基于参数估计的随机早期探测改进算法

石志强, 吴志美, 梁 进
(中国科学院软件研究所, 北京 100080)

摘 要: 流量控制是互联网有效运行的必要条件. 随机早期探测 RED(Random Early Detection) 是一种基于整条链路的积极队列管理技术, 它能显著地提高网络的利用率、稳定性和公平性, 是 IETF 倡导使用的流量控制技术. 它由 Hashem 首先提出雏型, Floyd 对之作出了重大改进, 形成了较为完善、实用的 RED 技术, 为了在应用中增大 RED 的适应范围, Fang 又提出基于参数自适应的 RED. 由于现有方法未能稳定、准确地刻画网络流量特征, 在流量特性差异较大时, 以上方法均难以有效控制网络流量. 在本文中, 作者首先给出了估计流量特性的解析方法, 该方法较参数自适应具有更好的稳定性和准确性; 并据此提出了一种 RED 改进算法, 最后通过动态仿真, 证明该改进算法提高了 RED 的适应性, 降低了网络时延, 减小了时延抖动.

关键词: 随机早期探测; 队列管理; 拥塞控制; 互联网; 传输控制协议
中图分类号: TP393 文献标识码: A 文章编号: 0372-2112 (2000) 11A-0088-04

The Improvement of RED Based on Estimating Parameter

SHI Zhi-qiang, WU Zhi-mei, LIANG-Jin
(The Institute of Software of Chinese Academy of Science, Beijing 100080, China)

Abstract: The flow control is vital for Internet. Random Early Detection(RED) is an active queue management based on the link, which improves efficiency, stability and impartiality, and is also advocated by IETF. It is presented by Hashem, then had been improved into rather perfection and practicality by Floyd, and further had been appended with the self-configuring by Fang to be more adaptive. After simulation, we find that both RED and the self-configuring can't effectively control the network flow while confronted with large variety of flows, because they can't figure the property of network flows accurately and stably. In this paper, we give the analytic way to estimate the parameter of RED, which is more accurate and stable than the self-configuring, and then present the amendatory algorithm of RED, which is demonstrated by simulation to be more adaptive, with lower transfer delay and jitter.

Key words: RED; queue management; congestion control; Internet; TCP

1 引言

从互联网诞生, 流量控制就一直伴随着互联网的发展, 为此, IETF 从终端设备和网络设备两个方面, 研究公平、高效利用互联网资源的技术. V. Jacobson[1] 提出了 TCP 流量控制的基本策略, 奠定了端到端流量控制的基础, 其后, 众多学者进一步提出了 New Reno TCP[8], SACK TCP[7] 等增强机制, 提高了网络资源利用率. 但随着互联网的发展和网络规模迅速扩大, 端到端的流量控制已难以保持网络高效公平地运行, 这既由于许多流量不遵循端到端流量控制, 也由于终端设备感知和控制网络流量的能力较弱. Hashem[2] 指出随机包丢弃(Random Drop) 和尾丢弃(Drop Tail) 易造成网络流量的全局同步振荡, 使网络资源利用率下降, 为此提出了随机早期探测(RED) 的雏型. 其核心是当队列长度超过某个固定长度后, 以某个固定的几率丢弃到达的报文. S. Floyd[3] 对 RED 原型做了两点重大改进: 指数平滑队列长度和实际丢包率自适应. RED 采用随机丢包, 因而具有公平特性, 即对任何连接的丢包量都正比于其流量大小, 根据这个特性, S. Floyd 在文[4] 和[5] 中提出了利用 RED 实现带宽公平分配和发现非 TCP 友好数据流的机制. 在仿真和实际应用中, RED 都表现出较好的网络性能, 得到 IETF[9] 的大力推荐, 也逐渐被大多数设备制造商采用. Fang[6] 研究了 RED 在实现中的问题, 发现实际丢包率自适应方法不能适应网络流量的变化, 于是提出了参数自适应方法, 增大 RED 的适应范围. 作者发现依靠参数自适应方法在网络流量差异较大时, 仍无法有效控制网络流量, 并提出了基于参数估计的 RED 改进算法.

2 随机早期探测及其现状

2.1 随机早期探测简介

S. Floyd[3] 提出的 RED 采用权重为 w_q 的指数平滑方法计算平均队列长度 avg. 设定平均队列 avg 的上下限为 max_{th} 和

收稿日期: 2000-06-10; 修回日期: 2000-08-30
基金项目: 国家重点基础研究发展规划(973 计划) 项目(No. G1998030407) ; 国家高技术研究发展计划(863 计划) 课题(No. 863-317-01-10-99)

min_{th}, 当 avg 小于 min_{th}, 不丢弃到达报文; 当 avg 大于 max_{th}, 丢弃所有到达报文, 这类丢包称为强迫性包丢弃; 当 avg 在 min_{th} 和 max_{th} 之间时, 以概率 $p_a/(1-count*p_a)$ 丢弃到达的报文, 这类丢包称为非强迫性包丢弃, 其中 $count$ 为连续未丢包数, $p_a=max_p*(avg-min_{th})/(max_{th}-min_{th})$, max_p 为最大丢失率; 因为 p_a 是一个很小的正数, $1-count*p_a$ 很接近于 1, 所以 p_a 是丢失率的主要决定因素, 对 p_a 的修正用于避免连续丢包, 故以后仅考虑 p_a. 由于 p_a 是随着当前队列长度的变化而变动的, 该机制的最大适应范围为丢包率在 $[0, max_p]$ 的流量, 当然, 实际范围要小得多. Fang[6] 用实验证明了参数 max_p 对 RED 的性能有重大影响, 并提出了估计 max_p 的自适应算法, 其核心是当平均队列长度大于 max_{th} 时, 参数 max_p 增大三倍, 当平均队列长度小于 min_{th} 时, 参数 max_p 减半, 并通过网络仿真软件"NS", 验证了算法的有效性.

2.2 网络拥塞的分类

首先本文中网络拥塞是指网络交换设备的包到达速度大于其发送速度的现象, 根据成因的不同, 分为突发性拥塞和持续性拥塞. 突发性拥塞是指由于数据通信的突发性特点, 短暂、偶然地包到达速度大于发送速度, 表现为平均队列长度通常小于 min_{th}. 持续性拥塞是指由于互联网无接入控制机制, 导致长期、持续地包到达速度大于发送速度, 表现为平均队列长度通常大于 min_{th}. 对于采用 RED 技术的网络设备, 进一步把持续性拥塞又分为轻微拥塞和严重拥塞, 轻微拥塞指通过调整 max_p, 可避免强迫性包丢弃; 严重拥塞指无论如何调整参数 max_p, 都无法避免强迫性包丢弃, 从而导致网络性能恶化.

2.3 丢失率的理论计算

Floyd[5] 给出了 TCP 连接的带宽与包丢失率的关系

$$B_i = MSS_i * C/RTT_i * \sqrt{p_i} \tag{1}$$

其中 B_i 是 TCP 连接 i 占用的带宽, MSS_i 是连接 i 的平均包长度, RTT_i 是连接 i 的往返时延, p_i 是连接 i 的丢包率, 常数 C 为 1.22. 若网关处有 n 个 TCP 连接, 假定它们的丢包率相同, 网关处的总带宽为 B, 则能保持流量平稳的包丢失率为

$$p = \left(\sum_{i=1}^{n} \frac{MSS_i}{RTT_i} * \frac{C}{B} \right)^2 \tag{2}$$

对于采用 RED 的网关, 若已知使得网络流量处于稳定状态的丢包率 p, 则根据线性假设, 可粗略估计其参数 max_p 的一个适当值为 $2*p$.

2.4 参数自适应的随机早期探测仿真结果

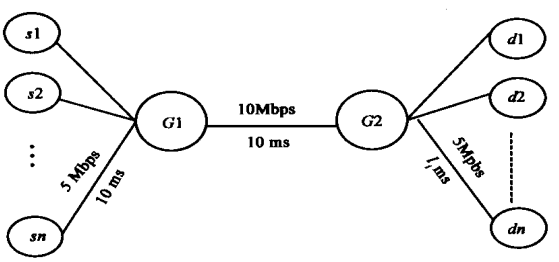


图 1　仿真环境

利用网络仿真软件"NS"(作者增加了文[6]中的自适应算法)仿真参数自适应的 RED 在网络处于轻微拥塞和严重拥塞时的性能. 仿真环境如图 1, 网关 $G1$ 和 $G2$ 采用文[6]参数自适应的 RED, 参数 w_q 为 0.002, min_{th} 为 30, max_{th} 为 170, 采用基于包的队列管理模式; 节点 d_i 与网关 $G2$ 间的时延 l_i 以及 TCP 连接数 n 是可变的. 案例 a 中, 网络处于轻微拥塞, TCP 连结数 n 为 40, 节点 d_i 与网关 $G2$ 间的时延均分为 5 组, 分别为 50ms、100ms、150ms、200ms、250ms, 假设网关 $G1$ 的平均队列长度为 80, 则根据公式(2)可得适当的丢包率为 0.0128, 参数 max_p 的一个适当值为 0.0256. 在案例 b 中, 网络处于严重拥塞, TCP 连结数 n 为 180, 节点 d_i 与网关 $G2$ 间的时延也均分为 5 组, 时延分别为 350ms、400ms、450ms、500ms、550ms, 同样计算可得参数 max_p 的一个适当值为 0.0643. 仿真长度为 800s, 在 50s 时启动参数自适应算法[6], max_p 的初值为 0.001, 仿真结果如图 2. 图 2(a)为案例 a 的平均队列长度随时间变化的曲线, 其总平均队列长度为 91.75, 可见, 在轻微拥塞时, 平均队列保持在一个过大的长度上, 从而导致网络延迟不必要增大, 其原因是 max_{th} 过大. 图 2(b)为案例 b 的平均队列长度随时间变化的曲线, 其非强迫性包丢失率为 0.019, 强迫性包丢失率为 0.011, RED 失效的关键原因在于 max_{th} 过小. 这样, 就处于两难的境地. 为了克服这个缺陷, 先给出一种估计参数 max_p 的解析方法.

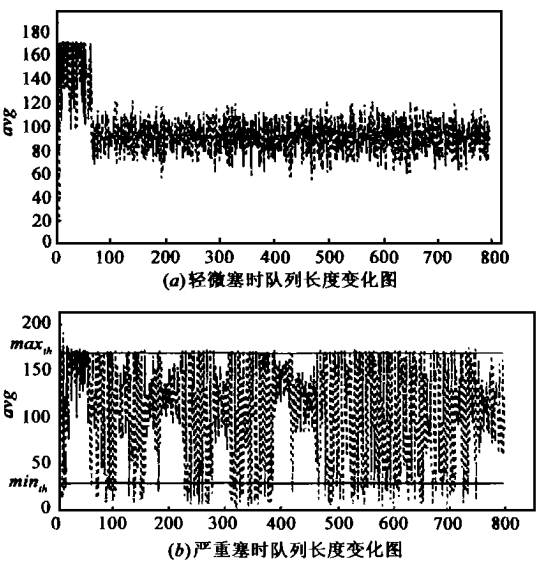


图 2　基于参数自适应的 RED 仿真结果

3　估计参数 max_p 的解析方法

图 3 是队列变化曲线的几个典型示意图, 曲线 a 的 max_p 过大, 带宽利用不足; 曲线 e 的 max_p 过小, 频繁出现强迫性包丢弃; 曲线 d 的 max_p 较小, 导致较长的网络时延; 曲线 c 的 max_{th} 较小, 导致带宽利用不足和频繁的强迫性包丢弃; 曲线 b 是 RED 处于最优运行状态. 对于曲线 a, 以时刻 x_2 为一个周期的开始, 时刻 y_2 为一个周期的结束, 周期长度记为 t_r; 在一个周期中时刻 z_2 与时刻 x_2 之间无包丢失的时间长度记为

t_w; 在一个周期内由于无包发送而浪费的带宽占总带宽的比例为 C_w(该数据在文[3] 的 RED 算法中也要求采集), 在一个周期内平均队列长度的最大值为 max_r.

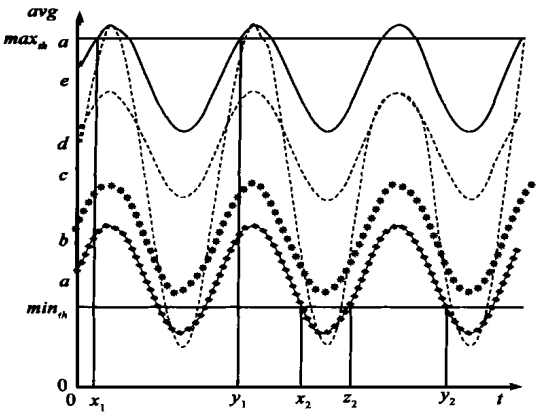


图 3 不同参数 max_p 对应的队列长度变化图

引理 1 对于 RED 的网关, 参数 max_p 的当前值为 max_{pc}, 若出现如图 3 曲线 a 的情况, 且 $max_r < max_{th}$, 则保持流量平稳的丢包率 p_n 为

$$p_n = (1 - C_w)^2 * \frac{t_r - t_w}{t_r} * \frac{max_r - min_{th}}{max_{th} - min_{th}} * \frac{max_{pc}}{2} \qquad (3)$$

证明 当网关出现曲线 a 的情况, 可知其最大丢包率为 $\frac{max_r - min_{th}}{max_{th} - min_{th}} * max_{pc}$; 由无包丢失的长度为 t_w, 整个周期的长度为 t_r, 可得当前的平均丢包率 p_c 为 $\frac{1}{2} * \frac{t_r - t_w}{t_r} * \frac{max_r - min_{th}}{max_{th} - min_{th}} * max_{pc}$; 又设当前浪费带宽为 B_w, 当前有效利用带宽为 B_c, B_n 为链路总带宽, 则由 $B_n = B_w + B_c$, 可得 $B_c = (1 - C_w) * B_n$, 根据公式(2), 化简可得公式(3).

对于曲线 e, 以时刻 x_1 为一个周期的开始, 时刻 y_1 为一个周期的结束, 在一个周期内收到的数据包个数为 num_{recv}, 丢失的数据包个数为 num_{loss}, 平均队列长度的最小值为 min_r.

引理 2 对于 RED 的网关, 参数 max_p 的当前值为 max_{pc}, 若出现如图 3 曲线 e 的情况, 且 $max_r > min_{th}$, 则保持流量平稳的丢包率 p_n 为

$$p_n = num_{loss} / num_{recv} \qquad (4)$$

证明 对于图 3 中的曲线 e, 由于无带宽浪费, 当前丢包率就是使网络流量平稳的丢包率, 故可得公式(4)

定义事件 A 为平均队列连续三次小于 min_{th} 且 $max_r < max_{th}$; 事件 B 为平均队列连续三次大于 max_{th} 且 $min_r > min_{th}$; 事件 C 为平均队列连续三次小于 min_{th} 但 $max_r > max_{th}$; 事件 D 为平均队列连续三次大于 max_{th} 且 $min_r > min_{th}$.

根据引理 1 和引理 2, 可得以下两个调整参数 max_p 的算法:

算法 1 当事件 A 出现, 则根据公式(3) 调整参数 max_p 为 $2 * p_n$.

算法 2 当事件 B 出现, 则根据公式(4) 调整参数 max_p 为 $2 * p_n$.

4 RED 改进算法

4.1 算法 3: 减少网络时延

对于轻微拥塞, 设参数 max_p 取值在 $[max_{pl}, max_{ph}]$ 内时都能满足公式(2)的丢失率要求, 首先, 在 RED 中添加两个定时器, 参数检测定时器 $T1$ 和延迟定时器 $T2$. 在 $T1$ 期间, 队列长度的最大值为 max_{rc}, 最小值为 min_{rc}, 定时器 $T2$ 用于等待参数改变发生效果, 仿真中 $T1$ 为 10s, $T2$ 为 3s.

引理 3 采用 RED 的网关, 若在 $T1$ 时间内, 有 $max_{rc} < max_{th}$ 且 $min_{rc} > min_{th}$, 当前参数 max_p 为 max_{pc}, 则使得网络流量平稳的参数 max_p 的最优值 max_{ph} 为

$$max_{ph} = \frac{min_{rc} + max_{rc} - 2 * min_{th}}{max_{rc} - min_{rc}} * max_{pc} \qquad (5)$$

证明 采用线性估计, 可得当前丢包率为

$$p_c = \frac{1}{2} * \left(\frac{min_{rc} - min_{th}}{max_{th} - min_{th}} + \frac{max_{rc} - min_{th}}{max_{th} - min_{th}} \right) * max_{pc}$$

假设队列长度的振荡幅度不变, 队列长度的最小值为 min_{th}, 最大值为 $min_{th} + max_{rc} - min_{rc}$, 参数 max_p 为 max_{ph}, 则可得其包丢失率为

$$p_n = \frac{1}{2} * \frac{max_{rc} - min_{rc}}{max_{th} - min_{th}} * max_{ph}$$

由于两种情况, 网络流量均处于稳定状态, 故有 $p_c = p_n$, 化简即可得公式(5).

定义事件 E 为在 $T1$ 时间内, $min_{rc} > min_{th}$ 且 $max_{rc} < max_{th}$; 事件 F 为在 $T1$ 时间内, $max_{rc} > max_{rn}$ 且 $min_{rc} < min_{rn}$; 事件 G 为在 $T1$ 时间内, $max_{rc} < max_{rn}$ 且 $min_{rc} < min_{rn}$; 事件 H 为在 $T1$ 时间内, $max_{rc} > max_{rn}$ 且 $min_{rc} > min_{rn}$

算法 3 当事件 E 发生时, 采用公式(5) 调整参数 max_p.

4.2 算法 4: 增大 RED 的适用范围

首先, 平均队列长度振荡是由数据流的固有随机波动和周期性波动构成的, 固有的随机波动难以避免的. 周期性波动是由于实际丢包率自适应方法的实际丢包率随时间周期波动, 导致包到达速度周期变化, 这种周期波动受网络时延和 TCP 连接数影响. 但这两个因素都无法改变, 基于此, 本文提出的算法 4 就是减小实际丢包率的变化幅度. 添加参数 $\gamma > 1$ 来限制实际丢包率的变化范围, 当 avg 在 min_{th} 和 max_{th} 之间时, 现有 RED 的实际丢包率为 $p_c = \delta * max_p$, 其中 $\delta = \frac{avg - min_{th}}{max_{th} - min_{th}}$, 现将之修改为 $p_c = \delta' * max_p$, 其中 $\delta' = \frac{1}{2} * \left(1 - \frac{1}{\gamma}\right) + \frac{\delta}{\gamma}$. 在 RED 中设定其初始状态为 RED_N, 当事件 C 或事件 D 满足时, 进入状态 RED_E.

算法 4 当状态为 RED_N, 事件 C 或 D 满足时, 设定 γ 为 α. 当状态为 RED_E, 事件 C 或 D 或 F 满足时, 设定 γ 为 $\alpha * \gamma$; 事件 G 或 H 满足时, 设定 γ 为 $\beta * \gamma$. (其中 $\alpha > 1$, $\beta < 1$, $\alpha * \beta > 1$)

4.3 RED 的改进算法

在综合算法 1, 2, 3, 4 后, 提出如图 4 所示的一种 RED 改

进算法.

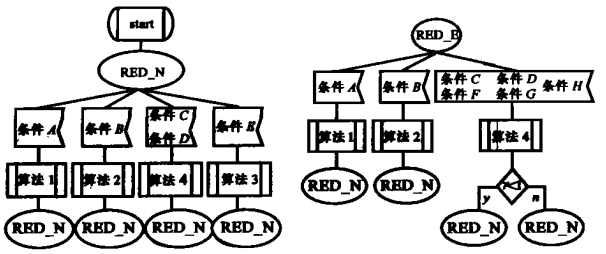
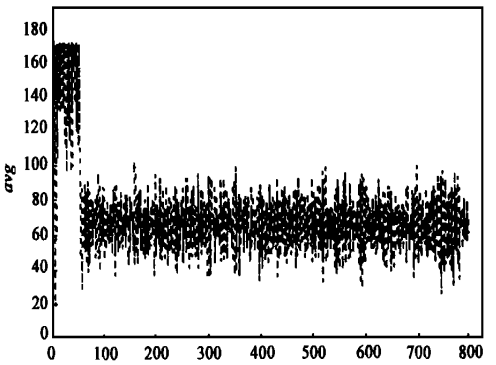
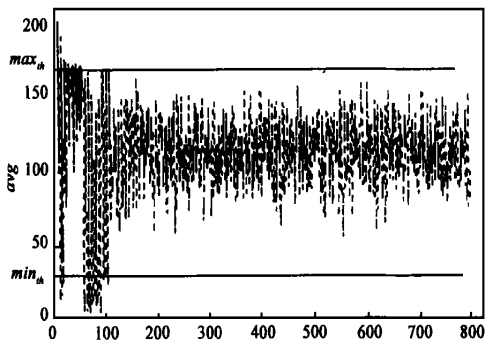


图 4　基于参数估计的 RED 改进算法



(a)轻微塞时队列长度变化图



(b)严重塞时队列长度变化图

图 5　基于参数估计的 RED 改进算法仿真结果

采用该改进算法仿真图 1, 结果如图 5. 图 5(a) 为案例 a 的仿真结果, 总平均队列长度由原来的 91. 75 减小为 65. 95. 图 5(b) 为案例 b 的仿真结果, 强迫性丢包率由原来的 0. 011 降低为 0. 00042, 非强迫性丢包率由原来的 0. 019 增大为 0. 020. 还作了进一步的仿真实验, 把案例 b 中的参数 max_{th} 降低为 120, 仿真结果是强迫性丢包率为 0. 0017, 非强迫性丢包率为 0. 021. 仿真表明改进算法显著地增大了 RED 的适应范围, 降低了队列振荡幅度, 减小了网络时延抖动.

5　结束语

由于网络流量的复杂性和时变性, 难以实时精确地估计适宜的丢包率, 采用自适应方法如实际丢包率自适应和参数自适应, 能在一定条件下保持网络有效运行, 但同时也带来延时、延时抖动增大等问题, 而且其适应范围也较小. 本文在丢包率自适应的基础上, 通过估计当前适宜的丢包率, 合理增大参数 max_p, 减小丢包率的变化范围, 从而达到减小网络时延和时延抖动, 避免强迫性包丢弃. 实时性和稳定性常常是相互对立的, 本文仅仅提出了基于四个基本算法的一种有效的、可能的改进, 如何利用这四个基本算法, 兼顾实时性和稳定性仍有待于进一步研究.

参考文献:

[1]　V. Jacobson. Congestion avoidance and control [A]. In Proc. ACM SIGGCOMM88, August 1988: 314- 329.

[2]　Hashem, E. Analysis of random drop for gateway congestion control [A]. Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge, MA, 1989: 103.

[3]　S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance [A]. ACM/ IEEE Transactions on Networking, August 1993.

[4]　Floyd, S., Fall, K., and Tieu, K. Estimating arrival rates from the RED packet drop history [A], ftp: // ftp. ee. lbl. org/ papers/ red-dropping. ps.

[5]　Floyd, S., and Fall, K. Promotin the use of end-to-end congestion control in the internet [A]. IEEE/ ACM Transactions on Networking, August 1999.

[6]　Fang, W. C., Kandlur, D. D., Saha, D., Shin, K. G. A self-configuring RED gateway [A]. INFOCOM' 99, March 1999.

[7]　M. Mathis. TCP Selective Acknowledgment Options, RFC2018 [S]. October 1996.

[8]　S. Floyd and T. Henderson. The newReno modification to TCP' s fast recovery algorithm [A]. April 1999.

[9]　B. Braden. Recommendations on queue management and congestion avoidance in the internet [A]. Internet RFC 2309, April 1998.

作者简介:

石志强　中国科学院软件研究所博士生, 主要研究方向网络流量控制, 网络仿真, 服务质量保证, 网络测试.

吴志美　中国科学院软件研究所多媒体通信和网络工程研究中心主任, 主任研究员, 博士生导师, 中国电子学会通信专业委员会委员, 中国计算机学会网络和数据专业委员会委员, 中国通信学会通信信号与信息处理委员会委员; 主要研究领域有多媒体通信, 网络融合与互通, 协议测试与验证.