

文章编号: 1672-2892(2012)01-0013-05

民航移动通信中 TCP 协议的改进

曾孝平, 王兴隆, 王志明, 李 渝

(重庆大学 通信工程学院, 重庆 400044)

摘 要: 针对民航移动通信中民航客机内移动终端与地面用户端到端通信的大时延、高随机误码和突发误码的特点, 在 TCP Veno 的基础上提出了增大初始发送速率, 拥塞快速恢复, 突发误码快速恢复等优化策略。用 OPNET 将本文算法与 Veno 进行了仿真比较, 仿真结果表明, 本文算法较好地克服了大时延和突发误码丢包问题, 显著提高了 TCP 吞吐量。

关键词: Veno 算法; 随机误码; 突发误码; 快速恢复

中图分类号: TN915.04

文献标识码: A

Improvement and research of TCP protocol in civil aviation mobile communication

ZENG Xiao-ping, WANG Xing-long, WANG Zhi-ming, LI Yu

(College of Communication Engineering, Chongqing University, Chongqing 400044, China)

Abstract: In order to realize reliable end-to-end communication in aeronautical mobile communication between mobile terminal in aircraft and ground user host, this paper proposes some strategies such as higher initial sending rate, fast recovery for congestion, fast recovery for burst error. Comparing this algorithm with TCP Veno by using OPNET, the simulation results show that the proposed algorithm can solve the problem of high latency and burst error loss, and significantly increases the TCP throughput.

Key words: Veno algorithm; random error; burst error; fast recovery

在民航客机上为乘客提供语音和数据移动通信业务在技术上已日趋成熟, 需求不断增加。由于飞机到地面主机要经过同步卫星中继, 所以端到端的往返时延非常大, 达到 480 ms~560 ms, 是地面网络的 6 倍以上, 进而造成了很大的带宽时延积; 飞机的高速移动性带来的大多普勒频移和卫星通信环境的复杂性(如天气变化)造成了非常高的随机误码和突发误码。

TCP 协议提供端到端的可靠性传输, 传统地面网络中的 TCP 协议(如 Reno^[1], Newreno^[2])用到的算法主要有: 慢启动、拥塞避免、快速重传和快速恢复, 主要适用于网络时延低、误码率低的环境。在大时延、高误码率的航空移动通信系统中, 现有协议性能急剧下降, 主要原因是大时延造成了慢开始时间过长, 拥塞窗口增长缓慢, 丢包后拥塞检测和恢复迟钝。因为地面网络有极低的误码率, 传统 TCP 将所有丢包都认为是拥塞造成的, 而在移动客机通信中存在高随机误码和突发误码, 如果将误码丢包也作为拥塞进行处理, 势必造成数据发送速率频繁减小, 严重影响吞吐量, 所以, 大时延和高误码率是必须克服的问题。

TCP Vegas^[3-4]引入了拥塞避免机制, 通过估计吞吐量的变化 *Diff* 来预判网络拥塞状况, 发送端计算 *Expected* 和 *Actual* 两个值:

$$Expected = cwnd / Base RTT \quad (1)$$

$$Actual = cwnd / RTT \quad (2)$$

式中: *cwnd* 是当前的拥塞窗口; *RTT* 是当前测量的往返时延; *Base RTT* 是往返时延的最小值; *Expected* 是期望达到的吞吐量; *Actual* 是当前的实际吞吐量。差值 *Diff*:

$$Diff = Expected - Actual \quad (3)$$

设置阈值 α 和 β , 采用退避的方式调整拥塞窗口:

$$cwnd = \begin{cases} cwnd + 1 & \text{if } Diff < \alpha \\ cwnd - 1 & \text{if } Diff > \beta \\ cwnd & \text{otherwise} \end{cases} \quad (4)$$

Vegas 尽量避免网络拥塞的产生, 相比 Reno 算法提高了 30% 到 70% 的吞吐量^[5], 但在长时延的网络中, 性能明显下降^[6], 而且存在严重的公平性问题。

Fu Ch P 等为了解决无线网络的 TCP 通信, 提出了 VenO 算法^[7-8], 在 Reno 算法的基础上将 Vegas 的拥塞检测机制运用到协议中, 通过对链路中堆积的包数目 N , 来估计网络的拥塞状况。当 $N < \beta$ 时, VenO 算法认为当前网络状况良好, 没有产生拥塞; 反之认为网络可用带宽已经用完, 拥塞产生。 N 计算过程如下:

如果 $RTT > BaseRTT$, 则说明链路上存在瓶颈, 使数据包在此堆积, 设堆积的包数为 N , 有:

$$RTT = BaseRTT + N / Actual \quad (5)$$

通过转换不难得到:

$$N = Actual \cdot (RTT - BaseRTT) = Diff \cdot BaseRTT \quad (6)$$

$Diff$ 由式(3)得到。

Veno 算法的优势在于较好地估计了网络的拥塞状况, 将随机误码丢包和拥塞丢包区分开来, 避免将随机丢包误判为拥塞而降低发送端的发送速率; 而且, 在公平性问题上可以与 Reno 算法媲美, 但是, VenO 算法没有考虑大时延和高突发误码问题。

本文在 TCP VenO 基础上, 针对网络大时延问题, 提出了拥塞快速检测算法、增大初始发送窗口和超时快速检测机制; 针对突发误码丢包问题, 提出了突发误码丢包快速恢复算法。

1 TCP VenO

Veno 拥塞控制算法包括慢开始、拥塞避免、快速重传和快速恢复, 算法的具体实现如下:

1) 慢开始阶段(同 Reno): 初始拥塞窗口($cwnd$)设为 1 个最大传输报文段(Maximum Segment Size, MSS), 每收到 1 个 ACK(Acknowledge character)将拥塞窗口增加 1 个 MSS, 直到拥塞窗口达到慢启动阈值($ssthresh$, 初始值为 64 KB), 即 $cwnd \geq ssthresh$, 进入拥塞避免阶段;

2) 拥塞避免阶段: 根据式(6)计算 N , 如果 $N < \beta$ (β 经验值为 3), 认为网络状态良好, 则每个 RTT 将拥塞窗口增加 1 个 MSS, 同 Reno; 反之, 认为网络状况趋于饱和, 可用带宽已经用尽, 则每 2 个 RTT 增加 1 个 MSS。即:

$$cwnd = \begin{cases} cwnd + 1 / cwnd & \text{if } N < \beta \\ cwnd + 0.5 / cwnd & \text{if } N \geq \beta \end{cases} \quad (7)$$

网络趋于拥塞后, 拥塞窗口以更慢的速度增加, 使拥塞窗口尽可能长时间地保持在较高的状态以获得更高的吞吐量。

3) 快速重传和快速恢复(fast retransmit and fast recovery): 当收到 3 个重复 ACK 后, 不等重传定时器超时就进入快速重传和快速恢复, 并进行如下操作:

a) 重传丢失的包, 将慢启动门限设为:

$$ssthresh = \begin{cases} cwnd \times (4 / 5) & \text{if } N < \beta \\ cwnd / 2 & \text{if } N \geq \beta \end{cases} \quad (8)$$

拥塞窗口设为:

$$cwnd = ssthresh + 3MSS \quad (9)$$

b) 每收到 1 个重复 ACK, 将 $cwnd$ 增加 1 个 MSS;

c) 收到确认了新数据的 ACK 后, 设置拥塞窗口: $cwnd = ssthresh$, 并退出快速恢复。

快速重传和恢复算法与 Reno 算法的不同之处在于步骤 1), Reno 算法将所有丢包都认为是拥塞产生的, 并将 $ssthresh$ 设置为拥塞窗口的一半, 而 VenO 算法将随机丢包与拥塞丢包做了区分, 估计到随机丢包后, VenO 网络未出现拥塞, 带宽仍未用完, 所以不需要降低发送速率(保守起见, 稍微降低发送速率), 避免了拥塞窗口大幅减小带来的性能下降。

Veno 算法很好地估计了丢包产生的原因, 避免了由于随机丢包被误判为拥塞, 使 $cwnd$ 频繁减半造成吞吐量下降。但是, VenO 算法在克服大时延、高突发误码方面作用乏力。

2 民航移动通信中 TCP 协议改进

为了实现民航移动网络端到端通信，针对网络大时延、高突发误码的特点，本文提出了增大初始发送速率，拥塞快速检测，突发误码丢包快速恢复和超时快速检测等机制。

2.1 增大初始发送速率

Veno 沿用 Reno 的慢启动算法， $cwnd$ 的增加是以 ACK 的回复为条件的，每收到 1 个 ACK ，将 $cwnd$ 增加 1 个 MSS ，默认情况下初始发送窗口值为 1 个 MSS ，而在大时延网络中这样的慢启动过程要经历更长的时间。为了在短时间内让拥塞迅速增加到较高水平，将初始发送窗口值设为^[9]：

$$cwnd = \min(4MSS, \max(2MSS, 4380 \text{ byte})) \quad (10)$$

增大发送速率的目的是，短时间内数据发送端收到更多 ACK ，使 $cwnd$ 尽快达到理想水平，在大带宽时延积、高随机误码和突发误码的民航移动通信中，丢包的原因主要是误码，而且慢启动的时间决定了卫星信道带宽的利用效率，所以将慢启动初始发送速率(即初始发送窗口)适当提高是可取的。

2.2 拥塞的快速检测

由于大时延的存在，如果希望能更快地检测到拥塞以便进行快速恢复，考虑到已经对网络的拥塞状况进行了有效的估计，在估计到网络带宽已经用尽后，如果收到重复 ACK ，那么极有可能是拥塞造成的。Veno 算法沿用了 Reno 快速重传和恢复机制，发送端收到 3 个重复 ACK 后，进行快速重传和恢复。本文对该策略的改进如下：

$N \geq \beta$ 说明可用带宽已经用尽，丢包更可能源于网络拥塞，而失序的可能性被削弱，所以收到 2 个重复 ACK ，即认为丢包是拥塞造成的，并进入快速重传和恢复，这是针对网络长时延而作的取舍。

如果 $N < \beta$ ，认为网络可用带宽未被用尽，如果收到重复 ACK ，则可能是失序或者误码产生的丢包，为了排除数据包失序的可能性，收到 3 个重复 ACK 后，认为误码产生了丢包并进行快速重传和恢复，这点与 Veno 算法相同，处理过程如下：

$$ssthresh = \begin{cases} cwnd \times (4/5) & \text{if } N < \beta \text{ and 3 dup } ACK \\ cwnd / 2 & \text{if } N \geq \beta \text{ and 2 dup } ACK \end{cases} \quad (11)$$

检测到丢包后进行快速重传和快速恢复。

2.3 突发误码丢包的快速恢复

由于高突发误码的存在，造成 1 个拥塞窗口内多个包丢失，而 Veno 算法沿用了 Reno 的重传策略，对 1 个拥塞窗口内多个包的丢失处理能力非常弱。如果出现突发误码而造成的连续丢包，Veno 算法会多次启动拥塞控制进程使拥塞窗口迅速下降，甚至超时重传进入慢开始阶段，这样极大降低了吞吐量。为了避免这个问题，引入 NewReno^[4]的思想，设置 $recover$ 变量，记录快速重传时已发送的最大报文序号，即：

$$recover = send_max \quad (12)$$

快速重传后，当收到确认了新数据的 ACK (New ACK)后，判断 ACK 和 $recover$ 的关系，并对 Veno 算法的快速重传和恢复阶段的步骤 3)作如下修改：

```
if(New ACK ≤ recover) //部分确认
    then:
        重传最早未确认的包;
        cwnd=cwnd-ACKedBytes+MSS;// ACKedBytes:确认的字节数等待下一个 ACK;
else if(New ACK > recover) //完全确认
    then:
        cwnd=ssthresh;
        退出快速恢复;
```

突发误码往往造成 1 个窗口内多个包的连续丢失，引入 $recover$ 变量，1 次重传后不立即退出快速恢复，而是等到 $New\ ACK > recover$ 才退出，解决了突发误码连续丢包产生的拥塞恢复问题，使拥塞窗口得到快速恢复。

2.4 超时快速检测机制

传统 TCP 的粗粒度超时检测机制在时钟滴答(500 ms)的倍数处进行超时检测，考虑到丢包非常严重，或者由于拥塞窗口太小，导致发送端不能收到 3 个重复 ACK 而进入超时重传，为了更及时而准确地检测到超时，引入

Vegas 的超时检测机制，当收到 1 个重复 ACK，就检测往返时延是否超过了超时时间，如果已超时，就重传丢失的包。

由于突发误码的存在，而且拥塞往往造成多个包的连续丢失，1 个包丢失而快速重传后，其后面的几个包可能已经超时。快速重传或超时重传后，检测已经发送的 1 或 2 个数据包是否超时，如果超时就进行重传，这样可以更快检测到超时。

3 仿真结果与性能分析

在 OPNET 网络仿真平台下，对本文提出的协议进行了仿真，搭建的网络仿真模型如图 1 所示。飞机以 1 000 km/h 的速率飞行，作为客户端，通过同步卫星、关口站、PSTN 向服务器请求数据(数据量为 6.4 MB)，以 1% 的误码率设置随机误码。突发误码丢包和误码丢包的设置情况如表 1 所示。

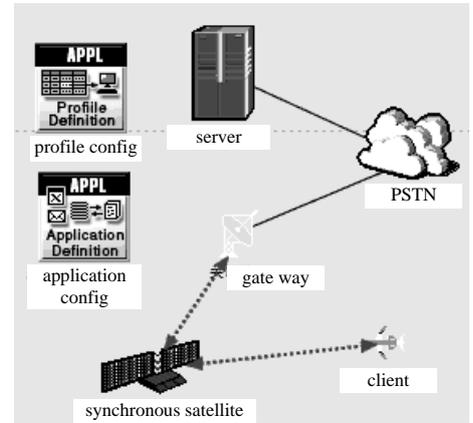


Fig.1 Network simulation model

图 1 网络仿真模型

将 Reno, Veno 算法和本文改进算法在上述设置的通信环境下同时进行仿真。仿真后提取 3 种算法的重传次数分布图(图 2)、拥塞窗口变化图(图 3、图 4)和吞吐量变化图(图 5)。

图 2 给出了 3 种算法的重传次数，可以看到，时间 31 s, 52 s, 85 s 左右分别重传了 5 次，3 次，3 次，是突发丢包产生的重传；在时间 43 s, 64 s, 71 s, 93 s, 106 s 左右都丢了 1 个包，是随机误码产生的。3 种算法在同样的误码率情况下丢失了相同的包，但分别表现出不同的拥塞控制策略，在不用的时间处做了重传操作(下文详细说明)。

表 1 突发误码丢包和误码丢包的设置
Table 1 Settings for Burst loss and Random loss

packet	time/s	type
5	31	B
1	43	R
3	52	B
1	64	R
1	71	R
3	85	B
1	93	R
1	106	R
5	119	B

注：B(Burst)为突发误码；R(Random)为随机误码

图 3 将 Veno 算法和 Reno 算法的拥塞窗口进行了比较，可以看出，在随机误码处(43 s, 64 s, 71 s, 93 s, 106 s)，Reno 算法拥塞窗口减半；而 Veno 算法在丢包时先估计了链路缓存的包个数，在随机误码处，估计的包数 $N < \beta (\beta = 3)$ ，所以按 Veno 的拥塞控制算法，拥塞窗口仅降低了 1/5；然而在突发误码时间处(31 s, 52 s, 85 s)，Veno 算法和 Reno 算法由于连续丢包造成了超时，都进入了慢启动进程，这说明面对突发误码产生多个包丢失，Veno 算法并未表现出更好的拥塞窗口恢复能力。

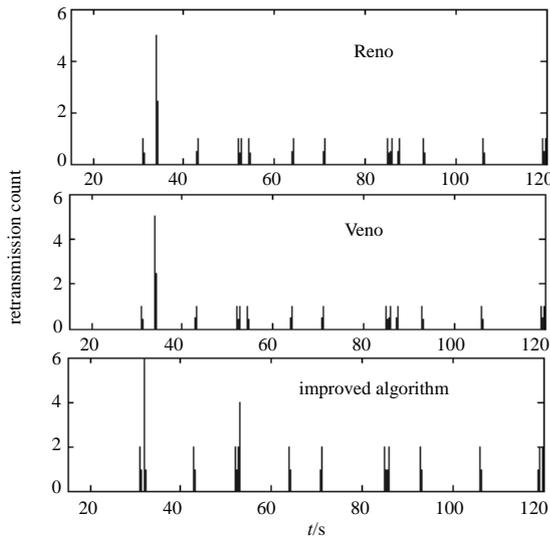


Fig.2 Retransmission count distribution of the three algorithms
图 2 三种算法的重传次数分布图

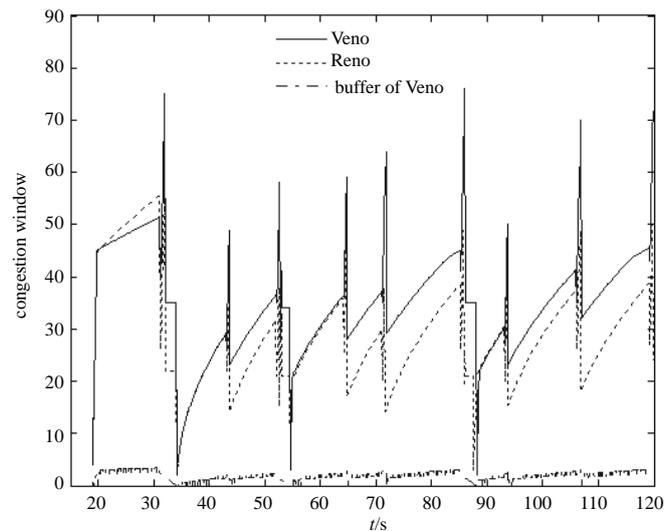


Fig.3 Congestion window of Reno and Veno
图 3 Reno 和 Veno 的拥塞窗口

图 4 将本文算法和 Veno 算法的拥塞窗口进行了比较，在随机误码处(43 s, 64 s, 71 s, 93 s, 106 s)，改进算法保持了与 Veno 算法同样的恢复能力，只将拥塞窗口减小了 1/5。而在突发误码处(31 s, 52 s, 85 s)，改进算法由于采用了突发误码的快速恢复机制，收到部分确认($New ACK \leq recover$)不会立即退出快速恢复，一直等到完全确认(New

ACK>recover)到达,才退出快速恢复,拥塞窗口得到了很好的恢复和保持,而且未进入慢开始进程,这说明本文算法面对突发误码产生的连续丢包拥塞窗口保持了更高的水平,明显优于 Veno 算法。

图 5 给出了本文算法、Veno 算法和 Reno 算法的平均吞吐量的比较结果,可以明显看到本文算法保持了更高的吞吐量并趋于稳定,优于其他 2 种算法。

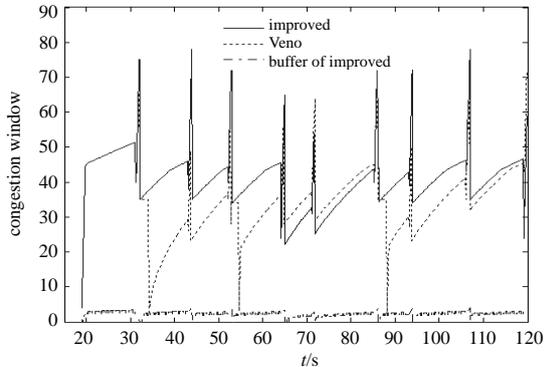


Fig.4 Congestion window of this improved algorithm
图 4 本文改进算法的拥塞窗口

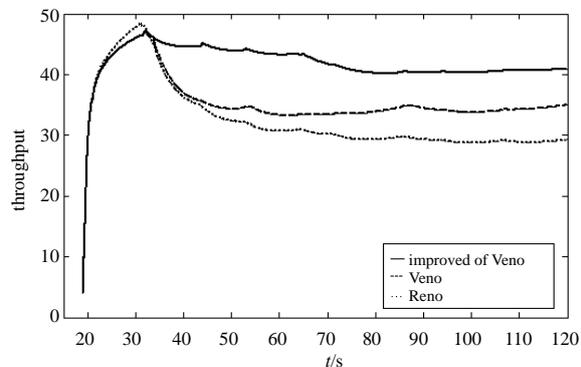


Fig.5 Throughput comparison
图 5 吞吐量比较

4 结论

为了实现民航移动通信中端到端的可靠通信,针对民航移动通信的特点,本文在 Veno 算法的基础上对拥塞控制策略进行了改进:设置了更大的初始发送速率,引入拥塞快速检测机制、突发误码丢包的快速恢复和更准确的超时检测机制,克服了民航移动通信系统中端到端大时延、高随机误码和突发误码的问题。本文提出的改进算法在拥塞窗口的恢复和吞吐量方面,明显优于 TCP Veno 算法和 Reno 算法,显著提高了 TCP 性能。

参考文献:

- [1] Jacobson V,Karels M J. Congestion avoidance and control[C]// ACM SIGCOMM'88, New York:[s.n.], 1988:273-288.
- [2] Floyd S,Henderson T,Gurtov A. The New Reno Modification to TCP's Fast Recovery Algorithm[S]. RFC3782, 2004.
- [3] Brakmo L S,Peterson L L. TCP Vegas:End to End Congestion Avoidance on a Global Internet[J]. IEEE JSAC, 1995,13(8): 1465-1480.
- [4] Shihada B,Zhang Q,Ho P-H,et al. A novel implementation of TCP Vegas for optical burst switched networks[J]. Optical switching and networking, 2010,7(3):115-126.
- [5] MO J,La R J,Anantharam V,et al. Analysis and Comparison of TCP Reno and Vegas[C]// Proceedings IEEE INFOCOM'99. New York:[s.n.], 1999:1556-1563.
- [6] Sing J,Soh B. TCP New Vegas:Improving the Performance of TCP Vegas Over High Latency Links[C]// 4th IEEE International Symposium on Network Computing and Applications. Cambridge,MA:[s.n.], 2005:73-82.
- [7] Fu Ch P,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.
- [8] WU Yanjie,BAI Guangwei,WU Zhigang. Performance analysis and evaluation of TCP Veno[J]. Computer Engineering and Design, 2009,30(12):2934-2940.
- [9] Allman M,Floyd S,Partridge C. Increasing TCP's Initial Window[S]. RFC2414, 1998.

作者简介:



曾孝平(1955-),男,四川广安人,教授,博士生导师,主要研究方向为电路与系统、信号与信息处理.email:zxp@ccee.cqu.edu.cn.

王兴隆(1986-),男,山东临沂人,在读硕士研究生,主要研究方向为通信信号处理。

王志明(1986-),男,陕西咸阳人,在读硕士研究生,主要研究方向为传感器网络路由协议。

李渝(1987-),男,重庆市人,在读硕士研究生,主要研究方向为无线多址接入技术。