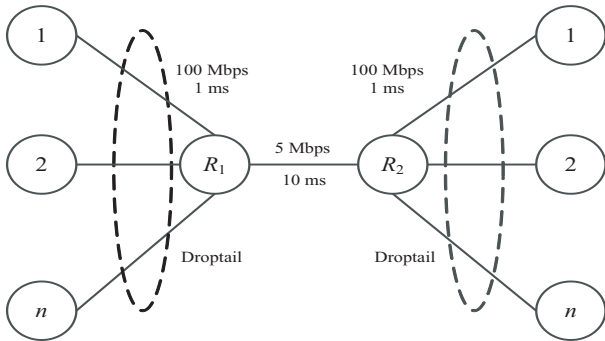


图 3 网络拓扑

图中, 共有  $n$  个源端和  $n$  个目的端,  $R_1$  和  $R_2$  为中间路由器。各源端和目的端到路由器的链路带宽为 10 Mbps, 延时为 1 ms, 队列算法均为 Droptail。路由器  $R_1$  和  $R_2$  之间带宽为 5 Mbps, 延时为 10 ms, 这样形成了一条瓶颈链路。TCP 源端发送 FTP 流, UDP 端发送

CBR 流, 分组大小为 800 个字节。队列参数 MINth 为 150, MAXth 为 300, 缓冲队列长度为 500。

仿真实验: 仿真共取了 80 个源端, 其中 78 个为 TCP 源端, 分组发送速度为 0.1 Mbps, 2 个为 UDP 源端, 分组发送速度由 0.6 Mbps, 0.7 Mbps, 0.8 Mbps, 0.9 Mbps, 1 Mbps 逐步增加。每个速度分别持续 20 s。所有源端均从 0 s 开始发送直到 100 s 结束, 整个过程持续 100 s。然后在 UDP 流个数不变的情况下, 观察在 UDP 速度逐渐增加时网络中 TCP 流和 UDP 流的吞吐量。

图 4 是 UDP 流实际占用带宽与 TCP 流实际占用带宽在 CHOCe, gCHOCe, sgCHOCe 三种算法下的比较结果。

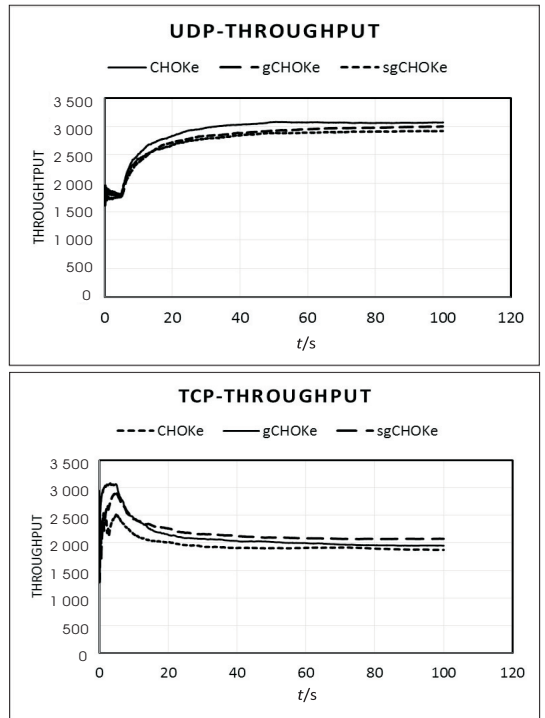


图 4 UDP 流吞吐量和 TCP 流吞吐量比较

从图 4 可以看到, 在 UDP 流个数恒定, 发送速度持续增加的情况下, 三种算法中 sgCHOCe 相比 CHOCe 和 gCHOCe, 在抑制 UDP 流的能力方面有很大的提高, 很好地削减了非适应性流; 而在保护 TCP 流方面, sgCHOCe 算法也较好地保护了 TCP 这样的适应性流。因此 sgCHOCe 在带宽公平性指标上效果最好并且很好地解决了 gCHOCe 存在的非适应性流击中有效性的问题。

而公平性指数<sup>[14]</sup>更能直观地反映网络的公平性, 它介于 0 到 1 之间, 公平性指数由 Jain 等提出, 具体计算公式如下:

$$FI = \frac{(\sum_{i=1}^n x_i)^2}{n \times \sum_{i=1}^n x_i^2} \tag{2}$$

其中,  $x_i$  为吞吐量;  $n$  为参与的个数。

公平性指数越接近 1, 网络越公平, 反之, 越不公平。在相同仿真条件下, 分别对 CHOKe, gCHOKe, sg-CHOKe 算法进行仿真, 计算得到相同条件下各算法在各个时间段下的公平性指数, 见表 1。

表 1 各算法公平性指数对比表

	sgCHOKe	gCHOKe	CHOKe
0 ~ 20 s	0.734 322	0.647 522	0.542 733
20 ~ 40 s	0.786 512	0.642 345	0.623 455
40 ~ 60 s	0.893 455	0.893 566	0.820 344
60 ~ 80 s	0.933 455	0.933 002	0.921 335
80 ~ 100 s	0.972 344	0.973 014	0.963 345

从表 1 可以看出, sgCHOKe 算法的公平性指数在大部分时间都是最高的, 尤其是在 0 ~ 40 s 这个阶段效果更为明显, 此时 UDP 速率刚开始递增。sgCHOKe 算法由于针对 UDP 非适应性流的采样机制和击中机制来双重处理 UDP 流, 使得网络中的 UDP 流所能占用的带宽会被进一步遏制, 相比 CHOKe 和 gCHOKe 算法, 它的网络公平性指数更高。

## 4 结束语

文中在 gCHOKe 的基础上提出了 sgCHOKe 算法, 加入了采样机制提高了 CHOKe 击中的比较次数, 并且通过一种新的分组丢弃策略, 将可能性高的非适应性流大量丢弃, 并对击中分组进行再抽取进行击中比较的方式, 来保护适应性流, 惩罚非适应性流。通过 NS2 仿真软件的实验与分析, 结果表明 sgCHOKe 算法拥有更高的带宽公平性, 很有效地提高了非适应性流击中的概率, 保护了适应性流并惩罚了非适应性流, 拥有更高的网络公平性指数。

## 参考文献:

[1] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance[J]. IEEE Transactions on Networking, 1993, 1(4):397-413.

(上接第 97 页)

graphs[J]. IEEE Trans on Visualization and Computer Graphics, 2008, 14(3):526-530.

[10] 巩敦卫, 耿娜, 张勇. 密集障碍物环境下基于凸包和微粒群优化的机器人路径规划[J]. 控制理论与应用, 2012, 29(5):609-616.

[11] Eluru N, Chakour V, El-Geneidy A M. Travel mode choice and transit route choice behavior in montreal: insights from McGill University members commute patterns [J]. Public Transport, 2012, 4(2):129-149.

[2] 龚静, 吴春明. S-CHOKe: 一种增强 CHOKe 公平性的主动式队列管理算法[J]. 电子学报, 2010, 38(5):1100-1104.

[3] 姜明, 边浩, 张少丽. 改进的基于 CHOKe 击中历史的公平主动式队列管理[J]. 计算机应用, 2010, 30(2):289-291.

[4] Pan R, Prabhakar B, Psounis K. CHOKe: a stateless active queue management scheme for approximating fair bandwidth allocation[C]//Proceedings of IEEE INFOCOM 2000. Piscataway: IEEE, 2000:942-951.

[5] 吴春明, 姜明, 朱森良. 几种主动式队列管理算法的比较研究[J]. 电子学报, 2004, 32(3):429-434.

[6] 刘艺, 孙鹤旭. 基于 BLUE 的主动队列管理算法分析[J]. 沈阳工程学院学报: 自然科学版, 2008, 4(1):56-59.

[7] Eshete A, Jiang Yuming. Protection from unresponsive flows with geometric CHOKe[C]//Proc of ISCC. [s. l.]: [s. n.], 2012.

[8] 王建新, 周雄伟, 杨湘. 一种惩罚非适应流的无状态主动队列管理算法[J]. 系统工程与电子技术, 2006, 28(12):1935-1939.

[9] 汤德佑, 骆嘉伟, 张大方, 等. 一种提高稳定性和公平性的主动队列管理机制[J]. 计算机研究与发展, 2005, 42(7):1136-1142.

[10] Chhabra P, John A, Saran H, et al. Controlling malicious sources at internet gateways[C]//Proc of IEEE international conference on communication. Anchorage: IEEE, 2003:1636-1640.

[11] 李学锋, 郑毅. 一种基于抽样统计改进的 CHOKe 算法[J]. 计算机时代, 2013(8):49-51.

[12] 姜明, 边浩, 陈勤. HCHOKe: 改进的公平主动队列管理算法[J]. 计算机工程, 2010, 36(10):115-117.

[13] 王秀丽. AQM 算法在 NS2 中的实现及其性能评价[J]. 计算机科学, 2009, 36(5):60-64.

[14] Jain R, Chiu D M, Hawe W. A quantitative measure of fairness and discrimination for resource allocation in shared systems [R]. [s. l.]: [s. n.], 1984.

[12] Golmakani H R, Birjandi A R. A two-phase algorithm for multiple-route job shop scheduling problem subject to makespan [J]. International Journal of Advanced Manufacturing Technology, 2013, 67(1):203-216.

[13] 杨玲玲, 马良. 非线性规划的混沌粒子群优化算法[J]. 计算机技术与发展, 2012, 22(10):15-17.

[14] 何娟, 涂中英, 牛玉刚. 一种遗传蚁群算法的机器人路径规划方法[J]. 计算机仿真, 2010, 27(3):170-174.

[15] 王越, 吕光宏. 改进的粒子群求解多目标优化算法[J]. 计算机技术与发展, 2014, 24(2):42-45.

# 基于采样的gCHOKe主动队列管理算法的研究

作者: [张萌](#), [周井泉](#), [唐上进](#), [ZHANG Meng](#), [ZHOU Jing-quan](#), [TANG Shang-jin](#)  
作者单位: [南京邮电大学 电子科学与工程学院, 江苏 南京, 210003](#)  
刊名: [计算机技术与发展](#)   
英文刊名: [Computer Technology and Development](#)  
年, 卷(期): 2015 (9)

引用本文格式: [张萌](#). [周井泉](#). [唐上进](#). [ZHANG Meng](#). [ZHOU Jing-quan](#). [TANG Shang-jin](#) [基于采样的gCHOKe主动队列管理算法的研究](#)[期刊论文]-[计算机技术与发展](#) 2015 (9)