文章编号:1001-9081(2010)02-0309-03

无线异构网络环境中基于拥塞状态区分的 TCP Vegas 改进算法

李 鹏¹, 陈元琰¹, 罗晓曙²

(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 电子工程学院, 广西 桂林 541004)

(lipengq@qq.com)

摘 要:在无线网络环境中, TCP Vegas 应用时会受到无线信道干扰和噪声的影响, 对往返延时(RTT)难以进行准确的估计, 导致其性能大幅度降低。结合 TCP New Vegas 和 TCP Vegas A+ 等的优点提出一种改进算法 TCP Vegas-P。该算法针对慢启动过早结束和拥塞避免阶段拥塞出现在反向链路上导致吞吐量下降的问题, 以及在与 NewReno 共存时公平性恶化的问题进行了综合的改进。经仿真实验, 改进的算法在无线网络中能够进行比较好的 RTT 估计, 对解决上述 Vegas 存在的问题达到了良好的效果。

关键词:无线网络; TCP Vegas; 往返延时; 相对延时; 非对称网络

中图分类号: TP393.07; TN925.93 **文献标志码:** A

TCP Vegas-P: Enhanced TCP Vegas congestion control algorithm based on congestion status differentiation over wireless heterogeneous network

LI Peng¹, CHEN Yuan-yan¹, LUO Xiao-shu²

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

2. College of Electronic Engineering, Guangxi Normal University, Guilin Guangxi 541004, China)

Abstract: In wireless network, channel interference and noise cause significant performance degradation of TCP. In order to solve the problems existing in the application of TCP Vegas, based on TCP New Vegas algorithm, combining the advantages of TCP Vegas A+ algorithm, an improved algorithm named TCP Vegas-P was proposed. It was improved to solve the premature end of the slow start phase and congestion avoidance in the reverse link congestion resulting in the throughput decline, and the fairness deterioration when it was accompanied by NewReno. The simulation results show that the improved algorithm in the wireless network can better estimate Round-Trip Time (RTT), and resolve the problems about Vegas, and can be well applied to wireless network environment.

Key words: wireless network; TCP Vegas; Round-Trip Time (RTT); relative delay; asymmetric network

0 引言

尽管较之其他算法, Vegas 在无线的网络环境中有良好的性能。然而当 TCP Vegas 在无线网络中应用时, 因线路质量差(噪声干扰产生的数据包丢失频繁发生), 总是难以准确地估计传播延迟的最小值(Base Round-Trip Time, BaseRTT), 这也是 Vegas 出现缺陷的主要原因。Vegas 中存在的很多问题都与 RTT 估计的精度有关, 而且 TCP Vegas 是根据传播延迟的最小值 BaseRTT 来修改窗口尺寸, 又由于 RTT 与网络运行情况有密切关系, 所以 RTT 估计和 BaseRTT 估计的精度对 Vegas 各项性能指标的改进是至关重要的。

F. Kurose 等人^[1]指出: 当一个数据包经过一系列中间节点从源节点到达目的节点时, 其总的时延包括节点的处理时延、排队时延、传输时延、传播时延, 而这其中包括了会引起判断网络可用带宽中的主要误差的一些延时。比如在慢启动的窗口指数增长中, 数据包在网络中的暂时堆积是导致慢启动提前结束的主要原因; 在拥塞避免阶段, 反向链路的拥塞情况会影响 Vegas 的吞吐量, 特别是当拥塞发生在非对称网络中的 ACK 应答链路上时, 情况更加严重。本文针对 TCP Vegas 在无线网络环境中因难以精确地估计 RTT 和 BaseRTT 所带

来的问题, 分别对慢启动、拥塞控制及快速重传进行了改进, 最后将其结合得到了 TCP Vegas-P 算法。经仿真实验, 改进的 Vegas 算法相比原来 Vegas 算法, 在延时估计上能滤去干扰, 同时结合低通滤波器来处理 RTT, 使得估计曲线更平滑, 实现了 RTT 的精确估计, 进而达到在无线网络环境中改进算法性能的目的。

1 TCP Vegas 算法简介

为了提高吞吐率、降低分组丢弃率, TCP Vegas^[2] 采用了 3 种与众不同的技术^[3]。然而自 1995 年提出至今, 仍然没有取代 TCP Reno 成为现今最广泛使用的 TCP 拥塞控制算法。文献[4]中详细地说明了 TCP Vegas 协议中存在的问题: 1) 更变路径的问题; 2) 持续拥塞的问题; 3) 不公平性的问题; 4) 网络非对称性; 5) 不兼容性。

每当收到一个 ACK, Vegas 计算期望吞吐量和实际吞吐量的差值 $Diff$, 见式(1), (2), $Diff$ 的值就是这条链路瓶颈的队列长度, Vegas 希望能让队列长度维持在 α 和 β 之间, 避免数据包丢失, 从而进一步提高链路的利用率。BaseRTT 表示所有已测量 RTT 的最小值; $0 < \alpha < \beta$ 是两个门限, 经验值分别为 1 和 3。关于这两个参数的实际意义, 在文献[3]中说明为

收稿日期: 2009-07-29; 修回日期: 2009-09-13。

基金项目: 广西自然科学基金资助项目(0728099); 广西教育厅研究生创新项目(2008106020812M261)。

作者简介: 李鹏(1984-), 男, 湖北洪湖人, 硕士研究生, 主要研究方向: 计算机网络拥塞控制; 陈元琰(1961-), 男, 福建仙游人, 教授, 博士, 主要研究方向: 计算机图形学、计算机网络; 罗晓曙(1961-), 男, 湖北应城人, 教授, 博士, 主要研究方向: 非线性动力学。

沿途路由器中缓存内积累的包数; $Diff = UExpected - UActual$, 即期望速率与实际速率的差值, 而期望速率和实际速率分别由下式确定:

$$UExpected = \frac{cwnd_{out}}{BaseRTT}, UActual = \frac{cwnd_{out}}{RTT} \quad (1)$$

$$\Delta = Diff \times BaseRTT = cwnd_{out} \times \frac{RTT - BaseRTT}{RTT} \quad (2)$$

从上面的描述可以看出, 一旦 TCP Vegas 的窗口尺寸收敛于一个平衡点, 它将不会再像 TCP Reno 那样出现很大的波动。

2 改进的 TCP Vegas-P 算法

2.1 慢启动算法

在慢启动中, 最大的问题就是慢启动过早结束。在网络连接开始或超时重传时, 启动慢启动算法并没有可用带宽的预知信息, 为了找出一个不会导致数据包丢失的合适的可用带宽, 文献[5]以介于指数增长和直线增长之间的稳定增长来增加拥塞窗口, 在大的带宽时延乘积网络中有良好的效果, 但是没有考虑无线网络中随机丢包的情况。文献[6]分析指出了慢启动分为两个阶段: 增长期和观测期。

在增长期内, 由于拥塞窗口呈指数的增长, 短时间内数据包发送比较密集, 很容易出现数据包在网络中暂时堆积, 导致排队时延暂时性增大。而且随着窗口的增大这种现象越来越凸现, 在 $cwnd$ 仍较小时提前发生 $\Delta > \gamma$ (通常设为 1) 会导致慢启动过早地结束。在 TCP Vegas-P 中, 利用 RTT 和 $cwnd$ 之间成正比的线性关系, 从 RTT 中消除掉暂时性排队时延增量 $k \cdot cwnd$, 使得网络能够充分利用带宽, 吞吐量得到提高。本文改进的慢启动算法, 是以观测期 ($\Delta < \gamma/2$) 和整合期 ($\gamma/2 < \Delta < \gamma$) 为阶段来进行的, 在 $\Delta < \gamma/2$ 时, $cwnd$ 和 Δ 都还较小, 网络也未达到满负荷, 在这个阶段计算 k 比较准确。在 $\gamma/2 \leq \Delta < \gamma$ 时, 暂时性排队时延增量随着窗口急剧增加而增大, 在这个阶段通过在观测期的 k 去除掉 RTT 中的暂时性排队时延增量得到准确的 Δ , 消除了慢启动中导致过早结束的主要误差。慢启动的伪代码如下:

```
if (Status_ == 0) // 慢启动在观测期内时
    if (cwnd <= 3) Rtt* = RTTf, W = cwnd;
    // RTT 和 cwnd 之间成正比
    k = (RTTf - Rtt*) / (cwnd - W + 0.5);
    if (Δ >= γ/2) Status_ = 1;
else // 慢启动在整合期内时, 消除掉暂时性排队时延增量
    Δ = cwnd * (RTTf - tff - incr * t) / (RTTf - incr * cwnd);
    if (Δ > γ)
        Ssthresh_ = 2, cwnd -= 1/cwnd;
    if (cwnd < 2) cwnd = 2;
    incr = 0;
    else incr = 8/cwnd;
```

其中: $Status_$ 是用来表示慢启动的两个阶段, 为 0 表示观测期, 为 1 表示整合期; $Ssthresh_$ 为慢启动阈值, RTT_f 表示数据报在正向链路上所需要的传输时间, tff 表示正向固定传输延时。

2.2 拥塞避免算法

目前拥塞控制对无线网络的改进主要基于 TCP Reno, 而较少基于 TCP Vegas。本文提出的 TCP Vegas-P 的拥塞避免算法, 针对在无线网络环境中很难准确估计 RTT 和 $BaseRTT$ 所带来的问题, 进行了相应的改进。特别是当 TCP Vegas 连接已经建立并收敛到稳定状态时, 反向链路出现流量瓶颈造成 RTT 增大而导致发送端降低其传输速率, 最终使链路吞吐量下降。结合了文献[7], 使用了“相对延时”来修改 Vegas

原有的拥塞避免中估计链路数据量的算法, 从中去除了反向链路对吞吐量的影响。如果在计算期望速率和实际速率时, 只考虑正向链路的传输时间, 那么调整后的 $UExpected$ 和 $UActual$ 的计算式表示如下:

$$UExpected = \frac{cwnd_{out}}{tff}, UActual = \frac{cwnd_{out}}{RTT_f} \quad (3)$$

相对延时 D_q 为连续两个分组的 ACK 接收时间间隔与这两个分组的发送时间间隔之差。在判断网络拥塞状态时, 运用相对延时 D_q 对拥塞状态进一步细分的方法, $D_q = 0$ 说明分组没有经过队列时延, 网络处于稳定状态; $D_q > 0$ 说明分组由于中间节点的队列时延增加而导致网络拥塞程度加剧, 网络处于拥塞增加状态; $D_q < 0$ 说明分组所经过的中间节点的队列时延在减少, 网络处于拥塞减小状态。这样, 在计算期望吞吐量和实际吞吐量时, 去除了反向链路上传输时延的影响, 即使当反向链路存在一定的拥塞, 导致往返时间 RTT 增大, 也不会引起链路上队列中数据包数量太多而导致吞吐量的下降。拥塞避免机制的伪代码如下:

```
if (Status_ != 0)
    if (Δ > β) // 队列长度大于 3 时
        if (Th(t) > Th(t-m)) incr = 8/cwnd, ++β, ++α;
        else if (Th(t) <= Th(t-m))
            // 通过相对延时, 对拥塞状态进一步分析
            if (Dq <= 0) incr = 0;
            else if (Dq > 0) cwnd -= 1/cwnd;
            if (cwnd < 2) cwnd = 2;
            --α, --β;
    else if (Δ < α) // 队列长度小于 1 时
        if (α > 1 && Th(t) > Th(t-m)) incr = 8/cwnd;
        else if (α > 1 && Th(t) <= Th(t-m))
            incr = 8/cwnd, --α, --β;
        else if (α == 1) incr = 8/cwnd;
    else // 队列长度介于 1 和 3 之间时
        if (Th(t) > Th(t-m)) incr = 8/cwnd, ++β, ++α;
        else if (Th(t) <= Th(t-m))
            if (Dq <= 0) incr = 8/cwnd, ++β, ++α;
            else if (Dq > 0) incr = 0;
    else
        Status_ = 0;
```

其中: $Th_{(t)}$, $Th_{(t-m)}$ 分别在时间 t 实际的吞吐率及时间 t 之前的一个 RTT 内测得的实际吞吐率。

3 仿真实验和性能分析

为了分析和验证改进算法的性能, 本文采用 NS2 对其进行仿真。网络拓扑结构如图 1 所示, 无线网络是 WLAN。网络有关参数取为: 有线链路(从发送端到基站 BS)的带宽为 100 Mbps, 单向传输时间为 30 ms; 无线链路(从基站 BS 到接收端)带宽为 2 Mbps ~ 45 Mbps, 单向传输时间为 0.01 ms; 发送数据包大小为 1000 B, 无线链路的位误码率在 0% ~ 5% 范围内变化, 主动队列管理 (AQM) 算法模拟使用 DT (Drop Tail) 队列管理策略。

3.1 不同误码率下几种算法并存时平均吞吐量比较

TCP Vegas、TCP Vegas-P 与 Newreno 共存时, 在不同的误码率下进行了吞吐量的比较。在实验中分别用 Vegas、Vegas-P 算法来连接 S_1 和 D, 用 NewReno 算法来连接 S_2 和 D, 进行 FTP 传输。从表 1 中可以看到, 随着误码率的增大, Vegas-P 的吞吐量虽有所下降, 但是相比 Vegas, 吞吐量一直保持着良好的增量。可以得出, 改进之后的 Vegas-P 在与 Newreno 共存时, 其性能有了较好的改善。

表1 两种算法共存时平均吞吐量比较

算法	误码率/%					
	0	1	2	3	4	5
Vegas/	3447/	1386/	1030/	903/	662/	552/
Newreno	214	247	274	267	266	229
Vegas-P/	4058/	1997/	1298/	931/	674/	614/
Newreno	229	250	254	263	222	223

图1 网络拓扑结构

3.2 信道误码率与吞吐量的关系比较

本部分对 NewReno、Vegas 和 Vegas-P 在不同误码率 (0% ~ 5%) 下单条流的平均吞吐量进行了统计比较,其结果显示在图 2 的信道误码率与吞吐量的关系上。分别用 NewReno、Vegas 及 Vegas-P 算法来连接节点 S_1 、D,并进行 FTP 传输。本文假设窗口大小是 100 B,路由器的缓存大小为 30,模拟时间为 100 s。可以看出:TCP Vegas 和 TCP Vegas-P 的吞吐量一直好于 Newreno。当误码率超过典型的 1% 时,TCP Vegas-P 优于其他两种算法。特别是在误码率为 2% 时,TCP Vegas-P 吞吐量有较好的优势。由此可以看出,在误码率较高时,Vegas-P 的吞吐量高于 TCP Vegas 和 Newreno。

图2 信道误码率与吞吐量

3.3 单向传输时间与吞吐量

该实验是误码率为 5% 时对 NewReno、Vegas 和 Vegas-P 在双向传输时间从 10 ms 变化到 200 ms 的吞吐量比较情况。如图 3 所示,随着传输时间的增加,虽然 Vegas-P 的吞吐量有明显的下降,但是在传输时间约为 135 s 之前,Vegas-P 的吞吐量要远远大于 NewReno 和 Vegas 的吞吐量。而且在这之后,Vegas-P 的吞吐量又开始慢慢回升,虽然在整体性能上不如 NewReno 稳定,但可以得出该算法对无线中高的误码率在长传输时延情况下有较好的适应能力。

3.4 信道容量与吞吐量

该部分是误码率为 5% 时对 NewReno、Vegas 和 Vegas-P 在无线链路容量由 2 Mbps 变化到 40 Mbps,误码率为 5% 下吞吐量的统计比较情况。从图 4 可以看出,随着信道容量的增加,TCP Vegas-P 可以获得更多的带宽。

3.5 非对称网络中性能比较

为了测试 Vegas-P 在拥塞避免阶段的改进关于非对称网络中拥塞出现在反向链路上的性能,本部分在实验中加入了一个 UDP 数据流。节点 D 和 S_2 之间传输 UDP 数据流,实验中 UDP 传输的数据源是 CBR (Constant Bit Rate) 业务。设定包大小为 1000 B,发送速率为 0.9 Mbps。其中节点 D 为 CBR

源, S_2 为 UDP 接收端。在仿真实验中,节点 S_1 从 0 s 开始建立到节点 D 的 TCP 连接,并开始发送数据,到 20 s 结束;节点 D 从 5 s 开始向节点 S_2 发送 UDP 数据报,15 s 结束。TCP Vegas 的拥塞避免机制和改进后 Vegas-P 的拥塞避免机制的吞吐量比较如图 5 所示,当反向链路不存在拥塞时,两者的吞吐量一样。但是 5 s 后,节点 D 开始发送数据,反向链路开始出现大量数据,可以看出,Vegas 原有的拥塞避免算法会导致吞吐量的明显下降,而改进后的 Vegas 拥塞避免算法能获得比原有的 Vegas 高得多的吞吐量,而在 15 s,节点 D 停止发送数据,Vegas 的吞吐量开始回升了。

图3 单向传输时间与吞吐量 (rate = 5%)

图4 信道容量与平均吞吐量 (rate = 5%)

图5 当拥塞出现在反向链路中吞吐量比较

4 结语

针对 TCP Vegas 在无线网络环境中难以精确地估计 RTT 和 BaseRTT 所带来的问题,本文分别对慢启动、拥塞控制及快速重传进行了改进。在前人研究成果的基础上,得到了新的改进算法 TCP Vegas-P。经仿真实验证明,在慢启动阶段,Vegas-P 去除了 RTT 中的暂时性排队时延增量得到较准确的期望的吞吐量和实际吞吐量的差值,消除了慢启动中导致慢启动过早结束的主要误差。在拥塞避免阶段,主要解决了在非对称网络中拥塞发生在反向链路上时吞吐量下降的问题,即当拥塞发生在反向链路时,由拥塞所产生的时延不应该计算到总的 RTT 中。另外引入了相对延时,对网络状态进行进一步细分,以较为准确地判断网络拥塞状态,合理调整拥塞窗口,使得在于 NewReno 共存的网络环境中,能竞争到较大的

(下转第 326 页)

平均连通度 <10),会有一些未知节点的可连接节点较少,缺乏足够的参考节点信息,从而导致定位误差较大。当锚节点固定时,随着网络平均连通度的增加,四种算法的定位误差都明显下降。当连通度为12或者更高时,本文改进算法的平均定位误差可以降至35%以下。同时可以看出,在连通度相同的情况下,本文改进算法的定位精度始终优于前三种算法。

图3 四种算法在不同连通度下的平均定位误差

图4 不同连接度下的平均定位误差

分析:图4说明了本文改进算法在不同连通度下的平均定位误差的比较情况。从仿真结果可知,四种不同的平均连通度下,定位精度都随着锚节点的增加而提高。当锚节点增加到40个时,定位精度提高的幅度逐渐变小,平均定位误差变化趋于平稳。

4 结语

本文对DV-HOP算法产生误差的原因进行了理论分析,并针对误差原因提出了相应的改进方法。经过仿真验证,改进算法的定位精度优于原算法,同时具有良好的扩展性。当节点数目不断增加时,定位精度明显提高。当锚节点数量较少时,改进算法的定位性能尤为突出。

由于改进算法的计算量会有所提高,从而会提高算法的开销。未来的研究将集中于如何在保持较好的定位精度的同时,降低通信计算量。

参考文献:

- [1] 孙利民,李建中,陈渝,等. 无线传感器网络[M]. 北京:清华大学出版社,2005:146-168.
- [2] HE T, HUANG C D, BLUM B M, *et al.* Range-free localization schemes in large scale sensor networks[C]// Proceedings of the 9th Annual International Conference on Mobile Computing and Networking. San Diego: ACM Press, 2003: 81-95.
- [3] PRIYANTHA N B, CHAKRABORTHY A, BALAKRISHNAN H. The cricket location-support system[C]// Proceedings of International Conference on Mobile Computing Networking. New York: ACM, 2000: 32-43.
- [4] SAVVIDES A, HAN C C, SRIVASTAVA M B. Dynamic fine-grained localization in Ad-Hoc networks of sensors[C]// Proceedings of 7th Annual International Conference on Mobile Computing and Networking. New York: ACM, 2001:166-179.
- [5] 王福豹,史龙,任丰原. 无线传感器网络中的自身定位系统和算法[J]. 软件学报,2003,16(5):857-868.
- [6] CHEN HONG-YANG, SEZAKI K, DENG PING, *et al.* An improvement DV-HOP algorithm for wireless sensor networks[EB/OL]. [2009-06-01]. <http://www.ee.cityu.edu.hk/~hcsoc/icica08.pdf>.
- [7] 林金朝,刘海波,李国军,等. 无线传感器网络中DV-HOP节点定位改进算法研究[J]. 计算机应用研究,2009,26(4):1272-1275.
- [8] CHAN Y T, HO K C. A simple and efficient estimator for hyperbolic location[J]. IEEE Transactions on Signal Processing, 1994, 42(8): 1905-1915.
- [9] 魏木生. 广义最小二乘为问题的理论和计算[M]. 北京:科学出版社,2006:107-114.
- [10] 于宁. 无线传感器网络定位优化方法[D]. 北京:北京邮电大学,2008:15-17.

(上接第311页)

带宽,并获得良好的公平性。在改进的算法中,将TCP Vegas-P中所有涉及到增加 $cwnd$ 的行为改变为 $cwnd + 8/cwnd$,减小 $cwnd$ 的行为改变为仅降低 $1/cwnd$,使得网络更加平稳^[8]。本文给出了详细的算法描述,综合改进了Vegas算法在无线网络中的性能,提高了带宽竞争性。文中改进的TCP Vegas-P协议还具有对现有协议的改动较少,无须增加TCP包头内容的优点。

参考文献:

- [1] KUROSE J F, ROSS K W. Computer networking - A top-down approach featuring the Internet[M]. Reading, Massachusetts: Addison Wesley/Longman, 2001:38-44.
- [2] BRAKMO L S, O'MALLY S W, PETERSON L. TCP Vegas: New techniques for congestion detection and avoidance[J]. IEEE/ACM Transactions on Networking, 1994, 24(4):1024-1035.
- [3] BRAKMO L S, LARRY 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.
- [4] 雷永惠. 延时估计及改进的TCP Vegas算法[D]. 大连:大连理工大学,2005.
- [5] HO C Y, CHAN Y C, CHEN Y C. Gallop-Vegas: An enhanced slow-start mechanism for TCP Vegas[J]. Journal of Communications and Networks, 2006, 8(3): 351-359.
- [6] 顾明,张军,苏东林. 大带宽时延网络TCP Vegas自适应慢启动算法[J]. 电讯技术, 2007, 47(2): 27-30.
- [7] 王建峰,黄国策,陈才强,等. GEO卫星网中基于拥塞状态区分的TCP Vegas改进算法[J]. 计算机科学, 2008, 35(11): 70-73.
- [8] 拱长青,赵志刚,王光兴,等. LEO卫星网络中TCP Vegas拥塞控制算法研究[J]. 小型微型计算机系统, 2006, 27(1): 54-57.