

FERED:公平性增强的 RED 算法

武航星¹ 慕德俊¹ 龚贤武² 潘文平¹

(西北工业大学自动化学院 西安 710072)¹ (长安大学信息工程学院 西安 710064)²

摘要 当前大多数 AQM 机制的实现算法都更多地强调了算法的效率和稳定性而忽视了公平性。已有研究表明, TCP 的 RTT 不公平性问题和多拥塞链路环境下的不公平性问题广泛存在于众多的著名 AQM 实现算法中, 比如 RED, REM, PI 和 AVQ 等。虽然 FRED 和 Balanced RED 可以解决这些不公平性问题, 但它们的实现都需要在路由器上保留每流状态信息, 算法可扩展性存在问题。在实际网络测量试验结果的基础上, 提出了利用 IP 数据报头中的 TTL 字段信息来增强公平性的思路, 并据此对 RED 算法进行了扩展, 实现了一个公平性增强的 RED 算法(FERED)。NS2 仿真试验结果显示 FERED 可以显著增强公平性, 同时保留了 RED 算法可以很好地控制队列长度的优点, 而且 FERED 实现简单, 无需在路由器保留每流状态信息。

关键词 公平性, AQM, RED

FERED: Fairness Enhancement RED

WU Hang-xing¹ MU De-jun¹ GONG Xian-wu² PAN Wen-ping¹

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)¹

(School of Information Engineering, Chan'an University, Xi'an 710064, China)²

Abstract Most of existing AQM schemes focus on the efficiency and stability, but neglect the fairness. Some researches have shown that TCP RTT unfairness and unfairness led by multiple congestion links widely occur in many famous AQM schemes, such as RED, REM, PI, AVQ etc. Although FRED and Balanced RED can solve unfairness problem, both of them keep per-flow state in router. Based on measurement results in real Internet, a fairness enhancement RED, in which TTL field in IP header is used to improve fairness of RED, was proposed. The results in NS2 show that FERED improves fairness greatly and inherits the merit of RED to control queue length. Moreover FERED can be implemented simply and no per-flow state is needed in router.

Keywords Fairness, AQM, RED

1 引言

传输控制协议(TCP)是当前 Internet 中最广泛使用的传输协议,然而一些理论和试验研究已表明 TCP 的设计特性使得 TCP 协议在一些情况下会存在公平性问题。在文献[1]中,作者分析了 TCP 的 AIMD 拥塞窗口更新策略在具有不同 RTT 的 TCP 连接间所存在的 RTT 不公平问题,这一结论已被广泛承认。同时文献[2]中作者指出了 TCP 在多拥塞链路环境下存在的不公平性问题,经历较少拥塞链路的 TCP 数据流在与经历多个拥塞链路的 TCP 数据流竞争瓶颈链路带宽时,将处于明显的优势地位,可以获得更多的带宽资源,有时甚至可以使经历多个拥塞链路的 TCP 数据流出现“饿死”现象。从理论上来说,假设每一个瓶颈链路 i 上的丢包率为 p_i ,那么一个经过 n 个拥塞链路的数据流,其数据包被成功传送到接收方的概率为 $\prod_{i=1}^n (1 - p_i)$ 。简单地假定所有拥塞链路上的丢包率 $p_i = 0.1$,则经过 6 个拥塞链路的数据流将会有 47% 的数据包被丢弃,而对于经过单拥塞链路的数据流,其数

据包成功传送到发送方的概率为 90%。此外,在多拥塞链路环境下,在拥塞链路处的排队时延将使经历多个拥塞链路的数据流的 RTT 加大,这将使公平性进一步恶化。

AQM 机制作为端到端拥塞控制的又一重要方法自被提出以来一直是网络研究领域内的热点问题之一,也出现了很多比较有影响的 AQM 实现算法。然而这些算法的设计大多着重于算法的效率和稳定性,而忽视了算法的公平性性能。在文献[3]中,作者通过 NS2 下的仿真试验,证实了 RED^[4], REM^[5], PI^[6] 和 AVQ^[7] 这些著名的 AQM 实现算法在多拥塞链路环境下都存在着普遍和严重的公平性问题。虽然有一些 AQM 实现算法,比如 FRED^[8] 和 Balance RED^[9],可以有效地解决上述不公平性问题,但是它们的实现都需要在路由器上保留每流状态信息,算法的可扩展性存在问题。

在本文中,我们在实际网络测量试验结果的基础上,提出了利用 IP 数据报头中的 TTL 字段信息来增强公平性的思路,并据此对 RED 算法进行了扩展,实现了一个公平性增强的 RED 算法(FERED)。NS2 仿真试验结果显示 FERED 可

到稿日期:2008-04-01 本文受西安交通大学“机械制造系统工程国家重点实验室”开放基金资助。

武航星 博士,主要研究方向为网络拥塞控制和流量控制, E-mail: whx9711@163.com; 慕德俊 教授,博士生导师,主要研究领域为网络信息安全; 龚贤武 博士,主要研究方向为电子检测技术; 潘文平 博士,研究方向为网络服务质量。

以显著增强公平性,同时保留了 RED 算法可以很好地控制队列长度的优点,而且 FERED 实现简单,无需在路由器保留每流状态信息。

2 FERED

2.1 FERED 的设计动机

在文献[10]和文献[11]中,作者通过大量的实际网络中的实验测量得出近乎相同的结论:虽然一个数据流从发送方端系统到达接收方端系统所经历的路由器跳数及其往返时延之间不存在严格的线性关系,但从总体的趋势上来看,数据流所经历的路由器跳数和往返时延之间大致上还是存在着线性关系的,即数据流经历跳数越多,其往返时延很可能就越大。此外,在多拥塞链路的环境下,数据流的 RTT 中很大一部分将由拥塞链路的排队时延决定,经历的拥塞链路越多,其 RTT 必将相对越大。而且,在多拥塞链路环境下,经历多个拥塞链路的数据包其丢弃概率也将越大,这将使其处于更加不利的位置。另外,在文献[12]中,作者指出:当网络出现拥塞时,会出现数据包的丢弃,被丢弃数据包经过所有上游路由器时所占用的网络资源就被浪费掉了,这很大程度上降低了网络资源的有效利用率,使得本就十分匮乏的网络资源更加紧张。

基于上述考虑,我们认为在设计实现 AQM 机制的算法时可以将数据包所经历的路由器跳数作为一个参考,对经过路由器跳数较多的数据包进行一定的保护来增强算法的公平性性能。对经过路由器跳数较多的数据包进行一定的保护,在很大程度上就是对具有较大 RTT 的 TCP 连接进行保护,可以增强具有不同 RTT 的 TCP 连接间的公平性。尤其是在多拥塞链路的环境下,不但可以有效改善 TCP 的 RTT 不公平性问题,而且可以解决经过拥塞链路数目较多的数据包被丢弃概率相对较大而引发的不公平问题。同时,丢弃数据包时对已经历路由器数目较多的数据包进行保护,有利于提高网络资源的有效利用率。因此,我们认为将数据包经历的路由器跳数作为一个参考量来设计一个公平性增强的 AQM 机制实现算法是合理的。

2.2 FERED 的实现

在 RED 算法中,当一个数据包到达链路时,路由器将根据平均队列长度计算出一个丢包概率 P ,并以概率 P 丢弃该数据包。 P 的计算如下:

$$P = \begin{cases} 0, & Q_{min} > Q_e \\ \frac{Q_e - Q_{min}}{Q_{max} - Q_{min}} \times P_{max}, & Q_{min} \leq Q_e \leq Q_{max} \\ 1, & Q_e > Q_{max} \end{cases} \quad (1)$$

其中 Q_{min} 为最小阈值, Q_{max} 为最大阈值, P_{max} 为最大丢弃概率,这些值在 RED 中设置为常数。 Q_e 为计算出的平均队列长度。FERED 就是通过利用数据包 IP 报头中的 TTL 字段信息对式(1)得到的丢弃概率 P 进行一定的修正来实现的。其具体实现如下:

当数据包到达链路时,首先计算该数据包已经经历的上游路由器跳数 $v_hops_$,

$$v_hops_ = 32 - v_TTL_ \quad (2)$$

其中, $v_TTL_$ 为 IP 报头 TTL 字段中的值。在 NS2 中, TTL 字段的初始值设为 32。但是在实际的网络中,对于不同的操作系统, TTL 字段的初始值可能为 32, 64 或 128 等。由于在实际网络中,很少有数据包经历的路径跳数会超过 30 跳,因此对于在不同的操作系统发出的数据包所经历的路由器跳数

还是可以通过 TTL 字段值得到的^[11]。

然后利用指数加权滑动平均(EWMA)的方法计算出经过该链路的数据包所经历的路由器的平均跳数 $avg_hops_$:

$$avg_hops_ = avg_hops_ \times (1 - \alpha) + v_hops_ \times \alpha \quad (3)$$

接着,我们需要计算丢弃概率修正因子 f_i :

$$f_i = \frac{avg_hops_}{v_hops_} \quad (4)$$

可以看出,对于经历较少路由器的数据包,对应的 $f_i > 1$;反之, $f_i < 1$ 。

最后,我们利用式(4)得到的修正因子对式(1)得到的丢弃概率 P 进行修正:

$$P = P \times f_i \quad (5)$$

显然, $f_i < 1$ 时, P 减小,这意味着在一定程度上保护了经历较多路由器的数据包,从而可以改善 TCP 的 RTT 不公平性问题和多拥塞链路环境下的不公平性问题。显然 FERED 的实现非常简单,无需在路由器保留每流状态信息。

3 仿真试验

在这一部分,我们用 NS2. 29^[13] 下的仿真试验来验证 FERED 对公平性的增强。我们采用图 1 所示的“停车场”网络拓扑结构,这是多拥塞链路的一种典型配置,其中具有 4 个瓶颈链路。具有不同 RTT 的 TCP 连接经历了不同的拥塞链路。试验中的数据源分为 3 类:第一类由 $N1$ 个贪婪持久的 FTP 数据组成,第二类由 $N2$ 个突发的 HTTP 数据组成,第三类由 $N3$ 个非适应性 UDP 数据流组成,其空闲和突发时间周期服从负指数分布。空闲和突发平均时间分别为 10s 和 1s。试验中,我们使 $N11 = N21 = N31 = N41 = n$, $N12 = N22 = N32 = N42 = n$, $N13 = N23 = N33 = N43 = 2$, 通过使 n 的值在 5 到 40 之间变化来模拟链路负载从轻载到重载的变化。为使所有的瓶颈链路具有相同的负载,我们使 $N10 = N21 + N31 + N41$, $N20 = N31 + N41$, $N30 = N41$ 。模拟中我们假定所有瓶颈链路的带宽和广播延时分别为 2Mbps 和 10ms,其他链路的带宽和广播延时均为 10Mbps 和 5ms。在 $t=0s$ 时启动 FTP 和 UDP 业务流,HTTP 业务流随机启动。整个模拟持续 200s。RED 和 FERED 参数配置如下:

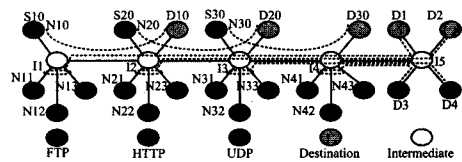


图 1 “停车场”网络拓扑结构

RED:最大/最小门限值分别设置为 45/15packets,其他参数根据文献[14]设置。

FERED:加权指数 $\alpha=0.025$,采用去尾丢弃。

首先,我们随机地跟踪节点 I3 上的队列长度来评价 FERED 算法对队列长度的控制能力。其结果显示在图 2 中。

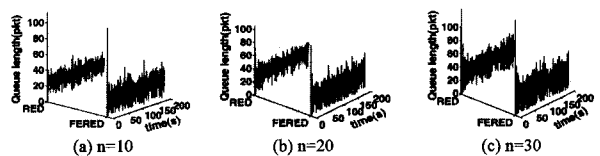


图 2 不同负载下 I3 处的队列长度

从图 2 中我们可以看出,FERED 继承了 RED 算法对队

列长度的控制能力。无论是在轻载还是在重载的情况下, FERED 都可以取得和 RED 相似的队列长度控制能力。

为了评价 FERED 算法对公平性的增强, 我们从经历不同数目拥塞链路的 4 组数据源 S1, S2, S3, S4 中分别随机选取一个 FTP 数据流, 跟踪它们在不同负载情况下的拥塞窗口的变化来验证 FERED 对公平性的增强。这 4 个 FTP 数据流在 $n=10, 20, 40$ 时的拥塞窗口变化分别显示在图 3—图 5 中。

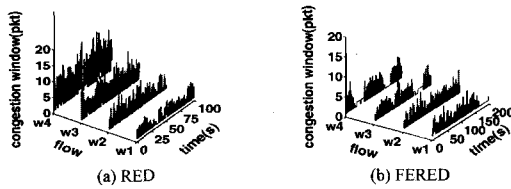


图 3 $n=10$ 时各数据流的拥塞窗口

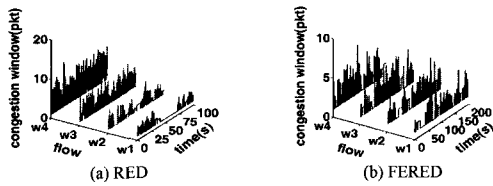


图 4 $n=20$ 时各数据流的拥塞窗口

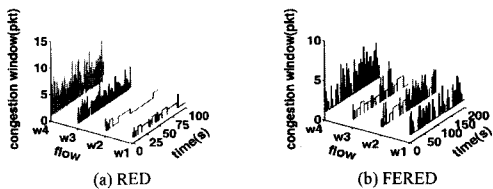


图 5 $n=40$ 时各数据流的拥塞窗口

如图 3 所示, 当 $n=10$ 时, 各个链路处于轻度负载之下, RED 算法也可以取得较好的公平性。但是随着链路负载的增加, 当 $n=20$ 时, 如图 4 所示, 多瓶颈链路环境开始出现, 在 RED 算法中数据流 w1 和数据流 w2 将处于明显的不利位置。随着链路负载继续增加, 当 $n=40$ 时, 如图 5 所示, 对于 RED 算法中数据流 w1 和数据流 w2 的拥塞窗口很少有机会能得到增长, 几乎出现“饿死”现象。相比之下, FERED 算法明显可以克服存在于 RED 算法中的不公平性问题, 在任何链路负载情况下, 都可以取得较好的公平性。

为了定量表示 FERED 算法对公平性的增强, 我们利用文献[15]中的公平性指数定义:

$$FI = \left(\sum_{i=1}^n x_i \right)^2 / \left(n \sum_{i=1}^n x_i^2 \right) \quad (6)$$

其中, x_i 为 FTP 数据流 w1, w2, w3 和 w4 所成功发送的数据包的数量。计算出不同链路负载情况下 RED 和 FERED 的公平性指数, 其结果显示在图 6 中。

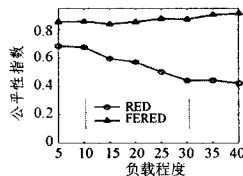


图 6 不同负载下 RED 和 FERED 的公平性指数

显然, 和 RED 算法相比, FERED 极大地提高了 TCP 数据流之间的公平性, 在链路负载从轻载到重载的变化过程中,

FERED 算法的公平性指数始终在 0.8 以上。

结束语 在本文中, 我们基于实际网络测量中得出的结论: 从总体的趋势上来看, 数据流所经历的路由器跳数和往返时延之间大致上还是存在着线性关系的, 即数据流经历跳数越多, 其往返时延很可能就越大。利用 IP 数据报头中的 TTL 字段信息对 RED 算法进行了扩展来增强公平性, 实现了一个公平性增强的 RED 算法 (FERED)。NS2 下的仿真试验结果显示, FERED 可以显著增强公平性, 同时保留了 RED 算法可以很好地控制队列长度的优点, 而且 FERED 实现简单, 无需在路由器保留每流状态信息。但是 FERED 所存在的不足就是当经历较多路由器跳数的数据包却具有较小 RTT 的情况下可能会加重 TCP 的 RTT 不公平性, 但是正如网络测量结论, 我们认为这种情况发生的概率较小, 因此 FERED 还是具有较大应用价值的。

参考文献

- [1] Chiu D, Jain R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks[J]. Computer Networks and ISDN Systems, 1989, 17(1): 1-14
- [2] Floyd S. Connections with multiple congested gateways in packet-switched networks part 1: one-way traffic[J]. Computer Communication Review, 1991, 21(5): 30-47
- [3] Ren F Y, Huang X M, Lin C. Design a Two-Category Classifier with Multi-Dimension for Active Queue Management// Proc. of First ACM SIGCOMM Asia Workshop. Beijing, China, April 2005: 150-157
- [4] Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance. ACM/IEEE Transactions on Networking, 1993, 1(4): 397-413
- [5] Athuraliya S, Lapsley D E, Low S H. Random early marking for Internet congestion control // Proceedings of IEEE Globecom. 1999
- [6] Hollot C, Misra V, Towsley D, et al. On designing improved controllers for AQM routers supporting TCP flows. UMass CMPSCI Technical Report 00-42. 2000
- [7] Kunniyur S S, Srikant R. An Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management. IEEE-ACM Transactions on Networking, 2004, 12 (2): 286-299
- [8] Lin D, Morris R. Dynamics of random early detection[J]. ACM Computer Communication Review, 1997, 27(4): 127-137
- [9] Anjum F, Tassiulas L. Balanced RED: An algorithm to achieve fairness in Internet // Proceedings of IEEE INFOCOM1999. New York, USA, 1999
- [10] Fei A, Pei G, Liu R, et al. Measurements on delay and hop-count of the internet // IEEE GLOBECOM'98—Internet Mini-Conference. 1998
- [11] Fujii K, Goto S. Correlation between hop count and packet transfer time // APAN/IWS2000. February 2000
- [12] Kurose J F, Ross K W. Computer Networking: A Top-Down Approach Featuring the Internet. 2nd Edition. Boston: Addison-Wesley, 2003
- [13] UCN/LBL/VINT. Network Simulator-NS2. <http://www.mash.cs.berkeley.edu/ns>
- [14] <http://www.icir.org/floyd/red.html#parameters>
- [15] Jain R, Chiu D M, Hawe W. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems. Technical Report TR301. Digital Equipment Corp, 1984