

# 一种基于噪声模型带宽估计的拥塞控制改进算法

史士杰 薛凯 邵艳 张国勇

(中国矿业大学信电学院 徐州 221008)

**摘要** 无线传输中,拥塞控制算法 Ntcp 估计带宽波动性较大且估计值偏低。提出了 Ntcp 改进算法 Ntcp',它通过对发送端发送数据量更精确的计算,来消除带宽估计的波动性和纠正估计值偏低的问题。理论分析和仿真结果表明 Ntcp' 算法效果显著。

**关键词** 拥塞控制,带宽估计,无线网络,TCP

**中图分类号** TP393 **文献标识码** A

## Improved Algorithm of Congestion Control Based on Bandwidth Estimation

SHI Shi-jie XUE Kai SHAO Yan ZHANG Guo-yong

(School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221008, China)

**Abstract** In the wireless environment, the method of bandwidth estimation for TCP based on channel noise model(Ntcp) has two disadvantages, one is the low estimated bandwidth and the other is the high fluctuation of the estimated bandwidth. With that, we proposed an improved algorithm(Ntcp') based on the amount of data sent by the sender. The analysis and simulated results show that Ntcp' works well.

**Keywords** Congestion control, Bandwidth estimated, Wireless network, TCP

### 1 引言

鉴于 TCP 协议在无线网络中的广泛应用, TCP 传输协议中的拥塞控制算法日益成为关注的焦点。传统拥塞控制算法,如 Reno, New Reno, SACK<sup>[1-3]</sup> 均采用加性增加乘性减小(AIMD)的滑动拥塞窗口机制来探测网络有效带宽, AIMD 机制可以保证发送的数据不过分超过网络容量。但无线网络中一些数据包的丢失并非拥塞引起,此时 AIMD 机制仍会降低发送端拥塞窗口,进而造成网络吞吐量偏低,不能有效利用带宽资源。为此人们提出各种拥塞控制算法来自适应地调整网络拥塞窗口(cwnd)和拥塞阈值(ssthresh),如 TCP Veno, Freeze TCP, TCPW<sup>[4-6]</sup> 等,大大提高了无线网络性能。

基于噪声模型的 TCP 有效带宽估计算法——Ntcp<sup>[7]</sup> 即作为一种基于带宽估计的加性增加自适应减小窗口(AIADM)的拥塞控制算法,此算法对估计带宽的依赖性较大,在随机丢包情况下 Ntcp 的带宽估计值波动性较大且偏低。本文利用 OPNET 仿真工具,在 Ntcp 模型基础上,提出一种基于发送端发送数据量的带宽估计算法 Ntcp', 并推导了 Reno, Ntcp 和 Ntcp' 算法下丢包率与吞吐量之间的关系,为如何提高无线网络中拥塞控制算法的性能提供了依据。

### 2 Ntcp 算法介绍

一般的通信系统有信道和噪声,一个端到端 TCP 连接亦存在可导致丢包的背景流量和误码。因此,可将 TCP 连接中导致拥塞的背景流量和传输误码等效为加性噪声。在通信信

道模型中,用“0”表示无数据发送,“1”表示有数据发送。由已知的离散信道容量模型可以推知 Ntcp 有效带宽估计等价模型<sup>[7]</sup>为:

$$\hat{B}_{we} = H_t(x) - H_t(x/y) = r[H(x) - H(x/y)] \quad (1)$$

其中,

$$\begin{cases} H(x) = -\sum_{j=0}^1 [P(x_j) \log_2 P(x_j)] \\ H(x/y) = -\sum_{j=0}^1 \{P(y_j) \sum_{i=0}^1 [P(x_i/y_j) \log_2 P(x_i/y_j)]\} \end{cases} \quad (2)$$

由于发送“0”和“1”的概率相同,可得:

$$\begin{cases} H(x) = 1 \\ H(x/y) = -\frac{1}{2} \{ [P_e \log_2 P_e] + [1 - P_e] \log_2 [1 - P_e] \} \end{cases} \quad (3)$$

因此估计的有效带宽为:

$$\begin{aligned} \hat{B}_{we} &= r \left\{ 1 + \frac{1}{2} P_e \log_2 P_e + \frac{[1 - P_e]}{2} \log_2 [1 - P_e] \right\} \\ &\approx r \left\{ 1 + \frac{1}{2} P_e \log_2 P_e \right\} \quad \text{当 } P_e \ll 1 \end{aligned} \quad (4)$$

式中,  $r$  是单位时间内传输的数据包数,则有:

$$r = \frac{N}{T_i} = \frac{(ack_i - ack_{i-1})}{T_i * packte\_size} \quad (5)$$

式中,  $N$  为两次相邻的数据丢失之间成功传输的数据包;  $ack_i$  为第  $i$  个丢失数据的确认号;  $T$  为第  $i$  个丢包周期的持续时间;  $packte\_size$  为一个数据包的大小。

传统拥塞控制算法在监测到丢包和计时器超时,会盲

到稿日期:2010-06-08 返修日期:2010-10-11 本文受江苏省产学研联合创新资金项目(BY2009114)资助。

史士杰(1986-),男,硕士生,主要研究方向为无线网络拥塞控制、无线网络安全, E-mail: sscumt@163.com; 薛凯(1986-),男,硕士生,主要研究方向为计算机网络通信; 邵艳(1987-),女,硕士生,主要研究方向为无线通信; 张国勇 工程师,主要研究方向为计算机网络。

目地降低拥塞窗口和慢启动阈值。Ntcp 算法则是根据计算出的实际可用带宽,动态调整拥塞窗口和慢启动阈值,避免 TCP 链路中数据量出现剧烈波动,有效提高了网络利用率。另外,Ntcp 仍采用 TCP Reno 的慢启动、快速重传和超时重传措施,因此具有很好的兼容性和友好性。

### 3 改进算法(定义为 Ntcp')

#### 3.1 算法介绍

当发送端检测到数据丢失信息后进行数据重传时,重传数据占用一定网络带宽。进行带宽估计时不考虑重传数据,会降低带宽估计值。当网络丢包率较大时,表现更为严重。另外,丢包发生时接收到的数据序列号失序,接收端会暂时缓存或丢弃这些数据。在重传开始至网络恢复正常的这段时间内传输的数据,在计算带宽时如果不被考虑,则网络中流量瞬时减至很小,致使带宽估计值波动较大。针对以上问题,我们在 Ntcp 算法基础上提出了 Ntcp',根据发送端持续发送的数据量进行带宽估计来减小带宽估计的波动性,可更真实地估计带宽值。

Ntcp'中,定义  $r$  为单位时间内传输的数据包数,则  $r = d/\Delta t$ ,  $d$  是一个丢包周期内传输的数据量,  $\Delta t$  为一个丢包周期的持续时间。带宽估计由包丢失事件触发,每一个丢包周期计算一次。算法采用两个变量:一个计时器变量和一个计数器变量。在一次数据丢包之后,计时器记录下此时的丢包时刻,计数器从 0 开始计数。在以后的过程中,发送端每发送出去一个数据包,计数器就加 1。直到再次发现有数据丢失时,得到相邻两次丢包的时间差  $\Delta t$  和计数器 count 的值,求得  $r$ ,由式(6)表示。然后计时器记下第二次数据丢失的时刻,计数器归 0。

$$r = \frac{d_i}{\Delta t} = \frac{\text{count}_i}{T_i} \quad (6)$$

将式(6)代入式(4),得到 Ntcp'的估计带宽:

$$Bwe = \frac{\text{count}_i}{T_i} \left\{ 1 + \frac{1}{2} P_e \log_2 P_e \right\} \quad (7)$$

#### 3.2 性能分析

本文在瓶颈带宽分别为 10M 和 50M 及存在 UDP 流的情况下进行 Ntcp'带宽估计的仿真,结果如图 1、图 2 所示。

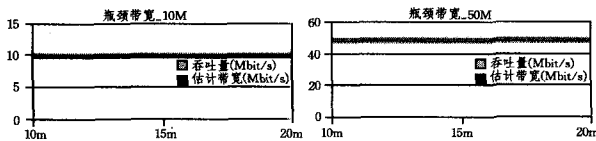


图 1 不同瓶颈下的带宽估计

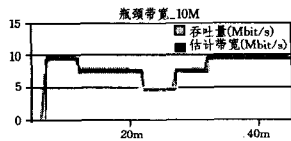


图 2 UDP 流下的带宽估计

由图 1 和图 2 知,Ntcp'算法估计的带宽值能有效反映链路的状态,采用该算法的连接可有效提高网络资源利用率。存在 UDP 流的情况下,估计的带宽亦能准确跟踪网络实际带宽。

### 4 吞吐量与丢包率关系

#### 4.1 Reno 算法吞吐量和丢包率关系

Reno 算法下丢包率与拥塞窗口的关系及最大窗口

( $w_{max}$ )、最小窗口( $w_{min}$ )、平均拥塞窗口( $\bar{w}$ )和吞吐量( $T_h$ )的相互关系由式(8)表示:

$$\begin{cases} \frac{1}{p_e} = \frac{(w_{max} + w_{min})(w_{max} - w_{min} + 1)}{2} \\ \bar{w} = \frac{w_{max} + w_{min}}{2} \\ w_{min} = \frac{w_{max}}{2} \\ T_h = \frac{\bar{w}}{Rtt} \end{cases} \quad (8)$$

从而网络的吞吐量为:

$$T_h = \frac{\bar{w}}{Rtt} = \frac{3}{2 * Rtt} \sqrt{\frac{2}{3 p_e}} \quad (9)$$

#### 4.2 Ntcp, Ntcp' 算法吞吐量与丢包率关系

(1)假设中间路由由节点缓存足够大。

(2)假设接收端缓存足够大。

(3)假设每个包的丢失都是独立的,不具有相关性,即按发送数据比例丢失数据包。

Ntcp, Ntcp'算法下丢包率与拥塞窗口的关系以及  $w_{max}$ ,  $w_{min}$ ,  $\bar{w}$  和  $T_h$  的相互关系由式(10)表示:

$$\begin{cases} \frac{1}{p_e} \approx \frac{w_i^2 \max - w_i^2 \min}{2} \\ \bar{w}_i = \frac{w_i \max + w_i \min}{2} \\ \frac{1}{p_e} * \frac{1}{w_i \max - w_i \min + 1} = w_i \min \\ T_h = \frac{\bar{w}_i}{Rtt} \end{cases} \quad (10)$$

从而得吞吐量与  $p_e$  之间的关系为:

$$T_h = \frac{\bar{w}}{Rtt} = \frac{1}{Rtt} \left( \frac{1}{4 p_e^2} + \sqrt{\left( \frac{1}{4 * p_e^2} \right)^2 - \left( \frac{1}{6 * p_e} \right)^3} \right)^{\frac{1}{3}} + \frac{1}{Rtt} \left( \frac{1}{4 p_e^2} - \sqrt{\left( \frac{1}{4 * p_e^2} \right)^2 - \left( \frac{1}{6 * p_e} \right)^3} \right)^{\frac{1}{3}} \quad (11)$$

### 5 仿真分析

#### 5.1 理论吞吐量与实际吞吐量比较

瓶颈带宽分别为 10M 和 50M,  $Rtt = 0.1s$ , 丢包率分别为 0.1%, 0.5%, 1%, 2%。Reno, Ntcp 和 Ntcp' 3 种算法下,根据以上推导出的吞吐量与丢包率关系,计算出的吞吐量与实际仿真的吞吐量进行比较,如图 3、图 4 所示。

由图 3、图 4 知,不同丢包率下的经典 Reno 算法,理论推导吞吐量与实际仿真吞吐量相吻合,同一丢包率条件下吞吐量几乎相同。Ntcp 算法亦如此。对于本文所提的 Ntcp' 算法,不同丢包率条件下理论推导吞吐量与实际仿真吞吐量仍然接近,足以表明本文推导的网络吞吐量和丢包率之间的关系是正确的。在 10M 和 50M 两种不同瓶颈带宽条件下仿真时吞吐量相当,说明当  $Rtt$  一定时, Reno, Ntcp 和 Ntcp' 算法下网络的吞吐量仅与丢包率有关。

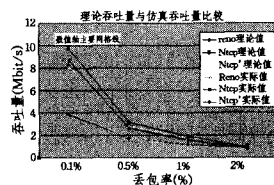


图 3 10M 瓶颈带宽下理论吞吐量与仿真吞吐量比较

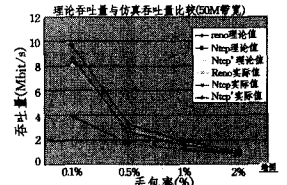


图 4 50M 瓶颈带宽下理论吞吐量与仿真吞吐量比较

## 5.2 Ntcp, Ntcp' 带宽估计的性能比较

设瓶颈带宽为 10M, 往返时延  $Rtt=0.1s$ , 在丢包率分别为 0.1%, 0.5%, 1% 和 2% 条件下进行 Ntcp 和 Ntcp' 带宽估计的仿真, 比较结果如图 5 所示。

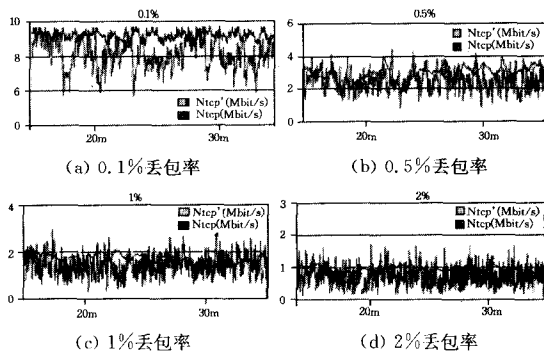


图 5 Ntcp, Ntcp' 带宽估计

由图 5 可知, 不同随机丢包率下, Ntcp' 估计带宽的波动比 Ntcp 算法下小得多, 估计的带宽值也较 Ntcp 高, 更接近实际值, 说明了 Ntcp' 改进算法非常有效。

## 5.3 Reno, Ntcp 和 Ntcp' 的性能

瓶颈带宽为 10M,  $Rtt=0.1s$  时, 设置不同的丢包率, 对 3 种算法进行对比仿真, 得到的吞吐量如图 6 所示。

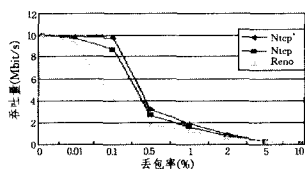


图 6 不同丢包率条件下的吞吐量

由图 6 可以看出同一丢包率时, Ntcp 算法下的吞吐量明显高于 Reno 算法, Ntcp' 算法下的吞吐量又较 Ntcp 高。可以说明 Ntcp 和 Ntcp' 算法对具有无线随机丢包的网络环境

具有较好的适应性, Ntcp' 改进算法在 Ntcp 的基础上又提高了网络的性能, 改进算法是有效的。

**结束语** 本文在 Ntcp 算法基础上提出了新的拥塞控制算法。理论分析和仿真结果表明 Ntcp 算法在无线网络下有较好适应性, 而改进后的 Ntcp' 算法表现出更好的性能。通过对发送端数据量的精确检测, 得到网络实际可用带宽, 很好地控制了拥塞窗口的大小, 提高了带宽的利用率。另外, 通过分析推导了 Reno, Ntcp 和 Ntcp' 3 种算法下吞吐量与丢包率的关系, 发现并实验证明了无论是传统的拥塞控制算法 Reno 还是本文中 Ntcp, Ntcp' 算法, 在忽略超时丢包情况下, 当往返时延一定时, 网络的吞吐量仅由丢包率决定。相信, 这对今后 TCP 拥塞控制算法的性能改进具有重大意义。

## 参考文献

- [1] 周慧斌, 周铁军, 管贵秋, 等. 基于仿真的 TCP 拥塞控制算法研究[J]. 计算机仿真, 2007, 24(12): 121-124
- [2] Allman M, Paxson V, Stevens W R. TCP Congestion Control [C]//RFC 2581. IETF, April 1999
- [3] Floyd S, Henderson T. The NewReno Modification to TCP's Fast Recovery Algorithm[C]//RFC 2582. IETF, April 1999
- [4] Claudio C, Mario G, Saverio M, et al. TCPWestwood: End-to-End Congestion Control for Wired/Wireless Networks [J]. Wireless Networks, 2002, 8(5): 467-479
- [5] Tom G, James M, Phatak D S, et al. A True End-to-End TCP Enhancement Mechanism for Mobile Environments[C]// Proceedings of IEEE INFOCOM, v3. 2000: 1537-1545
- [6] Cheng Peng-fu, Liew S C. TCP VenO: TCP Enhancement for Transmission over Wireless Access Networks[J]. IEEE Journal on Selected Areas in Communications, 2003, 21(2): 216-228
- [7] 李世银, 王秀娟, 徐冬, 等. 一种基于噪声模型的 TCP 有效带宽估计方法研究[J]. 系统仿真学报, 2008(19): 5058-5061

(上接第 55 页)

18.7%, 10.5% 和 17.6%。而误报率则明显降低, 误报率的最大降低率达到 84.6%。由此说明, PSO-based k-means 算法克服了 k-means 算法容易陷入局部最优解这一缺点, 聚类效果优于 k-means 算法, 从而提升了检测效果。

**结束语** k-means 算法初始聚类中心随机选择, 聚类结果随初始聚类中心的不同而波动, 从而导致聚类结果不稳定。本文提出的 PSO-based k-means 算法使用 PSO 算法优化生成初始聚类中心, 得到的聚类结果全局最优, 使得聚类结果不会陷入局部最优解。

在此基础上, 将基于 PSO 的改进 k-means 算法用于入侵检测系统的规则挖掘处理模块。实验结果表明, PSO-based k-means 算法的入侵检测率明显高于传统 k-means 算法, 而误报率则大大低于后者。显然, 这对改进入侵检测系统的性能有重要意义。

## 参考文献

- [1] McQueen J. Some methods for classification and analysis of multivariate observations [C]// Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. 1967

- [2] Alsabti K, Ranka S, Singh V. An efficient k-means clustering algorithm[C]//IPPS/SPDP Workshop on High Performance Data Mining. Orlando, Florida, 1998
- [3] 陆林华, 王波. 一种改进的遗传聚类算法[J]. 计算机工程与应用, 2007, 43(21): 170-172
- [4] 曹志宇, 张忠林, 李元韬. 快速查找初始聚类中心的 K-means 算法[J]. 兰州交通大学学报, 2009, 28(6): 15-18
- [5] Ester M, Kriegel H P, Sander J, et al. A density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [C]// Proceeding the 2nd International Conference on Knowledge Discovery and Data Mining (KDD). Portland, 1996
- [6] <http://archive.ics.uci.edu/ml/datasets.html>
- [7] 李国辉, 李恒峰. 基于内容的音频检索: 概念和方法[J]. 小型微型计算机系统, 2000, 21(1): 1173-1177
- [8] Vapnik V N. 统计学习理论[M]. 许建华, 张学工, 译. 北京: 电子工业出版社, 2004
- [9] 程佳. 支持向量机与 K-均值聚类融合算法研究[J]. 辽宁师范大学, 2008
- [10] Richard O, Duda P E, Hart D G, et al. 模式分类(第二版)[M]. 李宏东, 姚天翔, 等译. 北京: 机械工业出版社, 2007
- [11] 张里, 彭小峰. 数据挖掘在网络入侵检测系统中的应用[J]. 重庆工学院学报: 自然科学版, 2008, 22(8): 135-138