

文章编号:1001-9081(2007)10-2437-03

## 一种新的适用于异构网络的 TCP 算法

曲大鹏<sup>1</sup>, 黄东军<sup>2</sup>

(1. 辽宁大学 信息科学与技术学院, 沈阳 110036;

2. 中南大学 信息科学与工程学院, 长沙 410083)

(dapengqu@gmail.com)

**摘要:**提出了一种新的适用于异构网络的传输控制协议(TCP)算法。该算法不仅能够动态寻求网络中的性能最优点,尽量使网络流量保持在该点,而且能够判断网络中数据包丢失的原因,合理地调整参数,避免了 TCP-Reno 中盲目将发送窗口减半的机制。仿真实验的结果表明,新算法的性能优于 TCP-Reno 和其他同类改进协议。

**关键词:**传输控制协议;异构网络;NS2;拥塞控制机制

**中图分类号:** TP393.03 **文献标志码:** A

## New TCP algorithm over heterogeneous networks

QU Da-peng<sup>1</sup>, HUANG Dong-jun<sup>2</sup>

(1. College of Information Science and Technology, Liaoning University, Shenyang Liaoning 110036, China;

2. School of Information Science and Engineering, Central South University, Changsha Hunan 410083, China)

**Abstract:** A new TCP algorithm that was applicable to heterogeneous networks was proposed. It can not only find the best point in performance and keep the network load around it, but also determine the reasons of dropping packets and adjust parameters reasonably to avoid the mechanism of blind halving congestion window in TCP-Reno. The experimental results show its performance is better than TCP-Reno and other improved protocols of the same kind.

**Key words:** Transport Control Protocol (TCP); heterogeneous network; NS2; congestion control mechanism

## 0 引言

目前,互联网中广泛使用的传输控制协议(Transport Control Protocol, TCP)最初是为有线网络而设计的,当发生丢包后降低发送率。在拥塞几乎是造成数据包丢失的唯一原因的有线网络上,这一机制被证明是有效的。但是在异构网络下,无线网络部分的高比特错误率和信道衰落成为数据包丢失的主要原因,而 TCP 不能区分数据包丢失的原因,只是简单地降低发送率,造成性能大幅下降。近年来,随着异构网络(有线/无线网络)的快速发展,如何提高异构网络下的 TCP 性能已经成为了一个活跃的研究领域<sup>[1-4]</sup>。

本文提出了一种新的适用于异构网络的 TCP 算法——基于 Knee 的 TCP(TCP-Based on Knee, TCP-BK)。它对传统的 TCP 做了两个主要的改进:1)采取“预防”措施,根据动态探测得到的网络流量和性能参数,寻求网络中性能最优点,并且尽量使网络流量保持在该点附近,从而得到性能上的最优;2)采取“恢复”措施,根据动态探测得到的吞吐量和往返延时间(Round Trip Time, RTT)的乘积与拥塞窗口的大小关系来判断网络中数据包丢失的原因是网络拥塞还是无线错误,然后相应地调整参数,避免了传统 TCP 中的数据包丢失后盲目减小发送窗口的机制。

## 1 相关工作

针对异构网络下 TCP 的性能改进,已有多项改进方案,主要可分为三种:

### 1) 链路层方法

通过本地重传和前向纠错屏蔽发送端与链路相关的丢包,运行在物理层的上方,较早了解丢包的情况,在根部解决问题。但由于链路层协议与高层协议都有独立的差错控制功能,有一定的重复性,相互竞争会降低无线信道的利用率。文献[6]针对跨层设计进行了详细的论述。

### 2) 分离连接方法

在基站处将 TCP 连接分成两部分,有线连接部分使用传统的 TCP 协议,无线连接部分使用改进后的 TCP 协议。它破坏了端到端的 TCP 连接语义。以 Indirect-TCP<sup>[7]</sup>为代表,目前很少有此类方法。

### 3) 端到端方法

只修改 TCP 连接的端主机,保证了 TCP 连接的完整性,增强了 TCP 在无线链路上的性能。由于 IPv6 已经比较成熟,而且对于 IP 而言, TCP 是最佳的搭配,所以是目前研究的热点。TCP-Vegas<sup>[8]</sup>使用期望流量和真实流量之间的差值来估计可用带宽和网络拥塞级别,相应调整拥塞窗口。TCP-Westwood<sup>[9]</sup>通过在发送端测量返回的 ACK(ACKnowledge)速率测量可用带宽,在发生丢包后选择合适的慢启动阈值和拥塞窗口。

## 2 TCP-BK

### 2.1 网络状况分析

文献[10]介绍了与网络流量相对应的网络参数的变化(如图1所示),当网络流量很小的时候,网络中路由器收到

收稿日期:2007-04-05;修回日期:2007-06-18。

作者简介:曲大鹏(1981-),男,辽宁鞍山人,助教,硕士,主要研究方向:计算机网络;黄东军(1960-),男,湖南常德人,教授,博士,主要研究方向:计算机网络。

的分组数量也很小,新到的分组立即就会被处理,随着网络流量的增加,吞吐量快速增长而 RTT 缓慢增长;当网络流量超过某点(knee 点)时,即路由器新收到的分组不能立即被处理,只能被插入到路由器的缓冲区中。此时,继续增加网络流量,会造成吞吐量缓慢增长而 RTT 快速增长;当网络流量持续增长,超过某点(cliff 点),即路由器的缓冲区装满之后,再增加网络流量,分组或者在队列尾部被丢弃,或者当队列长度超出阈值限制后被丢弃,则造成吞吐量急剧下降,而 RTT 快速增长,网络趋向崩溃。

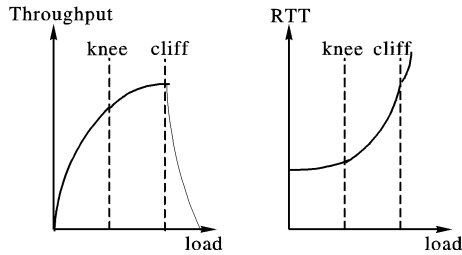


图1 与网络流量相对应的网络参数

从上面的讨论可以看出,在 knee 点处的网络性能是最好的。希望保持网络处于 knee 点附近。这里提出一个函数  $M = (\text{Throughput}/\text{RTT})$ 。在  $[0, \text{knee}]$  区间,RTT 基本保持不变而吞吐量快速增长,则函数  $M$  近似为一个增函数;在  $[\text{knee}, \text{cliff}]$  区间,吞吐量基本保持不变而 RTT 快速增长,则函数  $M$  近似为一个减函数, $M$  在 knee 点处达到最大值。

为节省篇幅,下面三节只介绍对传统 TCP 的改进部分,未说明部分沿用 TCP-Reno。

## 2.2 吞吐量计算

当 TCP 发送端发觉有数据包丢失,立即将测得的吞吐量和 RTT 的乘积与拥塞窗口相比较,后者减去前者得到的差值就是等候在瓶颈链路队列中的分组数量值,而如果差值为负,则表示网络中没有发生拥塞。

设  $t_k$  表示第  $k_{th}$  个 ACK 到达 TCP 发送端时的时刻,  $Th_{\text{sample}}(k)$  是当第  $k_{th}$  个 ACK 到达 TCP 发送端之前的 RTT 时间段内的吞吐量的采样值。计算如下:

$$Th_{\text{sample}}(k) = \frac{\sum_{j \in RTT_k} d_j}{RTT_k} \quad (1)$$

其中  $\sum_{j \in RTT_k} d_j$  表示在该 RTT 时间段内,接收端收到的所有数据。

本文采用恒值增益( $\beta = 0.6$ )

$$Th_{\text{estimate}}(k) = Th_{\text{estimate}}(k-1) \times \beta + Th_{\text{sample}}(k) \times (1 - \beta) \quad (2)$$

## 2.3 发送端对接受到新数据包的反应

从 2.1 的讨论可知,应该尽量使网络保持在 knee 点附近。即当发送端接收到一个新的 ACK 确认包时,它首先应该判断网络流量的状况。如果计算得到的  $M$  值大于以前的  $M$  值,表示现在的网络状况更接近于 knee 点,已经进入了临界区域,应该减小增长窗口(这里采用拥塞避免里的线性增长);如果网络状况在 knee 点之前,可以略增大发送窗口以达到更大的吞吐量(保持指数增长);如果网络状况在 knee 点之后,应该减小增长窗口以避免拥塞(将传统线性增长的周期乘 2)。这里合理地增长发送窗口不仅不会造成吞吐量的下降,而且可以推迟拥塞。具体见算法 1:

算法 1 发送端对接受到新应答包的反应

$$Th_{\text{estimate}}(k) \\ M_k = Th_{\text{estimate}}(k) / RTT_k$$

```

if ( $M_k > M_{\text{knee}}$ )
{ //find a larger M, it is more approached to the knee point
 $M_{\text{knee}} = M_k$ ;
 $Th_{\text{knee}} = Th_{\text{estimate}}(k)$ ;
 $RTT_{\text{knee}} = RTT_k$ ;
 $cwnd = cwnd + 1/cwnd$ ;
}
elseif ( $RTT_k > RTT_{\text{knee}}$ )
{ //the network condition is behind knee point
 $cwnd = cwnd + 1/2 * (1/cwnd)$ ;
}
elseif ( $Th_{\text{estimate}}(k) < Th_{\text{knee}}$ )
{ //the network condition is before knee point
 $cwnd = cwnd + 1$ ;
}
endif

```

## 2.4 发送端对新数据包丢失的反应

当发送端觉察到有 TCP 数据包丢失(收到 3 个重复的 ACK 确认包或者超时)时,它通过比较  $Th_{\text{estimate}} \times RTT$  和拥塞窗口的大小关系来判断 TCP 数据包丢失的原因,如果  $Th_{\text{estimate}} \times RTT$  大于拥塞窗口,表示是无线错误造成的丢包,尽量不减少拥塞窗口和慢启动阈值,以免降低吞吐量,造成带宽的浪费;反之,则是拥塞造成的丢包,则根据测得的网络流量在 knee 点的数据来调整拥塞窗口和慢启动阈值(slow start threshold, ssthresh),以达到最优。其中超时是比收到 3 个重复的 ACK 确认包更严重的现象。具体见算法 2、3:

算法 2 发送端对接受到 3 个重复的 ACK 确认包的反应

```

if (3 dup ack)
{ //the sender receives 3 dup ack
if ( $Th_{\text{estimate}}(k) * RTT_k < cwnd$ )
{ //the reason of dropping packets is congestion
 $ssthresh = (Th_{\text{knee}} * RTT_{\text{knee}}) / \text{max\_segsz}$ ;
if ( $cwnd > ssthresh$ )
 $cwnd = ssthresh$ ;
endif
}
endif
endif

```

算法 3 发送端对超时的反应

```

if (timeout)
{ //the sender timeouts
if ( $Th_{\text{estimate}}(k) * RTT_k > cwnd$ )
{ //the reason of dropping packets is link error
 $ssthresh = (Th_{\text{knee}} * RTT_{\text{knee}}) / \text{max\_segsz}$ ;
 $cwnd = ssthresh$ ;
}
else
{ // the reason of dropping packets is congestion
 $ssthresh = (Th_{\text{knee}} * RTT_{\text{knee}}) / \text{max\_segsz}$ ;
 $cwnd = 1$ ;
}
endif
}
endif

```

## 3 实验结果和分析

### 3.1 实验方法

下面使用网络模拟器 NS2 来验证 TCP-BK, 并与 TCP-Reno、TCP-Westwood、TCP-Vegas 等协议相比较。TCP-Reno 是

当前互联网中广泛使用的 TCP 协议, TCP-Westwood 和 TCP-Vegas 都属于前面讨论的端到端方法, 其中 TCP-Westwood 是由 UCLA 提出的改进协议, 目前在端到端方法中性能最好。这里只在一个基本的环境下进行实验, 并且在发送端只配置一种 TCP 实现。拓扑结构如图 2 所示, 由三段链路和四个节点组成, 其中第二个路由器是一个基站, 最后是 TCP 无线接收端, 最后一段为无线链路。发送端到中间路由器的带宽为 100 Mb, 延迟为 1 ms; 两个路由器之间的带宽为 100 Mb, 延迟为 49 ms; 路由器到接收端的带宽为 2 Mb, 延迟为 0.01 ms (在模拟中参数可能会根据具体情况作相应的变化, 见后面的说明)。本文中的各个实验均以 FTP 数据传输作为 TCP 流。



图2 模拟网络拓扑结构

### 3.2 实验结果

#### 3.2.1 无线衰落情况

无线衰落是异构网络中无线链路部分最常见的现象。Bad 状态表示无线衰落情况, 其错误率在 0~0.5 变化, Good 状态的错误率为 1E-5。TCP 连接处于两种状态的持续时间分别为 10 s 和 5 s。图 3 显示了 TCP-BK 与 TCP-Reno、TCP-Vegas 和 TCP-Westwood 在无线衰落情况下的性能比较。可以看出, 在错误率低的情况下, 性能差不多, 但随着错误率的增大, TCP-BK 的吞吐量大于 TCP-Reno、TCP-Vegas 和 TCP-Westwood。

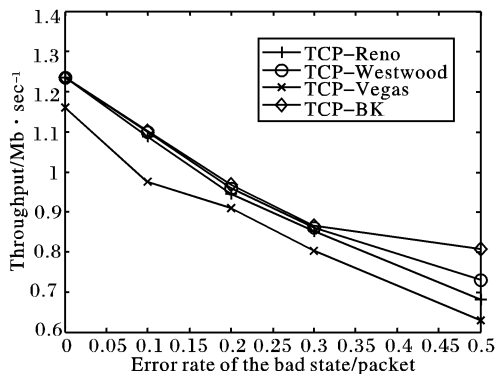


图3 Bad状态不同链路错误率下的吞吐量

#### 3.2.2 临时掉线情况

临时掉线是异构网络中无线链路部分另一种常见的现象, 主要是由于无线终端的移动性而造成其与基站短时无法通信。Bad 状态表示临时掉线情况, 其错误率为 1。Good 状态的持续时间为 10 s, 而 Bad 状态的持续时间在 0.05 s~0.5 s 变化。图 4 显示了 TCP-BK 与 TCP-Reno、TCP-Vegas 和 TCP-Westwood 在临时掉线情况下的性能比较。可以看出, TCP-BK 一直保持着性能上的优势。

无线衰落和临时掉线是异构网络中无线链路部分常见的两种现象, 在错误率低或临时掉线的时间特别短的情况下, 网络中数据包丢失的原因主要是网络拥塞, 四种协议的性能非常接近, 而随着错误率的增加或临时掉线的时间增长, 无线错误造成的数据包丢失越来越多, TCP-Reno 不能判断数据包丢失的原因, 只是简单地将拥塞窗口减半; TCP-Vegas 能判断出网络拥塞级别, 并进行调整, 但它在判断网络拥塞后的调整比较保守, 导致性能特别是在带宽竞争能力上较差; TCP-Westwood 能够通过探测得到的可用带宽判断丢包的原因, 但它只是在丢包后才做调整, 不能更好地适应时变的异构网络; TCP-BK 不仅能够动态地探测网络性能, 合理调整拥塞窗口,

而且能够判断出丢包的原因, 从而采取合理的措施, 所以较之其他三种协议提高了吞吐量。

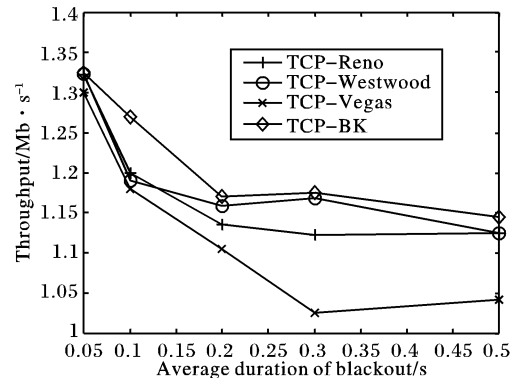


图4 不同掉线时间下的吞吐量

#### 3.2.3 瓶颈带宽对吞吐量的影响

随着网络技术的发展, 链路所能提供的带宽越来越大, 一个重要的问题便由此提出: 传输协议是否能够充分利用底层提供的带宽。传统的 TCP 由于其慢启动和拥塞避免机制, 导致它对丢包十分敏感。例如 TCP-Reno, 它的“窗口减半”的策略导致其链路利用率远小于链路所能提供的带宽。这不仅浪费网络资源, 而且使得一些新型的网络应用难以实现。这在异构网络中错误率较高的情况下显得更为突出。

下面检验几种 TCP 实现在不同瓶颈带宽上的吞吐量。对每个 TCP 实现分别单独进行实验, 无线链路部分的错误率设置为 0.1%。瓶颈带宽从 1 Mbps 逐渐变化到 20 Mbps。实验结果如图 5 所示。TCP-BK 的性能一直较之其他协议性能更高。

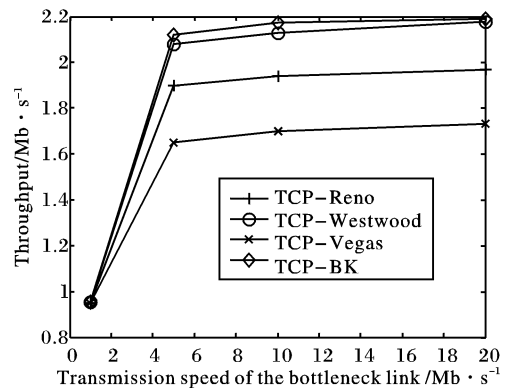


图5 不同瓶颈带宽下的吞吐量

#### 3.2.4 RTT 对吞吐量的影响

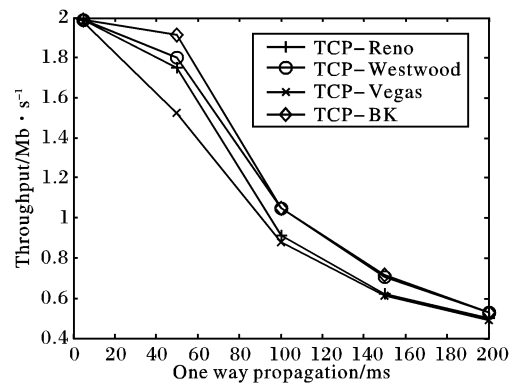


图6 不同 RTT 下的吞吐量

毫无疑问, RTT 对 TCP 吞吐量的影响也是巨大的。特别是在网络发生拥塞或链路发生错误时, 丢包导致 TCP 需要一

(下转第 2449 页)

含 635 条同类报警。这其实属于虚报警,因为是正常的网管行为,但是我们合理设置时间区间,可以完全避免这类重复和虚假报警,加快管理员处理报警的速度。

由于篇幅所限,本文没有给出每条具体报警的属性相似度比较的阈值设置,也没有给出每条报警具体归属于哪类报警的信息,原型中还没有实现关于动态配置接口的模块,但是通过比较聚合前后的报警数量,我们证明了算法是有效的。实现动态配置接口后,管理员可随时根据虚报警和重复报警的情况,修改相应报警的阈值,将这些重复报警聚合。

### 3 结语

本文提出一种基于报警分类和属性相似度的聚合算法,按照攻击类别将报警分为四类,根据属性值将其分为四类,在此基础上,对每种属性的计算采取不同的相似度计算方法,并结合分类报警设置不同的预期阈值,阈值可以通过外部接口动态调整,使聚合模块能产生一个比较好的结果。最终聚合结果将通过综合所有属性相似度加权平均获得。基于 Snort-ids 和 Darpa 99 的网络数据进行的实验结果表明,本文算法有效地减少了重复告警。

未来工作主要包括:1) 目前算法只考虑了单一 IDS 的报警分类,实际上只作为单个 IDS 的报警后台使用,如果作为多个 IDS 的聚合模块,则需要进一步考虑如何分类,因为各个 IDS 报警的 attack-class 属性未必相同;2) 多个 IDS 聚合时,各 IDS 属性名字也未必相同,还需要新的机制来进行各属性的综合比较,可以考虑采用 IDMEF 架构;3) 阈值如何自适应调整也是未来工作的一部分;4) 在超报警代表数量比较大的报警集合时,如何有效地合并新报警到超报警中也是一个需要继续研究的方向。

#### 参考文献:

(上接第 2439 页)

段时间来恢复到正常的发送速率,RTT 越大,则所需的恢复时间越长。图 6 显示了在瓶颈链路带宽固定为 2 Mbps,RTT 从 5 ms 变化到 200 ms 的情况下,TCP-BK 一直较其他三种协议性能更好。随着 RTT 的增大,所有协议的性能都急剧下降,这是由于端到端方法自身的反应机制造成的。

TCP-BK 在不同的瓶颈带宽和 RTT 下,能够保持较之其他三种协议更高的吞吐量,说明它具有比较强的适应性,主要是因为 TCP-BK 不仅采取了“预防”措施,而且采取了有效的“恢复”措施,两者相结合,使得网络流量尽量保持在 knee 处,取得了性能上的提高。

### 4 结语

本文提出了一种新的适用于异构网络的 TCP 算法——TCP-BK。它动态地寻找网络中的性能最优端 knee,根据吞吐量和 RTT 的实时计算来判断当前网络状态与 knee 点的关系,当发送端收到一个新的 ACK 确认包时,先判断网络状况,再采取不同的机制调整发送窗口。在发生数据包丢失后,先判断丢包的原因,再采取相应的恢复措施。NS 仿真实验结果表明与现在广泛使用的 TCP-Reno 和同类改进协议 TCP-Vegas、TCP-Westwood 相比,TCP-BK 取得了更好的性能。

#### 参考文献:

- [1] 周建新,邹玲,石冰心. 无线网络下 TCP 综述[J]. 计算机研究与发展, 2004, 41(1): 53 - 59.

- [1] 穆成坡,黄厚宽,田盛丰. 入侵检测系统报警信息聚合于关联技术研究综述[J]. 计算机研究与发展, 2006, 43(1): 1 - 8.
- [2] VALDES A, SKINNER K. Probabilistic alert Correlation[C]// Proceedings of 4<sup>th</sup> International Symposium on Recent Advance in Intrusion Detection (RAID) 2001, Lecture Notes in Computer Science 2212. Berlin: Springer-Verlag, 54 - 68.
- [3] DAIN O, CUNNINGHAM R. Fusing a heterogeneous alert stream into scenarios[C]// Proceedings of the 2001 ACM Workshop on Data Mining for Security Applications. New York: ACM Press, 2001: 1 - 13.
- [4] DEBAR H, WESPI A. Aggregation and correlation of intrusion-detection alerts[C]// Proceedings of 4<sup>th</sup> International Symposium on Recent Advance in Intrusion Detection (RAID) 2001, Lecture Note in Computer Science 2212. Berlin: Springer-Verlag, 85 - 103.
- [5] AUREL F, CUPPENS F. Using an intrusiondetection alert similarity operator to aggregate and fuse alerts[C/OL]. [2007 - 04 - 01]. <http://www.rennes.enst-bretagne.fr/~fcuppens/articles/sar05.pdf>.
- [6] DEBAR H. The intrusion detection message exchange format[EB/OL]. [2006 - 03 - 16]. <http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-16.txt>.
- [7] JULISCH K. Using root cause analysis to handle intrusion detection alarms[D]. Phd Thesis, Unversisty of Dortmund, 2003.
- [8] 穆成坡,黄厚宽,田盛丰. 基于模糊综合评判的入侵检测报警信息处理[J]. 计算机研究与发展, 2005, 42(10): 1679 - 1685.
- [9] ROESCH M. Snort user manual2. 0. 0[EB/OL]. [2006 - 05 - 23]. [http://www.snort.org/docs/snort\\_htmanuals/htmanual\\_261/](http://www.snort.org/docs/snort_htmanuals/htmanual_261/).
- [10] MIT Lincoln Labs. 1999 DARPA intrusion detectionevaluation[EB/OL]. [2007 - 03 - 15]. <http://www.ll.mit.edu/IST/ideval/index.html>.

- [2] BARAKAT C, ALTMAN E, DABBOUS W. On TCP performance in a heterogeneous network: a survey[J]. IEEE Communications Magazine, 2000, 38(1): 40 - 46.
- [3] PENTIKOUSIS K. TCP in wired - cum - wireless environments[J]. IEEE Communications Surveys, 2000, 3(4): 2 - 14.
- [4] YE T, KAI X, ANSARI N. TCP in wireless environments: problems and solutions[J]. IEEE Radio Communications 2005, 43(3): 27 - 32.
- [5] NS-2 network simulator( ver. 2) [CP/OL]. [2006 - 11 - 11]. <http://www.mash.cs.berkeley.edu/ns>.
- [6] GRANELLI F, KLIASOVICH D. Cross-Layering for performance improvement in multi - hop wireless networks[J]. Journal of Interconnection Networks, 2006, 7(1): 51 - 61.
- [7] BAKRE A, BADRINATH B R. I-TCP: indirect TCP for mobile hosts[C]// 15th International Conference on Distributed Computing Systems. Vancouver: IEEE Computer Society, 1995: 136 - 143.
- [8] BRAKMO L S, PETERSON L L. TCP vegas: end-to-end congestion avoidance on a global internet[J]. IEEE Journal on Selected Areas in Communication, 1995, 13(8): 1465 - 1480.
- [9] REN W, KENSHIN Y, SANADIDI M Y, et al. TCP with sender-side intelligence to handle dynamic, large, leaky pipes[J]. IEEE Journal on Selected Areas in Communications, 2005, 23(2): 235 - 248.
- [10] JAIN R. A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks[J]. ACM CCR, 1989, 19(5): 56 - 71.