LEO卫星网络中一种改进的Vegas算法

魏德宾 陶顺利 石怀峰 廖德林

(大连大学信息学院通信与网络重点实验室 大连 116622)

摘 要 针对SCPS-TP(Space Communications Protocol Standards Transport Protocol)协议的Vegas算法在LEO(Low Earth Orbit)卫星网络中吞吐量下降的问题,提出了一种自适应Vegas-AD(Adaptive)拥塞控制算法。该算法在分析Vegas的基础上,细化往返时延RTT的计算方法,使其能够更加精确地调整拥塞窗口;优化拥塞窗口的增长策略,提高了拥塞避免阶段的带宽竞争力;同时,提出基于网络拥塞程度的自适应窗口调整因子。仿真结果表明,Vegas-AD算法的带宽竞争力明显高于Vegas,并且该算法能较大幅度地提高网络吞吐量。

关键词 SCPS-TP,卫星网络,拥塞控制,Vegas

中图法分类号 TP393 **文献标识码** A

Improved Vegas Algorithm over LEO Satellite Network

WEI De-bin TAO Shun-li SHI Huai-feng LIAO De-lin

(Key Laboratory of Communication Networks, College of Information Engineering, Dalian University, Dalian 116622, China)

Abstract Aimming at throughput degradation caused by the Vegas algorithm of SCPS-TP (Space Communications Protocol Standards Transport Protocol) in LEO (Low Earth Orbit) satellite network, an adaptive congestion control algorithm named Vegas-AD (Adaptive) was proposed. The algorithm based on the analysis of the Vegas improves calculation method of RTT(Round-Trip Time) so that it can adjust the congestion window more accurately, and optimizes the growth strategy congestion window, which can improve the bandwidth competitiveness of the congestion avoidance phase. At the same time, an factor which can adaptively adjust value of the congestion window is put forward according to the degree of network congestion. The results show that the bandwidth competitiveness of Vegas-AD algorithm is significantly higher than that of Vegas, and the new algorithm can improve the network throughput greatly.

Keywords SCPS-TP, Satellite network, Congestion control, Vegas

1 引言

由于将传统传输层TCP协议应用于LEO卫星网络时会影响通信性能,降低系统吞吐量和资源利用率[1-2],因此空间数据系统咨询委员会(Consultative Committee for Space Data Systems,CCSDS)提出了SCPS-TP协议,其主要适用于具有传播时延长、间歇性连接、链路误码率高、数据处理能力和通信容量有限等特点的战场以及空间通信环境[3-4]。与传统TCP协议相比,SCPS-TP协议能够区分丢包原因,支持高误码、间歇性连接的LEO卫星网络。研究表明,SCPS-TP协议能有效地提高卫星网络的通信性能[5-6]。然而,SCPS-TP协议默认的Vegas拥塞控制算法在应用于LEO卫星网络时无法区分时延增大是由网络拓扑变化还是由网络拥塞引起的,并且在拥塞避免阶段存在带宽竞争不足的缺陷,从而导致带宽资源浪费及吞吐量降低等问题。

目前,针对Vegas算法无法区分时延增大原因并导致带宽资源浪费的问题,文献[7]利用星上最大处理时间作为拥塞窗口调整的依据,更加精确地反映了网络状态;文献[8]根据卫星间的距离计算出最小往返时延baseRTT,并据此进行拥塞控制;文献[9]通过监视RTT的剧烈变化来区分时延增大的原因。它们均能在一定程度上提高网络带宽资源利用率。针对Vegas算法在拥塞避免阶段存在带宽竞争不足并导致吞吐量过低的问题,文献[10-11]通过加快Vegas算法在拥塞避免阶段的窗口增长速度来提高算法的带宽竞争力。另外,Vegas算法在每次拥塞事件发生时,都会将门限与窗口值减半,如果在一个窗口内出现重复丢包,最终则会导致门限与窗口值以指数级别减小。LEO卫星链路具有高误码率、长传播时延等突出特性,在这段时间内不仅吞吐量非常低,而且会使恢复到拥塞之前状态的时间增加,同时也会降低进入拥塞避免窗口的起点,进而降低后续的吞吐量。

因此,针对以上问题,本文提出了一种新的拥塞控制算法——Vegas-AD。该算法在Vegas算法的基础上做了以下调整:1)分析影响RTT在LEO卫星网络中的具体因素,在计算RTT时,保留能反映当前网络拥塞状况的排队时延、传输时延和处理时延,解决了Vegas算法无法区分时延增大是由网络拓扑变化还是由网络拥塞引起的,从而导致带宽资源浪费等问题,使其能够更加精确地调整拥塞窗口,从而提高网络性能;2)根据AIMD(Additive Increase Multiplicative De-

本文受国家自然科学基金(61301151,91338104)资助。

魏德宾(1978-),男,硕士,副教授,主要研究方向为空间信息网络、系统仿真与建模,E-mail:15712387525@163.com;**陶顺利**(1989-),男,硕士生,主要研究方向为卫星网络拥塞控制技术;**石怀峰**(1989-),男,博士生,主要研究方向为空间信息网络协议技术、软件定义网络;**廖德林**(1992-),男,硕士生,主要研究方向为卫星路由算法技术。

crease)算法，即“加法增大，乘法减小”的原理，改进了拥塞避免阶段的窗口增长因子与策略，解决了 Vegas 算法竞争力不足的问题，增加了其带宽竞争力；3)改进了网络出现拥塞时窗口的乘法因子，使其能够根据网络的拥塞程度进行调整，而不是盲目地减半，从而提高了网络的吞吐量。

2　Vegas 算法分析

Vegas 算法包括慢启动阶段、拥塞避免阶段、快重传与快恢复阶段[12-13]。慢启动阶段负责拥塞窗口可靠而快速地增长；拥塞避免阶段控制拥塞窗口的稳定，避免增长过快而导致网络拥塞；快重传与快恢复负责丢包后的数据重传与拥塞窗口值的恢复，具体算法如下。

Step1　计算期望的吞吐量和实际吞吐量的差值。

$$Diff=Expected-Actual=\frac{cwnd_n}{baseRTT}-\frac{cwnd_n}{curRTT} \tag{1}$$

其中，$Expected$ 为期望的吞吐量，是理想情况下的吞吐量；$Actual$ 为实际的吞吐量；$cwnd_n$ 代表第 n 个 RTT 时的拥塞窗口；$baseRTT$ 代表当路由器缓存中没有数据包时的 RTT 值；$curRTT$ 为 RTT 的测量值。由于此时路由器中已有数据包，需要排队，因此 $curRTT>baseRTT$。

Step2　计算当前路由器中缓存的数据包个数，即吞吐量的差值与链路时延的乘积。

$$diff=Diff*baseRTT \tag{2}$$

(1)慢启动阶段：如果 $diff>\gamma$(默认值为 1)，表示窗口增长速度过快，则将拥塞窗口的大小按式(3)进行设置，并从慢启动阶段进入拥塞避免阶段。

$$cwnd_{n+1}=Actual*baseRTT \tag{3}$$

(2)拥塞避免阶段：Vegas 根据 $diff$ 进行窗口调整：

$$cwnd_{n+1}=\begin{cases}cwnd_n+1, & diff_n<\alpha\\ cwnd_n, & \alpha\leqslant diff_n\leqslant\beta\\ cwnd_n-1, & diff_n>\beta\end{cases} \tag{4}$$

其中，$diff_n$ 代表第 n 个 RTT 时估算出的路由器中属于此连接的数据包的个数。α(默认值为 2)表示在路由器中缓存的属于该连接的数据包的个数，该连接没有充分利用带宽，需要增加发送窗口；β(默认值为 4)表示在路由器中缓存的属于该连接的数据包，路由器可能遇到拥塞，应该减小发送窗口。

(3)快重传与快恢复阶段，计算出门限阈值与拥塞窗口值，并进入拥塞避免阶段。

$$\begin{cases}ssthresh=\max(2,\frac{cwnd_n}{2})\\ cwnd_{n+1}=ssthresh+3*MSS\end{cases} \tag{5}$$

其中，$ssthresh$ 为从慢启动阶段进入拥塞避免阶段的门限值，MSS 为最大报文段。

Vegas 算法应用于传统网络时具有以下优点：

(1)Vegas 通过计算期望值的吞吐量与实际吞吐量的差值来估计网络瓶颈处的可用带宽[14-15]。因此，Vegas 不依靠丢包就能预测网络拥塞，从而在丢包之前进行拥塞避免，能减少数据包的丢失，更有效地利用带宽。

(2)在拥塞避免阶段，Vegas 能根据往返时延 RTT 的变化主动地调整拥塞窗口，而基于丢包反馈的协议是一种被动式的拥塞控制机制，其依据网络中的丢包事件来进行网络拥塞判断。即便网络中的负载很高，只要没有产生拥塞丢包，协议就不会主动降低自己的发送速度。这种协议可以最大程度地利用网络剩余带宽，提高吞吐量。然而，基于丢包反馈的协议在网络近饱和状态下表现出侵略性，一方面提高了网络的带宽利用率；但另一方面，对于基于丢包反馈的拥塞控制协议来说，在提高网络利用率的同时意味着下一次拥塞丢包事件为期不远，因此这些协议在提高网络带宽利用率的同时也间接增加了网络的丢包率，加剧了整个网络的抖动性。

然而，Vegas 算法在应用于 LEO 卫星网络时存在以下问题：

(1)Vegas 算法以 RTT 为主要参数来控制发送窗口的变化，而 LEO 卫星通信网的拓扑结构是实时且高动态变化的，这会造成 RTT 的非拥塞原因增大。Vegas 算法本身并没有能力识别 RTT 的增大是由于网络拥塞造成的还是由于路径变化造成的。如果是路径的改变导致 RTT 的增加，那么 Vegas 算法也会减小发送窗口，从而浪费了网络带宽资源。

(2)在拥塞避免阶段，Vegas 规定每个连接能占用路由器的队列不超过 β，若超过则需要减速。因此，这种调整方式过于保守，它不能保证队列完全被利用[16]。另外，当与基于丢包的算法(如 Cubic)竞争链路带宽时，如果时延满足 $cwnd_n*(curRTT-baseRTT)>\beta*curRTT$，Vegas 算法的拥塞窗口则会减小，从而降低发送速率，而基于丢包的算法则不会降低其发送速率，因此 Vegas 分配到的带宽资源会逐渐地减小，从而严重地影响了其通信性能。

(3)在快重传与快恢复阶段，当链路出现拥塞时，Vegas 算法将窗口减半。如果在一个窗口内出现重复丢包，最终则会导致门限与窗口值呈指数级别减小。由于 LEO 卫星链路具有高误码率、长传播时延等突出特点，因此在这段时间内不仅吞吐量非常低，而且会使恢复到拥塞之前的状态的时间增加，同时也会降低进入拥塞避免窗口的起点，进而降低后续的吞吐量[17]。

3　Vegas-AD 算法设计

3.1　RTT 的计算

RTT 的计算公式如式(6)所示：

$$RTT=T_t+T_l+T_p+T_q \tag{6}$$

其中，发送时延 T_t(以一定的速率发送完一个报文所需的时间)和处理时延 T_p(节点进行报文存储转发处理所产生的时间)在 LEO 卫星网络中变化不大，依赖于最初设计的网络架构与性能；排队时延 T_q(报文发送前在发送队列中排队的时间)则依赖于网络的拥塞程度，与链路的利用率有关；传播时延 T_l(信号在传输通道上产生的时延)在 LEO 卫星网络中会随着网络拓扑结构的变化而变化。因此，在计算 RTT 时，将不能反映网络拥塞状况且随网络拓扑结构变化的传播时延剔除，从而使 RTT 能精确地调整拥塞窗口值。对于一个 N 跳的链路，其传播时延的分析与计算方法如下。

(1)单跳链路传播时延的分析

对于空间中任意两个节点，如果给出源节点发送数据包和目地节点收到数据包时的相对距离，就可以计算出数据包的传播时延。在 LEO 卫星网络中，传播时延主要由 3 部分组成：星地链路传播时延、轨道内星际链路传播时延和轨道间星际链路时延。

1)星地链路传播时延

如图 1 所示，设地面节点 S(源节点)在某时刻向处在位置 1 的卫星节点 D(目标节点)发送数据包，经过时间 t，卫星节点 D 在位置 2 处收到数据包，源节点发送数据包和目地节

点收到数据包时的相对距离为 d。同理，当卫星节点作为 S，地面节点作为 D 时，也可以得到相同的结果。

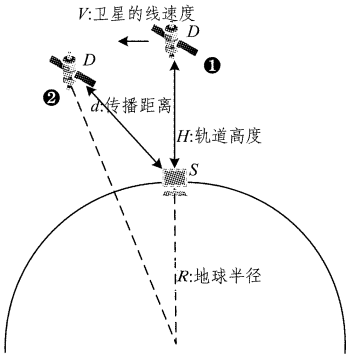

图 1

2)轨道内星际链路传播时延

如图 2 所示，对于同轨道内的两个卫星节点 S 和 D，由于它们始终保持相对静止，传播距离为 d，因此数据包在它们之间的传播时延基本不变。

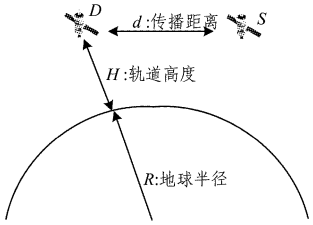

图 2

3)轨道间星际链路时延

如图 3 所示，对于不同轨道内的两个卫星节点 S 和 D，某时刻它们分别在各自轨道的位置 1 处，并且节点 S 向节点 D 发送数据。当节点 D 收到数据时，节点 S 和 D 分别处在各自轨道的位置 2 处，则数据包的传播距离为图 3 标记的 d。

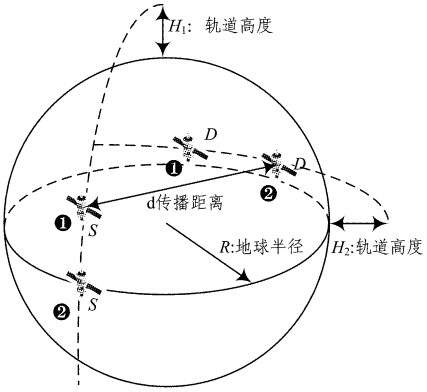

图 3

(2)单跳链路传播时延的计算

对于给定空间内任意一点 $P(A,B,H)$，可以建立如图 4 所示的数学模型。其中，A 和 B 分别为 P 点的经度和纬度，H 为 P 点距离地球表面的高度。将其换算成笛卡尔坐标系统时的坐标为 $P(x,y,z)$，具体算法如下：

$$\begin{cases} A'=\pm A,\text{西经取“}-\text{”，东经取“}+\text{”} \\ B'=90\pm B,\text{北纬取“}-\text{”，南纬取“}+\text{”} \\ x=(R+H)\cos B'\cos A' \\ y=(R+H)\cos B'\sin A' \\ z=(R+H)\sin B' \end{cases} \tag{7}$$

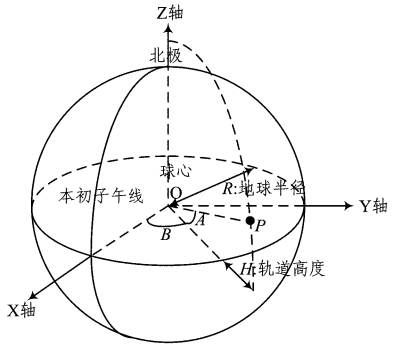

图 4

因此，设源节点发送数据包和目地节点收到数据包时的地理位置分别为 $S(A_s,B_s,H_s)$ 和 $D(A_d,B_d,H_d)$，它们对应的坐标分别为 $S(x_s,y_s,z_s)$ 和 $D(x_d,y_d,z_d)$，则单跳链路的传播时延为：

$$t_i=\frac{\sqrt{(x_d-x_s)^2+(y_d-y_s)^2+(z_d-z_s)^2}}{v} \tag{8}$$

其中，t_i 为单跳链路的传播时延，v 为电磁波在链路上的传播速度，值为 3×10^8 m/s。

(3)多跳链路 RTT 的计算

对于一个往返为 N 跳的链路，重新计算 RTT 的表达式为：

$$\begin{cases} T_l=\sum_{i=1}^{N}t_i \\ RTT=RTT-T_l \end{cases} \tag{9}$$

其中，T_l 为数据包的传播时延，t_i 为其单跳链路的传播时延。

3.2 拥塞时窗口因子的自适应调整

由于拥塞控制采用的是 AIMD 算法[18]，即“加法增大，乘法减小”的策略，当网络出现拥塞时，窗口调整方式如下：

$$\begin{cases} cwnd_{i+1}=\lambda cwnd_i,\lambda\in[0.5,0.8] \\ \lambda=\dfrac{baseRTT}{curRTT} \end{cases} \tag{10}$$

其中，$baseRTT$ 和 $curRTT$ 的意义在式(1)中已说明，此处其值不包括传播时延。λ 为网络拥塞时的“乘法因子”(传统拥塞控制算法的值为 0.5)，其表达式及取值范围证明如下。

(1)λ 表达式的证明

对于具有 N 个数据流的瓶颈链路，在拥塞事件即将发生时，其链路队列缓存和链路容量均到达了饱和[16]，那么这条链路的吞吐量可以表示为：

$$\sum_{i=1}^{n}R_i(k)^-=\sum_{i=1}^{n}\frac{cwnd_i(k)}{T_i+\dfrac{Q_{\max}}{B}} \tag{11}$$

其中，$R_i(k)^-$ 为第 i 个数据流源端的吞吐量，B 为这条瓶颈链路的带宽，$Q_{\max}$ 为缓存队列的大小，T_i 为第 i 个数据流在缓存队列为空时的往返时延。设第 i 个数据流发生拥塞时，其窗口调整因子为 λ_i，这里假设拥塞发生后链路的缓存队列在一定时间内为空，拥塞之后这条链路的吞吐量可以表示为：

$$\sum_{i=1}^{n}R_i(k)^+=\sum_{i=1}^{n}\frac{\lambda_i cwnd_i(k)}{T_i} \tag{12}$$

为了使 N 条链路的带宽资源得到充分利用，可使每条链路的 $R_i(k)^-=R_i(k)^+$，即：

$$\lambda_i=\frac{T_i}{T_i+\dfrac{Q_{\max}}{B}}=\frac{baseRTT_i}{curRTT_i} \tag{13}$$

(2)λ 取值范围的证明

设 $T(k)$为第 K 个拥塞事件到第 $K+1$ 个拥塞事件的时间(单位：s)，$\alpha_i(k)$和 λ_i 分别为第 i 个数据流的“加法因子”和“乘法因子”，$cwnd_i(k)$为第 K 个拥塞事件发生时的窗口值，在这里定义：

$$\alpha_i(k)=(cwnd_i(k+1)-\lambda_i cwnd_i(k))/T(k) \tag{14}$$

即：

$$cwnd_i(k+1)=\lambda_i cwnd_i(k)+\alpha_i(k)T(k) \tag{15}$$

对于数据流 i 和 j，当第 N 个拥塞事件发生时，由于拥塞控制采用的是 AIMD 算法[18] 且 $\lambda_i<1$，因此 $cwnd_i(n)$和 $cwnd_j(n)$必然收敛于相同的值。λ_i 越大，收敛时间越长，调节网络拥塞的效果越迟钝。当 $\lambda_i\in[0.5,0.8]$时，不仅可以保证不同的数据流在第 4 到 9 个拥塞事件发生后完全收敛，而且拥塞窗口可以根据网络的拥塞程度进行调整，而不是将窗口值盲目地减半，从而使得进入拥塞避免阶段的窗口值增加，提高了网络的吞吐量和带宽的利用率。

3.3　窗口增长策略的动态调整

当网络达到稳定时，第 K 次拥塞和第 $K+1$ 次拥塞时窗口值的峰值相差不大，因此令 $cwnd_i(k)=cwnd_i(k+1)$，由式(14)可知，$cwnd_i(k)=\frac{\alpha_i T(k)}{1-\lambda_i}$。对于不同的数据流 i 和 j，有 $\frac{cwnd_i(k)}{cwnd_j(k)}=\frac{\alpha_i(1-\lambda_j)}{\alpha_j(1-\lambda_i)}$，为了实现不同流之间的公平性，必须保证 $\frac{\alpha_i}{1-\lambda_i}$为常量[18](传统算法中 λ_i 为 1/2，α_i 为 1)；同时，为了增加该算法的带宽竞争力，将窗口的调整策略设置为：

$$cwnd_{n+1}=\begin{cases}cwnd_n+2-\lambda, & diff_n<\alpha\\ cwnd_n+4-5\lambda, & \alpha\leqslant diff_n\leqslant\beta\\ cwnd_n, & diff_n>\beta\end{cases} \tag{16}$$

其中，设 3 个阶段所经历的时间均为$\frac{1}{3}T(k)$，即满足$\frac{\alpha_i T(k)}{1-\lambda_i}=\frac{(2-\lambda_i+4-5\lambda_i)}{1-\lambda_i}*\frac{1}{3}T(k)$。从上式的调整策略可以看出，当路由器中缓存的且属于该连接的数据包的个数 $diff_n$ 在[α，β]内时，窗口的值不再保持不变，而是根据 λ 值动态地增加；而当路由器中缓存的且属于该连接的数据包的个数 $diff_n>\beta$时，窗口的值保持不变。因此，该算法保持了不同流之间的公平性，同时也提高了带宽竞争力和网络的吞量。

3.4　算法流程

Vegas-AD 算法主要包括 4 个阶段：慢启动阶段、拥塞避免阶段、快重传与快恢复阶段。另外，SCPS-TP 能够利用网络层 SCPSNP(SCPS network protocol)[19]协议来区分网络丢包的原因，这也是该协议应用于卫星网络时优于传统 TCP 协议的主要因素之一。因此，SCPS-TP 拥塞控制算法的流程也略有不同，具体过程如图 5 所示。

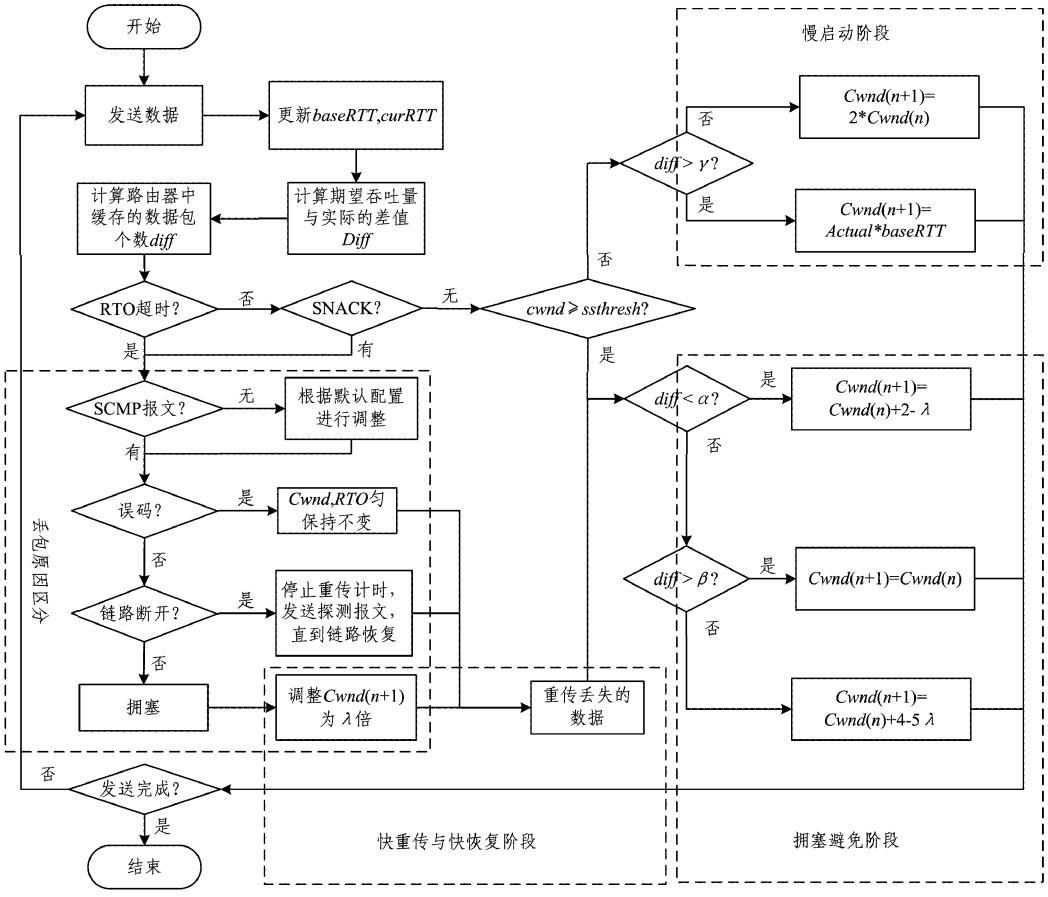

图 5　Vegas-AD 算法流程图

4　仿真分析

4.1　仿真实验场景及步骤

本文采用 NS-2 网络仿真软件(版本为 2.35)来验证 Vegas-AD 算法的性能，网络拓扑结构如图 6 所示。图 6 中，LEO 卫星网络采用具有全球覆盖能务的 Iridium 极道卫星星座，相关参数如表 1 所列。其详细的节点参数及拓扑结构配置文件为 ns-2.35/tcl/ex 目录下的 sat-iridium-nodes.tcl 与

sat-iridium-links.tcl。选择Berkeley(37.9,−122.3)和 Boston(42.3,−71.1)作为两个地面站,每个地面站都连接 N 个终端,与 Berkeley 地面站连接的终端设为源结点,与 Boston 地面站连接的终端设为目的结点。两地面站通过 LEO 卫星星座连接在一起,从而构成一条瓶颈链路。

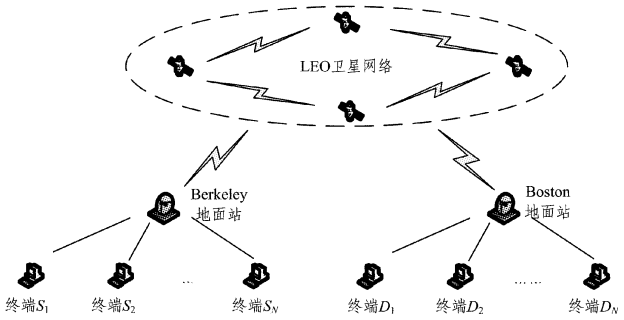

图 6 网络仿真拓扑图

表 1 Iridium 卫星星座参数

仿真参数	参数取值
轨道高度/km	780
轨道数	6
轨道平面卫星数	11
轨道倾角/°	86.4
轨道平面间隔/°	31.6
星际链路纬度阈值/°	60
每个卫星星际链路数	4
卫星链路误码	10^{-5}
缝间隔/°	22
截止高度角/°	8.2
星际链路带宽/(Mb/s)	25
上/下行带宽/(Mb/s)	1.5
卫星链路队列长度/packets	50
越缝星际链路	否

实验步骤如下:

Step1 下载 NS-2 及 SCPS 开源源码,编写算法模块并在 Linux 操作系统下编译安装。

Step2 创建相应的 tcl 脚本文件,设置节点参数及拓扑结构。

Step3 执行 tcl 脚本文件,生成相应的 trace 文件。

Step4 编写 awk 脚本,对 trace 数据文件进行统计与分析,得到仿真结果。

4.2 仿真结果分析

(1)单个源终端的情况

根据上述拓扑结构和配置参数,设置 N 为 1,即开启一条通信链路,分别使用 Vegas 与 Vegas-AD 拥塞控制算法进行仿真,得到的仿真结果如图 7—图 9 所示。

图 7、图 8 给出了终端与地面站的链路带宽为 10Mb/s 时,不同算法拥塞窗口的变化情况。其中 Vegas 算法拥塞窗口的平均值与标准方差分别为 10.854 和 1.835,而 Vegas-AD 算法的值分别为 15.782 和 1.121。可以看出,Vegas-AD 算法的拥塞窗口的平均值较大,变化较为稳定,波动较小。其主要原因在于,Vegas-AD 算法屏蔽了卫星拓扑结构改变带来的影响,在计算往返时延时删除了不能反映网络拥塞状况且随网络拓扑结构变化的传播时延,从而使往返时延更加稳定,精确地调整了拥塞窗口。而传统的 Vegas 算法在计算往返时延时将传播时延作为调整窗口值的依据,因此卫星网络拓扑变化会对 Vegas 算法性能造成较大的影响。在网络拓扑结构变化而非网络拥塞导致 RTT 增大时,由于 Vegas 算法未能正确区分 RTT 增大的原因,导致窗口值错误地减小,浪费了大量卫星资源。另外,在网络出现拥塞时,Vegas-AD 算法会根据网络的拥塞程度进行相应调整,而不是将窗口值盲目地减半,提高了进入拥塞阶段的窗口值和网络的吞吐量。图 9 给出了两种不同算法的吞吐量随时间变化的关系。在此仿真实例下,Vegas-AD 算法比 Vegas 算法的吞吐量提高了 11.7%。

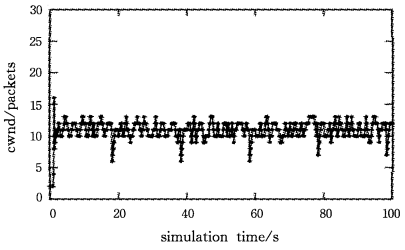

图 7 Vegas 算法的拥塞窗口值

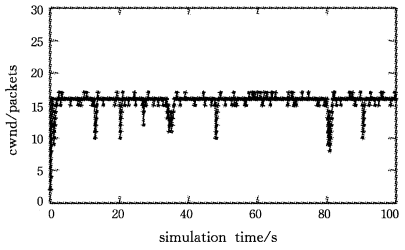

图 8 Vegas-AD 算法的拥塞窗口值

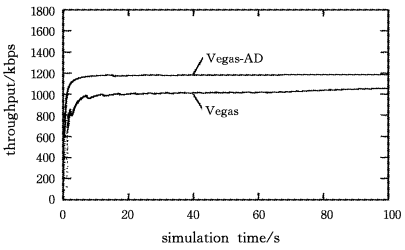

图 9 不同算法的吞吐量

图 10 给出了终端与地面站在不同链路带宽时,Vegas-AD 算法与 Vegas 算法的平均吞吐量。它们的平均吞吐量会随链路带宽的增加而增加,但由于星地链路上/下行的限制,其最终都会处于稳定状态。与 Vegas 算法相比,Vegas-AD 算法在链路带宽为 20Mb/s 时就已达到稳定,并且在链路带宽小于 30Mb/s 时优势较为明显。

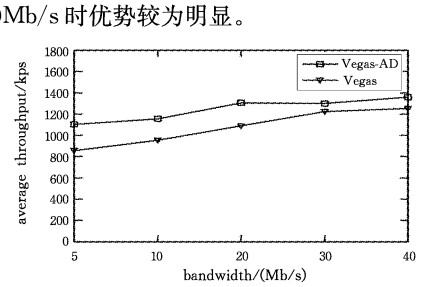

图 10 不同链路带宽的平均吞吐量

(2)多个源终端的情况

为了验证 Vegas-AD 算法的带宽竞争力,设置 N 为 3,即开启 3 条通信链路,分别使用 Cubic,Vegas 以及 Vegas-AD 拥塞控制算法得到的仿真结果如图 11 所示。

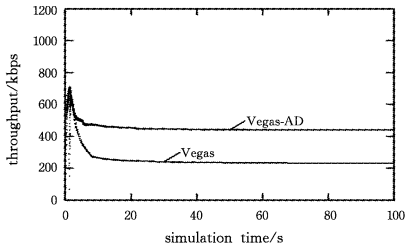

图 11　不同算法的吞吐量

图 11 给出的是终端与地面站的链路带宽为 10Mb/s 时不同算法的吞吐量随时间的变化关系。可以看出，Vegas-AD 算法的吞吐量明显高于 Vegas 算法，主要原因在于在刚开始时算法都处于慢启动阶段，窗口的增长速度比较快，因此它们的吞吐量都维持在较高的水平，而当它们吞吐量的和趋于星地上/下行链路带宽时，算法会根据自身情况进入拥塞避免阶段。在拥塞避免阶段，Vegas 规定每个连接能占用路由器的队列不超过 β，若超过则会减小拥塞窗口值。因此，这种调整方式过于保守，不能充分利用带宽资源。另外，如果时延增大，Vegas 算法的拥塞窗口会减小，从而降低发送速率；Cubic 算法则不会降低其发送速率，只要没有产生拥塞丢包，就不会主动降低自己的发送速率；Vegas-AD 算法则会根据 λ 值来调整拥塞窗口，最终 Vegas 分配到的带宽资源会逐渐地减少，而 Vegas-AD 算法则会保持带宽竞争力，其吞吐量明显高于 Vegas 算法。

图 12 给出了终端与地面站具有不同链路带宽时 3 种不同算法的平均吞吐量与带宽的关系。不难看出，随着链路带宽的增加，相比于 Vegas 算法，Vegas-AD 算法在带宽竞争力方面的优势更加明显。

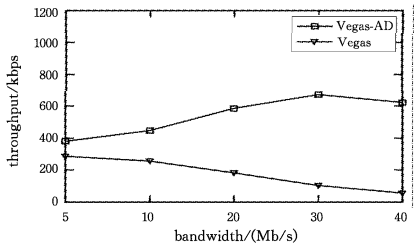

图 12　不同链路带宽的平均吞吐量

结束语　在 Vega 算法基础上，Vegas-AD 算法做了 3 方面的改善：优化了 RTT 的计算方法；改善了算法在拥塞避免阶段的增长策略；改进了网络出现拥塞时窗口的乘法因子。Vegas-AD 算法在计算 RTT 时屏蔽了 LEO 卫星拓扑结构改变带来的影响，将不能反映网络拥塞状况且随网络拓扑结构变化的传播时延剔除，从而使往返时延能精确地调整拥塞窗口值；优化了拥塞避免的窗口增长策略，提升了其带宽竞争力；提出了网络出现拥塞时窗口的自适应调整因子，使其能够根据网络的拥塞程度进行自适应调整，避免了将窗口值盲目地减半，扩大了进入拥塞阶段的窗口值。仿真结果表明，与 Vegas 算法相比，Vegas-AD 算法在 LEO 卫星网络中具有较高的吞吐量和较强的带宽竞争力。

参 考 文 献

[1] 戴帅. 肖楠. 梁俊. SCPS-TP 在卫星网络中的传输控制算法研究[J]. 现代防御技术，2013(6)：83-87.

[2] SARKAR M，SHUKLA K K，DASGUPTA K S. A Survey of Transport Protocols for Deep Space Communication Networks [J]. IEEE/International Journal of Computer Applications，2011，12(4)：58-67.

[3] The Consultative Committee for Space Data Systems. Space Communications Protocol Specification (SCPS)—Transport Protocol (SCPS-TP)：CCSDS 714.0-R-3[S]. Washington，DC：CCSDS，1997.

[4] The Consultative Committee for Space Data Systems. Space Communication Protocol Specification-Transport Protocol(SCPS-TP)：CCSDS 714.0-B-2[S]. Washington，DC：CCSDS，2006.

[5] 安建平，靳松，许军，等. 深空通信网络协议的发展与展望[J]. 通信学报，2016，37(7)：50-61.

[6] 廖勇. 统一信息网空间数据通信传输协议研究[D]. 重庆：重庆大学，2014.

[7] 戴帅，肖楠，梁俊，等. 基于处理时延的卫星网络 TCP 拥塞控制算法[J]. 现代防御技术，2014(3)：127-134.

[8] 杨力，李静森，魏德宾，等. 一种高动态卫星网络的拥塞控制算法[J]. 宇航学报，2014，35(8)：953-960.

[9] ZHOU K，YU Q，ZHU Z. Dynamic Vegas：A Competitive Congestion Control Strategy[C]// Proceedings of International Conference on Computer Science and Information Technology. Springer India，2014：333-340.

[10] 王斌，陈元琰，胡愚. TCP Vegas 拥塞避免机制的改进算法[J]. 计算机应用，2010，30(9)：2486-2500.

[11] CHENG R S，DENG D J. Congestion control with dynamic threshold adaptation and cross-layer response for TCP Vegas over IEEE 802.11 wireless networks[J]. International Journal of Communication Systems，2014，27(11)：2918-2930.

[12] ALL MAN M，PAXSON V，STEVENS R. TCP Congestion Control：RFC 2581[R]. 1999.

[13] BRAKMO L S，PETERSON L L. TCP Vegas：End to End Congestion Avoidance on a Global Internet[J]. IEEE Journal on Selected Areas in Communications，2002，13(8)：1465-1480.

[14] YANG L，WANG K，WEI D. Congestion control algorithm based on bandwidth estimation over satellite network[J]. ICIC Express Letters，2015，9(7)：2003-2008.

[15] AUTIN F，FREYERMUTH J M，VON SACHS R. Block-thresholded-adapted estimators via a maxisetapproach[J]. Scandinavian Journal of Statistics，2013，41(1)：240-258.

[16] LIN X，CHENG C，KUMARY S，et al. Impact of Queue Management Schemes and TCP Variants on the Performance of 10Gbps High Speed Networks：An Experimental Study[J]. Journal of Networks，2014，9(5)：1183-1192.

[17] SAKAR M，SHUKLA K K，DASGUPTA K S. Network State Classification Based on the Statistical Properties of RTT for an Adaptive Multi-State Proactive Transport Protocol for Satellite Based Networks[J]. International Journal of Computer Networks & Communications，2010，19(4)：88-105.

[18] ESMAEELZADEH V，BERANGI R，HOSSEINI E S，et al. Stochastic backlog and delay bounds of generic rate-based AIMD congestion control scheme in cognitive radio sensor networks[J]. Pervasive & Mobile Computing，2015，4(C)：46-57.

[19] The Consultative Committee for Space Data Systems. Space Communications Protocol Specification (SCPS)—Network Protocol (SCPS-NP)：CCSDS 713.0-R-3[S]. Washington，DC：CCSDS，1997.