

基于最优化流控的 TCP 拥塞控制改进算法 MReno* An Improved Algorithm of TCP Congestion Control Based on Optimization Flow Control MReno

陈元琰¹, 胡江伟¹, 罗晓曙², 苏 聪¹

CHEN Yuan-yan¹, HU Jiang-wei¹, LUO Xiao-shu², SU Cong¹

(1. 广西师范大学计算机科学与信息工程学院, 广西桂林 541004; 2. 广西师范大学物理与电子工程学院, 广西桂林 541004)

(1. College of Computer Science and Information Engineering, Guangxi Normal University, Guilin, Guangxi, 541004, China; 2. College of Physics and Electronic Engineering, Guangxi Normal University, Guilin, Guangxi, 541004, China)

摘要:应用最优化流控理论, 给出 TCP Reno 算法的一个对偶模型, 把该模型与 REM 算法相结合, 得到一个新的 TCP 网络拥塞控制算法(MReno), 并对其进行模拟实验。结果表明, MReno 算法可以使源端获得最大资源利用率, 能够保证网络的整体优化, MReno 算法比 Reno 算法有更好的公平性和稳定性。

关键词:最优化流控 对偶模型 TCP 拥塞控制 REM

中图分类号:TP310.6 **文献标识码:**A **文章编号:**1002-7378(2007)04-0223-03

Abstract:In this paper a dual mode of TCP Reno algorithm is elicited by means of optimization flow control theory, then a new TCP congestion control algorithm MReno(Marking Reno) is obtained by combining the dual mode of TCP Reno with algorithm REM (Random Exponential Marking). Simulation experimental results indicate that the MReno algorithm is able to get maximum resource utility for sources, ensure whole optimization of networks, and is fairer and more stable than the Reno algorithm.

Key words:optimization flow control, duality mode, TCP congestion control, REM

计算机网络技术在近十几年中得到了飞速的发展, 随之将会出现越来越严重的网络拥塞问题。网络拥塞已逐渐成为制约网络发展和应用的瓶颈。因此, 互联网的拥塞控制是一个非常关键的问题。拥塞控制的目标就是要达到链路吞吐量的最大化、分组延迟的最小化和各用户之间资源分配的合理化。为了解决网络拥塞, 各种网络拥塞控制理论以及算法也相继被提出。网络拥塞控制算法大体上分为两种: 一种是 TCP 源端算法即 Reno 算法^[1], 也就是 TCP 基于滑动窗口的和式增加积式减少(AIMD)的拥塞控制机制; 另一种是链路算法, 即路由器参与的主动队

列管理(AQM)算法, 具有代表性的主要有 RED 算法和 REM 算法^[2]。

Reno 算法是 TCP 源端算法中的一种, 它以丢包作为拥塞度量, 其基本思想是: 在每个回路响应时间内, 如果没有丢包, 它会持续把窗口增加 1, 如果有 1 个或多个丢包, 它会把窗口减小为当前窗口的一半。相对于早期的 TCP 源端算法, Reno 算法确实有很大进步。但 Reno 算法在稳定性、公平性和扩展性方面还存在很多不足。

REM 算法是基于隐式价格的一种 AQM 算法, 它依靠包在每个链路中的标记来携带链路的价格信息而在源端估计所使用的整个链路的价格信息, 即网络状况。它将价格转换为丢弃或者标记概率, 源端依此丢弃或标记概率来调整其发送速率。这里, 借用经济学中的“价格”一词来表达网络的反馈信号, 它表示一种拥塞度量, 也是一种提供给源端的控制

收稿日期: 2007-08-20

作者简介: 陈元琰(1961-), 男, 副教授, 主要从事计算机图形学和计算机网络方面的研究。

* 国家自然科学基金项目(70571017)和广西自然科学基金项目(0728099)资助。

信号,同样可以表示为源端使用网络资源所需要付出的代价,例如丢包、显式标记(ECN)、排队时延和显式价格等等。

本文以非线性最优化流控理论作为指导思想,把源端算法 Reno 和链路算法 REM 相结合,生成一种新的网络拥塞控制算法 MReno,以改进原有 TCP Reno 算法的不足。

1 TCP Reno 的一个对偶模型

基于最优化流控原理,Low^[3]提出了 TCP/AQM 的一个对偶模型,该模型把现有的 TCP 拥塞控制和 AQM 算法看作是求解具有适当效用函数的最优速率分配问题的分布式算法,源端发送速率和路由拥塞度量是相互影响的。源端根据反馈回来的拥塞度量调整其发送速率,而各源端发送速率的大小又会反过来影响拥塞度量,从而构成一个闭环拥塞控制系统。具体思想如下:

每个源 $s(s \in S)$ 使用一个链路的集合 $L(s)(L(s) \in L)$ 。集合 $L(s)$ 被定义成一个 $L \times S$ 的路由矩阵 R 。当 $l \in L(s)$ 时,矩阵 R_{ls} 的值为 1,其它情况为 0。

令 $x_s(t)$ 是源 s 在时间 t 的传输速率,在时间 t 链路 l 的拥塞度量是 $p_l(t)$,其值大于等于 0。令 $y_l(t) = \sum_s R_{ls} x_s(t)$ 是链路 l 总的源速率,令 $q_s(t) = \sum_l R_{ls} p_l(t)$ 是源 s 端到端的拥塞度量。

在每个阶段源端速率 $x_s(t)$ 依据函数 F_s (某种 TCP 算法)进行调整, F_s 只依靠变量 $x_s(t)$ 和 $q_s(t)$,对所有的 s 有

$$x_s(t+1) = F_s(x_s(t), q_s(t)). \quad (1)$$

在每个阶段链路拥塞度量 $p_l(t)$ 的调整只依靠 $p_l(t)$ 、 $y_l(t)$ 和可能的一些内部变量 $v_l(t)$,例如链路 l 的队列长度。这可以用函数 G_l 表示,对所有链路 l ,我们有

$$p_l(t+1) = G_l(y_l(t), p_l(t), v_l(t)), \quad (2)$$

其中 G_l 是非负的,所以 $p_l(t) \geq 0$ 。这样,就可以用 F_s 模拟 TCP 算法,用 G_l 模拟 AQM 的算法。

通过上述原理,我们可以得到 Reno 算法的对偶模型,其推导过程如下:

令 $w_l(t)$ 是窗口尺寸,令 D_l 是平衡时的回路响应时间,一般假定为常量。定义 $x_s(t) = w_l(t)/D_l$,单位为包/秒。令 $m_l(t)$ 是链路 l 在时刻 t 的标注概率(和最优化流控理论的价格成正比),假设每个源 s 经过所有链路的端到端的标注概率 $m_s(t)$ 是这个链

路标注概率的和,即 $m_s(t) = \sum_{l \in L(s)} R_{ls} m_l(t)$,当 $m_l(t)$ 较小的时候,这是完全合理的。在阶段 t ,假设每个包都被确认,它的传输速率为每单位时间 $x_s(t)$ 个包,并且近似地以相同的速率收到正的和负的确认(负的确认指标注的包)。

在每个回路响应时间如果没有包被标注,TCP Reno 窗口增加 1,如果有一个或多个包被标注,窗口将减半。作如下的改变:在阶段 t 每收到正的确认,窗口增加 $1/D_l$,收到负的确认,窗口减小为 $w_l/2D_l$ 。正的确认的概率为 $1 - m_l(t)$,负的确认的概率为 $m_l(t)$ 。那么在阶段 t 的窗口尺寸平均改变为

$$\frac{1}{D_l}(1 - m_l(t)) - \frac{w_l(t)}{2D_l} m_l(t),$$

除以 D_l 变成

$$\frac{1}{D_l^2}(1 - m_l(t)) - \frac{w_l(t)}{2D_l^2} m_l(t).$$

x_s 在时刻 $t+1$ 的速率 $x_s(t+1)$ 变为

$$x_s(t+1) = [x_s(t) + \frac{1 - m_l(t)}{D_l} - \frac{1}{2D_l} m_l(t) x_s(t)]^+, \quad (3)$$

其中 $[z]^+ = \max\{z, 0\}$,保证速率非负。(3) 式就是我们想要的 TCP Reno 的对偶模型。

下面基于上述理论,结合 REM,尝试生成一种新的拥塞控制算法。

2 新算法描述

REM 算法也是在最优化流控理论的基础上得到的主动队列管理算法,计算价格是 REM 算法的一个重要特征。对于队列 l ,价格 $p_l(t)$ 在阶段 $t+1$ 按照如下的公式被更新

$$p_l(t+1) = [p_l(t) + \gamma \beta_l b_l(t) + y_l(t) - c_l]^+, \quad (4)$$

其中 $\gamma > 0$ 和 $\beta_l > 0$ 是常量,并且 $[z]^+ = \max\{z, 0\}$, $b_l(t)$ 是在阶段 t 队列总的队列长度, $y_l(t)$ 是所有源 s 在阶段 t 经过链路 l 的速率之和, c_l 是链路带宽。

在阶段 t 链路 l 对每个包的标注概率 $m_l(t)$ 为

$$m_l(t) = 1 - \phi^{-p_l(t)}, \quad (5)$$

其中 $\phi > 1$ 是一个常数。如果一个包被标注,它的标注被传送到目的地并且经过确认被传送到源端。对于多链路网络,这个指数形式是必不可少的。源 s 的一个包经过所有的链路,它的端到端标注概率为

$$m_s(t) = 1 - \prod_{l \in L(s)} (1 - m_l(t)) = 1 - \phi^{-q_s(t)}. \quad (6)$$

(6) 式中 $q_i(t) = \sum_{l \in L(i)} p_l(t)$ 是一个包经过所有链路的价格的和。当 $q_i(t)$ 较大的时候, 这个端端的标注概率就高。反之, 标注概率就会小。把(6)式代入(3)式我们得到一个称之为 MReno(标注的 Reno)新的网络拥塞控制算法, MReno 算法调整其速率依赖于路由器算法 REM 的标注概率, 实际上还是依赖于 REM 算法的价格。其速率调整的表达式为

$$x_i(t+1) = [x_i(t) + \frac{\phi - q_i(t)}{D_i^2} - \frac{1}{2D_i}(1 - \phi - q_i(t))x_i(t)]^+ \quad (7)$$

3 模拟实验及分析

采用 NS-2 对 MReno 算法进行分析和验证其性能。网络的拓朴结构如图 1 所示。R1、R2 是路由器结点, S1、S2、S3、S4 为源端结点, 有两条链路, 分别为 S1 到 S3, S2 到 S4。源端和路由器的连接带宽为 10Mb, 瓶颈链路带宽为 2Mb, 模拟时间为 100s。

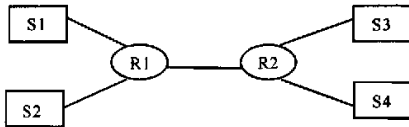


图 1 网络的拓朴结构

稳定性和公平性的模拟结果如图 2、图 3 和图 4 所示。

由图 2 可以看出, MReno 算法比 Reno 算法有更好的稳定性。由图 3 和图 4 可以看出, 在相同的条件下, MReno 算法比 Reno 算法能更好与 Vegas 共享带宽, 因此有更好的公平性。

所谓公平性, 是指在发生拥塞时各源端(或同一源端建立的不同 TCP 或 UDP 数据报)能公平地共享同一网络资源(或带宽、缓存等)。公平性是衡量算法好坏的标准之一。处于相同级别的源端应该得到相同数量的网络资源。不公平的根本原因在于拥塞

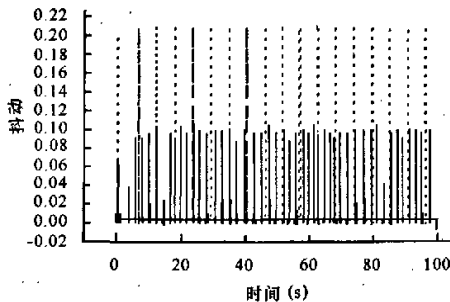


图 2 MReno 算法与 Reno 算法的稳定性比较
—: MReno 算法; ---: Reno 算法。

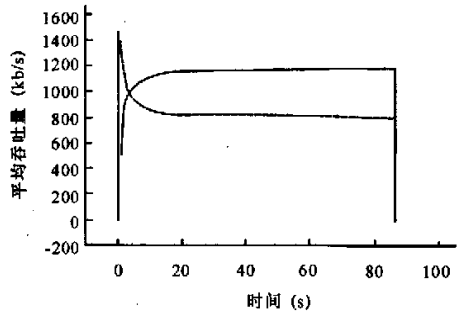


图 3 Reno 算法和 Vegas 的平均吞吐量比较
—: Reno 算法; ---: Vegas。

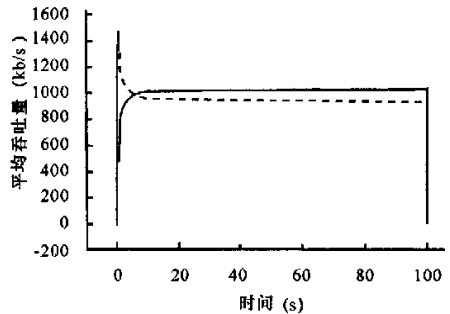


图 4 MReno 算法与 Vegas 的平均吞吐量比较
—: MReno 算法; ---: Vegas。

发生必然导致各数据流之间为争抢有限网络资源发生竞争, 争抢能力弱的数据流将受到更多损害。

4 结束语

MReno 算法根据路由器算法 REM 反馈回来的标记概率来调整源端发送速率, 而源端的发送速率反过来又会影响到 REM 算法的标记概率, 从而构成一个闭环的拥塞控制系统。该算法可以使源端获得最大资源利用率, 保证了网络的整体最优。从模拟实验可以看出, MReno 算法相对于 TCP Reno 算法, 有较好的稳定性和公平性。

参考文献:

- [1] CHIU D M, JAIN R. Analysis of the increase and decrease algorithms for congestion avoidance in computer network[J]. Computer Networks and ISDN Systems, 1987, 17(1): 1-14.
- [2] ATHURALIYA S, LI V H, LOW S H, et al. Rem: active queue management[J]. IEEE Network, 2001, 5(3): 48-53.
- [3] LOW S H. A duality model of TCP and queue management algorithms [J]. IEEE/ACM Transactions on Networking, 2003, 11(4): 525-536.

(责任编辑: 尹 闯)