

基于带宽估计的无线 TCP 拥塞控制

赵宇红, 刘海良, 张晓琳

(内蒙古科技大学信息工程学院, 内蒙古 包头 014010)

摘要:针对在无线网络中,由于链路本身的特性,丢包存在拥塞丢包及无线误码丢包两种不同情况,但出现丢包时,传统 TCP 协议不会分析丢包原因,而盲目地进行拥塞窗口(cwnd)及慢启动门限值(ssthresh)的调整,引起数据发送速率波动较大,导致 TCP 吞吐量下降的问题。为了改善无线网络传输性能,提出了一种基于带宽估计的无线 TCP 拥塞控制算法。该算法通过在每往返时延(RTT)确认的发送数据量来实时估计带宽,为获得更准确的估计值,计算过程中对 RTT 及带宽估计值进行了滤波平滑处理。在此基础上,计算出带宽利用率来分析丢包原因及拥塞程度,并据此对拥塞控制参数进行自适应动态调整。运用 NS-2 完成实验,从拥塞窗口、抖动、平均吞吐量等方面等经典的 TCP 版本进行比较,仿真结果表明,算法对拥塞控制参数的调整更加平滑,提高了系统吞吐量,增强了无线网络的传输性能。

关键词:无线网络;拥塞控制;带宽估计;无线误码丢包

中图分类号:TP393 **文献标识码:**B

Wireless TCP Congestion Control Based on Bandwidth Estimation

ZHAO Yu-hong, LIU Hai-liang, ZHANG Xiao-lin

(School of Information Engineering, Inner Mongolia University of Science and Technology,
Baotou Inner Mongolia 014010, China)

ABSTRACT: In the paper, an algorithm based on bandwidth estimation was proposed in order to improve the wireless network transmission performance. The algorithm calculated the amount of send data confirmed in each round trip delay (RTT) to estimate real-time bandwidth. In order to obtain a more accurate estimated value, the RTT and estimated bandwidth was smoothed in the process of calculation, and then the bandwidth utilization was calculated. The algorithm used the bandwidth utilization to analyze the reason of packet loss and congestion degree, and control parameters were adaptively and dynamically adjusted according to analysis results. Experiment is completed using NS-2 platform, and comparison was made based on the aspects of congestion window, jitter, and average throughput with classic TCP version. Simulation results show that the algorithm makes the change of congestion control parameters smoother, improves the system throughput and enhances the wireless network transmission performance.

KEYWORDS: Wireless network; Congestion control; Bandwidth estimation; Wireless error packet loss

1 引言

随着网络系统迅速发展,信息量急剧增加,网络拥塞的情况已经无法避免,对于网络中拥塞问题的控制也越来越迫切。其中,TCP/IP 协议下的网络拥塞控制机制是大多数学者研究的主要对象,并取得了很多成果。如文献[1]中提出了一种无线网络中固定码率(CBR)发送速率调整的拥塞控制算法。文献[2]提出了基于一种在不同阶段基于带宽估计的 TCP 慢启动改进算法。文献[3]中提出了一种在移动自组网中显示拥塞控制算法。文献[4]中提出了一种在高带宽时延网络中基于带宽估计的显示拥塞控制方法。文献[5]中提出一种无线网络中基于反馈的自适应网络编码控制算法。

文献[6]提出一种基于网络层资源队列管理的算法。文献[7]中提出了一种基于 RTT 自适应的 TCP Reno 改进算法。这些算法都从不同的方面对 TCP 的拥塞控制做了改进,但在数据传输出现丢包时对于 ssthresh 和 cwnd 的值的调整分别为减半处理和置 1 处理,依然存在控制参数盲目减小,设置单一化的问题。

作为一种在网络中普遍使用的协议, TCP 为网络中的用户提供了可信的、健壮的端到端网络通信服务。TCP 中的拥塞控制机制一直是保证网络通信正常的关键,最初的 TCP 拥塞控制机制是针对于有线网络环境设计的,对于链路质量相对较差的无线网络, TCP 拥塞控制机制的表现并不是很好。在无线网络环境中,网络拥塞不再是导致数据丢包的唯一原因,由于无线网络本身的传输信道受到外界影响而造成信号

收稿日期:2014-12-21 修回日期:2015-06-03

衰减或干扰时,也会造成数据包丢失,可以称之为无线误码丢包。但是,传统的 TCP 拥塞控制,并不会区分丢包类型,一旦有丢包发生时,TCP 会频繁的调用拥塞控制算法,减小拥塞窗口及慢启动门限值,无线网络中有时链路环境较差而导致误码丢包频发,这时将会出现控制参数变化很快且严重降低了无线网络的吞吐量。另外,传统 TCP 拥塞控制机制在进行参数调整时,仅仅依据丢包是否发生,并没有对拥塞程度加以区别,这样单一化的参数设置,也将严重影响网络性能。

针对上述分析,根据无线网络的特点,提出了一种基于带宽估计的拥塞控制算法(TCP-Bandwidth,简记 TCP-BD 算法),该算法实时估计网络中的带宽,计算出带宽利用率,据此判断丢包类型及拥塞程度,对不同丢包类型及拥塞情况下,自适应调整拥塞控制参数 $cwnd$ 和 $ssthresh$,解决了传统 TCP 算法在丢包时,参数盲目减小的问题,使得控制参数的变化更加平和。仿真结果证明,该算法能够对拥塞控制参数 $cwnd$ 和 $ssthresh$ 值作出合理的调整,很大程度上提高了网络的吞吐量。

2 TCP 拥塞控制算法

网络拥塞是某种网络处在持续过载的网络状态,在这种状态下用户对因特网中资源(如处理器处理能力强弱、存储空间大小和链路带宽大小等)的需求超过了网络本身所具有的容量。如图 1 显示了拥塞发生的情况,开始时随着负载的增加实际吞吐量增加,但随着负载逐渐接近网络最大容量时,导致网络缓冲区被填满并丢失数据包,从而出现实际吞吐量的大幅度降低,最终导致拥塞崩溃。

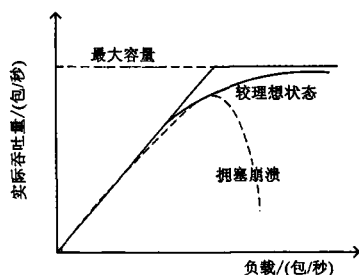


图 1 网络拥塞变化

为了解决网络的拥塞问题,TCP 提出了一系列的拥塞控制算法。最初由 V. Jacobson 在论文中提出了“慢启动”(slow start)算法和“拥塞避免”(congestion avoidance)算法^[8]。1990 年 TCP Reno^[9] 版本被提出,添加了“快速重传”(fast retransmit)算法和“快速恢复”(fast recovery)算法,防止了网络处于拥塞不严重状态时调用“慢启动”算法而造成过大降低数据发送速率的现象。TCP New Reno^[9] 版本中针对 Reno 中的“快速恢复”算法进行了改进,比较经典的还有 TCP SACK、TCP Vegas、TCP Westwood 等各种算法。

TCP 的拥塞控制算法主要由“慢启动”、“拥塞避免”、“快速重传”和“快速恢复”四个部分组成。当新的 TCP 连接

建立时, $cwnd$ 值被初始化为一个数据包大小。TCP 发送端按 $cwnd$ 值大小来发送数据包,每当发送端收到一个确认的 ACK, $cwnd$ 值就增加,增加量为一个数据包的大小, $cwnd$ 值的增长将随 RTT 呈指数级(exponential)增长。发送端利用来自接收端的 ACK 应答和发送端维护的超时定时器 RTO 两个参数进行数据丢包的判断,当重传计时器超时发生溢出时,即可判断对应分组丢失,当连续收到 3 个同一个分组的重复 ACK 时,也可迅速判定出该分组丢失。当两种情况之一发生时,便认为网络发生了拥塞,TCP 通过调用拥塞控制算法减小 $ssthresh$ 和 $cwnd$ 值来实现降低发送速率的目的。

3 TCP-BD 带宽估计算法

3.1 问题分析

在有线网络环境下,基本上网络拥塞是丢包的唯一原因,丢包发生时,通过调整 $ssthresh$ 值和 $cwnd$ 值的大小达到控制拥塞的目的。此时会根据两种情况完成相应的调整:

1) 如果发送端收到同一分组的 3 个重复的 ACK 时,慢启动门限值调整为当前拥塞窗口值的一半大小,即 $ssthresh = cwnd/2$ 。

2) 如果是由于超时引起的,则还会把拥塞窗口设置为 1,即 $cwnd=1$ 。

此时,如果 $cwnd$ 值不大于 $ssthresh$ 值,TCP 将会重新进入慢启动过程;如果 $cwnd$ 值大于 $ssthresh$ 值,TCP 将会执行拥塞避免算法,在每次收到一个 ACK 时, $cwnd$ 值将按照 $1/cwnd$ 个数据包大小来增加。但在无线环境下,数据丢包的原因不再仅仅是因为网络发生了拥塞,即当网络发生无线误码丢包时,TCP 发送端也会频繁的调用 TCP 的拥塞控制算法将 $ssthresh$ 的值设置为当前 $cwnd$ 大小的一半,并减小 $cwnd$ 值,即降低了发送端数据发送速率,而此时的网络并没有发生拥塞,如此操作导致网络吞吐量很大程度的下降,影响了网络的传输性能。

3.2 TCP-BD 算法描述

在无线网络环境中,不论出现拥塞丢包还是无线误码丢包,不论网络拥塞是严重还是轻微,TCP 都会调用拥塞控制算法,通过减小 $ssthresh$ 和 $cwnd$ 值的来降低发送端的发送速率,以达到减缓网络拥塞程度这一目标。为了更有效地进行拥塞控制,改善无线网络传输性能,本文借鉴了 TCP Westwood 等^[10-12] 的带宽估计算法,提出了一种基于带宽估计的无线网络拥塞控制算法 TCP-BD。其中,带宽估计算法公式如式(1)。

$$BWE = \frac{S_{data}}{SRTT_n - SRTT_{n-1}} \quad (1)$$

BWE 为网络带宽估计值, $SRTT_n$ 和 $SRTT_{n-1}$ 分别代表第 n 个和第 $n-1$ 个 RTT 经过平滑后的时延时间, S_{data} 是在这段时间内发送的数据总量。

为改善 RTT 剧烈抖动导致带宽估计值误差较大的影响,对 RTT 和 BWE 分别进行平滑处理。

1) RTT 的平滑,公式如(2) 所列。

第1次测量:

$$RTT = T, SRTT = T;$$

第*i*次测量:

$$SRTT_i = (1 - k)SRTT_{i-1} + k * T_{new} \quad (2)$$

其中, T_{new} 为新测量的 RTT , k 为加权因子, k 取 1/4 (来自 Jacobson88 推荐)。

2) BWE 平滑处理, 借鉴了 RTT 的平滑处理方法, 对 BWE 的平滑处理如式(3) - (6) 所列。

第1次测量:

$$BWE = \frac{S_{data}}{SRTT_n - SRTT_{n-1}} \quad (3)$$

$$SBWE = \frac{S_{data}}{SRTT_n - SRTT_{n-1}} \quad (4)$$

第*i*次测量:

$$SBWE_i = k * SBWE_{i-1} + (1 - k) * BWE_{new} \quad (5)$$

$$BWE_i = k * BWE_{i-1} + (1 - k) * SBWE_i \quad (6)$$

其中, $SBWE$ 为平滑处理后的带宽估计值, BWE_{new} 为最新测量的带宽值, k 为加权因子, 在不同的网络负载状态下, 经过大量仿真, k 取 7/8 时, 性能较好。

3.3 TCP - BD 算法过程

当无线网络出现丢包时, 根据带宽的利用率来动态调整 $cwnd$ 和 $ssthresh$ 的值。网络拥塞的判定依然依据收到重复确认或超时, 具体调整如下:

1) 收到三个重复 ACK

此时说明网络拥塞不是特别严重, 慢启动门限值不再为原来的 TCP 直接减半计算, 重新计算公式如式(7) 所示。

$$ssthresh = cwnd * (1 - \mu) \quad (7)$$

其中, μ 为动态调节因子, 计算公式如式(8) 所示。

$$\mu = \frac{BWE}{BWE_{max}} \quad (8)$$

式中, BWE 为当前的最新带宽估计值, BWE_{max} 为估计带宽值中的最大值, μ 的取值范围为(0,1]。

当 μ 值较小, 即网络带宽利用率较低时, 网络拥塞的可能性较小, 如果此时出现了数据丢包, TCP 将直接做减半处理, 会过大降低数据包的发送速率, 为了解决这样情况, 对 μ 值进行进一步划分, 重新调整计算公式如式(9) 所示。

$$ssthresh = \begin{cases} cwnd * (1 - \mu) & 0 < \mu \leq \frac{1}{2} \\ \frac{1}{2} * cwnd & \frac{1}{2} < \mu \leq 1 \end{cases} \quad (9)$$

从式(9) 中可以看出, 当 $\mu \in (0, 1/2]$ 时, μ 值较小, 表明网络可用带宽高, 拥塞发生时 $ssthresh$ 计算值越大, 从而保证了数据的发送速率; 当 $\mu \in (1/2, 1]$ 时, μ 值较大, 表明网络可用带宽低, $ssthresh$ 计算值做减半处理。算法伪代码如下:

```
if ( dupacks == 3 ) { /* 收到 3 个重复的 ACK */
    /* 算出带宽估计值 BWE 和  $\mu$  值, 根据  $\mu$  值来设置
    ssthresh 和 cwnd 的值 */
    if ( 0 <  $\mu$  ≤ 1/2 ) {
```

```
ssthresh = cwnd * ( 1 -  $\mu$  );
```

```
cwnd = ssthresh + 3MSS; /* 进入拥塞避免阶段 */
```

```
if ( 1/2 <  $\mu$  ≤ 1 ) {
```

```
ssthresh = cwnd * 1/2;
cwnd = ssthresh + 3MSS; /* 进入拥塞避免阶段 */
```

```
2) RTO(重传计时器) 超时
```

这种情况下, 网络拥塞程度比收到三个重复 ACK 时严重, 慢启动门限值应该更小, 已更好缓解网络拥塞的程度。计算公式如式(10) 所列。

$$ssthresh = \begin{cases} \text{null} & 0 < \mu \leq \frac{1}{4} \\ cwnd * (1 - \mu) & \frac{1}{4} < \mu \leq \frac{1}{2} \\ cwnd * \mu & \frac{1}{2} < \mu < \frac{3}{4} \\ cwnd * \frac{1}{2} & \frac{3}{4} \leq \mu \leq 1 \end{cases} \quad (10)$$

当 $\mu \in (0, 1/4]$ 时, μ 值较小, 网络带宽利用率较低, 网络拥塞的可能性较小, 如果网络中此时出现了数据丢包, 则认定为出现了无线丢包, 此时应该继续保持较大的数据发送速率, 为了不降低数据发送速率, TCP 拥塞控制不做任何处理; 当 $\mu \in [3/4, 1]$ 时, 网络带宽利用率很高, 如果发生拥塞导致数据丢包, 此时为了更好的缓解网络的压力, 应该使慢启动门限值变的更加小, 此时直接做减半处理; 当 $\mu \in (1/4, 3/4)$ 时, 如果此时网络出现了拥塞, $ssthresh$ 值的计算应该高于 $cwnd$ 的一半, 以保证数据的发送量。由于此种情况网络拥塞程度较为严重, 因此不论 μ 的取值如何, 统一将 $cwnd$ 的大小设置为 1, 即 $cwnd = 1$ 。算法伪代码如下:

```
if ( timeout ) { /* RTO 超时 */
    /* 算出带宽估计值 BWE 和  $\mu$  值, 根据  $\mu$  值来设置
    ssthresh 和 cwnd 的值 */
    if ( 1/4 <  $\mu$  < 1/2 ) {
        ssthresh = cwnd * (1 -  $\mu$ )
        cwnd = 1; /* 进入慢启动阶段 */
    }
    if (  $\mu$  ≤ 1/4 ) {
        /* 认为是无线丢包, 不做处理 */
    }
    if ( 1/2 <  $\mu$  < 3/4 ) {
        ssthresh = cwnd *  $\mu$ ;
        cwnd = 1; /* 进入慢启动阶段 */
    }
    if ( 3/4 <  $\mu$  ≤ 1 ) {
        ssthresh = cwnd * 1/2;
        cwnd = 1; /* 进入慢启动阶段 */
    }
```

在以上两种情况下, 若计算出的 $ssthresh$ 值出现小于 2 的情况, 为了更好的进行拥塞控制, $ssthresh$ 值设置最小值为 2, 计算如式(11) 所示。

$$ssthresh = \begin{cases} ssthresh & ssthresh \geq 2 \\ 2 & ssthresh < 2 \end{cases} \quad (11)$$

TCP-BD 算法的流程图如图 2 所示。

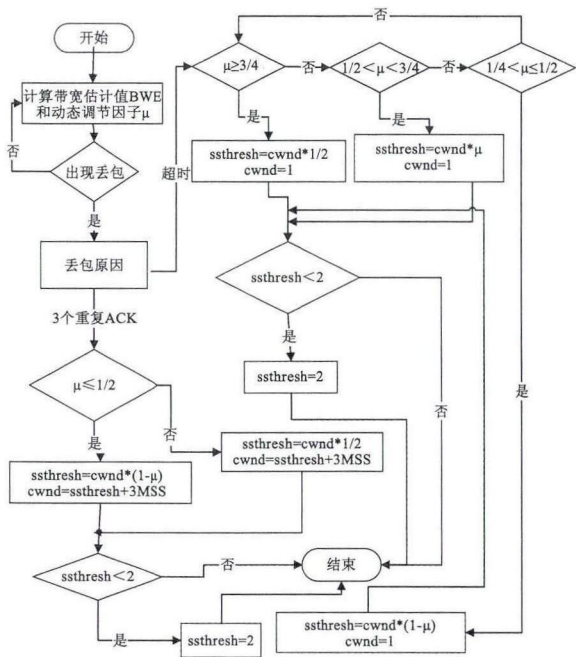


图2 TCP-BD 算法流程图

4 TCP-BD 算法的仿真分析

4.1 仿真环境

实验采用仿真工具 Network Simulator Version 2 (简称 NS2) 进行算法的仿真验证。仿真网络的拓扑结构如图 3 所示。节点 N0 为 TCP 发送方, 节点 N1 为 TCP 接收方, 节点 R0、R1 为中间路由节点。节点 N0 到节点 R0 之间、R1 到 N1 之间建立延时为 2ms、带宽为 10 兆的双向链路。在节点 R0 和 R1 之间建立延时为 2ms、带宽为 5M 的双向链路, 此链路作为瓶颈链路。节点 N0 和节点 N1 之间建立 TCP 背景业务, 通信业务为 FTP 数据流, 数据包大小为 1500 packets。

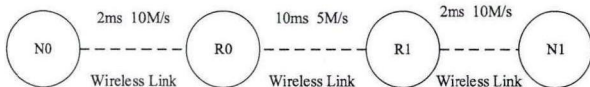


图3 仿真网络拓扑结构

4.2 仿真结果分析

图 4 中 delay-BD、delay-Westwood、delay-Reno 分别为 TCP-BD、TCP-Westwood、TCP-Reno 三种算法下的 delay 时延大小的变化值。从图中可以看出 delay-BD 算法下的 delay 值变化更加平滑, 震荡幅度更小, 在出现网络拥塞时, 可以提供更大的数据包发送量, 提高 TCP 传输性能, 也反映出拥塞控制算法的有效性。

图 5 中 th-BD、th-Westwood、th-Reno 分别为 TCP-BD、TCP-Westwood、TCP-Reno 三种算法下得到的系统吞吐量。仿真结果表明, TCP-BD 算法得到的系统吞吐量最大。

为了更好的模拟 TCP-BD 算法在无线网络下的性能, 在加入不同误码率后分别对 TCP-BD、TCP-Westwood、TCP-Reno

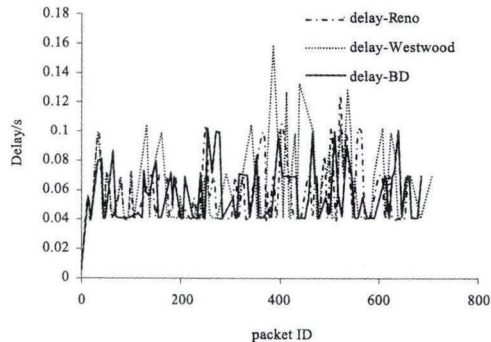


图4 delay 变化对比

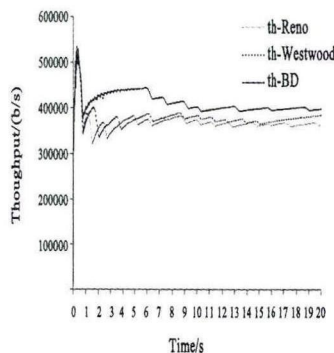


图5 吞吐量结果对比

no 三种算法进行了平均吞吐量的对比, 如图 6 所示。

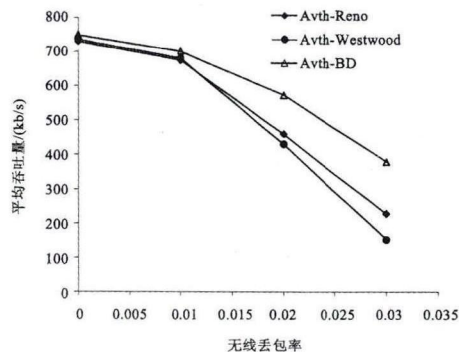


图6 不同链路丢包率下平均吞吐量对比

图 6 中 Avth-Reno、Avth-Westwood、Avth-BD 分别为 TCP-Reno、TCP-Westwood、TCP-BD 算法在无线误码率为 0.01、0.02、0.03 下的平均吞吐量对比图, 从图 6 中可以看出 TCP-BD 算法的平均吞吐量最高。仿真结果表明, 本文提出的 TCP-BD 算法能有效的改进无线网络环境中因无线丢包导致的 cwnd 和 ssthresh 值盲目减小的问题, 使 cwnd 值的变化更加平滑, 更加充分的利用网络带宽发送数据, 很大程度上提高了 TCP 的传输性能。

5 结束语

由于在 TCP 发送端采用可用带宽估计算法,并在计算中平滑了 RTT 和带宽值,从而保证估计带宽的误差较小,进而计算出带宽利用率,当 TCP 在遇到丢包时可以在一定程度上把无线误码丢包和拥塞丢包区分开;同时可区别网络拥塞的程度,并据此完成控制参数的分段自适应动态调整。从仿真图中,可清晰地看到算法对于吞吐量、时延及时延抖动这些网络质量的关键参数度量上都有了很大的改善,也显示出改进后的 TCP 在无线环境中的总体性能上有了较大的提高。本文提出的方案实现简单,且能较准确地区分丢包原因和拥塞程度,并采取不同的控制策略,使 TCP 的自适应能力及网络性能进一步增强。

参考文献:

- [1] 彭玉艳,杜文才,任佳. 基于贝叶斯网络的 Ad Hoc 网络拥塞控制[J]. 计算机仿真, 2014,(5): 312-315.
- [2] 邓晓衡,等. MP-Start:基于带宽测量的分段 TCP 慢启动机制[J]. 通信学报, 2008,28(11): 92-102.
- [3] S K Tiwari, Y K Rana, A Jain. An ECN Approach to Congestion Control Mechanisms in Mobile Ad hoc Networks[J]. Network and Complex System, 2014,4(6): 18-22.
- [4] J Wang, P Dong, J Chen, J Huang, S Zhang, and W Wang. Adaptive explicit congestion control based on bandwidth estimation for high bandwidth delay product networks[J]. Computer communication, 2013,36(10): 1235-1244.
- [5] K Zhang, L Yin, J Lu. Feedback-based adaptive network coded cooperation for wireless networks[J]. EURASIP Journal on Wireless Communications and Networking, 2011,(1): 1-15.

- [6] 秦光. 计算机网络拥塞的高效控制方法研究[J]. 计算机仿真, 2012,(9): 154-157.
- [7] V Jacobson. Congestion avoidance and control[C]. ACM SIGCOMM Computer Communication Review. ACM, 1988,18(4): 314-329.
- [8] M Allman, V Paxson, W Stevens. TCP Congestion control[M]. IETF, RFC2581, 1999.
- [9] S Floyd, T Henderson, A Gurtov. The New Reno modification to TCP's fast recovery algorithm[R]. RFC 2582, April, 1999.
- [10] C Casetti, et al. TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks[J]. Wireless Networks, 2002, 8(5): 467-479.
- [11] WangJianxin, et al. Adaptive explicit congestion control based on bandwidth estimation for high band-width-delay product networks[J]. Computer Communications, 2013,36(10): 1235-1244.
- [12] C Casetti, et al. TCPWestwood: End-to-End Congestion Control for Wired/Wireless Networks[J]. Wireless Networks, 2002, 8(5): 467-479.

[作者简介]



赵宇红(1974-),女(汉族),内蒙古包头市人,硕士研究生导师,副教授,主要研究领域为复杂网络建模及智能控制;

刘海良(1989-),男(汉族),山东省潍坊市人,硕士研究生,主要研究领域为网络拥塞控制;

张晓琳(1966-),女(汉族),内蒙古包头市人,博士,硕士研究生导师,教授,主要研究领域为数据库理论与技术。

(上接第 100 页)

7 结论

经过理论分析和数值仿真得出结论如下:

1)补燃室头部回流区保证燃气在补燃室头部的充分燃烧,形成局部高温区,促进硼粒子的点火燃烧,提高硼粒子的反应完成度;

2)在相同的补燃室结构条件下,通过改变燃气喷头结构可以有效增强补燃室头部回流区,达到形成局部高温区,促进硼粒子的点火燃烧的目的;

3)就本文计算的头部进气固冲发动机而言,通过燃气喷头与补燃室结构的匹配优化,可以将燃烧效率提高 10% 以上。

参考文献:

- [1] 鲍福廷,等. 固体火箭冲压组合发动机[M]. 北京:中国宇航出版社, 2006.
- [2] B Natan. Ignition and Combustion Characteristics of individual Boron Particles in the Flowfield of a Solid Fuel Ramjet [R]. AIAA 87-2034.

- [3] B Natan, A Gany. Ignition and combustion of boron particles in the flowfield of a solid fuel ramjet[J]. Journal of propulsion and Power, 1991,7(1): 37-43.
- [4] R A Stowe. Two Phase Flow Combustion Modelling of a Ducted Rocket[R]. AIAA 2001-3461.
- [5] 胡建新. 含硼推进剂固体火箭冲压发动机补燃室工作过程研究[D]. 国防科技大学, 2006.



[作者简介]

何顺强(1980-),男(汉族),北京市人,硕士研究生,主要研究领域为固体火箭冲压发动机;

胡凡(1979-),男(汉族),湖南长沙市人,硕士研究生,主要研究领域为固体火箭冲压发动机;

刘杰(1984-),男(汉族),北京市人,硕士研究生,主要研究领域为固体火箭冲压发动机。