

一种改进的随机早期检测算法

吕兆齐

(南京邮电大学 计算机学院, 江苏 南京 210003)

摘要: IETF 推荐使用主动队列管理, 如 RED(Random Early Detection), 来解决网络中的拥塞控制问题。但 RED 中平均队列长度的计算方法, 导致对拥塞到来和拥塞恢复反应较慢, 进而导致在没有拥塞的时候都有较高的可能性发生不必要的分组丢弃, 同时也使网络吞吐量恢复较慢。提出一种新的称为 MRED 的算法, MRED 的主要目标是优化用来进行拥塞避免的平均队列长度的计算, 进而提高对网络负载变化的响应速度, 降低分组丢弃概率。理论分析和仿真结果表明 MRED 算法提高了响应速度、吞吐量、队列长度稳定性, 降低了时延, 并表现出良好的鲁棒性。

关键词: 主动队列管理; 拥塞控制; 负载变化

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2010)12-0111-04

An Improved Random Early Detection Algorithm

LÜ Zhao-qi

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: The IETF has addressed the problem of congestion control by advocating the deployment of active queue management mechanisms, such as RED, in the network. But the calculation of the average queue length in the random early detection (RED) algorithm results in a poor response time when the congestion is coming or RED recovers from congestion and causes packets to be unnecessarily dropped with a high probability even during the congestion free period, resulting in slow recovery of the throughput. Propose a new algorithm, called Modified RED (MRED). The primary objective of MRED is to optimize the calculation of the average size of the queue used for congestion avoidance and to consequently reduce the time required by RED to load changing and reduce the total loss of packets at the queue. Analysis and simulation results have been used to show that the proposed MRED algorithm significantly improves the response time, throughput, stability of the queue length and delay of RED gateways and shows good robustness.

Key words: active queue management; congestion control; load changing

0 引言

随着网络技术和多媒体技术的飞速发展, Internet 日益呈现出复杂、异构等特点, 网络用户不断增长, 新业务不断涌现, 网络承载的数据量越来越大, 负荷越来越重, 频频出现网络拥塞的问题, 而用户对网络的服务质量(Quality of Service, QoS)要求日益增强。队列管理是对网络传输节点中队列缓冲资源的管理, 在网络传输控制中发挥着重要作用, 是网络服务质量(QoS)控制的核心技术之一, 也是实现网络拥塞控制的重要手段, 一直受到广大研究者的关注。

传统的路由器中最常用的队列管理策略是尾丢弃(Tail Drop), 即当队列满时, 如果还有分组到来则丢弃

队尾的分组。在 TCP 网络中, 尾丢弃策略具有满队列、lock-out、TCP 全局同步等缺陷, 导致分组突发性的大量丢弃, 部分流独占带宽使得公平性较差等问题。

为了克服这些缺陷而提出的主动队列管理(Active Queue Management, AQM)在队列满之前标记或丢弃分组, 这样端节点便可以在队列溢出前对拥塞做出反应。AQM 算法逐渐成为研究热点, 其中具有代表性的算法有: RED^[1]算法、Blue^[2]算法、REM^[3]算法、AVQ 算法, 以及与现代控制理论相结合的 PI 控制器^[4]算法等。RED 被 IETF 推荐为在下一代网络中使用的主动队列管理算法, 自然受到研究者的关注, 产生了不少 RED 的改进算法, 如: Adaptive RED^[5], SRED^[6]等。

1 RED 算法分析

RED 算法是 Floyd 和 Jacobson 在文献[1]中提出的一种 AQM 算法, 采用完全共享策略和单队列结构

收稿日期: 2010-04-04; 修回日期: 2010-07-18

基金项目: 国家 863 计划项目(2009AA01Z202)

作者简介: 吕兆齐(1982-), 男, 山东聊城人, 硕士研究生, 研究方向为基于 IP 的下一代通信网络。

对到达的分组进行排队, 根据平均队列长度 avg_q 来检测即将到来的拥塞, 并以 avg_q 的值决定丢弃或标记分组。

该算法利用类似低通滤波器带权值的方法计算平均队列的大小:

$$avg_q = (1 - w_q) \times avg_q + w_q \times q \quad (1)$$

如果平均队列长度 avg_q 小于最小阈值 min_{th} , 则分组直接进入缓冲队列; 如果平均队列长度 avg_q 位于最小阈值 min_{th} 和最大阈值 max_{th} 之间, 则按如下式子计算分组丢弃概率 P_b :

$$P_b = max_p \times (avg_q - min_{th}) / (max_{th} - min_{th}) \quad (2)$$

其中 max_p 是最大丢弃(或标记)概率。

$$P_a = P_b / (1 - count \times P_b) \quad (3)$$

其中 $count$ 表示上次丢弃分组后收到的分组数目, 在算法的实际实现中, 以概率 P_a 丢弃或标记分组, 这样可以平滑分组的标记(或丢弃)过程; 如果平均队列长度 avg_q 大于最大阈值, 则强制性地丢弃到达的分组。

RED算法的优点是实现机制比较简单, 消除了全局同步现象, 提高了系统对突发流量的适应性, 随机选择分组丢弃的机制能够使用户数据分组丢弃概率与占用的带宽成正比, 从而保证了一定的公平性。正是由于 RED 算法的优越性能, 并且在提出主动队列管理的研究时, RED 是唯一能够实现其技术目标的算法, Braden 等人在 RFC 2309^[7] 中将其推荐为 AQM 的唯一候选算法, 并被 IETF 推荐为在下一代网络中进行拥塞控制和 QoS 控制的 AQM 算法。

RED 算法也存在一定的问题, 对控制参数非常敏感, 它的性能在很大程度上取决于 max_p , w_q 等参数的设置, 而要正确选取这些参数需要考虑实际网络中的分组长度、数据速率、突发性以及允许的时延等因素, 因此难以优化参数设定。RED 算法对网络负载的变化也很敏感, 在用户流增大的情况下, RED 的性能会急剧下降, 在特定的网络负载情况下仍然会出现多个 TCP 流的同步, 造成队列振荡、吞吐量降低和时延抖动加剧^[8]。在 Internet 中有很多对响应时间敏感的短期数据流(如 http), 要求在路由器中使用的主动队列管理算法有较高的响应速度, 而 RED 算法对响应时间考虑较少。此外, 基于单队列结构的 RED 算法没有考虑不同等级服务之间、不同用户流之间的差别, 难以提供有效的公平性保障, 也就难以支持区分服务。

从 RED 算法的计算平均队列长度的公式(1)中可以看到, RED 算法计算 avg_q 所采用的低通滤波机制过滤掉了当前实际队列长度中高频部分信息, 并且难以反映实际队列长度的变化趋势。由于权重 w_q 是一个固

定不变的参数, 当 w_q 设置的较小时, 平均队列长度中包含当前实际队列长度的信息就较少, 必然导致对当前实际队列长度变化的不敏感, 不能及时探测拥塞, 如果 w_q 较大, 平均队列长度就能更多地反映当前实际队列长度的变化, 但又带来了抖动的问题。

针对 RED 算法存在的对控制参数过于敏感, 配置难以优化的问题, 文献[5]提出的 Adaptive RED 算法自动配置 w_q 、 min_{th} 、 max_{th} 等参数, 并根据平均队列长度动态调整最大丢弃(或标记)概率 max_p , 提高了 RED 算法的性能稳定性, 但仍存在对负载变化响应慢的问题, 因为在 ARED 算法中用来计算平均队列长度的权重 w_q 也是固定不变的, 而最大丢弃概率又是根据平均队列长度计算的, 导致 ARED 算法仍然存在对负载变化响应慢的问题。

LPF/ODA 算法^[9]是一种提高响应时间的 RED 改进算法, LPF/ODA 的主要改进是减少从长期拥塞恢复所需时间。LPF/ODA 仍旧采用 RED 算法计算平均队列长度的方法, 但当检测到在一定时期内瞬时队列长度处于较低的水平而平均队列长度高于最小阈值 min_{th} , 就认为长期拥塞结束, 把平均队列长度减半, 以指数方式减小平均队列长度, 从而提高对拥塞结束的响应, 减少不必要的分组丢弃。但 LPF/ODA 算法仅考虑在拥塞结束时的情况, 不够全面, 且其减小平均队列长度的方法值得商榷。

2 RED 改进算法

在 RED 算法中平均队列长度的计算是整个算法的基础, 其中权重 w_q 的设置非常关键, w_q 的大小决定对突发流量的容忍程度。RED 算法根据平均队列长度标记(或丢弃)分组的优点是, 平均队列长度变化的很缓慢, 不会突然增大而导致大量分组丢弃, 提高了系统对于突发流量的适应性^[10]。 w_q 越小, 能容忍的突发流量就越大, 但是如果 w_q 设置的太小, 平均队列长度对实际队列长度变化的反应就很慢, RED 就不能在拥塞发生的初期就发现拥塞, 也就达不到 RED 将总体队列长度维持在一个较低水平的目的, 同样也不能使平均队列长度在拥塞结束后快速减小从而减少不必要的分组丢弃。进而导致经常出现网络负载增大, 拥塞来临时, 平均队列长度较低, 而实际队列长度迅速增大, 而由于权重 w_q 设置的太小, 平均队列长度增长缓慢, 而实际队列长度早已超过最大阈值, 甚至发生满队列, 造成强制丢包; 另外一种情况是, 长期拥塞结束时, 实际队列长度迅速减小, 低于最小阈值, 处于空闲状态, 而此时平均队列长度由于变化缓慢, 仍旧保持较大的队列长度, 因此仍在以一定的概率标记或丢弃分组, 造成

没有必要的丢弃,造成吞吐量降低。这两种情况同时还存在队列长度剧烈变化的问题,使得时延预测变得难以估计,QoS 控制实现更加困难。而若用实际队列长度代替平均队列长度来决定分组标记或丢弃的概率,又会造成剧烈的抖动。所以一个较好的解决办法就是在适当的时候动态调整权重 w_q 的大小,使之能够克服这样的缺陷。

改进的 RED 算法 MRED 仍然根据平均队列长度 avg_q 决定标记或丢弃概率,但通过检测实际队列长度的大小和变化趋势来动态调整权重的大小。MRED 算法仅在长期拥塞到来或结束时调整权重的大小,其它情况下仍然保持原来的权重不变,所以仍然保留了 RED 算法能够容纳突发流量的优点。

文中提出的 MRED 算法仅在平均队列长度的计算上有所改进,MRED 算法仅在长期拥塞的开始和结束通过调整权重 w_q 来调整平均队列长度 avg_q 的计算方式:

记录一定时间 $freeze_time$ 内平均队列长度 avq_i ($i = 1, \dots, N$),实际队列长度 q_i ($i = 1, \dots, N$)。

当网络负载增大,长期拥塞到来时的特征表现为,实时队列长度迅速增长,并在一定的时期内保持较高的队列长度,平均队列长度维持在相对较低的水平,即 $q_i > avq_i$ ($i = 1, \dots, N$),这时应该迅速加大权重: $w = \min\{w * \alpha, w_{max}\}$ ($\alpha > 1$, w_{max} 由最小阈值与路由器允许的突发流量决定),使平均队列长度迅速增大,使平均队列长度更多的包含当前队列长度的变化信息,减少对拥塞的响应时间,减少满队列从而造成强制丢弃的情况,平滑队列长度,从而提高吞吐量,也使时延抖动减小。当网络流量平稳时,即实际队列长度在平均队列长度附近上下浮动时,迅速减小权重: $w = w/\beta$ ($\beta > 1$) 到原来的值 w_{min} 。

当负载减轻时,实际队列长度迅速减小,平均队列长度会缓慢的减小,若平均队列长度高于 RED 算法的最小阈值 min_{th} ,实际队列长度迅速下降并在一定的时期内保持较低的队列长度,即一定时间内 $q_i < avq_i$ ($i = 1, \dots, N$),这时应该迅速加大权重: $w = \min\{w * \alpha, w_{max}\}$ ($\alpha > 1$),使平均队列长度迅速降低,减少不必要的分组标记或丢弃。而当平均队列长度降至最小阈值之后,恢复原来 RED 算法的平均队列计算方法,即权重 w 恢复到原来较小的值 w_{min} 。若负载减轻时,即若在一定时间内, $q_i < avq_i$ ($i = 1, \dots, N$),但平均队列长度小于最小阈值,则权重不改变。

其他情况下,仍然按原来 RED 算法计算平均队列长度的方法,使用一个较小的不变的权重 w_{min} 来计算平均队列长度。

3 仿真实验

文中采用由加州大学伯克利分校(UC Berkeley)开发的网络模拟软件 NS-2^[11]进行仿真,与 RED 算法进行对比,验证 MRED 的性能。仿真网络拓扑结构如图 1 所示。

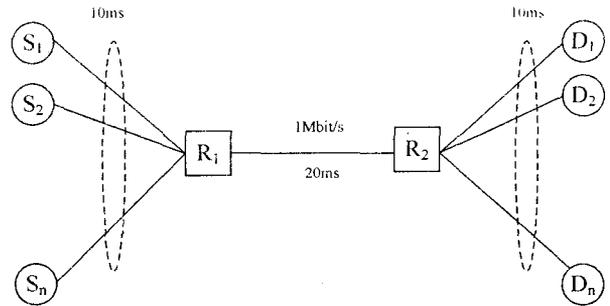


图 1 仿真网络拓扑结构

R_1 和 R_2 是路由器,节点 R_1 与节点 R_2 之间的链路为瓶颈链路, $S_1 \sim S_n$ 为发送端, $D_1 \sim D_n$ 为接收端, $S_1 \sim S_n$ 到路由器 R_1 的链路和 $D_1 \sim D_n$ 到 R_2 的链路传输时延为 10ms,采用 DropTail。瓶颈链路的带宽为 1Mb/s,链路时延为 20ms,在瓶颈链路上采用 RED 或 MRED 算法。

模拟场景为负载变化网络,在 0~1s 时,随机启动 10 个 FTP 连接,在 20 s 时加入 10 个 FTP 连接,在 40 s 时退出 20 个 FTP 连接,总共仿真时长为 80s。缓冲区长度为 100 packet,链路时延为 20ms,瓶颈带宽为 1Mb/s。RED 算法的参数设置为: $min_{th} = 20$, $max_{th} = 60$, $max_p = 0.1$, $w_q = 0.002$ 。MRED 算法的参数设置为: $min_{th} = 20$, $max_{th} = 60$, $max_p = 0.1$, $freeze_time = 0.05s$ 。仿真结果见图 2~图 4。

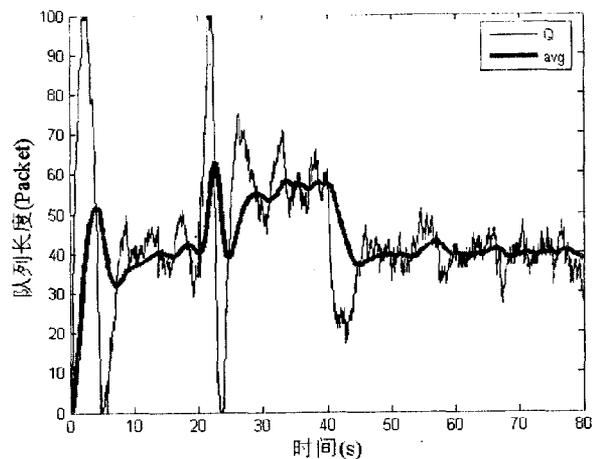


图 2 RED 瞬时队列(Q)和平均队列(avg)长度变化

仿真结果图 1 表明,在负载变化的网络中,RED 出现队列大幅震荡,说明对于 TCP 流仍然存在全局同步现象。这是因为 RED 算法在设置权重较小时,对负

载变化的响应比较迟缓,这一点在大时滞网络中更为明显^[12]。而动态调节权重 MRED 算法能很好地跟随实际队列的变化,在拥塞发生初期就对其进行控制,在长期拥塞结束后能减少不必要的丢包,并能有效地减少拥塞时间。在仿真实验中的 0s, 20s, 40s 处, MRED 明显比 RED 收敛速度快。

对比图 1 和图 2 可知, MRED 算法比 RED 更多时间地把队列长度控制在最小阈值和最大阈值之间, 并且减少了满队列和空队列的时间, 队列长度震荡较小, 很明显 MRED 有更低的时延和时延抖动, 这使得时延预测和控制更易实现。

由于 MRED 算法能够及时响应拥塞的结束, 减少了不必要的丢包, 从图 4 中可以看出, MRED 提高了吞吐量。

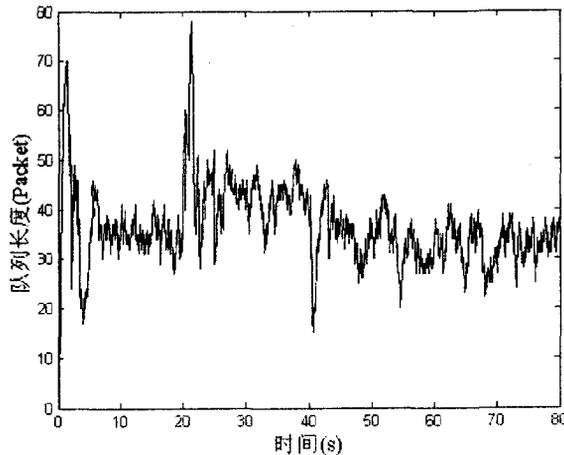


图 3 MRED 瞬时队列长度变化

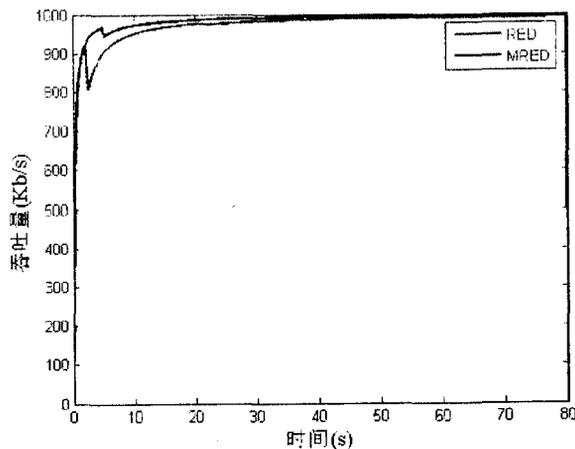


图 4 瓶颈链路的吞吐量

4 结束语

文中提出了一种新的 RED 改进算法 MRED, 其主要方法就是利用 RED 实际队列长度的变化趋势来动

态调整 RED 平均队列长度的计算。使得在拥塞发生初期能更及时地作出反应, 同时在拥塞结束后能使平均队列长度更接近实际队列, 从而减少不必要的丢包, 提高了吞吐量。仿真实验证明, MRED 保留了 RED 算法控制机制简单和适应突发流量的优点, 而且相比 RED 能有效地减少队列拥塞时间, 在拥塞发生时和拥塞结束后在吞吐量、时延、时延抖动、丢包率等几个指标上有更好的性能。

参考文献:

- [1] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance[J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413.
- [2] Feng W, Shin K, Kandlur D, et al. The blue active queue management algorithms[J]. IEEE/ACM Transactions on Networking, 2002, 10(4): 513-528.
- [3] Athuraliya S, Li V H, Low S H, et al. REM: Active queue management[J]. IEEE Network, 2001, 15(3): 48-53.
- [4] Hollot C, Misra V, Towsley D. On designing improved controllers for AQM routers supporting TCP flows[C]//In: Proceedings of the INFOCOM 2001. Alaska: IEEE Computer Society, 2001: 1726-1734.
- [5] Floyd S, Gummadi R, Shenker S. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management[R]. Berkeley: AT&T Center for Internet Research at ICSI, 2001.
- [6] Ott T J, Lakshman T V, Wong L H. SRED: Stabilized RED [C]//Proc. of IEEE INFOCOM'99. New York: IEEE, 1999: 1346-1355.
- [7] Braden B, Clark D, Crowcroft J, et al. Recommendations on queue management and congestion avoidance in the Internet [S]. RFC 2309, 1998.
- [8] Arpaci M, Copeland J A. An adaptive queue management method for congestion avoidance in TCP/IP Networks[C]//Proc. of IEEE GLOBECOM'00. San Francisco, USA: [s. n.], 2000.
- [9] Zheng Bing, Atiquzzaman M. Low pass filter/over drop avoidance (LPF/ODA): an algorithm to improve the response time of RED gateways[J]. Int. J. Commun. Syst., 2002, 15(10): 899-906.
- [10] 陈柏秀, 谭献海, 朱效稳. 自相似流量下自适应 AQM 算法性能研究[J]. 计算机技术与发展, 2009, 19(9): 28-31.
- [11] The Network Simulator - ns-2 [EB/OL]. 2007-03-10. <http://www.isi.edu/nsnam/ns>.
- [12] 任丰原, 林 闯, 任 勇, 等. 大时滞网络中的拥塞控制算法[J]. 软件学报, 2003, 14(3): 503-511.