

一种公平带宽分配的 AQM 算法

邱建勇

(华中科技大学 计算机科学与技术学院, 湖北 武汉 430074)

摘要: 研究了响应流与非响应流公平共享带宽问题, 当各微流竞争路由器的同一个输出链路时, 非响应流趋向于压制响应流, 使得带宽公平性遭受破坏。提出一种新的主动队列管理算法——动态阈值 RED (DTRED) 算法, 通过动态地调节队列参数来获得更加公平的带宽分配。

关键词: 带宽公平性; TCP; UDP; 主动队列管理; 随机早期检测

中图分类号: TP393.09

文献标识码: A

文章编号: 1673-9833(2007)03-0061-03

AQM Algorithm with Fair Distribution of Link Capacity

Qiu Jianyong

(College of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: The problem of fair bandwidth sharing among responsive and unresponsive traffic flows is investigated. When these flows competing for the same output link in a router, unresponsive flows tends to occupy more than their fair share of the link capacity. A new active queue management algorithm named Random Early Detection with Dynamic Thresholds (DTRED) is put forwards which dynamically adapts queue parameters to achieving a more fair distribution for the link capacity.

Key words: Fairness; TCP; UDP; AQM; RED

现今 Internet 上的大部分流量都是 TCP 流。TCP 的拥塞控制机制, 能够根据网络的拥塞情况自动调节发送速率。具有这种属性的传输协议被称为是响应的 (responsive), 相应的传输流被称为响应流。UDP 是在实时服务中最常用的协议, 随着网络多媒体应用的发展, Internet 上以 UDP 为传输协议的应用日益增多。由于 UDP 对网络拥塞不做任何反应, 故是非响应的 (unresponsive)。当拥塞发生时, 路由器缓冲将会溢出, 新到的数据包将会被丢弃。TCP 发送端视数据包丢弃为网络拥塞的标志, 并降低发送速率。而 UDP 发送端对网络拥塞一无所知, 继续保持原来的发送速率。这种拥塞控制机制的不同, 引出响应流与非响应流共享可用带宽时的带宽公平分配问题。

解决带宽公平性问题的一种方法是采用主动队列管理 (Active Queue Management, 简称 AQM) 算法。最早且广泛被采用的 AQM 算法是随机早期检测 (Random Early Detection, 简称 RED) 算法^[1]。但 RED 算

法不能抑制非响应流, 因为它对所有到达的数据包使用同样的丢弃概率, 因而不能很好地解决链路带宽的不公平分配问题。有些 AQM 算法, 如 FRED (Flow Random Early Detection)、LQD (Longest Queue Drop)、CHOKe 等, 能够确定并采用优先丢弃的方法抑制非响应流。本文描述一种新的 AQM 算法——DTRED (Dynamic Thresholds RED) 算法。

1 DTRED 算法

在拥塞期间, 流的输出速率能够从缓存的占用情况获知。“贪婪”的非响应流使用超过其公平共享部分的链路带宽以及超出其公平共享部分的缓存, 为了确定这些“贪婪”的非响应流, DTRED 算法对每一个“活动”流 (active flow) 保持一个每流状态 (per-flow state)。当一个流至少有一个数据包在缓冲队列中时, 我们认为这个流是“活动”的。对每个“活动”流, 保持一个状态记录 (Entry) 在 Cache 中, 这些记录组成一个

收稿日期: 2007-03-19

作者简介: 邱建勇 (1975-), 男, 湖北黄冈人, 华中科技大学硕士生, 主要研究方向为计算机网络。

活动流表 (Active Flow Table), Entry 中包含对应“活动”流 (单流) 的即时队列长度 q^i , 平均队列长度 q_{avg}^i , 最大丢弃概率 p_{max}^i 。同 RED 一样, 缓存中所有“活动”流组成一个总队列, 其当前队列长度 q 和平均队列长度 q_{avg} , 也同时被记录。而且, DTRED 也保持一对最大、最小阈值 $th_{max}=3th_{min}$, 不同的是, 在 DTRED 算法中, 它们的值是在每一个数据包到达时动态改变的。如果新到的数据包属于“活动”流 i , th_{max} 按下式计算:

$$th_{max} = (1 - \alpha p_{max}^i)(B - q), \quad (1)$$

其中, B 为缓冲大小, p_{max}^i 是“活动”流 i 的最大丢弃概率, α 是一个常量系数。DTRED 采用下式计算到达的数据包的丢弃概率:

$$p = \begin{cases} 0, & \text{if } q_{avg}^i \leq th_{min}; \\ \frac{q_{avg}^i - th_{min}}{th_{max} - th_{min}}, & \text{if } th_{min} \leq q_{avg}^i \leq th_{max}; \\ 1, & \text{if } th_{max} \leq q_{avg}^i. \end{cases} \quad (2)$$

不同的是, DTRED 将某一“活动”流的平均队列长度 q_{avg}^i 与阈值相比较, 而 RED 则将进入缓冲队列的所有流的总平均队列长度 q_{avg} 与其相应的阈值相比较。当一个数据包被允许进入队列, 最大丢弃概率将相应被重新计算和更改:

$$p_{max}^i = \begin{cases} (1, p_{max}^i + \delta)_{max}, & \text{if } q_{avg}^i > \frac{q_{avg}}{N}; \\ p_{max}^i, & \text{if } q_{avg}^i = \frac{q_{avg}}{N}; \\ (p_{min}, p_{max}^i - \delta)_{min}, & \text{if } q_{avg}^i < \frac{q_{avg}}{N}. \end{cases} \quad (3)$$

其中, δ 是一个常量, p_{min} 是 RED 算法中定义的参数, 而 N 则是缓冲队列中“活动”流的数量。为了尽早提供反馈给响应流, 最大丢弃概率 p_{max}^i 不能小于 p_{min} 。一种方法是设定常量 $\delta = p_{min}$ 。那些平均队列长度 q_{avg}^i 大于缓冲总队列长度 q_{avg} 的 $1/N$ 的“活动”流, 其新到数据包的最大丢弃概率 p_{max}^i 将逐渐增加。这些流被初步判定为潜在的非响应流或者“贪婪”流。同时, 它们的阈值将根据公式 (1) 重新计算而逐渐减小。而当其平均缓存占有量逐渐减少到缓存总队列长度的 $1/N$ 以下时, 它们的最大丢弃概率便又逐渐减小, 阈值却逐渐增加。这种机制的目的是在这些“活动”流之间平均分配缓存空间。

重新计算阈值是 DTRED 算法的主要时间复杂度问题, 空间需求仅需一个保存活动流表的 Cache。

2 算法评估

我们使用 ns-2^[2]来模拟 DTRED 算法并与 RED^[3]、FRED^[4]、LQD^[5]和 CHOKe^[6]进行性能比较。假设在

ns-2 中已经实现了上述几种算法, 所有的模拟情形都使用如图 1 所示的网络拓扑结构。

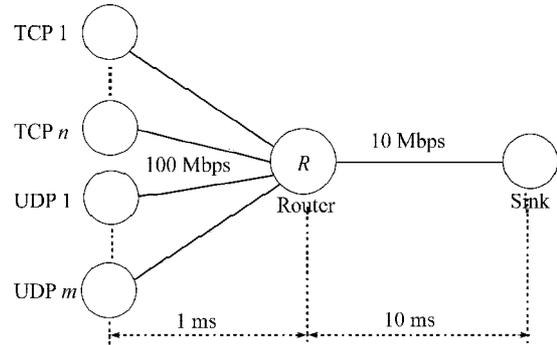


图 1 模拟网络拓扑结构

Fig. 1 Simulated network topology

网络中传输的都是 FTP 数据流。TCP (TCP New Reno) 和 UDP 源端发送的数据包长度都固定为 500 Bytes, 并且 UDP 源端发送的是 CBR (Constant Bit Rate) 流。缺省缓冲大小是可容纳 200 个数据包。UDP 的发送速率是 2.5 Mbps。

在第一个模拟情形中, 我们使用 16 个 TCP 流源与 4 个 UDP 流源竞争, 汇集到路由器的同一个输出端口, 形成一个聚集, 缓冲排队输出。我们使 UDP 流量与输出链路容量的比率从 5% 递增至 120%, 观察每个流的情况, 获得平均吞吐量, 并计算公平指数 Jain Fairness Index^[7]如图 2 所示, 由图 2 可以看出, 公平指数值非常接近 1, 可见 DTRED 算法很好地解决了带宽公平分配问题。并且 DTRED 算法所得的公平指数不会因为 UDP 的流量比率变化而降低, 即具有稳定的公平性。由图 2 还可看出, FRED 算法的公平性也很好, 而 RED、LQD 和 CHOKe 算法则表现不佳, 无法解决当 UDP 流量增加时的公平性问题。

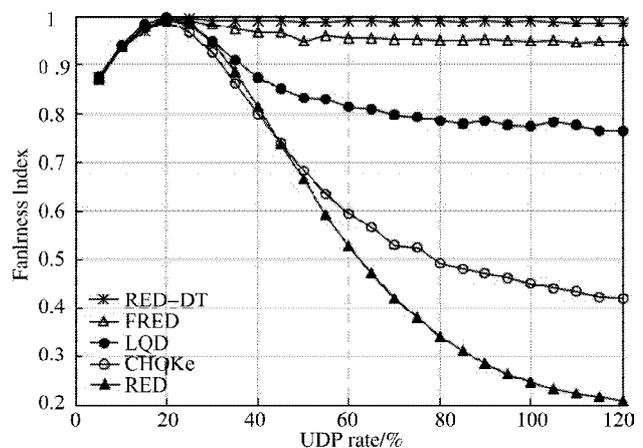


图 2 不同 UDP 负载比率下各算法的公平指数比较

Fig. 2 Fairness index for various UDP loads

第二个模拟情形是为了评估缓冲大小对公平指数的影响。如图 3 所示, DTRED 算法在缓存时延在 40 ms

至 200 ms 时, 性能都表现良好。因此, DTRED 算法的缓存大小适应性好。FRED 算法则至少需要 120 ms 的缓存空间才能获得相同的效果。而 LQD、CHOKe 和 RED 算法在增大缓存空间时, 甚至根本就不会提高带宽公平性, 并且维持在较低水平的公平指数上。

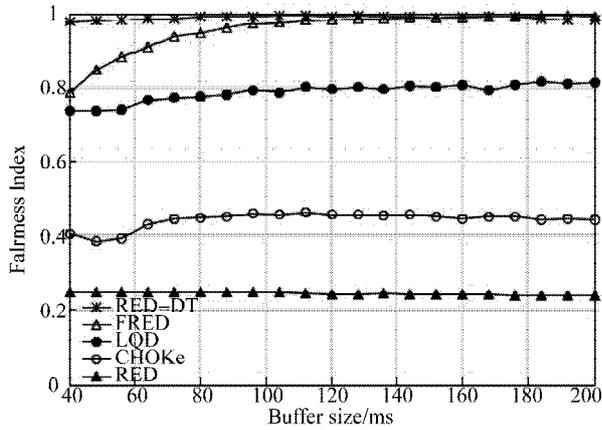


图3 不同缓存大小下各算法的公平指数的比较

Fig. 3 Fairness index for various buffer sizes

在计算公式 (2) 中的 α 是 DTRED 算法中最重要参数, 如果 $\alpha = 0$, 阈值 th_{max} 和 th_{min} 就与 p_{max} 无关。较大的 α 意味着“贪婪”流 i , 通过更快地降低其阈值, 将会被更多地受到抑制。不同的 α 得到的 Jain Fairness Index 如图 4 所示。

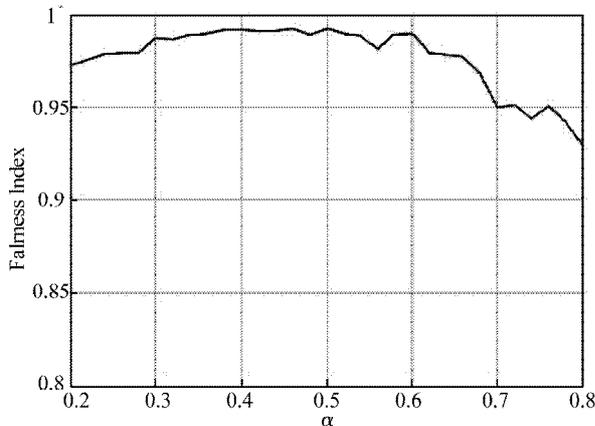


图4 α 取值对带宽公平指数的影响

Fig. 4 Fairness index for various parameter α

可见较好的 α 选择是在 0.3 与 0.6 之间。

3 结论

本文提出了一种新的主动队列管理算法 DTRED, 其目标是为在拥塞的路由器中的竞争流提供公平的带宽分配。模拟结果显示, 在与其他 AQM 算法如 RED、FRED、LQD 及 CHOKe 比较后, DTRED 算法在带宽公平性方面显示出较优的表现, 并在性能和复杂性之间取得了较好的平衡。

参考文献:

- [1] Floyd S. RED: discussions of setting parameters[EB/OL][1997-11-01]. [http://www.icir.org/floyd/RED parameters.txt](http://www.icir.org/floyd/RED%20parameters.txt).
- [2] Anon. The Network Simulator-ns-2:[EB/OL]. [发布日期不详]. <http://www.isi.edu/nsnam/ns/>.
- [3] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance[J]. IEEE Transactions on Networking, 1993, 1(4): 397-413.
- [4] Lin D, Morris R. Dynamics of random early detection[C]// Proc. of ACM SIGCOMM '97. Cannes: [s.n.], 1997: 127-137.
- [5] Suter B, Lakshman T V, Stiliadis D, Choudhury A. Design considerations for supporting TCP with per-flow queuing[C]// in Proc. of IEEE INFOCOM'98. San Francisco: CA, 1998: 299-306.
- [6] Pan R, Prabhakar B, Psounis K. CHOKe: a stateless active queue management scheme for approximating fair bandwidth allocation[C]//in Proc. of IEEE INFOCOM'00. Tel-Aviv/Israel: [s.n.], 2000: 942-951.
- [7] Jain R. The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling[M]. New York: John Wiley & Sons, 1991.