

基于跳到跳信息的卫星网络传输控制协议研究

王路^{1,2}, 胡月梅¹, 刘立祥¹, 胡晓惠¹

(1. 中国科学院 软件研究所, 北京 100190; 2. 中国科学院 研究生院, 北京 100049)

摘要: 设计了一种适用于卫星网络的传输控制协议 TPSN。该协议使用异步跳到跳确认, 快速恢复成段丢失数据, 并采用基于检测窗口的端到端选择性否定应答(SNACK)机制, 减少协议控制信息, 保证数据可靠传输。在跳到跳可靠性保证机制基础上传输网络负载, 实现在避免网络拥塞条件下高效利用网络带宽资源, 并保证具有不同端到端往返时延的异种数据流之间的公平性。仿真结果表明 TPSN 能够在长延迟、误码率高、链路频繁切换的卫星网络中保证高效的带宽利用、可靠的数据传输以及各数据流之间的公平。

关键词: 卫星网络; 长延迟; 高误码率; 动态拓扑

中图分类号: TP393

文献标识码: A

文章编号: 1000-436X(2012)06-0091-12

Transmission control protocol based on hop-by-hop for satellite networks

WANG Lu^{1,2}, HU Yue-mei¹, LIU Li-xiang¹, HU Xiao-hui¹

(1. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;

2. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: A transmission control protocol named TPSN (transmission control protocol for satellite networks) was proposed, which applied asynchronous hop-by-hop ACK, and SNACK based on checking window size to reduce overhead of control message and achieve reliable transmission. In addition, TPSN transmitted load information on the base of hop-by-hop scheme to maintain high network resources utilization, fairness among different data flows with heterogeneous RTT and congestion free. The good performance of TPSN, in terms of link utilization, transmission effectiveness and fairness, was verified via a simulated satellite network with high link error rate, large delay and intermittent links.

Key words: satellite networks; large delay; high link error rate; dynamic topology

1 引言

卫星网络^[1]面临着长延迟、动态拓扑、链路误码率高等挑战。早在 1976 年, Gitman^[2]发现在地面无线网络中使用跳到跳重传方式能够比端到端重传方式获得更短的传输延迟和更高的带宽利用率。近些年, 出现了许多利用跳到跳机制保证无线传感器网络数据可靠性的传输协议, 如 PSFQ^[3]、

OSDRMP^[4]等。PSFQ (pump slowly, fetch quickly)协议中间节点参与数据分组的检测和恢复, 通过跳到跳的方式保证数据可靠交换。OSDRMP 协议使用延迟的前向乱序重传请求恢复丢失数据。文献[5]对 PSFQ 进行改进, 扩大传感器节点缓存空间, 使传感器节点能够同时传输多个数据分组。文献[6]提出一种跳到跳传输控制协议体系, 通过跳到跳确认信息保证数据可靠传输。DTSN^[7]引入 ACK 和

收稿日期: 2010-07-02; 修回日期: 2010-11-30

基金项目: 综合信息系统组网与传输基金资助项目

Foundation Item: The Foundation of the Networking and Transport Scheme in Integrated Information System

NACK 机制, 保证数据可靠传输, 并将数据缓存在中间节点, 每个存储丢失数据的节点均能够重传恢复丢失数据, 提高丢失数据的恢复效率。

目前, 互联网使用的传统传输控制协议 TCP^[8, 9]认为数据丢失即表示发生拥塞, 需要减小发送窗口, 这种拥塞检测机制将链路误码造成的数据丢失错误地解释为网络发生拥塞, 所以不适用于链路误码率较高的无线传输环境, 并且 TCP 采用的拥塞避免、快速恢复等拥塞窗口调节方式在高带宽网络环境下很难达到高利用率。为解决 TCP 拥塞控制算法遇到的问题, 已经提出许多拥塞控制方法, 如 AQM/ECN^[10]、XCP^[11]、VCP^[12]、MLCP^[13]、BMCC^[14]等。这些拥塞控制机制由网络中间节点将网络负载情况通过端到端确认信息传递给发送端, 发送端根据接收信息调整发送速率。AQM/ECN 通过在 IP 分组头中加入 2bit, 用于标识网络状态, 成功地降低了网络分组丢失率及队列长度, 但是窗口调节等方面与传统 TCP 几乎相同, 在高带宽情况下很难实现高带宽利用率。为能够达到带宽的高利用率, XCP 提出每个路由节点计算自身平均流速率并将该速率发送给发送端, 发送端根据网络中节点反馈信息调节发送速率。这种方式将拥塞控制交给整个网络进行, 发送端接收到精确的网络内部信息后, 调整数据发送速率达到很好的网络性能。但是 XCP 需要在数据中加入较多比特传递中间节点信息, 增加了控制信息对网络资源的占用。VCP 通过分析 AQM/ECN 和 XCP, 提出中间节点计算负载并对负载分级, 在 ECN 原有 2bit 的基础上增加 1bit 传递负载信息。还有一些研究^[15, 16]将跳到跳思想引入拥塞控制, 相邻节点通过交换存储空间、路由表等信息, 降低网络状态反馈时延, 邻居节点快速调整发送速率, 避免网络拥塞, 减少数据丢失量。

为满足卫星网络需求, 空间数据系统咨询委员会 (CCSDS, consultative committee for space data systems) 制定了针对星际网络的传输层协议标准 SCPS-TP 协议^[17]。SCPS-TP 协议针对星间链路特点对数据丢失原因进行分类, 并引入 TCP Vegas^[18]拥塞控制机制及 SNACK 确认机制。

拓扑不断变化的卫星网络往返时延抖动较大, 导致无法准确测量往返时延。传统的超时重传依赖于往返时延测量的准确, 造成不必要的数据重传。快速重传方式主要针对单个数据分组丢失设计, 无

法有效恢复成段丢失数据, 同时快速重传方式也不能适用于长延迟网络, 原因是源端可能需要等待很长时间才能收到 3 个重复的 ACK。所以传统的数据重传方式不适用于卫星网络。快速、高效的跳到跳反馈有效地减少了高误码率带来的负面影响, 非常适用于卫星网络环境。但是目前跳到跳可靠传输机制主要针对地面无线网络设计, 使用较为单一确认机制, 并没有考虑卫星网络长延迟、动态拓扑导致数据成段丢失, 以及由于卫星移动、链路失效等原因造成数据无法通过跳到跳重传恢复的情况。拥塞控制方面, 显示负载反馈拥塞控制算法能够有效提高网络带宽资源利用率, 但是这些机制均是建立在传统的端到端可靠控制基础上, 这种端到端拥塞控制机制在长延迟、动态拓扑、误码率较高的卫星网络中, 由于端到端之间的长延迟导致反馈到源端的信息可能已经过期, 无法反应网络当前真实情况; 而且在卫星链路高误码率情况下, 反馈信息可能在传输过程中已经丢失。传统的端到端拥塞控制机制产生的另外一个问题是由于端到端往返延迟的不同, 造成各个数据流分配网络带宽资源的不公平。目前, 采用跳到跳机制的拥塞控制算法, 能够加快网络信息的反馈, 但是主要为地面无线网络设计, 节点负载信息在相邻节点间传输, 相邻节点能够快速调节发送速率, 但是源端对网络中邻居节点间的反馈信息反应较为迟钝, 不适用于长延迟、拓扑动态变化的卫星网络。为解决上述问题, 本文提出一种基于跳到跳信息的适用于卫星网络的传输控制协议 TPSN(transmission control protocol for satellite networks)。

为增强数据传输的可靠性, 提高发送端对网络状态反应灵敏度, TPSN 协议使用跳到跳异步确认机制, 快速恢复成段丢失数据。在此基础上, 使用端到端的基于检测窗口的选择性否定确认机制 (SNACK), 重传通过跳到跳机制无法恢复的数据, 保证卫星网络中数据传输的高可靠性。为了能够避免网络拥塞, 提高网络资源利用率, 保证不同数据流间的公平性, TPSN 协议周期性地计算卫星链路负载情况, 并在跳到跳基础上, 采用一种快速、灵活的传递方式, 将负载状态信息传递至源端, 源端根据负载情况调节发送速率。

2 卫星网络传输控制协议 TPSN

协议处理流程如图 1 所示。源端 A 向目的 A'

发送数据，中间卫星 N_i 接收到数据后发送跳到跳确认消息，卫星 N_{i-1} 接收到跳到跳确认信息后重传丢失数据；负载信息借助于跳到跳机制反馈到源端 A，源端 A 接收到网络负载信息后调整数据发送速率；目的端 A 接收到数据后按照协议规定发送端到端 SNACK 信息，源端 A 接收到端到端 SNACK 信息后调整滑动窗口，并重传丢失数据。TPSN 协议主要工作在传输层和网络层，其中端到端可靠性保证机制工作在传输层，跳到跳可靠性保证机制工作在网络层，拥塞控制机制工作在传输层和网络层，2 层借助于跨层思想交互有用信息，共同保证数据可靠传输、避免网络拥塞，提高网络资源利用率。

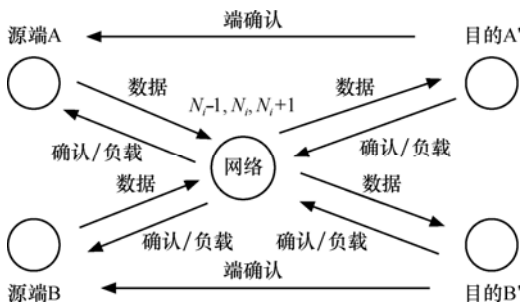


图 1 协议处理原理

2.1 数据可靠性确认机制

2.1.1 跳到跳异步确认重传机制

卫星网络中，链路的高误码率、链路的频繁切换经常导致数据丢失，而且这种网络环境下，链路错误会不断累积。假设数据分组长度为 L byte，误码率为 p ，数据分组通过一跳成功传输的可能性为 $P_{OH} = (1 - p)^{8L}$ ，数据分组通过 n 跳成功传输的可能性迅速减少为 $(1 - P_{OH})^n$ ，所以随着数据传输路径经过卫星跳数增加即传输延迟加长，数据成功到达率呈指数衰减。跳到跳确认方式，能够快速重传丢失数据，避免错误累积。同时中间卫星参与可靠性保障，使数据分组的恢复重传不需要等待 2 个端节点间的往返时间，中间卫星即可快速恢复重传丢失数据。定义 N 为数据传输链路上卫星个数。假设一次重传即能够恢复丢失数据，使用端到端数据确认重传机制，丢失数据成功传输时间 $T_{REtoE} = 3 \sum_{i=1}^N T_i$ ，当中间卫星参与数据重传，丢失数据成功传输时间 $T_{RHtoH} = \sum_{i=1}^N T_i + 2 \sum_{j=1}^M T_j$ ，其中， M 为数据传输过程中数据丢失次数， T_i 是数据跳到跳传输时延， T_j 是丢

失数据跳到跳传输延迟。由于 $M < N$ ，所以 $T_{RHtoH} < T_{REtoE}$ 。数据确认与重传过程，如图 2 所示。综上所述可以得到的结论是跳到跳机制非常适用于误码率高、链路频繁切换、端到端延迟长的卫星网络。

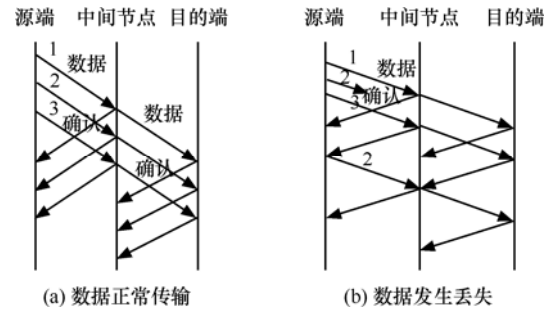


图 2 跳到跳数据确认

为能够快速恢复成段丢失数据，减少跳到跳确认对网络资源占用，确认信息采用异步跳到跳累积确认方式。累积确认能够防止确认信息丢失，减少确认信息对网络带宽占用。为能够恢复成段丢失数据，确认信息包括标识数据接收情况的窗口位图。窗口位图由一系列比特组成，每个比特代表一个数据分组接收情况，当比特设置为 1 代表数据已正确接收，比特设置为 0 代表数据未能正确接收。由于采用跳到跳确认，丢失数据恢复频率快，窗口位图取值可以较小，此处窗口位图设置为 8bit。单跳卫星间维护本地数据编号。确认信息包括卫星地址、本地数据编号及 8bit 窗口位图。本地数据编号与 8bit 窗口位图组合表明本地数据编号之前所有数据分组，包括数据本身及本地数据编号之后 8 个数据分组的接收情况。如(A, 18, 01101111)，表示卫星 A 已正确接收本地编号为 18 之前所有数据分组，包括 18 号数据分组，以及 20、21、23、24、25、26 号数据分组，本地编号为 19、22 的数据丢失。这种确认方式适用于高误码率情况下数据成段丢失情况，而且每个确认信息与传统累积确认信息相比仅增加 8bit，网络负载较小。接收卫星接收到数据后，填充窗口位图，发送确认信息。跳到跳确认信息主要作为网络中其他数据的附加信息传输，如路由信息、网络管理信息等。虽然中间卫星在接收到跳到跳确认消息前需要将数据缓存在存储介质中，但由于单跳往返时延(RTT)较小，收到确认信息后缓存数据能够立即被释放，因此数据发送队列不会由于需要等待跳到跳确认信息而不断增长。

跳到跳异步确认重传机制采用 2 种数据重传方式。第一种方式与传统超时重传相似，虽然传统超时重传方式不适用于卫星网络中端到端的超时重传，但是由于相邻卫星之间链路状态较为单一，所以可以用于跳到跳的数据重传控制。卫星通过跳到跳往返时延设置数据重传时钟，卫星重传超时数据。这种数据重传方式数据恢复时间较长，针对这个问题，TPSN 协议提出第二种数据重传方式，由于 2 个相邻卫星间链路状态较为单一，且跳到跳确认采用异步方式发送，TPSN 协议认为接收到包含乱序信息的跳到跳确认信息即表明有数据分组丢失，当发送卫星第一次接收到数据乱序信息后，立即重传数据，当第二次或者更多次收到相同数据乱序信息时，为避免不必要的的数据重传，将根据链路分组丢失率判定是否重传数据。

假设 $O(t)$ 是发送卫星时刻 t 的数据发送速率，不包括重传数据； $I(t+d)$ 是接收卫星时刻 $t+d$ 的数据接收速率，不包括重复数据； d 是数据单向传输时延， $d = RTT/2$ ； p_{avg} 是数据分组平均长度； δ_p 是时间间隔。为能够恢复丢失数据分组，并且避免重复重传数据，数据重传时间间隔 ΔT_R 应满足如下公式：

$$\int_{t-\delta_p}^t (O(t) - I(t+d)) dt \leq \frac{\delta_p}{\Delta T_R} p_{avg} \quad (1)$$

所以，
$$\Delta T_R \leq \frac{p_{avg} \delta_p}{\int_{t-\delta_p}^t (O(t) - I(t+d)) dt}$$
，假设单

位时间内发送卫星发送数据分组平均个数为 N ，链路分组丢失率为 p_{lost_rate} ，则 $\Delta T_R \leq \frac{1}{N p_{lost_rate}}$ 。为保

证数据可靠，每发送 $\frac{1}{p_{lost_rate}}$ 个数据分组就应该对数据进行重传。

根据上述分析，发送方第二次或者更多次接收到同一数据乱序信息时，如果 $\Delta N \geq \frac{1}{p_{lost_rate}}$ ，则重传乱序数据，其中， ΔN 为上次重传开始到目前为止发送端发送数据分组数量，不包括重传数据。

2.1.2 基于检测窗口的选择否定应答

某些时间段内，对于拓扑结构不断变化的卫星网络，2 个端节点之间可能并不存在传输路径，此时中间卫星如果发生故障，数据可能无法通过中间

卫星进行恢复。为保证数据传输的可靠性，引入端到端确认机制，并根据链路切换及高误码率特性，端节点确认信息采用选择否定确认(SNACK)信息，告知源端目前已经接收到数据信息以及需要重传数据信息，如图 3 所示。SNACK 采用累积确认方式，确认信息长度为 64bit，第一个 32bit 表示接收方希望接收到下一段数据编号 N_{rcv_nxt} ，第二个 32bit 是 $N_{rcv_nxt} - 1$ 开始向前 32 个数据段接收情况，格式如图 4 所示。每个 SNACK 数据每次可以确认 32 段 ($N_{rcv_nxt} - 32$ 到 $N_{rcv_nxt} - 1$) 数据接收情况。如(58, 252.15.192.240)，此处后 32bit 使用类似于 IP 每 8bit 计数方式，能够表示 26 号到 57 号数据分组接收情况，其中有 3 个缺口，长度分别为 6、6 和 4，与 26 号数据偏移量分别为 6、18 和 28。

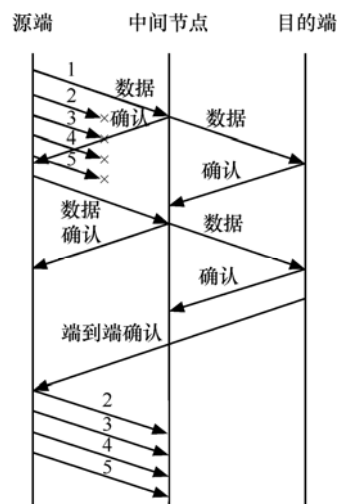


图 3 端到端数据确认

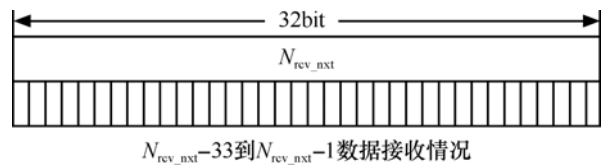


图 4 SNACK 格式

由于卫星网络端到端的长延迟以及动态拓扑导致较大端到端延迟抖动，数据往返时延(RTT)无法精确计算，所以源端不再设置超时重传时钟，端到端的数据重传由 SNACK 控制完成。为保证源端及时重传恢复跳到跳无法恢复的丢失数据，减少端确认信息对反向链路资源的占用，适应带宽不对称链路特性，目的端在 2 种情况下发送端节点间确认信息：1) 发现数据分组乱序；2) 周期性发送端节点确认信息，目的是更新滑动窗口，并释放已正确

接收数据占用的资源。

方式 1 如果目的端每收到一个数据分组即检测是否乱序，并发送确认信息，由于网络数据突发特性，即短时间内发送大量数据，一旦发生数据丢失，会在短时间内产生大量端节点间确认信息，这些确认信息可能指向同一范围内的丢失数据，结果不仅产生大量冗余确认信息，而且发送端会不断重传未确认数据。为减小卫星负担，卫星并不是每收到一个数据分组即检测数据是否乱序。TPSN 设置数据乱序检测窗口 $w_{\text{chk_wnd}}$ ，只有接收数据量达到窗口值时，接收卫星才检测数据是否乱序。采用固定窗口值的方式能够缓解每个数据分组都检测引起的压力，但是并不能解决由于数据分组丢失造成的重复确认信息及冗余数据分组重传。因此 TPSN 采用动态窗口值，变化方式采用乘性增、乘性减。根据 2.1.1 节分析，当链路分组丢失率为 $p_{\text{lost_rate}}$ 情况

下，发送端平均每发送 $\frac{1}{p_{\text{lost_rate}}}$ 个数据分组即重传数据才可能保证数据可靠传输，所以检测窗口 $w_{\text{chk_wnd}}$ 初始值设置为 $\frac{1}{p_{\text{lost_rate}}} p_{\text{avg}}$ 。当检测到数据乱序时，

为防止多次传输同一乱序数据信息，检测窗口值翻倍，减慢数据检测速率。当数据顺序到达，则检测窗口值 $w_{\text{chk_wnd}}$ 乘性减为原先值的 0.9 倍，并保证不低于初始值，加快数据检测速率。发送端重传数据时间间隔同 2.1.1 节。

上述过程中，由于卫星网络动态拓扑特性，目的端需要正确判断乱序是由于数据分组丢失，而不是路径改变引起。目的端通过数据分组的生存时间 TTL 字段判断数据分组乱序原因是数据分组丢失，而不是路径改变。目的端接收顺序到达数据后读取分组头 TTL 字段，并存储在 $TTL_{\text{in-order}}$ 中。如果后续接收到乱序数据的 TTL 字段等于或者小于 $TTL_{\text{in-order}}$ ：

$$TTL \leq TTL_{\text{in-order}} \quad (2)$$

目的端认为接收乱序数据的原因是传输路径发生改变。如果接收到的 TTL 不满足式(2)，即数据经过更长路径传输，则目的端认为数据丢失^[19]。

方式 2 当数据一直顺序接收时，按照方式 1 不会发送端节点间确认信息，这时候需要周期性发送端节点间确认信息，目的是及时通报源端数据接收情况，更新滑动窗口，并释放已经正确接收数据占用的缓存。为及时调节滑动窗口，发送时间间隔

由源端与目的端间的往返时间 RTT 和链路分组丢失率决定。假设 p_{avg} 为平均数据分组长度， $p_{\text{lost_rate}}$ 为链路分组丢失率， R_{rate} 为接收方平均接收数据速率，不包括重复数据，则确认信息发送时间间隔为

$$T_{\text{interval}} = \min \left(\frac{RTT}{2}, \frac{p_{\text{avg}}}{p_{\text{lost_rate}} R_{\text{rate}}} \right)$$

2.2 基于跳到跳机制的拥塞控制机制

2.2.1 链路负载计算

近期一些研究^[12,13]中利用负载因子改进拥塞控制算法，但是只考虑了单路径传输情况，即一个节点只有一个下一跳，负载计算不适用于节点存在多条路径情况。TPSN 对节点队列进行管理，发送到不同下一跳节点的数据存放在不同链路队列中，节点负载计算针对链路进行，每条链路都有对应的负载。如果负载计算针对节点进行，可能由于节点各链路带宽不一致导致节点负载判断失效。如图 5 所示，路径 1 带宽为 30Mbit/s 和 20Mbit/s，路径 2 带宽为 200Mbit/s，通过路径 1 及路径 2 数据流量均为 30Mbit/s，容易判断路径 1 负载重，路径 2 负载轻。如果按照文献^[12, 13]中的计算方法计算节点负载，则无法作出正确判断。

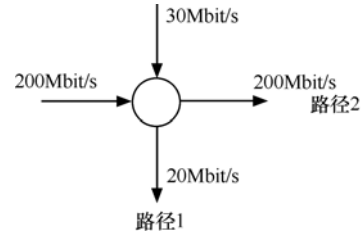


图 5 负载计算

中间节点每隔时间 t_p 计算节点链路负载 ρ_l 。参考文献^[12, 13]， ρ_l 计算公式为

$$\rho_l = \frac{\lambda_l + k_q \bar{q}_l}{\gamma_l C_l t_p} \quad (3)$$

其中，与文献^[12, 13]对 λ_l 和 \bar{q}_l 定义不同，此处 λ_l 指 t_p 时间内进入节点从该链路传输的数据总量； \bar{q}_l 是在这段时间内该链路的队列长度平均值，以字节为单位； k_q 是系数； γ_l 是链路的目标使用率，取值接近于 1； C_l 是链路数据发送能力^[12]。为了能够更加充分地利用节点存储空间，令 $k_q = 0.8$ 。平均队列长度计算时间间隔取值为 $t_p / 20$ 。

节点间通过使用发送窗口控制数据发送量，所

以计算负载的时间间隔 t_p 应该由单跳节点间往返时延决定。由于网络数据具有突发性，因此 t_p 取值应比节点间往返时间稍大才能反应网络中数据流量情况，同时应该避免存储数据队列较长后才进行负载计算、通告工作。此处 $t_p = RTT + 2D$ ，其中 D 是往返时延的平均方差。

2.2.2 拥塞窗口的调节

TPSN 根据网络节点传递给发送方的负载情况调节拥塞窗口，窗口变化采用乘性增、线性增、对数增、乘性减方式进行调节。

当瓶颈节点链路传回负载低于 80% 时，说明网络空闲，发送方窗口采用乘性增长方式快速增长。

乘性增：

$$w_{con}(t + RTT) = w_{con}(t)(1 + \delta) \quad (4)$$

当瓶颈节点传回负载大于 80% 时，系统已经处于高利用率，发送方通过线性增、对数增、乘性减使窗口保持在较稳定状态，确保资源的高利用率。当负载大于 80%，低于 95% 时，拥塞窗口采用线性增长方式调节；当负载大于 95%，低于 100% 时，拥塞窗口减慢增长，采用对数增方式调节；当负载大于 100% 时，发送方采用乘性减方式调整拥塞窗口。

线性增：

$$w_{con}(t + RTT) = w_{con}(t) + \alpha \quad (5)$$

对数增：

$$w_{con}(t + RTT) = w_{con}(t) + \alpha / (w_{con}(t))^{\frac{1}{2}} \quad (6)$$

乘性减：

$$w_{con}(t + RTT) = w_{con}(t)\beta \quad (7)$$

其中， α 取值为 1，增长方式与 TCP 相同^[8]。窗口乘性增长率 δ 由网络剩余资源决定，网络剩余资源可以表示为 $1 - \rho_l$ ，其中 ρ_l 是当前链路负载情况，则

$$\delta = k(1 - \rho_l) / \rho_l \quad (8)$$

其中， k 是常量。源端接收到的负载并不是 ρ_l 的真实值，而是它的编码，负载编码见 2.2.3 节表 1，而且负载信息借助于跳到跳确认机制在延迟较短的节点之间传递，对异常情况反映迅速，所以并不需要过于保守的窗口增长率^[12]。当 $k \leq 1/2$ 时，TPSN 协议能够达到稳定状态，证明详见第 3 节性能分析，所以常量 $k = 0.5$ ， $\rho_l = 0.80$ ，窗口增长率 $\delta = 0.125$ 。由于加入 [95%, 100%) 负载范围，此处 $\beta = 0.95$ ，目

的是使源端采用对数增、乘性减方式调节窗口，使得发送速率保持稳定，并且充分利用网络资源。

TPSN 协议将链路负载分为 4 个级别，与文献 [12] 相比，通过增加 [95%, 100%) 负载范围使得在高利用率情况下窗口变化更加平缓，配合上对数增和乘性减，能够更长时间维持窗口稳定，保证网络资源利用率最大化。

2.2.3 动态快速负载传递

传递负载 ρ_l 真实值会占用较多网络带宽，为减少负载信息对网络资源占用，网络中传递的并不是负载真实值，而是采用分级编码方式处理后的负载，如表 1 所示。

表 1 负载编码

负载	编码
负载 < 80%	001
80% < 负载 < 95%	010
95% < 负载 < 100%	011
负载 > 100%	100

卫星网络中使用传统负载传递方式会造成反馈信息获取不及时，发送方不能及时调整发送策略，传统负载传递时间 $2 \sum_{i=1}^N T_i$ ，其中， T_i 是单跳传

输延迟， N 是传输路径上节点个数，如图 6(a) 所示。为能够实现数据传输路径上的链路负载信息快速传递至发送端，负载随着节点间确认信息捎带传递，基于跳到跳机制的负载反馈时间为 $2T_1$ ，其中 T_1 是源端与第一跳节点间链路延迟，如图 6(b) 实线部分所示。负载反馈时间与单跳往返时间有关，与端

到端延迟无关。显然 $2T_1 < 2 \sum_{i=1}^N T_i$ ，极大地减小链路

信息反馈时延。中间节点在收到负载信息后会将自身的链路负载情况与接收到的负载进行比较，将较大负载值写入数据捎带至上一跳，传递至源端节点的负载表示该路径上瓶颈链路的负载，发送端根据接收到负载信息调整拥塞窗口。同时跳到跳负载信息传递方式没有传统端到端方式需要端到端往返延迟的收敛过程，原因是一旦有数据传输就会有负载传递，即使是新加入数据流同样能够获取最新的链路负载信息。

将负载信息捎带于不断传递给发送端的数据中，发送端不仅可以更快掌握数据传输路径网络状

况，同时避免了传统端到端拥塞控制由于两端节点往返时延不同产生的网络资源分配不公平现象。如图 6(a)所示，传统负载反馈信息均是由目的端发出，中间节点不更新负载信息，随着两端节点往返时延增大，负载反馈信息到达频率减小，造成两端节点距离远的应用网络资源占用率低，需要在发送端根据往返时间等信息对参数进行调整，才能保证各个应用间的公平性。借助于跳到跳机制传递负载信息，最新负载反馈信息不断在各节点间传送，发送端接收负载信息频率与两端节点往返时延无关，只与单跳节点间往返时延相关，当单跳节点间往返时延差别较小，不会产生由于端节点间往返时延不同造成的不公平现象，证明详见第 3 节性能分析。

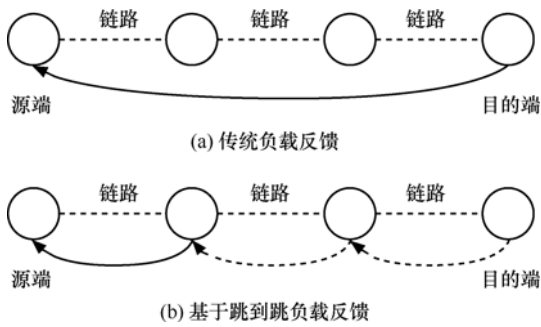


图 6 负载传递

3 性能分析

文献[12]中提出一种简单的协议稳定性分析数据模型。如图 7 所示，TPSN 协议模型中间节点根据 $t - RTT_{one-hop}$ 时刻发送数据计算链路负载 ρ_l ，并将负载传回发送端，发送端在 t 时刻根据接收到的负载调整发送窗口。根据 TPSN 特点对文献[12]中数据流分析模型进行调整：

$$\dot{w}_{con_i}(t) = \frac{1}{RTT_{one-hop}} \left(\delta(\rho_l) + \alpha + \frac{\alpha}{(w_{con_i}(t))^{1/2}} \right) \quad (9)$$

其中， $w_{con_i}(t)$ 是数据流 i 的拥塞窗口， $\delta(\rho_l)$ 、 α 和 $\frac{\alpha}{(w_{con_i}(t))^{1/2}}$ 分别是乘性增、线性增和对数增系数。协议正常运行状态下，窗口值大于 1， $w_{con_i}(t) > 1$ ，所以 $\alpha \geq \frac{\alpha}{(w_{con_i}(t))^{1/2}}$ ，式(9)可以简化为

$$\dot{w}_{con_i}(t) = \frac{1}{RTT_{one-hop}} (\delta(\rho_l) + \beta\alpha) \quad (10)$$

其中， $\delta(\rho_l) = k(1 - \rho_l) / \rho_l$ ， $1 \leq \beta < 2$ ，当 $k \leq 1/2$ 时，公式稳定值 $w_{con_i}^* = \gamma C RTT_{one-hop} + N \frac{\alpha}{k}$ 。上述模型与文献[12]中模型基本相同，所以此处不再详细证明稳定性，主要说明与文献[12]中的不同，TPSN 协议借助于跳到跳机制和对数增方式调节窗口，数据流 i 稳定速率为 $r_i^* = \frac{\gamma C}{N} + \frac{\beta\alpha}{k RTT_{one-hop}}$ ，其中

$RTT_{one-hop}$ 仅为单跳间往返时延，与端到端时延相比各种单跳时延间差值较小，协议不再因为端到端延迟差异导致各数据流间产生不公平现象。文献[12]中 $r_i^* = \frac{\gamma C}{N} + \frac{\alpha}{kT}$ ，其中 T 为端到端往返延迟，卫星

网络中由于节点移动性造成往返时延不仅易变且变化幅度较大，因此按照文献[12]处理方法很难达到公平性。同时由于 $\frac{\beta}{RTT_{one-hop}} > \frac{1}{T}$ ，所以 TPSN 能够比 VCP^[12]更加快速到达稳定状态。

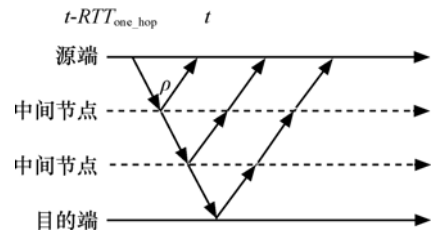


图 7 TPSN 模型

4 仿真实验

4.1 仿真环境

为了能够真实地模拟卫星网络环境，验证协议性能，使用事件驱动的 OMNeT++网络仿真软件建立卫星网络仿真验证系统，真实模拟卫星网络运行过程中网络环境变化，对协议进行性能验证与分析。

4.1.1 网络模型

卫星网络拓扑采用铱星星座，主要参数如表 2 所示。卫星经过极区时，星际链路会关闭。反向缝之间不存在星际链路。

4.1.2 节点模型

节点模型主要包括 2 类，一类是卫星节点模型，另一类是地面节点模型。主要包括 3 个功能模块，第 1 个是轨道链路模拟功能模块，第 2 个是传输控制协议功能模块，第 3 个是路由协议功能模块。

表 2 卫星网络拓扑主要参数

卫星个数	高度	轨道平面倾角	星际链路数	轨道平面数	反向缝星际链路	星座类型
66	780km	84.4°	轨道内 2 条, 轨道间 2 条	6	无	极地

轨道链路模拟功能实现卫星运动轨道以及卫星链路模拟。传输控制协议功能模块分别实现了本文提出的传输控制协议 TPSN、CCSDS 制定的 SCPS-TP 协议、TCP 协议以及 VCP 协议。路由协议功能模块实现参考文献[20, 21]中提出的基于地理位置信息的分布式路由算法。

4.1.3 仿真参数设置

每条链路带宽设置为 20Mbit/s, 数据发送时间间隔满足 Poisson 分布, 数据分组长度为 1KB。每条链路均维护一条先进先出队列, 队列长度为 200KB。数据通信均是由地面站发起, 通过卫星网络传输信息到目的地面站。当卫星高度为 780km 时, 卫星轨道周期为 1h40min, 2h 仿真时间能够反应整个网络变化过程对协议的影响, 所以除特殊说明, 仿真时间持续 2h。由于仿真实验在卫星拓扑结构中运行, 端到端往返时延由实际运行状态决定, 所以通过设置不同端节点距离调整端节点间延迟。仿真过程中, 2 个端节点间经过卫星个数随着运行时间推进会不断发生变化。

4.2 仿真结果分析

4.2.1 窗口增长及链路利用率

首先, 考虑协议对网络资源的利用。如图 8 所示, 纵坐标为各时刻窗口值与目的端接收缓存比值, 仿真时间持续 120s, 图中显示前 50s 窗口变化过程。各种情况下, TPSN 拥塞窗口初始阶段快速上升, 稳定后不因为距离、链路误码率等因素发生变化。对于 TCP, 由于节点移动导致两端节点间 RTT 发生变化, 发送的数据可能因为距离增大导致确认信息反馈时延增加, 造成 TCP 错误认为数据超时, 重传未确认数据、降低拥塞窗口, TCP 规定在重传数据确认到达之前不可以更新 RTT [22], 使得后续发送的数据分组仍然使用未更新的 RTO, 导致后续发送数据即使完全正常传输, 由于 RTT 未能及时更新, TCP 可能仍然错误认为数据需要超时重传。当链路误码率增高后, 情况更加恶劣, 除了超时影响之外, 还有数据分组丢失, 导致 TCP 一直处于慢启动阶段, 对网络资源利用率很低。TCP 在初始阶段拥塞窗口会增长, 但是当窗口增长到一定值后窗口开始下降, 链路利用率降低。说明 TCP 不适用于

高带宽、拓扑结构经常发生变化的卫星网络。SCPS-TP 采用 TCP-Vegas 拥塞控制方式, 这种方式虽然不会将链路切换或链路误码导致的数据丢失误判为链路拥塞, 但是非常依赖于 RTT 的测量以及 2 个边界值的选取; 数据确认采用 SNACK 方式, 随着两端节点距离增加, RTT 增大、链路误码率提升, SNACK 反馈频率会降低, 并且反馈信息可能丢失, 导致拥塞窗口增长缓慢, 数据发送量减少。针对上述问题, TPSN 借助于跳到跳机制保证窗口的及时更新。VCP 需要等待端到端负载信息才会对窗口进行调整, 而且乘性增参数设置为 0.062 5, 取值非常保守, 在长延时情况下窗口增长非常缓慢。

链路资源利用率计算公式:

$$l_{utilization} = \frac{N_{acked} p_{avg}}{l_{rate} t_{interval}} \quad (11)$$

其中, $t_{interval}$ 为采样时间间隔, N_{acked} 是时间 $t_{interval}$ 内被确认数据分组数量, p_{avg} 是数据分组平均长度, l_{rate} 是链路数据发送速率。

如图 9 所示, 对于 SCPS-TP 当链路误码率为 10^{-5} 时, 由于拥塞窗口上升非常缓慢, 链路利用率低。随着误码率的降低, 链路利用率立即恢复到很高水平。在链路误码率较低、端节点距离较短时, 由于数据分组丢失较少及端节点延迟较小, VCP 和 TPSN 协议利用率基本相同。但是 SCPS-TP 和 VCP 都存在相同问题, 发送端均需要等待接收端确认信息, 所以当距离增大到 10 000km, 由于往返时间加大, 确认信息反馈缓慢, 利用率会下降。对于 SCPS-TP 和 VCP, 随着误码率上升, 由于仅使用端确认保证机制, 数据更容易丢失, 利用率同样下降。TPSN 协议利用跳到跳确认机制使负载信息传递不再受端节点间距离的影响, 能够一直保持很高的链路利用率。

4.2.2 公平性及队列稳定性

传统端到端的拥塞控制机制中往返时延 RTT 小的应用由于反馈迅速, 首先抢占网络资源, 造成各种应用业务间网络资源分配不公平。针对这种情况已经有很多协议对其进行专门处理, 主要处理方式是通过加入参量减少由于 RTT 不同对窗口调整

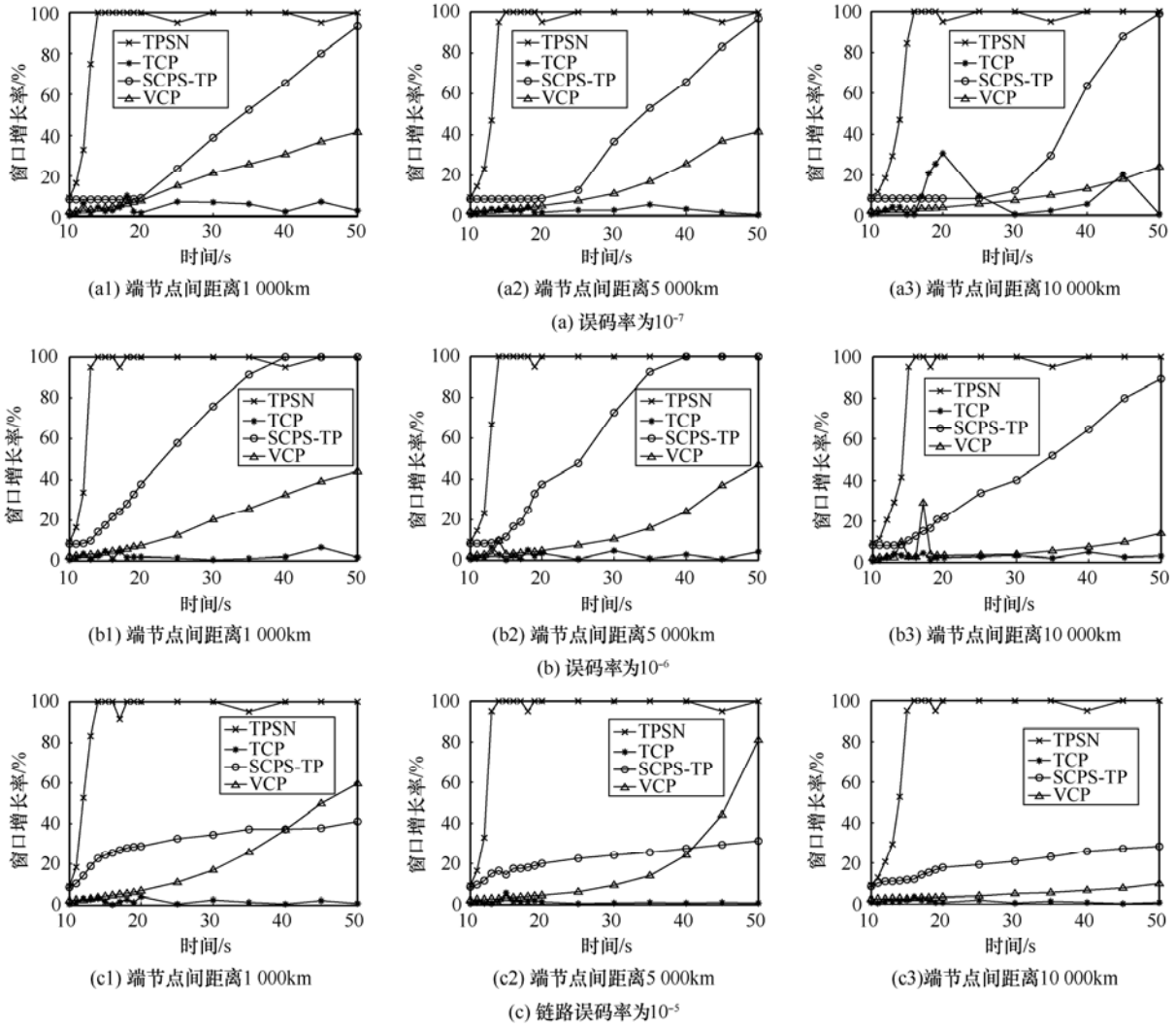


图8 拥塞窗口变化

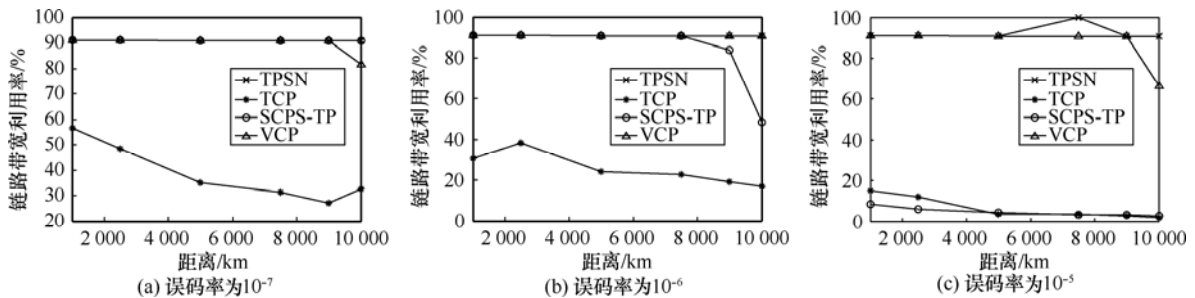


图9 利用率对比

的影响。TPSN 协议负载反馈无需等待目的端反馈信息，中间节点即将负载信息反馈至发送端，窗口变化频率由单跳之间往返时间决定，与两端节点距离无关，所以不需要参数处理即能够保证各应用数据流之间的公平性。公平性验证的各节点拓扑如图 10 所示，每加入一个新的目的节点，源节点分别为目的节点建立新的数据流。各目的节点与源端距离

变化范围是 1 000km 到 10 000km。随着具有不同端到端距离的数据流加入，各协议公平性变化如图 11 所示。参考文献[23]，公平性计算公式为

$$I_{\text{fairness}} = (\sum_{i=1}^N x_i)^2 / (N \sum_{i=1}^N x_i^2) \quad (12)$$

其中， N 是数据流总数， x_i 是第 i 个数据流发送数据量。

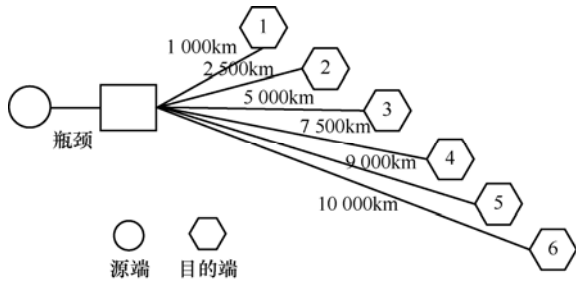


图 10 公平性测试拓扑

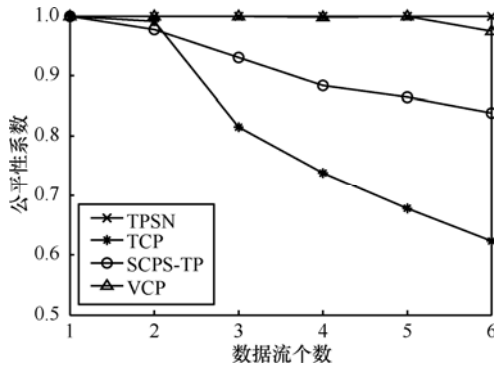


图 11 公平性

由图 11 可以看出，随着与目的节点距离不同的数据流数量增加，TCP 和 SCPS-TP 公平性下降很快。VCP 对 RTT 不同造成的不公平现象进行处理，公平性一直保持很好，仅略低于 TPSN 协议，但是随着数据流个数增加和端节点间延迟增大，TPSN 协议公平性优势逐渐体现出来，同时 VCP 公平性处理与 TPSN 协议相比较复杂，增加了节点处理负担，而 TPSN 协议从设计上保证各种不同 RTT 应用数据流的公平性，节点无需额外地处理开销就能够保证不同 RTT 节点间的公平性。

通过节点队列变化情况说明协议在不同环境下的稳定性。队列长度取值是数据传输路径上传输队列长度，仿真中队列采样时间间隔为 0.1s，图 12 中绘制的是传输路径上最长的队列长度，随着端节点距离增大队列长度略有增长，但是基本

稳定在 15KB 到 20KB 之间。这是因为 TPSN 协议通过使用拥塞控制以及灵活的确认机制，使 TPSN 协议节点发送队列长度一直处于平稳且较低水平。

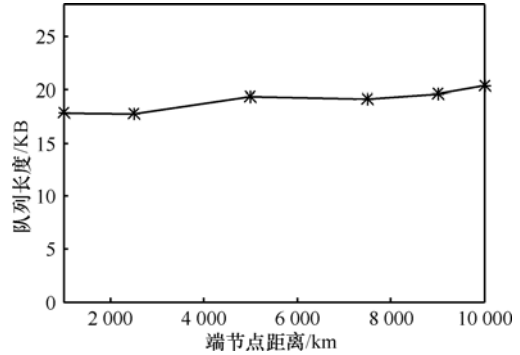


图 12 TPSN 协议队列变化

4.2.3 有效传输率

本文通过计算数据有效传输率验证 TPSN 协议对网络资源的有效利用，有效传输率计算公式为

$$E_{rate} = \frac{d_{rcv}}{d_{rcv} + d_{retrans}} \quad (13)$$

其中， E_{rate} 表示有效传输率， d_{rcv} 表示接收方成功接收数据量，不包括重复数据， $d_{retrans}$ 表示重传数据量。此处测试方法是发送方数据发送时间间隔为 0.1s，当接收方接收到 1 000 个数据分组时，统计发送方重传数据分组个数，通过式(13)计算有效传输率。

由于 VCP^[12]没有数据可靠性保证部分的描述，此处的比较对象是 TPSN 协议、TCP 和 SCPS-TP。

图 13 中横坐标表示发送方与接收方之间距离，单位是 km，纵坐标表示按照式(12)计算得到的有效传输率。由于 TPSN 协议采用跳到跳的确认方式，使各种丢失的数据在中间节点即能够重传恢复，减少源端对数据的重传，距离、误码率因素不会影响

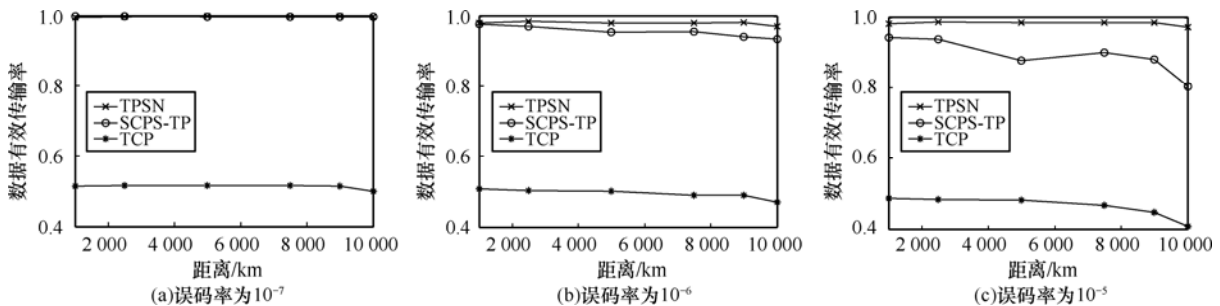


图 13 数据有效传输率

TPSN 协议对网络资源的有效使用, 而 TCP 由于并不针对长延迟、高误码率、拓扑不断变化的无线网络设计。如图 13(a)和图 13(b)所示, 链路误码率处于 10^{-7} 和 10^{-6} 时, TCP 有效传输率基本维持在 50% 的较低水平, 原因是卫星网络环境下数据需要通过多跳移动节点转发, 传输路径状态频繁的变化, 造成 RTT 的频繁变化, 导致 TCP 错误的超时和重传, 降低了链路的利用率; 随着误码率上升, 网络资源有效利用率基本保持不变, 如图 13(c)所示。当误码率较低时, 如图 13(a)和图 13(b)所示, SCPS-TP 与 TPSN 协议数据有效传输率基本相同。但是由于 SCPS-TP 仅采用端节点间确认重传方式, 所以当链路误码率增加到 10^{-5} , 端节点距离增加到 9 000km 以上后, 如图 13(c)所示, 数据丢失量增大, 发送端重传数据量增大, 数据有效传输率略有降低。

5 结束语

本文提出了一种基于跳到跳信息的适用于卫星网络的传输控制协议 TPSN, 该协议针对卫星网络高误码率、长延迟、拓扑变化、资源有限等特点设计了跳到跳累积确认机制, 能够快速恢复成段丢失数据。为能够及时恢复跳到跳确认无法恢复的丢失数据, 减少控制信息对反向链路带宽占用, 设计了基于检测窗口的选择否定应答机制。

针对卫星网络环境可能出现的拥塞情况, 本文在跳到跳机制的基础上, 设计了一种基于应用数据流的拥塞控制机制。这种方式能够使发送方快速掌握传输路径上实时状态, 做出及时、正确的窗口调整, 在长延迟、高误码率、拓扑经常变化的无线环境下能够充分利用网络资源。

本文中跳到跳确认信息处于网络层, 如何更好地与路由协议合作, 更加充分利用网络资源是今后研究工作的重点。

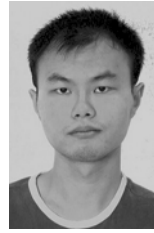
参考文献:

- [1] FARSEROTU J, PRASAD R. A survey of future broadband multimedia satellite systems, issues and trends[J]. IEEE Communications Magazine, 2000, 38(6): 128-133.
- [2] GITMAN I. Comparison of hop-by-hop and end-to-end acknowledgement schemes in computer communication networks[J]. IEEE Transactions on Communication, 1976, 24(11): 1258-1262.
- [3] WAN C Y, CAMPBELL A T, KRISHNAMURTHY L. Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks[J]. IEEE Journal on Selected Areas in Communication, 2005, 23(4): 862-872.
- [4] DATTA D, KUNDU S. Reliable and efficient data transfer in wireless sensor networks via out-of-sequence forwarding and delayed request for missing packets[A]. Proceedings of the International Conference on Information Technology[C]. Las Vegas, NV, IEEE Computer Society, 2007. 128-133.
- [5] PARAMASIVAN B, KRISHNAN R S. An enhanced reliability scheme for wireless sensor networks using PSFQ[A]. International Symposium on Ad Hoc and Ubiquitous Computing[C]. Surathkal, 2006. 106-111.
- [6] HEIMLICHER S, BAUMANN R, MAY M, *et al.* The transport layer revisited[A]. Proceedings of 2nd IEEE International Conference on Communication Systems, Software and Middleware[C]. Bangalore, 2007.
- [7] FRANCISCO R, ANTONIO G, PAULO R P, *et al.* Performance evaluation of DTSN in wireless sensor networks[A]. Proceedings of the 4th EuroNGI Workshop on Wireless and Mobility[C]. Barcelona, Spain, 2008.
- [8] JACOBSON V. Congestion avoidance and control[A]. Proceedings of SIGCOMM[C]. Stanford, CA, USA, 1988.
- [9] POSTEL J. Transmission Control Protocol[S]. RFC 793, 1981.
- [10] ATHURALIYA S, LI V, LOW S. REM: active queue management[J]. IEEE Network, 2001, 15(3): 48-53.
- [11] KATABI D, HANDLEY M, ROHRS C. Congestion control for high bandwidth-delay product networks[A]. Proceedings of ACM SIGCOMM[C]. Pittsburgh, PA, 2002.
- [12] XIA Y, SUBRAMANIAN L, STOICA I. One more bit is enough[J]. IEEE/ACM Transactions on Networking, 2008, 16(6): 1281-1294.
- [13] QAZI A, ZNATI T. On the design of load factor based congestion control protocols for next-generation networks[A]. IEEE INFOCOM[C]. Phoenix, AZ, 2008.
- [14] QAZI A, ZNATI T, ANDREW L L H. Congestion control using efficient explicit feedback[A]. IEEE INFOCOM[C]. Rio de Janeiro, 2009.
- [15] DANIEL S. Hop-by-Hop Transport Control for Multi-hop Wireless

Networks[D]. England: Brigham Young University, 2007.

- [16] YUNG Y, SANJAY S. Hop-by-hop control over wireless multi-hop networks[J]. IEEE/ACM Transactions on Networking, 2007, 15(1): 133-144.
- [17] Recommendation for space data system standards: SCPS-TP[EB/OL]. <http://public.ccsds.org/publications/archive>, 2006.
- [18] GRIECO L A, MASCOLO S. Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control[J]. ACM SIGCOMM Computer Communication Review, 2004, 34(2): 25-38.
- [19] TARIK T, DAISUKE M, ABBAS J, *et al.* Explicit load balancing technique for NGENO satellite IP networks with on-board processing capabilities[J]. IEEE/ACM Transactions on Networking, 2009, 17(1): 281-291.
- [20] EKICI E, LAN F A, MICHEAL D B. A distributed routing algorithm for datagram traffic in LEO satellite networks[J]. IEEE/ACM Transactions on Networking, 2001, 9(2): 137-147.
- [21] SVIGEL J A, MOHORCIC M, KANDUS G. Routing in ISL networks considering empirical IP traffic[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(2): 261-272.
- [22] KARN C P. Improving round-trip time estimates in reliable transport protocols[J]. ACM SIGCOMM Computer Communication Review, 1995, 25(1): 66-74.
- [23] CHIU D, JAIN R. Analysis of the increase and decrease algorithms for congestion avoidance in computer networks[J]. Computer Networks and ISDN Systems, 1989, 17(1): 1-14.

作者简介:



王路(1984-), 男, 江苏泰兴人, 中国科学院博士生, 主要研究方向为无线网络协议。



胡月梅(1979-), 女, 安徽舒城人, 中国科学院工程师, 主要研究方向为无线网络协议。



刘立祥(1973-), 男, 安徽安庆人, 博士后, 中国科学院副研究员, 主要研究方向为无线网络通信及路由技术。



胡晓惠(1960-), 男, 内蒙古人, 中国科学院研究员、博士生导师, 主要研究方向为信息系统集成与仿真技术。