基于 TCP Vegas 的网络拥塞控制改进算法 *

王云涛, 方建安, 张晓辉, 严伟锋
(东华大学 信息科学与技术学院, 上海 201620)

摘 要: 由于 TCP Vegas 在与 TCP Reno 算法共存的网络环境中不能公平地竞争到带宽,TCP Vegas-A 拥塞控制算法有效地改进了 TCP Vegas 算法在带宽竞争力弱方面的缺陷。对 TCP Vegas-A 算法进行了仿真研究并提出一种拥塞控制改进算法 TCP NewVegas。基于 NS2 的仿真实验证明 TCP NewVegas 算法提高了与 TCP Reno 共存时在网络中的带宽竞争能力。

关键词: 网络拥塞控制; 竞争能力; TCP Vegas-A; TCP NewVegas; NS2
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-3695(2009)12-4645-03
doi:10.3969/j.issn.1001-3695.2009.12.068

Enhanced TCP congestion control algorithm based on TCP Vegas

WANG Yun-tao, FANG Jian-an, ZHANG Xiao-hui, YAN Wei-feng
(*School of Information Science & Technology, Donghua University, Shanghai* 201620, *China*)

Abstract: Due to Vegas' incapability of seizing bandwidth fairly in heterogeneous network environments, the TCP Vegas-A congestion control algorithm improves TCP Vegas algorithm in this aspect's weak bandwidth competitive ability effectively. This paper studied the TCP Vegas-A algorithm and proposed an enhanced TCP congestion control algorithm TCP NewVegas. The simulations based on NS2 show that TCP NewVegas algorithm enhanced the band width competitive ability with TCP Reno in heterogeneous network environments.

Key words: network congestion control; competitive ability; TCP Vegas-A; TCP NewVegas; NS2

因特网的迅速发展带来了信息流量的急剧膨胀,由此带来了严重的网络拥塞问题。所谓网络拥塞(congestion)是指在分组交换网络中传输分组的数目太多时,由于存储转发节点的资源有限而造成网络传输性能下降的情况。网络拥塞发生时,一般会出现数据包时延增加,丢包概率增加,网络吞吐量下降,网络效率下降,严重的网络拥塞会造成网络拥塞崩溃。

目前伴随着新型网络的不断涌现和业务流量的日益增加,网络拥塞问题更加严重。TCP 协议是当今 Internet 广为使用的传输协议,而且 TCP 所采用的拥塞控制算法是保证当今网络不断发展同时又避免拥塞崩溃的主要方法。其中的 TCP Reno[1] 算法是目前网络上广泛应用的算法,TCP Vegas[2] 算法是一种在理论上已被证明更高效的拥塞控制算法,然而在和 TCP Reno 算法共同应用在网络上时,它存在着竞争带宽能力不足的严重缺陷,这也是它一直不能实际应用的主要原因。TCP Vegas-A[3] 算法针对 TCP Vegas 算法进行了改进,有效地提高了对网络资源的竞争能力,本文研究了 TCP Vegas-A 算法,通过对它进行仿真分析,提出了 TCP NewVegas 算法,与 TCP Vegas-A 算法相比,它更有效地提高了在算法共存时对网络资源的竞争能力。

1 TCP Reno 算法与 TCP Vegas 算法

1.1 TCP Reno 算法

TCP Reno 算法是由 Jacobson 于 1960 年提出的拥塞控制算法[4],它是目前网络上应用最为广泛的 TCP 拥塞控制算法,它主要包括慢启动、拥塞避免、快速重传、快速恢复四个阶段。算法可简单地表示如下:

```
initial( ); //发送窗口(win);源端预设的发送窗口大小(awin)
Win = min(cwnd,awin),cwnd = 1;// ssthresh = 64KB(缺省值)
  if (cwnd < ssttresh)
  Cwnd = cwnd + 1;//慢启动
  else   cwnd = cwnd + 1/cwnd;//拥塞避免
relay timeout;
ssthresh = max (2, min (cwnd/2, awin));
Cwnd = 1;
```

TCP Reno 算法是在拥塞发生之后进行处理,不能有效地预防网络拥塞的发生,这是它的缺点之一。此外,TCP Reno 总是尽可能多地增加拥塞窗口来占有带宽,对目前多种协议并存的因特网是不利的。

1.2 TCP Vegas 算法

TCP Vegas 算法采用了基于测量的 TCP 技术,与其他拥塞控制算法相比,它主要作了以下三个方面的改进[5]。

1.2.1 新的重传机制

TCP Vegas 算法采用了新的重传机制,即用一个重复的 ACK(确认比特位),而不是 Reno 中的三个重复 ACK 来启动超时判定程序。这样可以更及时地检测到拥塞的发生。

1.2.2 新的拥塞避免机制

TCP Vegas 算法通过观测回路响应时间(RTT),比较实际

收稿日期: 2009-03-11; **修回日期**: 2009-04-28 **基金项目**: 国家自然科学基金资助项目(60874113)

作者简介:王云涛(1984-),男,山东聊城人,硕士研究生,主要研究方向为网络拥塞控制、混沌保密通信(2008yuntao327@sina.com);方建安(1966-),男,上海人,院长,教授,博导,主要研究方向为复杂系统建模、网络控制系统分析与综合、混沌控制与同步、智能控制与系统等;张晓辉(1985-),男,福建莆田人,硕士研究生,主要研究方向为保密通信;严伟锋(1984-),男,浙江温州人,硕士研究生,主要研究方向为混沌通信.

吞吐数量和期望吞吐量,来调整拥塞窗口大小。TCP Vegas 的拥塞避免算法可表示如下:

```
expected = cwnd/BaseRTT; //计算期望的吞吐量
actual = cwnd/RTT; //计算实际的吞吐量
Diff = expected - actual
```

$$cwnd(t+1)=\begin{cases} cwnd(t)+1 & Diff<\alpha/BaseRTT \\ cwnd(t) & \alpha/BaseRTT<Diff<\beta/BaseRTT \\ cwnd(t)-1 & Diff>\beta/BaseRTT \end{cases}$$

式中:BaseRTT 表示所有观测回路响应时间的最小数值,它一般是建立连接后所发送的第一个数据包的 RTT;cwnd 表示目前的拥塞窗口大小,即发送的数据包数;α 和 β 是定义的两个阈值(一般设 α = 1,β = 3);expected 表示期望的吞吐量;actual 表示实际的吞吐量。TCP Vegas 算法的目标是在网络队列中保持一定数量的数据包,使数据包的数量介于 α 与 β 之间。

1.2.3　新的慢启动机制

TCP Vegas 算法在慢启动阶段采用了更加谨慎的方式来增加窗口大小,即要求每隔一个 RTT 才进行指数增长,而之间的 RTT 期间,拥塞窗口保持不变。这样就减少了不必要的分组丢失。

1.3　TCP Vegas 算法与 TCP Reno 算法兼容性

通常情况下,Vegas 具有比其他 TCP 拥塞控制算法优越的性能 ,但是由于 Vegas 采用了主动的拥塞避免机制,在实际丢包以前减小了拥塞窗口。而 Reno 采用了反应式的拥塞控制机制。它增加拥塞窗口直到发现丢包。在 Reno 和 Vegas 共存时后,Reno 会窃取 Vegas 的带宽[6]。这种不兼容性限制了 Vegas 的应用。

1.4　仿真实验

算法仿真采用了 NS2 网络仿真软件,网络拓扑结构如图 1 所示。网络由两个传输节点 Source0 和 Source1、路由器 R0 和 R1、接收端 Sink0 和 Sink1 组成,分别传送和接收使用不同协议的 FTP 数据流。其中,Source0 使用 TCP Vegas(α = 1,β = 3),Source1 使用 TCP Reno。Source0、Source1 到 R0 之间,R1 到 Sink0、Sink1 之间的带宽均为 10 Mbps ,传输延迟为 4 ms ,瓶颈带宽设置在 R0 ~ R1 ,带宽为 1.5 Mbps ,传输延迟为 4 ms。所有链路的队列管理机制都采用 DropTail,运行时间为 70 s。

首先对 TCP Vegas 与 TCP Reno 共存时各链路的平均吞吐量进行分析,仿真结果如图 2 所示。从图 2 可以看出,在链路带宽、传输延迟等因素相等的条件下,TCP Reno 的平均吞吐量为 1 250(数据包/s),远远大于 TCP Vegas 的平均吞吐量 250(数据包/s)。

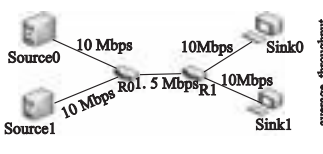

图1　网络拓扑结构

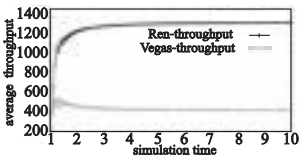

图2　TCP Vegas与TCP Reno 共存时的平均吞吐量

再对 TCP Vegas 与 TCP Reno 共存时两种算法的拥塞窗口进行分析,仿真结果如图 3 所示。

从图 3 可以看出,虽然与 TCP Reno 相比,TCP Vegas 算法的拥塞窗口较稳定,但是它的主动拥塞避免机制也使得它将过多的带宽让给了 TCP Reno,从而在算法共存竞争带宽时处于下风。

1.5　α 与 β 对公平性的影响

经研究证明增大 TCP Vegas 的两个参数 α 和 β 可以提高其对带宽的竞争能力[7]。这是由于 α 和 β 限制了路由器缓存中的报文数量,增大 α 和 β 的数值就允许 TCP Vegas 算法在缓存中存放更多的数据包,也就可以获得更大的吞吐量。通过改变 α 和 β 的数值,分别进行仿真,TCP Vegas 与 TCP Reno 算法的吞吐量如表 1 所示。

表 1　α 和 β 对公平性的影响

α	1	2	4	6	8	12
β	3	4	6	8	10	14
Reno(*T*)	238	214	186	153	147	54
Vegas(*T*)	78	103	134	168	183	284

表 1 中 Reno(*T*)代表 Reno 的平均吞吐量;Vegas(*T*)代表 Vegas 的平均吞吐量。从表 1 可以看出,随着 α 和 β 的增大。TCP Vegas 平均吞吐量也逐渐增大,这说明增大 α 和 β 确实对增强 TCP Vegas 的带宽竞争能力有较大作用。

2　TCP Vegas-A 算法

TCP Vegas-A 算法也叫做自适应 TCP Vegas 算法,它是一种针对 TCP Vegas 算法的改进拥塞控制算法。在 TCP Vegas 中 α 和 β 分别被设置为 1 和 3,而 TCP Vegas-A 算法随着网络情况动态调整 α 和 β 的值,使之随时适应网络的要求。它主要修改了 TCP Vegas 在拥塞避免阶段的策略实现,慢启动和拥塞恢复阶段的策略与 TCP Vegas 保持一致。

TCP Vegas-A 在拥塞避免阶段的算法可表示如下:

```
if Diff < α{
if α > 1 and Th(t) > Th(t-rtt){
Cwnd = cwnd + 1;
}
else if α > 1 and Th(t) < Th(t-rtt){
Cwnd = cwnd - 1; //减小拥塞窗口
α = α - 1, β = β - 1;
}
else if α = =1
Cwnd = cwnd + 1;//增大拥塞窗口,拥塞窗口呈指数级增长
}
else if α < Diff < β{
if Th(t) > Th(t - rtt){
Cwnd = cwnd + 1;
α = α + 1, β = β + 1;
}
else if Th(t) < = Th(t - rtt){
No update of cwnd, α, β;//cwnd, α, β 均保持不变
}
}
else if Diff > β{
if(α > 1){
Cwnd = cwnd - 1;
α = α - 1, β = β - 1;
}
else{
      no update of cwnd, α, β
}
}
```

Th(*t*)是在时刻 *t* 时的实际吞吐速率,Th(*t*-rtt)是在 *t* 时刻之前的一个 RTT 时刻的实际吞吐速率,采用图 1 中的网络拓扑结构,在链路 1 中用 TCP Vegas-A 算法代替 TCP Vegas 算法,其他保持不变,对 TCP Vegas-A 算法与 TCP Reno 算法共存时的拥塞窗口进行分析,仿真结果如图 4 所示。

从图 4 中可以看出,虽然与 TCP Vegas 的窗口相比,TCP Vegas-A 拥塞窗口有一定的增加,但是仍然保持一个较小的数值,使得在与 TCP Reno 共存时仍处于劣势。尤其是在仿真的最后 50 ~ 70 s 拥塞窗口有较大的波动,影响了算法本身的稳定性。故此算法存在一定的不足。

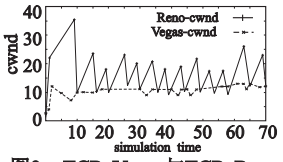

图3　TCP Vegas与TCP Reno 共存时拥塞窗口变化

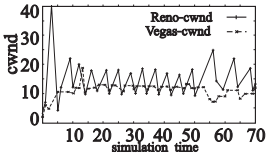

图4　TCP Vegas-A与TCP Reno 共存时拥塞窗口变化

3　TCP NewVegas 算法

TCP Vegas-A 算法主要是在 TCP Vegas 算法的基础上加入了对实际吞吐率的变化。如果吞吐量持续增加,说明网络带宽没有被完全利用。此时不考虑 α、β 的限制,增加拥塞窗口的大小以便充分探测网络可用空间。TCP Vegas-A 算法加强了对带宽的竞争能力,但是它也使窗口变化不稳定。这主要是由于拥塞窗口的增减程度过大,尤其在算法的最后单纯地减小拥塞窗口的大小和 α、β 数值,使得该算法占有带宽能力显著下降。

本文对 TCP Vegas-A 算法作出改进,提出了 TCP NewVegas 算法。算法的思想是针对它的稳定性问题,可以考虑适当减小拥塞窗口的变化程度。在 TCP Vegas-A 算法中,拥塞窗口的变化都是指数级地增加或减小,显得过于剧烈,可以考虑使其呈线性变化,这样可以保持拥塞窗口的稳定性。针对算法在最后部分对带宽的占有显著下降问题,可以加入吞吐量的变化来进一步进行控制。如果 t 时刻的吞吐量大于之前的吞吐量说明还有可以利用的带宽,增加拥塞窗口和 α、β 的数值;反之,线性减小拥塞窗口的大小。

TCP NewVegas 算法可表示如下:

```
if Diff < α{
if α > 1 and Th(t) > Th(t-rtt){
Cwnd = cwnd + 1;
}
else if α > 1 and Th(t) < Th(t-rtt){
Cwnd = cwnd - 1/cwnd;//减小拥塞窗口,拥塞窗口呈线性变化
no update of α,β;
}
else ifα = = 1
Cwnd = cwnd + 1;
}
else if α < Diff < β{
if Th(t) > Th(t - rtt){
Cwnd = cwnd + 1;
α = α + 1, β = β + 1;
}
else if Th(t) < = Th(t - rtt){
no update of cwnd, α, β;//cwnd、α、β 均保持不变
}
}
else if Diff > β{
if α > 1 and Th(t) > Th(t - rtt){
Cwnd = cwnd + 1;
α = α + 1,β = β + 1;
}
else If α > 1 and Th(t) < Th(t - rtt){
Cwnd = cwnd - 1/cwnd;
no update of α,β;
}
```

TCP NewVegas 算法的思想是当 diff < α 时,若吞吐量继续增大,则增大拥塞窗口(增大一个 cwnd);若吞吐量减小,说明有一定程度的网络拥塞发生则减小拥塞窗口(减小 1/cwnd);当 α < diff < β 时,若吞吐量增大,则增大拥塞窗口,同时增大 α 和 β 的数值,否则保持不变;当 diff > β 时,若吞吐量继续增加,则增大拥塞窗口 α 和 β;否则,减小拥塞窗口的数值。算法的目的是最大限度地利用网络空余带宽,始终保持带宽占有的稳定性。

采用图 1 中的网络拓扑结构,在链路 1 中用 TCP NewVegas 算法代替 TCP Vegas-A 算法,其他保持不变,对 TCP NewVegas 算法与 TCP Reno 算法共存时的拥塞窗口进行仿真。结果如图 5 所示。

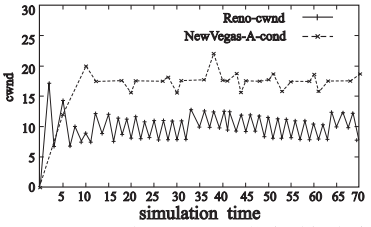

图5　TCP NewVegas与TCP Reno共存时拥塞窗口变化

从图 5 中可以看出,TCP NewVegas 拥塞窗口比较稳定,而且远大于 Reno 窗口的平均值,说明 TCP NewVegas 算法在带宽竞争能力上显著增加。

4　结束语

本文对 TCP Vegas 拥塞控制算法进行了研究,重点研究了 TCP Vegas-A 算法,针对此算法的不足作出相关改进,提出了 TCP NewVegas 算法。通过仿真软件 NS2 对 TCP NewVegas 算法的仿真分析,证明了此算法与 TCP Vegas-A 算法比较而言,在带宽竞争能力和稳定性方面有很大的提高。

参考文献:

[1] THOMPSON K, MILLER G, WILDER R. Wide-Area Internet traffic patterns and characteristics[J]. IEEE Network, 1997, 11(6): 10-23.

[2] CHENG Lai-yuan, LI Yao-chang. Performance comparison between TCP Reno and TCP Vegas[J]. Computer Communications, 2002, 25: 1765-1773.

[3] SRIJITH K N, JACOB L, ANANDA A L. TCP Vegas-A: improving the performance of TCP Vegas[J]. Computer Communications, 2005, 28(6): 429-440.

[4] BONALD T. Comparison between TCP Reno and TCP Vegas: efficiency and fairness[J]. Performance Evaluation, 1999, 36(37): 307-332.

[5] BRAKMO L S, PETERSON L L. TCP Vegas: end to end congestion avoidance on a global Internet[J]. IEEE Journal on Selected Areas in Communications, 1995, 13(8): 1465-1480.

[6] De VENDICTIS A, BAIOCCHI A, BONACCI M. Analysis and enhancement of TCP Vegas congestion control in a mixed TCP Vegas and TCP Reno network scenario[J]. Performance Evaluation, 2003, 53(3-4): 225-253.

[7] 秦楠,郑应平. 基于 TCP Vegas 与 TCP Reno 的一种改进拥塞控制算法[J]. 计算机工程与科学,2007,29(11):23-25.

[8] 徐昌彪,隆克平,杨士中. 基于双重 AIMD 的 TCP 拥塞控制[J]. 计算机研究与发展,2003,23(8):6-9.