

一种自适应比例系数的 TCP 拥塞控制策略*

陶 洋,杜军恒,武 俊

(重庆邮电大学 软件技术中心,重庆 400065)

摘 要: 互联网业务的快速发展使得网络拥塞日益严重,因此如何减轻网络拥塞也就成为当前研究的热点。针对这一问题,首先叙述了 TCP 拥塞控制的基本原理及其相关算法;然后对拥塞控制算法进行了详细分析并提出了自适应比例系数的新策略;最后对新策略进行 NS2 仿真。仿真结果表明,新方法提高了包投递率,降低了端到端时延。

关键词: 拥塞控制;自适应;NS2

中图分类号: TP39

文献标识码: B

文章编号: 1674-7720(2011)12-0058-03

An adaptive ratio strategy of TCP congestion control

Tao Yang, Du Junheng, Wu Jun

(Center of Software Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: With the rapid development of Internet, the problem of network congestion is becoming more and more serious. Therefore, how to lighten network congestion grows gradually the hotspot at present. To solve this trouble, in this paper, the basic principle of TCP congestion control as well as related algorithm is stated and then, some of the algorithm is improved because of it's faults after being analysed. Finally, we simulate the new method and it's result shows that the new strategy not only increases the delivery rate of packets but also reduces the end-to-end delay.

Key words: congestion control; adaptive; NS2

21 世纪以来,计算机网络发生了突飞猛进的发展,各种新业务在互联网上的应用也越来越广泛,这就给带宽有限的网络带来了巨大的负担,从而引发了严重的网络拥塞问题。

TCP 是互联网上应用极其广泛的基于滑动窗口的拥塞控制协议。它使用了一种可靠的设计思路,这种思路具有较为简单、扩展性强等特点。它能够依据网络的实际具体情况自动地调整拥塞窗口的大小,使得 TCP 数据的发送能够依据网络的负担自适应地变化。但近年来,互联网上的用户越来越多,业务量也随之增加,网络拥塞问题亦越来越严重。因此,人们对拥塞控制的研究重视程度亦越来越高,先后提出了许多新的算法,如 TCP Tahoe、TCP Reno、TCP New-Reno、TCP sack、TCP Vegas^[1]。这些算法着重研究了快重传和快恢复机制,而慢开始和拥塞避免并无过多涉及。但慢启动和拥塞避免亦不能忽视,因为这些策略的好坏对拥塞控制机制的影响同样是

巨大的。针对该问题,本文提出了一种自适应比例系数的拥塞控制策略。

1 TCP 拥塞控制原理及相关算法

互联网拥塞控制主要由传输层完成,它通过改变一些主要的参数来降低发送端数据的发送速率。这些参数主要包括拥塞窗口 cwnd(发送端一次最多发送的数据包的个数)和慢开始门限 ssthresh(慢开始阶段与拥塞避免阶段的分界点)。

早期的拥塞控制协议中,发送端每发送一个报文就必须收到接收端返回的 ACK 后才能发送下一个报文。发送端在等待接收端发送 ACK 的期间内不发送任何报文;当报文丢失时,必须等到计时器超时后才能重发丢失的数据包。显然,这种策略严重地浪费了网络的带宽,网络传输数据的效率极其低下。

现在使用的拥塞控制协议采用了 AMID^[2]控制算法,能够达到较好的拥塞控制效果,提高了网络数据的传输

* 基金项目:重庆邮电大学自然科学基金项目(A2009-30)

网络与通信 Network and Communication

效率,这种策略一般分为4个阶段^[3],具体如下:

(1)慢开始阶段。在TCP建立连接之初,如果发送方将cwnd设置为较大值,发送方有可能将较大的数据字节全部注入到网络中,但此时并不清楚网络状况,因此就有可能引起网络拥塞。经实际证明,最好的方法是试探一下,即由小到大逐渐增大发送端的cwnd数值。通常发送方在开始发送数据报文段时将cwnd设置为一个最大的MSS数值,发送方每接收到接收方发来的一个ACK,就把cwnd增加一个值,直到cwnd增加到sssthresh(慢开始门限值),此时进入拥塞避免阶段,采取拥塞避免算法。在采取拥塞避免算法之前,拥塞窗口cwnd的值以指数方式增长。

(2)拥塞避免阶段。当cwnd值达到sssthresh(慢开始门限值)时,此后进入拥塞避免阶段,发送端的cwnd每经过一个RTT(往返时延)就增加一个MSS的大小(而不管在时间RTT内收到了几个接收端发送的ACK)。这样,发送端cwnd就不再像慢开始阶段那样进行指数增长,而是按线性规律增长(称为加法增大),这比慢开始算法的拥塞窗口增长率缓慢得多。在没有丢包之前,该策略将一直持续下去。当假定cwnd的数值增长到 N ($N=24$)时,网络出现超时(表明网络拥塞了)。sssthresh置为 $N/2$ (为发送窗口数值的一半,这种算法称为乘法减小),cwnd重置为1,并执行慢开始算法。当cwnd=12时改为执行拥塞避免算法,拥塞窗口按线性规律增长,每经过一个RTT就增加一个MSS的大小。

(3)快重传阶段。前面的慢开始和拥塞避免算法是在TCP早期使用的拥塞控制算法,后来人们对其进行了改进,因此就有了快重传和快恢复。假设发送端发送了M1~M4这4个报文段,当接收端收到M1和M2后,就发出确认ACK2和ACK3。现假定M3丢失了,接收端收到下一个M4,发现其序号不对,但仍收下放在接收缓存中,同时发出确认,但发出的是重复的ACK3(不能发送ACK5,因为ACK5表示M4和M3都已经收到了)。这样发送端知道现在可能是网络出现的拥塞造成了分组丢失或是M3仍滞留在网络中某处,还需经过较长的时延才能到达接收端。发送端接着发送M5和M6,接收端收到了M5和M6后,还要分别发出重复的ACK3。这样发送端共接收到了4个ACK3,其中3个是重复。如果快重传算法规定,当发送端连续收到3个重复的ACK时即可断定有分组丢失了,就应立即重传丢失的报文而不必继续等待为M3设置的重传计时器超时。由此可知,快重传并非取消重传计时器,而是在某些情况下更早地重传丢失的报文段。

(4)快恢复阶段。与快重传配合使用的还有快恢复算法,当不使用快恢复算法时,发送端若发现网络出现拥塞就将拥塞窗口置为1,然后执行慢开始算法。但这样做的缺点是网络不能很快地恢复到正常状态。快恢复算法可以较好地解决这一问题。当发送端连续接收到3

个重复的ACK时,就采用乘法减小设置sssthresh,与慢开始不同的是cwnd设置为 $sssthresh+3 \times MSS$,然后执行拥塞避免算法,这样可以使网络快速恢复到正常水平(与cwnd设置为1相比),大大地提高了网络的吞吐量。

TCP拥塞控制算法效果图如图1所示。

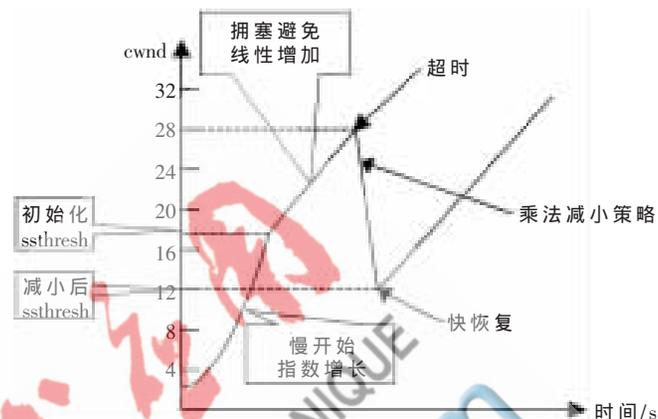


图1 TCP拥塞控制算法图例

2 乘法减小及其存在问题

当假定cwnd的数值增长到 N ($N=24$)时,网络出现超时(表明网络拥塞了)。sssthresh置为 $N/2$,sssthresh的新数值为12,这种算法称为“乘法减小”。进一步说,“乘法减小”是指不论在慢开始阶段还是拥塞避免阶段,只要出现一次超时(即出现一次网络拥塞),就将慢开始门限值sssthresh设置为当前的拥塞窗口值的1/2。当网络频繁出现拥塞时,sssthresh值就会下降得很快,以大大减少注入到网络中的分组数。但是网络拥塞极其频繁时,这种固定比例的减小策略就会存在问题,这种减小力度不够大,仍会造成网络十分拥堵,甚至使得网络发生崩溃;当网络很少出现拥塞时,这种固定的减小策略会使得注入网络的分组数不够多,从而造成网络带宽得不到充分利用,造成网络资源的浪费。

3 一种新的变比例因子的乘法减小策略

针对固定比例因子的“乘法减小”策略容易出现的一些问题,本文提出了一种新的变比例因子的策略,其主要思路是在数据传送初期设置4个参数:3个时间参数 t_1 、 t_2 、 Δt 和比例因子 p ,依据3个时间参数的变化来动态调整比例因子 p 的大小,从而使得比例因子 p 根据网络的状况自适应地变化。

具体思路:引入3个时间参数 a 、 b 、 c 和 p ($0 < p < 1$)。 t_1 是上次超时的时间, t_2 是本次超时的时间, Δt 是本次超时与上次超时的时间差, p 是减小比例系数并赋初值为0.5。当出现超时时,比较 $t_2 - t_1$ 与 Δt 的大小,(1)若 $t_2 - t_1 < \Delta t$,判断 p 是否小于0.2,如果 $p < 0.2$,则 p 的值大小不变,且将 $t_2 - t_1$ 的值赋给 Δt ,将 t_2 的值赋给 t_1 ;否则将 $p - 0.2$ 的值赋给 p ,将 $t_2 - t_1$ 的值赋给 Δt ,将 t_2 的值赋给 t_1 ;(2)若 $t_2 - t_1 \geq \Delta t$,判断 p 是否小于0.8,如果 $p > 0.8$,则 p 的值大小不变,将 $t_2 - t_1$ 的值赋给 Δt , t_2 的值赋

网络与通信 Network and Communication

给 $t1$, 否则将 $p+0.2$ 的值赋给 p , $t2-t$ 的值赋给 Δt , 且将 $t2$ 的值赋给 $t1$ 。新策略的算法描述大致如下。

```

t1=初值;
t2=初值;
p=0.5 ;
if(超时)
{
    t2=getcurrenttime();
    if(t2-t1>=Δt)
    {
        if(p>0.8)
        {
            p=0.9;
            ssthresh=(int)p×cwnd ;
            Δt=t2-t1;
            t1=t2;
        }
        else
        {
            p+= 0.2;
            ssthresh=(int)p×cwnd ;
            Δt=t2-t1;
            t1=t2;
        }
    }
    else
    {
        if(p<0.2)
        {
            p=0.1 ;
            ssthresh=(int)p×cwnd ;
            Δt=t2-t1;
            t1=t2;
        }
        else
        {
            p-=0.2;
            ssthresh=(int)p×cwnd ;
            Δt=t2-t1;
            t1=t2;
        }
    }
}

```

本文所提出的动态变化系数策略如上所述, 新策略可以根据网络状态的好坏动态地调整系数 p 的大小。

4 仿真

4.1 仿真场景

为了证明新策略的有效性, 本文使用 NS2^[4] 网络仿真软件对未改进的策略与改进后的策略进行了仿真对

比。本文使用的网络拓扑图如图 2 所示。

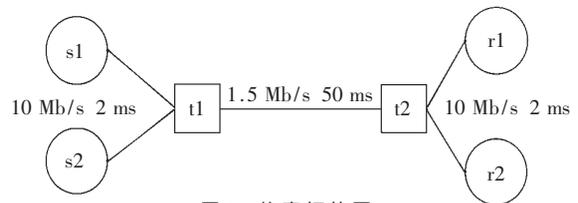


图 2 仿真拓扑图

图 2 中, $s1$ 、 $s2$ 为发送端, $t1$ 、 $t2$ 为干路的路由器, $r1$ 、 $r2$ 为接收端。在上面的拓扑图中, 建立了两条链路, 一条为 $s1$ 将数据通过干路路由器转发, 将数据发送到 $r1$; 另一条为 $s2$ 将数据通过干路路由器转发, 将数据发送到 $r2$ 。两条链路均采用基于 TCP 的 FTP 连接, 传输速率为 10 Mb/s , 延迟为 2 ms , 分组大小为 1000 B 。干路路由器 $t1$ 、 $t2$ 之间构成瓶颈链路, 传输速率为 1.5 Mb/s , 延迟为 50 ms 。

4.2 仿真结果分析

图 3、图 4 是新旧策略包到达率和端到端延时比较, 可以看到, 新策略的到达率比原策略大, 这是因为新策略可以根据网络状态的好坏动态调整乘法比例因子。当网络发生拥挤时, 新策略降低乘法因子, 减少数据包的发送, 减小网络负担, 因此数据包到达的可能性提高, 封装端到端时延减小。

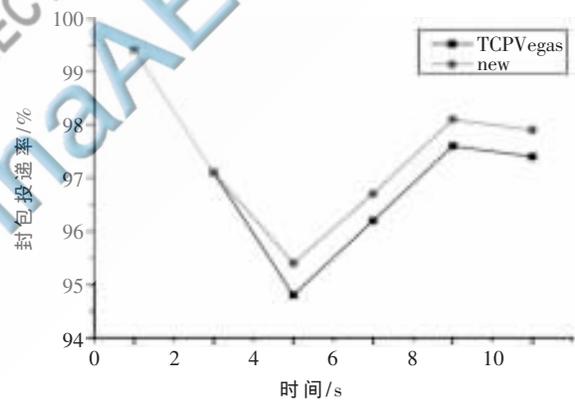


图 3 封包投递率

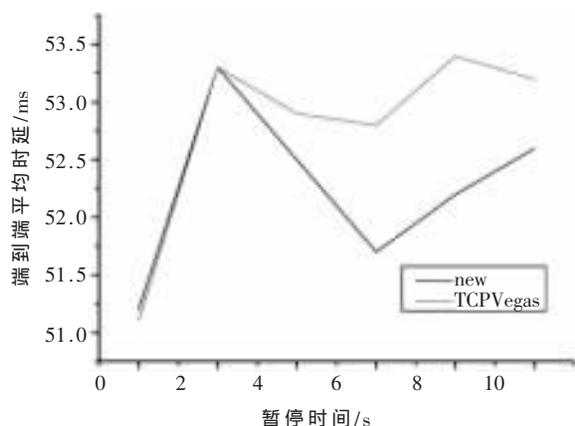


图 4 端到端平均时延

网络与通信 Network and Communication

本文阐述了拥塞控制原理及算法并对算法做出了一些改进,提出了自适应比例因子乘法减小策略。通过NS2仿真分析可知,这种策略明显提高了网络的性能。

参考文献

- [1] 王云涛,方建安,张晓辉,等.基于TCP Vegas的网络拥塞控制改进算法[J].计算机应用研究,2009,26(12):56-58.
- [2] Yang Ming, Hang Fuyan. AIMD-based congestion control for layered multicast[S]. IEEE, 2002, 02: 833-837.

- [3] 谢希仁.计算机网络[M].北京:电子工业出版社,2006.
- [4] 徐雷鸣,庞博,赵耀.NS与网络模拟[M].北京:人民邮电大学出版社,2003.

(收稿日期:2010-10-22)

作者简介:

陶洋,男,1964年生,博士后,主要研究方向:通信协议及软件。

杜军恒,男,1986年生,硕士研究生,主要研究方向:通信协议及软件。

