

文章编号:1001-9081(2008)11-2763-04

基于 GEO 卫星链路丢包区分的 TCP Westwood 改进算法

王建峰, 黄国策, 康巧燕

(空军工程大学 电讯工程学院, 西安 710077)

(wjf4206@126.com)

摘要: 在分析 TCP Westwood (TCPW) 算法优缺点的基础上, 针对其应用于同步轨道 (GEO) 卫星链路时存在的不足, 结合 Vegas、Veno 及 LogWestwood + 等改进算法的优势, 基于预测的下一时刻的网络带宽, 把窗口调整与带宽利用情况相结合, 提出了一种新的适合于 GEO 卫星链路的基于丢包区分的 TCP Westwood 改进算法。改进算法将每个阶段的窗口调整与带宽估计、网络状态紧密联系起来, 结合网络状态和带宽估计判断拥塞窗口的合理性, 动态地调整拥塞窗口, 使拥塞即将发生时, 窗口能及时下降到适宜的水平, 尽量避免由于拥塞而导致的分组丢失。仿真结果表明, 改进算法提高了 TCP westwood 在 GEO 卫星链路中应用时的性能, 具有较好的吞吐量、公平性、友好性和较低的丢包率。

关键词: 同步轨道卫星; 拥塞控制; TCP Westwood; 丢包区分; 带宽估计

中图分类号: TP393 **文献标志码:** A

Enhanced TCP westwood algorithm based on loss differentiation over GEO satellite networks

WANG Jian-feng, HUANG Guo-ce, KANG Qiao-yan

(Institute of Telecommunication Engineering, Air Force Engineering University, Xi'an Shaanxi 710077, China)

Abstract: The advantages and disadvantages of TCP Westwood (TCPW) algorithm were analyzed, especially when it was used in Geostationary Earth Orbit (GEO) satellite networks. In order to solve the existing problems, a new improved TCPW (TCPW-N) algorithm considering loss differentiation was proposed. TCPW-N algorithm took full advantage of Vegas, Veno and the existing improved TCPW algorithm (namely LogWestwood +), and TCPW-N was suitable for GEO satellite networks. TCPW-N algorithm can judge the packet loss reason, and adjust the congestion window according to the different packet loss reasons and the estimated network bandwidth. Performance analysis and simulation results show that TCPW-N algorithm has effectively resolved the existing problems of TCPW. Compared with TCPW and its other improved algorithm, TCPW-N algorithm has better throughput performance, lower packet loss rate, and better fairness and friendliness when applied to GEO satellite networks.

Key words: Geostationary Earth Orbit (GEO) satellite; congestion control; TCP Westwood (TCPW); loss differentiation; bandwidth estimation

0 引言

TCPW (TCP Westwood)^[1] 针对无线网络的特点对传统拥塞控制协议 TCP Reno 进行了改进, 它提出了一种较为精确的带宽估计算法, 通过监控到达的肯定应答 (Acknowledgment, ACK) 流确定网络当前的可用带宽。TCPW 的动态估算公式如下:

$$BE_k = \alpha_k BE_{k-1} + (1 - \alpha_k) \left(\frac{b_k + b_{k-1}}{2} \right)$$

$$\alpha_k = \frac{2\tau - \Delta t}{2\tau + \Delta t}$$

$$b_k = \frac{d_k}{t_k - t_{k-1}}$$

其中: d_k 是第 k 个 ACK 所响应的数据量, t_k 是第 k 个 ACK 的到达时间, BE_k 是 t_k 时刻的带宽估计值, $1/\tau$ 是滤波器的截止频率。

当检测到分组丢失后, TCPW 根据带宽估计来确定新的

窗口和慢启动门限值, 而不是像传统 TCP Reno 那样盲目降低窗口, 从而消除了高误码率的影响, 在随机错误较多的无线链路上, TCPW 能获得较好的性能。

同步轨道 (Geostationary Earth Orbit, GEO) 卫星网络是一种特殊的无线网络, 除高误码率外, 另一个主要特征是长时延。TCPW 虽然能较好地处理高误码率的影响, 但对长时延的影响考虑不足, 在时延较大的环境中, 表现不稳定。文献 [2] 的仿真表明, 当时延较小时, 一定比例的随机错误对提升 TCPW 的性能是有益的。但随着时延的增加, TCPW 的吞吐量迅速下降, 波动幅度明显增加。TCPW 的窗口校准是发生在丢包之后, 在时延较小时, TCPW 通过丢包后的处理可及时校准窗口值, 提高吞吐量。随着时延的增大, 上一往返时间 (Round-Trip Time, RTT) 的测量值和当前实际网络情况的相关性下降, 校准的精度下降, 导致 TCPW 的性能下降。

TCPW 在慢启动和拥塞避免阶段仍采用 Reno^[3] 的增加机制, 这是一种较为盲目的增加机制, 未考虑网络的实际可用

收稿日期: 2008-05-15; 修回日期: 2008-07-28。 基金项目: 军队科研基金资助项目 (KJZ06618)。

作者简介: 王建峰 (1978-), 男, 陕西武功人, 讲师, 硕士, 主要研究方向: 卫星通信、TCP 拥塞控制; 黄国策 (1962-), 男, 陕西高陵人, 教授, 博士生导师, 主要研究方向: 卫星通信、移动通信; 康巧燕 (1980-), 女, 福建永春人, 讲师, 博士, 主要研究方向: 宽带网络技术、网络拥塞控制。

带宽,也未区分分组丢失的原因。

针对 TCPW 存在的问题,有不少研究者提出了相应的改进算法。如文献[2]针对卫星网长时延的特点,提出一种较保守的 TCPW 改进方案,每次带宽估计计算后确定当前拥塞窗口是否适宜,一旦当前窗口大于估计带宽,就迅速降低窗口。文献[4]针对动态带宽和大传播时延提出 TCPW-A 算法,采用一种灵活的探测机制,对 TCPW 的慢启动和拥塞避免机制进行了改进。文献[5]针对卫星网络中由于链路误码引起的多包丢失的恢复问题,而提出 TCPW BR 算法。文献[6]针对 TCPW 在慢启动和拥塞避免阶段的盲目递增,结合 TCPW 和 TCP Vegas^[7]提出一种 TCP 增强算法。文献[8]针对 TCP 慢启动存在的问题,结合 TCP Vegas 和 TCPW 的关键技术,提出一种动态门限变速增加的 TCP 慢启动算法,采用动态设定慢启动门限和分段的窗口增加粒度。文献[9]针对大的带宽时延乘积网络中,TCPW 的拥塞避免阶段拥塞窗口(Congestion Window, CWND)增长速度过慢问题,提出 LogWestwood + 算法,即在拥塞避免阶段拥塞窗口采用对数增长的方法。

这些改进算法都有较好的性能,但不能很好地适应 GEO 卫星网络环境,本文结合上述改进算法及 Veno 算法^[10]的优势,提出一种新的基于丢包区分的 TCPW 改进算法,对 TCPW 的慢启动、拥塞避免、快速重传和快速恢复分别进行改进,称为 New TCP Westwood,简记为 TCPW-N。

1 基于丢包区分的 TCPW 改进算法

1.1 算法基本思想

首先,采用和 Veno 类似的网络状态区分算法,在慢启动阶段判断网络带宽是否利用充分,在拥塞避免阶段区分网络状态,在快速重传和快速恢复阶段判断分组丢失原因。然后,针对不同的情况,分别比较当前拥塞窗口与预测到的下一时刻可用带宽的大小关系,并采用不同的拥塞窗口调整策略。在调整窗口的过程中,针对慢启动后期拥塞窗口增长过快容易造成多包丢失问题,采用较保守的对半增加的方法。在拥塞避免阶段,当判断网络处于正常状态时,拥塞窗口采用 LogWestwood + 算法的较为激进的对数增加方法,以节省窗口恢复到最佳值的时间,这对于长时延的 GEO 卫星网络尤其重要;在快速重传和快速恢复阶段,根据判断的不同分组丢失原因,采用不同的窗口调整策略。

改进算法借鉴了 TCP Veno 的状态区分算法,用于区分丢包原因,但在算法的实现上有很大的不同。TCPW-N 在考虑丢包原因的基础上,还考虑了网络带宽的利用情况(通过比较当前的拥塞窗口与预测到的下一时刻的可用带宽),把每个阶段的窗口调整与带宽估计、网络状态紧密联系起来。结合网络状态和带宽估计更加合理的调整拥塞窗口,使网络在非拥塞状态时,以较快的速度增长,更快地到达拥塞窗口最佳值;而在拥塞即将发生时,窗口能及时下降到适宜的水平,而不是只有丢包时才进行窗口减小的操作,尽量避免由于拥塞而导致的分组丢失。

1.2 慢启动改进算法

队列中的报文积压长度为 N ,设定一个 β 值,通过对 N 和 β 的比较来区分当前连接所处的状态(有实验表明^[10],门限值 β 取 3 较为合适)。当 $N < \beta$ 时,判断网络带宽未充分利用,若当前拥塞窗口小于预测到的下一时刻的可用带宽,拥塞窗口遵循指数增长规律,即每接收到一个 ACK 就增加 1;否则,

拥塞窗口每个 RTT 增加门限值与拥塞窗口之差的一半,即 $(ssthresh - cwnd)/2$,逐步逼近门限阈值 $ssthresh$,直到其差值小于调节因子 δ ($2 \leq \delta \leq ssthresh/2$),将拥塞窗口设为门限值,从而进入拥塞避免阶段。当 $N \geq \beta$ 时,判断网络带宽充分利用,窗口增长速度调整为网络带宽未充分利用时增长速度的一半。具体算法描述如下:

```

For every ACK
  { If ( cwnd < ssthresh )
    { If ( N < β )                                // 网络带宽未充分利用
      { If ( cwnd_k < BE_{k+1} )
        { cwnd_{k+1} = cwnd_k + 1; }
      Else if ( cwnd_k ≥ BE_{k+1} && ssthresh_{current} - cwnd_k ≥ δ )
        { cwnd_{k+1} = cwnd_k + min(1, (ssthresh_{current} - cwnd_k)/(2cwnd_k)); }
      Else { go to congestion avoidance; }
    }
    Else if ( N ≥ β )                                // 网络带宽利用充分
      { If ( cwnd_k < BE_{k+1} )
        { cwnd_{k+1} = cwnd_k + 1/2; }
      Else if ( cwnd_k ≥ BE_{k+1} && ssthresh_{current} - cwnd_k ≥ δ )
        { cwnd_{k+1} = cwnd_k + min(1/2, (ssthresh_{current} - cwnd_k)/(4cwnd_k)); }
      Else { go to congestion avoidance; }
    }
    Else { go to congestion avoidance; }
  }
}
}

```

1.3 拥塞避免改进算法

借鉴 Veno 的状态区分算法,当判断网络处于正常状态时,若当前拥塞窗口小于预测到的下一时刻可用带宽,窗口采用 LogWestwood 算法的对数增加方法。若当前拥塞窗口大于预测到的下一时刻可用带宽,窗口的增长采用标准的线性增长方式。当判断网络处于拥塞状态时,若当前拥塞窗口小于预测到的下一时刻可用带宽,窗口的增长仍采用线性增长方式,但拥塞窗口每 2 个 RTT 增加 1 个字段。若当前拥塞窗口大于预测到的下一时刻可用带宽,拥塞窗口保持不变。具体算法描述如下:

```

If ( a new ACK is received)
  { If ( N < β )                                // 判断网络处于正常状态
    { If ( cwnd_k < BE_{k+1} )
      { cwnd_{k+1} = cwnd_k + max((cwnd_{max} - cwnd_k)/(2cwnd_k), 1/cwnd_k); }
    Else if ( cwnd_k ≥ BE_{k+1} )
      { cwnd_{k+1} = cwnd_k + 1/cwnd_k; }
    }
    Else if ( N ≥ β )                                // 判断网络处于拥塞状态
      { If ( cwnd_k < BE_{k+1} )
        { cwnd_{k+1} = cwnd_k + 1/2cwnd_k; }
      Else if ( cwnd_k ≥ BE_{k+1} )
        { cwnd_{k+1} = cwnd_k; }
      }
    }
  }
}

```

其中 $cwnd_{max}$ 为检测到最后一个分组丢失事件时最大的拥塞窗口大小^[9]。

1.4 快速重传和快速恢复改进算法

当接收端接收到 3 个重复的 ACK,进入快速恢复阶段,此时可能达到了网络的容量或是链路随机错误导致一个或多个数据包丢失,通过预测网络可用带宽,与拥塞窗口相比较,作出相应的拥塞窗口和门限阈值调整。若 $N < \beta$,认为丢包均

是由于链路误码造成的,并非网络发生拥塞,此时,若当前窗口小于可用带宽,拥塞窗口和门限阈值保持不变;若当前窗口大于可用带宽,发送速率只适当降低,以避免造成带宽的浪费。当 $N \geq \beta$ 时,则表明网络趋于饱和导致数据包丢失,并未发生严重拥塞,应将发送速度调整到一个适当的值,确保网络带宽充分利用,又不致引发拥塞。此时,同样根据当前窗口与可用带宽的关系区分调整。

当网络重传计时器发生超时,进入快速重传阶段,网络可能发生严重拥塞或者卫星链路处于严重误码状态,导致持续的数据包丢失,或者 ACK 丢失或延时。根据当前拥塞窗口与预测到的网络可用带宽的关系,可进一步判断。若 $N < \beta$,认为网络并未发生拥塞,认为网络并未发生拥塞,丢包可能由于链路误码,或者可能是由于 ACK 丢失或延时,此时发送速率应降低,调整到一个合适的值,但不必启动慢启动。当 $N \geq \beta$ 时,则表明网络严重拥塞应重新启动慢启动解除拥塞。具体算法描述如下:

```

If (the sender receive 3 duplicated ACKs)
  { If ( $N < \beta$ ) // 判断包丢失是由于链路误码引起
    { If ( $cwnd_k < BE_{k+1}$ )
      {  $ssthresh_{new} = ssthresh_{current};$ 
        $cwnd_{k+1} = cwnd_k;$  }
     Else if ( $cwnd_k \geq BE_{k+1}$ )
      {  $ssthresh_{new} = 4cwnd_k/5;$ 
        $cwnd_{k+1} = ssthresh_{new};$  }
    }
   Else if ( $N \geq \beta$ ) // 判断包丢失是由于网络拥塞引起
    { If ( $cwnd_k < BE_{k+1}$ )
      {  $ssthresh_{new} = \max((BE_{k+1} * RTT_{ave})/Seg\_size, cwnd_{k+1}/2);$ 
       //RTTave 为测量到的 RTT 的平均值
        $cwnd_{k+1} = ssthresh_{new};$  }
     Else if ( $cwnd_k \geq BE_{k+1}$ )
      {  $ssthresh_{new} = \max(2, (BE_{k+1} * RTT_{min})/Seg\_size);$ 
        $cwnd_{k+1} = ssthresh_{new};$  }
    }
  }
}

If (coarse timeout expires)
  { If ( $N < \beta$ )
    // 认为链路误码丢包或 ACK 丢失而导致定时器超时
    {  $ssthresh_{new} = \max((BE_{k+1} * RTT_{min})/Seg\_size, cwnd_{k+1}/2);$ 
      $cwnd_{k+1} = ssthresh_{new};$  }
   Else if ( $N \geq \beta$ )
    // 认为网络拥塞引起的丢包导致了定时器超时
    {  $ssthresh_{new} = \min((BE_{k+1} * RTT_{min})/Seg\_size, cwnd_{k+1}/2);$ 
      $cwnd_{k+1} = 1;$  }
  }
}

```

2 算法特点分析

从前边的描述中,可看出改进算法 TCPW-N 具有如下特点:

1) 在慢启动阶段,改进算法对慢启动阶段的网络状态根据队列中的报文长度 N 进行区分,并把窗口调整与估计带宽紧密结合起来,根据网络状况,采用不同的窗口增加策略,避免了原有 TCPW 中拥塞窗口的盲目增加导致的分组拥塞丢失。此外,对于分段,不是采用固定值 $ssthresh/2$,而是根据当前的拥塞窗口与预测到的下一时刻的可用带宽的关系来分段,具有动态自适应性。

2) 在拥塞避免阶段,改进算法区分网络状态,在正常工作状态,若当前拥塞窗口小于可用带宽,为了更快达到最优窗口大小,窗口采用对数增长方式;若当前拥塞窗口大于可用带

宽,保持原有的线性增长方式。在拥塞状态,若当前拥塞窗口小于可用带宽,降低窗口增加速度,每收到两个 ACK 窗口增加 1;若当前拥塞窗口大于可用带宽,保持原有拥塞窗口值不变。这可为其他的链路提供多出的缓存空间,使 TCPW-N 连接充分地利用链路、更长时间处于较大的窗口状态,同时减少了拥塞次数和丢包数,提高网络利用率和吞吐量。

3) 在快速重传和快速恢复阶段,采取更为保守的减小窗口的方法。当发送端接收到 3 个重复 ACK 时,在区分丢包原因的基础上,根据当前窗口与可用带宽的关系,采用相应的窗口调整方法,避免了窗口盲目减小而造成的带宽浪费,使窗口能更快的恢复到最佳值。当重传计时器超时,首先判断超时是由于拥塞丢包或链路误码丢包(或 ACK 丢失)引起的,针对不同的超时原因,采用不同的重传策略,避免了不必要的慢启动,减少了带宽的浪费,提高了网络的性能。

(4) TCPW-N 算法的实现只需对源端代码进行修改,保持了 TCP 传输协议的端到端语义,实现简单,且有较好的兼容性和灵活性,易于在实际网络中广泛配置。

3 仿真结果与分析

通过 NS2 仿真验证改进算法的有效性。仿真拓扑结构如图 1 所示, M 个源节点通过 10 MB、5 ms 的链路与路由器 1 相连, M 个目标节点通过 10 MB、5 ms 的链路与路由器 2 相连,两路由器之间的链路模拟 GEO 卫星链路,带宽为 2 MB,延时为 270 ms,路由器与 GEO 卫星之间绑定均匀分布的误差模型。为了反映卫星链路资源受限的特性,两路由器之间链路的队列长度设为缺省值 20 个分组,每个分组的大小为 1 kB,仿真时间为 100 s。

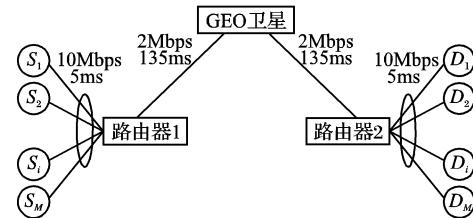


图 1 网络仿真拓扑图

3.1 单个 TCP 连接的性能

考虑单个连接,取 $M = 1$,此时只有 S_1 到 D_1 之间有 TCP 流,分布在 S_1 和 D_1 之间运行 TCPW 算法、TCPW-S 算法^[2] 和 TCPW-N 算法。图 2 和表 1 给出了各种 TCP 连接的吞吐量随分组误码率变化的性能,可以看出,改进算法 TCPW-N 的吞吐量性能明显优于 TCPW 和 TCPW-S,当误码率较小时,随着分组误码率增加,吞吐量性能的改善越明显,这是因为 TCPW-N 算法能较好地区分分组丢失的原因,并且针对不同的分组丢失,结合带宽估计,以更适当的方式调整拥塞窗口大小。但是,随着分组误码率的增大,当超过 1% 时,改进算法相对于 TCPW-S 和 TCPW 的吞吐量性能改善逐渐降低,这是因为当分组误码率较大时,分组丢失主要是由于信道误码引起的,改进算法中的区分分组丢失功能逐渐不起作用。

表 1 给出了不同分组误码率条件下, TCPW-N/TCPW-S 和 TCPW-N/TCPW 的吞吐量比值和丢包数比值的比较。从表中可看出,改进算法 TCPW-N 相对于 TCPW 和 TCPW-S 的吞吐量性能的改善。此外,从表中还可看出,在相同的信道条件下,改进算法 TCPW-N 的丢包数相对于 TCPW 和 TCPW-S 有较大的减少。当分组误码率较小时,随着误码率的增加,丢包数减小的比率增大;当分组误码率较大时,TCPW-N 连接的

丢包数仍然小于 TCPW 和 TCPW-S,但是,随着误码率的继续增大,丢包数减小的比率逐渐减小。

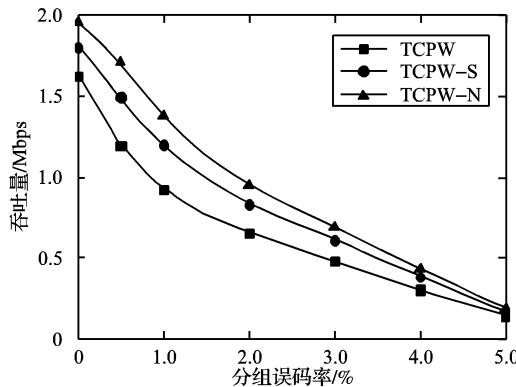


图 2 不同 PER 下各算法吞吐量性能比较

表 1 TCPW-N、TCPW-S 和 TCPW 的性能比较

PER/%	TCPW-N/TCPW-S		TCPW-N/TCPW	
	吞吐量比值	丢包数比值	丢包数比值	吞吐量比值
0.00	1.081	0.893	1.202	0.741
0.05	1.094	0.868	1.246	0.708
0.10	1.121	0.852	1.353	0.625
0.50	1.143	0.841	1.425	0.614
1.00	1.151	0.833	1.487	0.603
5.00	1.139	0.926	1.341	0.782

3.2 多个 TCP 连接的性能

取 $M = 2$, 考虑两个 TCPW-N 连接在不同分组误码率下竞争带宽的吞吐量性能, 以考查多个 TCPW-N 连接竞争带宽的公平性。图 3 和图 4 分别给出了分组误码率为 0% 和 5% 时, 两个 TCPW-N 连接的吞吐量性能。由图可知, TCPW-N 算法对于同种类的 TCP 连接(即 TCPW-N 连接)具有很好的公平性。

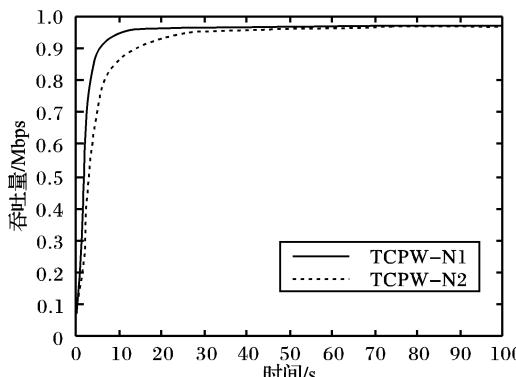


图 3 PER = 0% 时两个 TCPW-N 连接性能比较

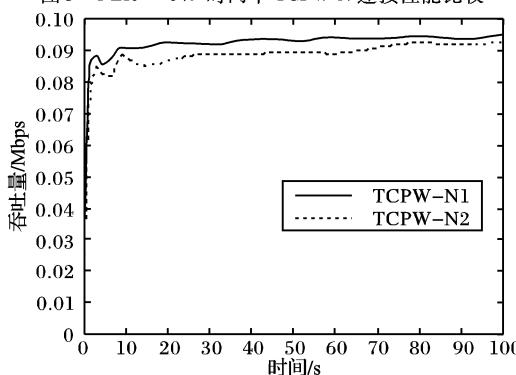


图 4 PER = 5% 时两个 TCPW-N 连接性能比较

取 $M = 4$, 考虑两个 TCPW-N 连接和两个 Reno 连接在不同分组误码率下竞争带宽的吞吐量性能, 以考查改进算法

TCPW-N 对现有的广泛应用的 TCP Reno 连接的 TCP 友好性。图 5 和 6 给出了分组误码率为 0% 和 5% 时, 两个 TCPW-N 连接的总吞吐量和两个 Reno 连接的总吞吐量随时间的变化情况。由图可知, 不管分组误码率为 0% 还是 5%, TCPW-N 连接与 Reno 连接各自所获得的总带宽, 基本上等于单个连接时 TCPW-N 连接和 Reno 连接各自所获得的带宽的一半。由此可见, 与 TCP Reno 具有较好的 TCP 友好性。

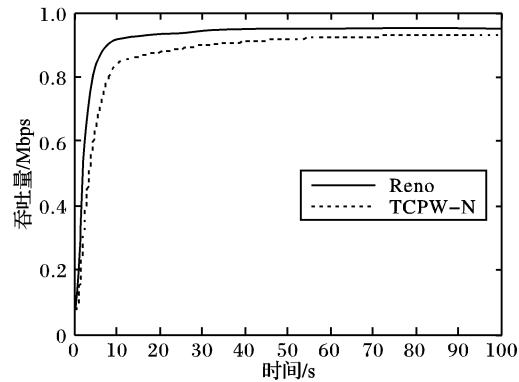


图 5 PER = 0% 时 TCPW-N 与 Reno 连接性能比较

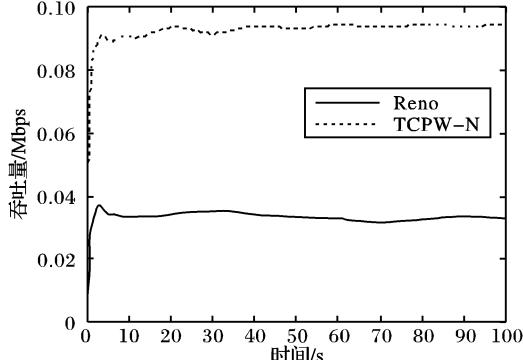


图 6 PER = 5% 时 TCPW-N 与 Reno 连接性能比较

4 结语

针对 TCP Westwood 算法应用于 GEO 卫星网络中存在的不足, 引入了丢包区分的方法, 提出了一种适用于 GEO 卫星链路的 TCP Westwood 拥塞控制改进算法。改进算法把每个阶段的窗口调整与丢包区分和带宽利用情况相结合, 合理地调整拥塞窗口, 当误码丢包时保持拥塞窗口不变或小幅调整, 当拥塞即将发生时, 窗口能及时下降到适宜的水平。仿真结果表明, 改进算法改善了 TCP Westwood 存在的不足, 具有较好的吞吐量、公平性、友好性和较低的丢包率。

参考文献:

- [1] MASCOLO S, CASETTI C, GERLA M, et al. TCP westwood: Bandwidth estimation for enhanced transport over wireless links [C]// Proceedings of the 7th annual international conference on Mobile computing and networking. New York: ACM Press, 2001: 287 - 297.
- [2] 黄蕾, 刘立祥. TCP Westwood 针对卫星网的改进方案[J]. 计算机工程, 2007, 33(8): 103 - 105.
- [3] ALLMAN M, PAXSON V, STEVENS W. TCP congestion control, RFC 2581[S]. California: The Internet Society, 1999.
- [4] KENSHIN Y, WANG REN, SANADIDI M Y, et al. TCP westwood with agile probing: Dealing with dynamic, large, leaky pipes[C]// Proceedings of the 2004 IEEE International Conference on Communications. Washington, DC: IEEE Computer Society, 2004: 1070 - 1074.

(下转第 2770 页)

从图 5~6 可知,当 ε 很小时,搜索算法尽可能要接近目标节点,而不关心从初始节点到当前行驶节点路径的代价,随着 ε 的增大,算法对降低移动物体从初始节点到当前行驶节点穿越代价的要求也随之增高,从而能够保证移动物体获得更为安全的反监控路径,但同时,不可避免地增加了搜索范围,降低了算法效率。而本算法随着 ω 的增大,启发式算法中的启发权重也就越大,此时,算法拥有较高的执行效率,但不保证能为移动物体找到最优的反监控路径;当 ω 的取值较小时,启发式算法中的启发权重也就较小,此时,算法趋向于盲目的搜索路径而导致算法的低效率。

通过数据比较,取搜索范围和路径实际优劣程度达到均衡较优的权重因子 $\varepsilon = 1$ 和 $\omega = 1$, 算法的搜索范围比 Dijkstra 路径搜索算法缩小约 40%。

为了进一步揭示在传感环境中 A* 算法的搜索区域和实际生成的反监控路径的关系,下面首先给出相关定义:

定义 6 搜索相关度:假定 A* 算法执行完成后,生成的最优路径 ψ 由 m 个三角形网格顶点 p_1, p_2, \dots, p_m 构成,即 $\psi = \{p_1, p_2, \dots, p_m\}$,而访问过的三角形网格顶点集合为 $\Phi = \{C_1, C_2, \dots, C_n\}$,则算法的搜索相关度 (Searched Related Extend, SRE) 定义为:

$$SRE = \frac{1}{n} \times \sum_{i=1}^n \frac{\gamma}{[d(p_{min}, Ci) + 1]^o} \quad (4)$$

其中 $p_{min} = \{p_m \mid d(p_m, C_i) \leq d(p_j, C_i), p_m \in \psi, \forall p_j \in \psi\}$ 。 γ, o 为技术环境常量。

为了更进一步地验证算法的优越性,下面对区域中随机部署的传感器节点个数,传感半径变化的情形下,从 p_s 到 p_d 的反监控路径长度和搜索相关度 SRE 的变化情况进行了模拟实验。为了简化起见,区域 F 仍为长 \times 宽为 600×500 的长方形区域,移动物体将从位置点 $p_s(20, 20)$ 移动到位置点 $p_d(580, 480)$,划分因子 $\rho = 500$,式(1)和(2)中的参数取值 $\lambda = 0.05, \sigma = 1$,式(4)中的 $\gamma = 1, o = 1$,基于前面的分析,搜索算法的权重因子取 $\varepsilon = 1$ 和 $\omega = 1$,且实验结果均取 50 次独立实验平均值。

图 7~8 显示了传感区域 F 内,随机分布着 60 个传感器节点,每个传感器节点的传感半径都为 30 和 100 个传感器节点,每个传感器节点的传感半径都为 20 的情况下,算法搜索反监控路径的结果。不难发现,算法仅仅搜索了很小的范围便生成了反监控路径。为了更具体地分析传感器的传感半径,传感器节点个数与搜索面积,路径长度,SRE 之间的关系,下面给出更多的实验结果和数据分析。

图 9(a)描述了传感器的节点个数和传感器的传感半径与搜索范围的关系,图 9(b)描述了传感器的节点个数和传感器的传感半径与生成的反监控路径长度的关系,图 9(c)描述了传感器的节点个数和传感器的传感半径与搜索相关度 SRE 的关系。

从图 9 可以看出,随着目标区域内分布的传感器节点个数

的增多和传感半径的扩大,算法需要访问更大面积的区域来获得较优的反监控路径,当目标区域的覆盖率很高时,法的搜索开销较大;而当目标区域内分布的传感器节点较为稀疏,或传感半径较小时,算法访问的网格顶点节点较少,其搜索相关度较高,移动物体只需要当前位置探测到的传感器节点的 40×33 分布情况,便能很好地完成局部的反监控路径的生成。显然可知,当目标区域覆盖率较低的情况下,本算法可通过适当的变化应用于通过局部信息动态获得反监控路径的情况。

4 结语

基于三角形网格划分,利用 A* 启发式路径搜索算法探索了传感环境下如何得到移动物体反监控穿行路径的问题,理论分析和实验数据表明,算法较以往的工作能够提高效率并能得到更优的反监控路径,保证了移动物体能以更安全更快速的方式通过传感区域。另外,本算法同样也具有很好的扩展性来适应分布式的应用。

参考文献:

- [1] 林亚平,王雷,陈宇,等. 传感器网络中一种分布式数据汇聚层次路由算法[J]. 电子学报, 2004, 32(11): 1801~1805.
- [2] PHIPATANASUPHORN V, RAMANATHAN P. Vulnerability of sensor networks to unauthorized traversal and monitoring[J]. IEEE Transactions on Computers, 2004, 53(3): 364~369.
- [3] 姚焯善,王雷,汤念,等. 基于正三角形剖分的传感器网络快速 k-覆盖判定算法[J]. 系统仿真学报, 2007, 19(10): 2366~2369.
- [4] MEGUERDICHIAN S, KOUSHANFAR F, QU G, et al. Exposure in wireless Ad Hoc sensor networks[C]// Proceedings of the 7th Annual International Conference on Mobile Computing and Networking. New York: ACM Press, 2001: 139~150.
- [5] 张文哲,李明禄,伍民友. 一种基于局部 Voronoi 图的目标穿越算法[J]. 软件学报 2007, 18(5): 1246~1253.
- [6] RAHMAN M M, HONG C S. Probabilistic exposure identification for wireless sensor network[C]// Proceedings of the 26th KIPS Fall Conference 2006. Cheongju, Korea: [s. n.], 2006: 1359~1362.
- [7] MEGUERDICHIAN S, KOUSHANFAR F, POTKONJAK M, et al. Coverage problems in wireless Ad Hoc sensor networks[C]// Proceedings of Annual Joint Conference of the IEEE Computer and Communications Societies: INFOCOM 2001. New York: IEEE Press, 2001: 1380~1387.
- [8] VELTRI G, HUANG Q, QU G, et al. Minimal and maximal exposure path algorithms for wireless embedded sensor networks[C]// Proceedings of the 1st International Conference On Embedded Networked Sensor Systems. New York: ACM Press, 2003: 40~50.
- [9] TIAN D, GEORGANAS N. A coverage-preserving node scheduling scheme for large wireless sensor networks[C]// Proceedings of the 1st International Workshop on Wireless Sensor Networks and Applications: WSNA 2002. Atlanta: ACM Press, 2002: 32~41.
- [10] NILSSON N J. Problem solving methods in artificial intelligence, artificial intelligence: A new synthesis [M]. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 2000.

(上接第 2766 页)

- [5] YANG GUANG, WANG REN, SANADIDI M Y, et al. TCPW with bulk repeat in next generation wireless and satellite networks[C]// Proceedings of the 2003 IEEE International Conference on Communications. Washington, DC: IEEE Computer Society, 2003: 674~678.
- [6] 彭华, 邓亚平. 基于 TCP Westwood 的一种 TCP 增强算法[J]. 计算机应用, 2006, 26(6): 221~223.
- [7] BRAKMO L S, O'MALLY S W, PETERSON L L. TCP Vegas: New techniques for congestion detection and avoidance[C]// Proceedings of the 1994 Conference on Communications Architectures, Protocols And Applications. New York: ACM Press, 1994: 24~35.
- [8] 李士宁,方济平,覃征. 一种动态门限变速增加的 TCP 慢启动算法[J]. 西安电子科技大学学报: 自然科学版, 2006, 33(4): 640~645.
- [9] KLIJAZOVICH D, GRANELLI F, MIORANDI D. TCP westwood + enhancement in high-speed long-distance networks[C]// Proceedings of the 2006 IEEE International Conference on Communications. Washington, DC: IEEE Computer Society, 2006: 710~715.
- [10] FU CHENG-PENG, 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.