

基于地理位置信息的无收敛多测度卫星网络 路由算法研究

王 路^{1,2}, 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: 设计了一种基于地理位置信息的无收敛多测度的卫星网络路由算法 CFR, 算法根据地理位置关系、延迟和链路丢包率三种测度计算路由, 满足不同应用的服务质量要求。当有数据到达时, 卫星根据地理位置关系、网络拓扑和链路状态实时计算数据传输路径, 避免了路由收敛过程。在此基础上, 为有效均衡卫星网络负载, 卫星将链路负载信息通告给数据传输路径上各颗卫星, 卫星根据负载调节数据传输路径, 有效地均衡了网络负载, 减少了数据丢失, 增加了吞吐量。

关键词: 卫星网络; 无收敛; 多测度; 负载均衡; 地理位置信息

中图分类号: TP393 **文献标识码:** A **文章编号:** 1000-1328(2011)07-1542-09

DOI: 10.3873/j.issn.1000-1328.2011.07.016

Geographical Location-Based Convergence-Free Routing Using Multiple Metrics for Satellite Networks

WANG Lu^{1,2}, 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 novel satellite network routing algorithm named CFR is proposed in this paper. In the CFR geographical location, delay and packet drop rate are used as metrics to calculate routes to meet different QoS requirements. When packets arrive, instead of using global routing tables, CFR calculates routes in real time, in order to achieve convergence free. In addition, an explicit load balancing mechanism is proposed to achieve load balance. The link load information is exchanged among satellites transmitting packets from the same data flow. In response, a less congested path is selected when there are satellites with heavy link load. The retrieved path does not include the congested link, and a portion of data is communicated via the retrieved path. CFR is able to guarantee a good performance in terms of a better distribution of traffic among satellites, lower packet drops and higher throughput.

Key words: Satellite network; Convergence free; Multiple metrics; Load balancing; Geographical location

0 引 言

卫星网络能够实现全球覆盖,且数据传输具有广播特性,是为用户提供宽带接入的非常不错的选择,特别适用于沙漠、极地等不适于铺设有线网络的地区的用户。卫星网络中,卫星之间及卫星与地面

用户之间通过同一星座内的链路(Intra Satellite Link, ISL),不同星座间的链路(Inter Orbit Link, IOL)以及卫星与地面用户之间的链路(User Datagram Link, UDL)实现互联。

LEO 卫星往返延时小,对终端设备功率要求低,而且实现与部署单层 LEO 卫星网络相对较为容

易,所以已经有许多学者对单层 LEO 卫星网络路由算法开展了研究^[1-2]。这些算法通常针对单一星座的单层 LEO 卫星网络设计。随着技术发展,人们发现多层卫星网络能够获得更高的网络服务性能,之后出现了多层卫星网络路由算法,如 MLSR^[3]、SGRP^[4]、SOS^[5]等。MLSR 是一种层次化的链路状态路由算法。SGRP 是一种层次化的、基于快照的多层卫星网络集中式路由算法。这些多层卫星网络路由算法通常由 LEO 卫星传输数据, MEO 和 GEO 卫星完成控制与管理功能,并没有充分发挥 MEO 和 GEO 的传输能力。还有一类利用地理位置信息的卫星网络路由算法,如 LDPB^[6],该算法利用卫星广播特性,通过卫星间位置关系,计算数据传输参考路径,将数据发往目的端方向。

由于卫星的快速移动特性(LEO 卫星运动速度超过 25,000 千米/小时^[7]),目前多数卫星网络路由算法都存在收敛过程,包括收集网络信息,更新路由表。收敛过程中由于路由表的失效可能导致数据被丢弃,需要重传。卫星网络动态拓扑特性可能造成路由表的频繁收敛,产生较重的网络负担。在基于快照的路由算法中,卫星需要存储快照时间片内的路由信息,随着卫星网络规模增加,存储空间不断加大。当链路发生故障时,当前快照时间片及后续时间片内路由结果将失效,路由计算节点需要较长时间重新计算和更新卫星路由信息。

卫星网络除了多层结构、动态拓扑特点外,还具有用户分布不均匀的特点,城市上空卫星用户数量远远高于偏远地区卫星用户数量。针对用户分布的不均匀性,提出了一些负载均衡机制,如 CEMR^[8]、ELB^[9]等。CEMR 根据卫星数据传输时延以及排队时延计算路由,但是卫星仅考虑本地信息,没有考虑数据后续传输过程中链路拥塞以及丢包情况的发生。ELB 设计了一种显式的负载均衡机制,并对数据流进行分级,相邻卫星之间交换负载信息,卫星根据数据流优先级对数据进行分流,使各卫星之间负载均衡,避免数据丢失。但是该算法只有当邻居卫星负载过重时才均衡网络负载,源端对网络负载情况反应较为迟钝。同时 ELB 采用跟踪数据传输时延的方法,避免路由环路,实际上这种方法无法完全避免路由环路的产生。

针对上述卫星网络的特点以及已有算法尚未解决的问题,本文提出一种基于地理位置信息的无收敛多测度卫星网络路由算法 CFR(Convergence-Free Routing),该算法首先根据地理位置信息确定数据传输方向,获得下一跳候选集;然后使用链路延迟和链路丢包率两种路由测度,从候选集中确定下一跳,并根据数据传输路径上瓶颈链路负载情况进行数据分流,能够满足不同业务的 QoS 需求,保证网络负载均衡。

1 基于地理位置信息的无收敛多测度卫星网络路由算法

CFR 根据卫星网络拓扑特性,使用卫星及地面节点地理位置信息,将数据发往距离目的端更近的卫星,直到卫星能够成功覆盖目的端。

GPS 可以非常容易地获取地球表面节点的经纬度信息,本文根据经纬度信息划分地球表面,地球表面节点地址使用所处区域经纬度信息表示。划分地球表面的目的不再是为卫星分配虚拟地址^[3-4],而是便于发送端判断目的端位置。为减小地面节点与卫星之间切换频率,源端和目的端使用最长覆盖时间作为接入策略。当卫星能够覆盖目的端时,即将数据发往目的端。

处于同一层的极地卫星网络,存在两条相对运动速度很快的轨道,两条轨道间不存在轨间链路,称之为“缝”,如极轨 LEO 卫星网络的“缝”。由于“缝”的存在,卫星接收到数据后需要判断目的端是否在“缝”的另一侧,如果是,则数据需要过极传输到与目的端一侧,或者发送到另一层卫星网络,消除当前层的“缝”对数据传输的影响。由于其它层的卫星同样可能存在“缝”,所以如果存在过极路径,则卫星不会将数据发往另一层卫星。由于同一层卫星网络中“缝”总是位于两条固定的轨道之间,所以卫星所处的轨道与“缝”在赤道交点的经度差是固定的。由于卫星轨道存在一定倾斜角度,所以卫星所处轨道与赤道交点的经度并不等于卫星所处位置的经度。定义(latitude, longitude)为当前位置的经纬度坐标,obliquity_orbit 为轨道倾角,该卫星星座内任意位置在赤道交点的经度 p_longitude 计算公式如下:

$$p_longitude = longitude + \arcsin(\tan(latitude)/\tan(obliquity_orbit)) \quad (1)$$

假设当前卫星经纬度为 $(lat_a, long_a)$, 该层卫星网络两条“缝”分别称为左缝与右缝, 左缝与赤道交点的经度为 $long_{left_sew}$, 右缝与赤道交点的经度为 $long_{right_sew}$, 该颗卫星所处轨道和左缝在赤道交点的经度差, 简称左缝经度差:

$$left_sew_distance = long_a + \arcsin(\tan(lat_a)/\tan(obliquity_orbit)) - long_{left_sew} \quad (2)$$

该颗卫星所处轨道和右缝在赤道交点的经度差, 简称右缝经度差:

$$right_sew_distance = long_{right_sew} - long_a + \arcsin(\tan(lat_a)/\tan(obliquity_orbit)) \quad (3)$$

左缝经度差和右缝经度差在卫星星座结构不发生变化的情况下保持不变。如果目的端所处经度与当前卫星所处轨道在赤道交点的经度差在左缝经度差 $left_sew_distance$ 和右缝经度差 $right_sew_distance$ 之间, 则可以判断当前卫星与目的端处于“缝”的同侧, 否则当前卫星与目的端处于“缝”的两侧。设 $Neighbor(src)$ 存放卫星 src 邻居卫星列表, $Nei(src)$ 存放根据当前卫星与“缝”位置关系更新后的邻居卫星列表。如果当前卫星与目的端位于“缝”的同侧, 则 $Nei(src) = \{x \mid x \in Neighbor(src)\}$ 。如果当前卫星与目的端位于“缝”的两侧, 则 $Nei(src)$ 中的卫星需要与当前卫星位于同层, 且保证数据能够过极。定义对称点的概念, 对称点是指轨道上一点与地球圆心连线在同一轨道另一侧的交点。保证数据过极的判定方法是假设北极为正纬度, 南极为负纬度, 如果目的端比当前卫星的对称点纬度高, 则数据应该经过北极完成过极, 否则应该经过南极完成过极。做出上述判断后, 当前卫星根据过极的方向从同轨道的邻居卫星中选择下一跳, 直到数据成功过极。如果不存在上述条件的卫星, 则 $Nei(src)$ 仅包含不同层卫星。

判断当前卫星与目的端是否处于“缝”的两侧后, 更新得到邻居卫星集合 $Nei(src)$ 。如果当前卫星与目的端位于缝的两侧:

$$N(x) = \{x \mid x \in Nei(src)\} \quad (4)$$

其中, src 是当前卫星, $Nei(src)$ 是 src 邻居卫星集

合。如果当前卫星与目的端位于“缝”的同侧, 则计算邻居卫星与目的端之间的距离, 产生候选下一跳节点集 $N(x)$:

$$N(x) = \{x \mid distance(x, dest) < distance(src, dest), x \in Nei(src)\} \quad (5)$$

其中, src 是当前卫星, $Nei(src)$ 是 src 邻居卫星集合, $distance(x, dest)$ 是 x 与 $dest$ 之间距离, $distance(src, dest)$ 是 src 与 $dest$ 之间距离。

节点间距离计算使用地球坐标系, 由于卫星可能位于不同层, 计算当前卫星与目的端距离时, 使用星下点坐标计算。假设节点 a 星下点坐标为 $a(x_a, y_a, z_a)$, 节点 b 星下点坐标为 $b(x_b, y_b, z_b)$, 则节点 a 与节点 b 星下点之间距离:

$$distance(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \quad (6)$$

卫星计算得到候选下一跳集合 $N(x)$ 后, 需要从中选择卫星 $next$ 作为下一跳。由于 MEO 和 LEO 卫星网络动态拓扑特性, 如果维护全局拓扑信息, 不仅会产生大量节点间交互, 而且路由计算使用的网络拓扑可能已经过期, 路由结果无法达到最优, 甚至可能产生错误路径。针对上述问题, CFR 不再维护全局链路状态信息, 而是根据相邻卫星及链路状态等信息从集合 $N(x)$ 中选择下一跳。下一跳 $next$ 的选取根据数据类型及网络负载情况实时计算, 路由测度的选取详见 1.1 节路由选择的测度, 负载均衡机制详见 1.2 节传输路径负载均衡机制。

根据局部信息, 卫星分布式实时计算路由, 不再需要收敛过程, 能够很好地处理动态拓扑、节点故障、链路失效等产生的问题。但是面临另外两个问题, 实时路由计算给卫星带来较重的计算负担和数据传输过程中可能产生路由环路。为减少卫星路由计算负担, 卫星缓存路由计算结果, 直到邻居节点和链路丢包率发生变化。为避免路由环路, 数据包头中记录数据传输经过的卫星链路信息, 这些信息组成数据传输链路集合 F 。计算下一跳时, 不会选择数据已经经过的卫星链路, 避免产生路由环路。

虽然在包头中添加传输路径能够避免路由环路, 但是增加了网络负载。CFR 采用一种本地映射机制, 减少数据包头长度。当卫星 N_1 首次接收到传输链路集合为 F 的数据包或者下一跳 N_2 发生变化

时,卫星 N_1 将集合 F 映射为标签 l_f ,并将该集合 F 和映射标签 l_f 发送给下一跳 N_2 , N_2 接收到消息后发送确认信息,当 N_1 接收到 N_2 返回的确认信息后, N_1 不再将链路集合 F 写入发往 N_2 的数据包包头,而是仅将标签 l_f 写入包头。标签仅在两颗相邻卫星之间有效,并无全局含义。由于标签是在数据传输过程中按需产生,同颗卫星存储的标签数量 M 为该卫星传递数据经过各种卫星链路情况的集合。假设卫星传递数据经过的不同卫星链路个数为 n ,则 M 的最大值 M_{\max} 为:

$$M_{\max} = C_n^1 + C_n^2 + C_n^3 + \dots + C_n^n = 2^n - 1 \quad (7)$$

实际情况中 n 值不会很大,且 $M < M_{\max}$,所以同颗卫星存储标签数量最大值 M 较小。卫星为每个标签设置有效时钟 T 。超过时间 T ,标签及所代表的链路集合将被删除。时间 T 可以根据不同网络进行选择,本文中 T 设置为 1 分钟。

1.1 路由选择的测度

为满足不同数据流的 QoS 要求,本文将数据流分为三种类型,与参考文献[9]相似。三种类型数据并无固定优先级顺序,而是根据数据流需求决定路由测度的选择顺序,并用二进制数编号:

(1) 延迟敏感业务:典型应用包括实时交互式数据流,如语音业务,视频交互业务,编号 01;

(2) 吞吐量敏感业务:典型应用包括视频点播业务、大量数据文件传输业务,编号 10;

(3) 尽力而为服务:没有特殊业务要求的数据流业务均可以归为此类,编号 11。

根据上述划分,CFR 路由选择主要考虑两个方面因素,传输延迟和数据丢包率。

本文中数据传输延迟包括数据处理时延 T_{process} 、数据排队时延 T_{queue} 以及数据传播时延 $T_{\text{propagation}}$ 。实时性要求高的业务,如语音业务,首先考虑数据传输时延 α :

$$\alpha = T_{\text{process}} + T_{\text{queue}} + T_{\text{propagation}} \quad (8)$$

为最大化吞吐量,应减少数据成功传输所需要发送次数^[10]。数据成功传输所需发送次数与链路丢包率相关,假设链路丢包率为 l ,则数据成功传输所需发送次数 $n = \frac{1}{1-l}$ 。数据服务类型大致可以分为两类,第一类是可靠性保证服务,这类服务需要发送方与接收方双向合作,保证数据正确传输,第二

类是无可靠性保证服务,这类服务仅需要发送方发送数据。由于卫星链路的不对称性,前向与反向链路的链路丢包率可能不相等,对于第一类服务,为保证较高吞吐需要同时考虑正向与反向两条链路情况,对于第二类服务仅需要考虑正向链路情况。假设影响吞吐量因素为 χ ,正向链路丢包率为 l_f ,反向链路丢包率为 l_r 。 χ 计算公式如下:

$$\chi = \frac{1}{(1-l_f) \cdot (1-l_r)} \quad (9)$$

当为第二类无可靠性保证服务时, $l_r = 0$ 。

路由的下一跳 next 将综合考虑上述两种路由测度,计算公式 $\text{next} = f(\alpha, \chi)$ 。不同业务具有不同 QoS 要求,在数据包头中加入 4 比特,用于标识数据流对路由测度选取的优先级,前面 2 比特除表示数据对路由测度要求外,还表示该数据流类型。卫星 Sat_i 接收到包头信息 1001 后,判定该数据流为吞吐量敏感数据流,首先根据链路丢包率从下一跳集合 $N(x)$ 中选出子集 $N_{s_1}(x)$ 。如果 $N_{s_1}(x)$ 包括多颗候选下一跳,则卫星 Sat_i 根据链路传输延迟从 $N_{s_1}(x)$ 中选出下一条 next。

如果卫星 Sat_i 接收到尽力而为数据,假设卫星 Sat_i 与 $N(x)$ 中卫星互联链路队列长度为 $\text{que}_{i,j}$, $\text{Sat}_j \in N(x)$,则下一跳满足:

$$\text{next} = \{\text{Sat}_j \mid \min(\text{que}_{i,j}), \text{Sat}_j \in N(x)\} \quad (10)$$

即卫星 Sat_i 选取队列长度最短链路连接的卫星作为下一跳。

1.2 传输路径负载均衡机制

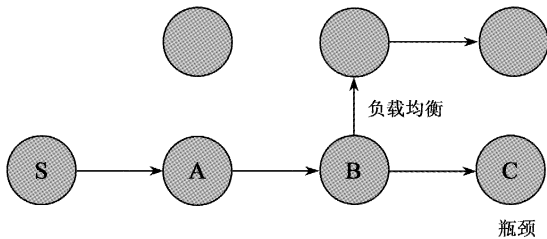
为能够均衡卫星网络负载,卫星每隔 t_p 时间计算链路负载 R_l 。 R_l 计算公式参考文献[11]:

$$R_l = \frac{\lambda_l + k_q \bar{q}_l}{\gamma_l C_l t_p} \quad (11)$$

其中, λ_l 和 \bar{q}_l 的定义与参考文献[11]不同,此处 λ_l 指 t_p 时间内进入卫星且从该链路传输的数据总量; \bar{q}_l 是在 t_p 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,本文中设置为 0.95; C_l 是链路数据发送能力^[11]。为能够更加充分利用节点存储空间,令 $k_q = 0.8$ 。平均队列长度计算时间间隔取值为 $t_p/20$ 。由于发送卫星数据发送量由负载信息决定,所以计算负载的时间间隔 t_p 应该由单跳卫星间往返时延决定。

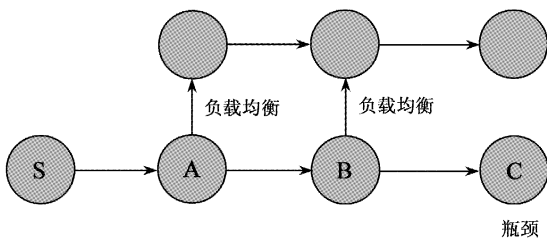
由于网络数据具有突发性,所以 t_p 取值应比卫星间往返时间稍大才能反应网络中数据流量情况,同时应该避免存储数据队列较长后才进行负载计算、通告工作。此处 $t_p = RTT + 2 \times D$,其中 D 是往返时延的平均方差。

假设 threshold 是链路负载门限值, $0 < \text{threshold} < 1$, 当负载 $R_l > \text{threshold}$ 后,表明卫星链路可能马上成为瓶颈链路,负载信息随同卫星链路状态信息告知邻居卫星。邻居卫星接收到链路负载信息后,读取数据传输链路集合 F ,如果传输链路集合 F 包括该条链路,则将链路负载与接收到的链路负载进行比较,将较大值传递到传输路径的上一跳卫星。卫星接收到链路负载后,进行上述相同操作步骤,这样数据传输路径上瓶颈链路的负载能够被所有上游卫星获得。卫星获得瓶颈链路负载 R_l ,且 $R_l > \text{threshold}$ 后,根据数据类型计算次优路径,分流数据,均衡网络负载,如图 1(b)。



(a) 传统负载均衡方式

(a) Traditional load balance



(b) 本文提出负载均衡方式

(b) CFR load balance

图 1 负载均衡

Fig.1 Load balance

假设, I_t 是卫星覆盖区域终端用户发往瓶颈链路数据传输速率总量, I_n 是邻居卫星发往瓶颈链路数据传输速率总量, O 是瓶颈链路输出数据速率总量, l_d 是星间链路数据传输延迟。假设节点接收到邻居节点 t 时刻负载 R_l , 此时节点负载:

$$R_l(t + l_d) = \frac{\lambda_t + k_q(\tilde{q}_t + (I_t + I_n - O)l_d)}{\gamma_l C_l t_p} \tag{12}$$

为使瓶颈节点快速恢复,并在正常状态维持 θ 时间,邻居节点收到负载信息后发往瓶颈链路数据速率总量 I_n^{new} 需要满足:

$$R_l(t + l_d + \theta) = \frac{\lambda_t + k_q(\tilde{q}_t + (I_t + I_n - O)l_d + (I_t + I_n^{new} - O)\theta)}{\gamma_l C_l t_p} < \text{threshold} \tag{13}$$

分流比例 φ 按照如下方式计算:

$$\varphi = 1 - \min(\max(0, \frac{I_n^{new}}{I_n}), 1) \tag{14}$$

前文已经对数据流进行了划分,卫星根据不同种类数据占总数据量比例分流数据。当卫星接收到链路负载信息超过门限 threshold 时,卫星节点首先为尽力而为数据流计算次优路径。如果分流比例 φ 超出尽力而为数据占总数据量的比例,则开始为吞吐量敏感数据计算次优路径,延迟敏感数据尽力按照最优路径进行传输。假设三种数据占总数据量的比例分别为 a 、 b 和 c 。数据分流比例如表 1。负载均衡机制可能使数据传输路径并不是根据测度计算的最优路径,但是通过负载均衡机制可以保持较短队列长度,较小数据排队等待时延。

表 1 不同类型数据分流比例

Table 1 Packet detouring ratio of three envisioned classes

	延迟敏感数据	吞吐量敏感数据	尽力而为数据
$\varphi < c$	0	0	$\frac{\varphi}{c}$
$c \leq \varphi < (b + c)$	0	$\frac{\varphi - c}{b}$	1
$\varphi \geq (b + c)$	$\frac{\varphi - b - c}{a}$	1	1

使用传输路径上瓶颈链路负载信息比使用本地负载信息进行数据分流能够更好地均衡网络负载。如图 1,使用本地负载信息,只有当数据到达瓶颈卫星邻居卫星时才进行分流,这样网络负载全部集中于瓶颈卫星周围,网络中其余节点可能仍然处于空闲状态。通过使用传输路径上瓶颈链路负载进行数据分流,使卫星能够根据传输链路负载情况更加快速地分流数据,使更多卫星参与到数据传输过程中,

资源得到更充分利用。假设 I 为进入瓶颈节点数据流总量,使用本地负载信息进行数据分流进入瓶颈链路数据量为 $I \cdot \varphi$,然而使用传输路径上瓶颈链路负载进行数据分流,瓶颈卫星前 N 跳能够获知前方路径存在瓶颈链路需要分流,时间 $\sum_{i=1}^N l_i$ 后进入瓶颈链路数据量为 $I \cdot \varphi^N$,其中 l_i 为链路 i 的数据传输延迟。假设使用局部负载信息正常运行持续时间为 θ_i ,进入瓶颈链路数据总量为 λ_i^l ,其中 $\lambda_i^l = I \cdot \varphi$,则卫星接收到邻居卫星链路 $(t + \theta_i)$ 时刻负载信息时,邻居卫星链路实际负载为:

$$R_i(t + l_d + \theta_i) = \frac{\lambda_i^l + k_q(\tilde{q}_i + (I_i + I_n - O)l_d + (I_i + I_n\varphi - O)\theta_i)}{\gamma_i C_i t_p} \quad (15)$$

其中 l_d 是两相邻卫星数据传输延迟。

假设使用整条传输路径瓶颈链路正常运行持续时间为 θ_g ,进入瓶颈链路数据总量为 λ_i^g ,其中 $\lambda_i^g = I \cdot \varphi^N$,则卫星接收到邻居卫星链路 $(t + \theta_g)$ 时刻的负载信息时,邻居卫星链路实际负载为:

$$R_i(t + l_d + \theta_g) = \frac{\lambda_i^g + k_q(\tilde{q}_i + (I_i + I_n - O)l_d + (I_i + I_n\varphi^N - O)\theta_g)}{\gamma_i C_i t_p} \quad (16)$$

当 $R_i(t + l_d + \theta_i) = R_i(t + l_d + \theta_g)$ 时,由于 $\lambda_i^l > \lambda_i^g$,所以 $\theta_g > \frac{\theta_i}{\varphi^{N-1}}$,又由于 $0 \leq \varphi < 1, N \geq 1$,所以 $\theta_g > \theta_i$,使用传输路径瓶颈链路负载的 CFR 能够比使用局部链路负载信息更加持久保持正常运行状态。

1.3 卫星网络信息收集与发布

根据上文中的描述,卫星节点需要收集邻居节点位置、链路延迟、链路丢包率、链路负载以及数据传输路径信息。本文采用主动方式收集和发布各种信息。

节点周期性主动发送探测信息,卫星切换时间由几分钟到几十分钟不等^[7],本文中主动探测信息发送周期设置为 1 分钟。卫星周期性发送 N 个探测信息,接收卫星收到探测信息后回复确认信息,卫星接收到确认信息后计算链路延迟 l_d ,并记录确认信

息个数 M ,正向链路丢包率 $l_f = 1 - \frac{M}{N}$ 。节点收集这些信息后,将这些信息发布到邻居卫星。卫星在收集并发布链路信息后,将信息存储到矩阵中,当链路信息发生变化,矩阵中信息自动更新。

本文中,产生信令开销的信息主要包括邻居节点位置、链路延迟、链路丢包率、链路负载、数据传输路径信息以及链路状态探测信息。由于链路状态探测信息发送周期是 1 分钟,所以邻居节点位置、链路延迟和链路丢包率,这三种信息的交互周期也是 1 分钟,发送频率较小。同时这些信息可以通过较少的比特位进行表示,而且仅在邻居卫星之间传递,不会向整网扩散。另外两种信息链路负载和数据传输路径信息通过在数据包头中增加少量比特位,随着传输的数据捎带传输。综上所述,这六种信息产生的信令开销较小,不会产生较重的网络负载。

2 性能分析

2.1 仿真环境

为验证协议性能,使用事件驱动的 OMNeT++^[12] 网络仿真软件建立卫星网络仿真系统,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。

2.1.1 网络模型

卫星网络拓扑采用依星星座,主要参数如表 2 所示。卫星经过极区时,轨道间链路会关闭。反向“缝”之间不存在星际链路。

2.1.2 节点模型

节点模型主要包括两类,一类是卫星节点模型,另一类是地面节点模型。主要包括三个功能模块,第一个是轨道链路模拟功能模块,第二个是传输控制协议功能模块,第三个是路由协议功能模块。轨道链路模拟功能实现卫星运动轨道,以及卫星链路模拟。传输控制协议模块的拥塞控制算法使用 VCP^[11],原因是 TCP 将数据包丢失视为网络发生拥塞,所以 VCP 较 TCP 更加适用于卫星网络环境。路由协议功能模块分别实现了 CFR、基于快照的路由算法^[13] 和最短路径算法 Dijkstra。链路队列使用主动队列管理算法 RED^[14],被标记的数据包会被丢弃。

表 2 卫星网络拓扑主要参数
Table 2 Network topology parameters

卫星个数	高度	轨道平面数	轨道平面倾角	极区切换纬度门限	星际链路数	反向缝星际链路	星座类型
66	780km	6	84.4°	70°	轨道内 2 条, 轨道间 2 条	无	极地

2.1.3 仿真参数设置

星间链路以及卫星与地面用户间链路的能力均设置为 25Mbps, 数据包长度为 1000bytes。节点链路队列采用先进先出队列, 队列缓存大小为 200Kbytes, 即最多可以存放 200 个数据包。链路误码率设置为零。RED 算法中, 平均队列计算权重 $w_q = 0.002$, 数据包标记概率最大值 $\max_q = 0.02$, 队列最小门限 \min_{th} 设置为 90Kbytes, 队列最大门限 \max_{th} 设置为 \min_{th} 的两倍^[14], 即 180Kbytes。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地面站。当卫星高度为 780km 时, 卫星轨道周期

表 3 终端分布

Table 3 Distribution of terminals

	北美洲	南美洲	欧洲	非洲	亚洲	大洋洲
终端数量	35	10	25	5	20	5

2.2 仿真结果

2.2.1 数据丢包率

数据丢包率如图 2 所示。数据丢失主要是指被标记的数据包。定义数据丢包率为 r_l , 数据包丢失个数为 l_n , 数据发送量为 s_n , 数据丢包率计算公式:

$$r_l = \frac{l_n}{s_n} \quad (17)$$

其中, 各种算法的 s_n 值相同, $s_n = 1.08 \times 10^{12}$ bits, 即等于发送速率为 1.5Mbps 时, 持续运行两小时的总数据发送量。

CFR 路由算法实时计算数据包传输路径, 并缓存路由结果, 减少算法计算复杂度, 对网络拓扑结构的变化能够作出快速反应, 同时负载均衡机制使得卫星间均衡负载, 有效降低了发生缓存溢出丢包的概率。最短路径算法和快照算法由于没有考虑负载均衡, 随着发送速率的线性增长, 导致超出队列最大门限 \max_{th} 的数据包数量成线性增长, 这些数据包会被标记并丢弃, 所以数据丢包率 r_l 表现为缓慢线性增长趋势。ELB 算法负载较轻时, 即发送速率小于 1.3Mbps 时, 数据丢包率缓慢上升, 表明负载均衡机

制作用发挥不明显。随着负载加重, 当发送速率高于 1.3Mbps 后, 数据丢包率减少, 表明队列长度超过最大门限 \max_{th} 的比例降低, 负载均衡机制发挥作用。但由于 ELB 的负载信息需要较长时间才能够到达发送端, 所以数据丢包率仍高于本文提出的 CFR 算法。

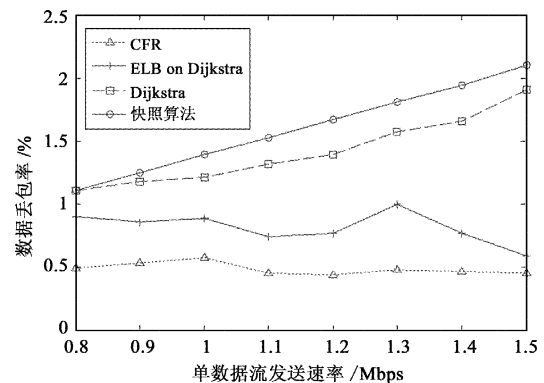


图 2 数据丢包率

Fig. 2 Packet drop rate

制作用发挥不明显。随着负载加重, 当发送速率高于 1.3Mbps 后, 数据丢包率减少, 表明队列长度超过最大门限 \max_{th} 的比例降低, 负载均衡机制发挥作用。但由于 ELB 的负载信息需要较长时间才能够到达发送端, 所以数据丢包率仍高于本文提出的 CFR 算法。

2.2.2 多级别数据流

数据传输延迟如图 3 所示, 虽然最短路径路由

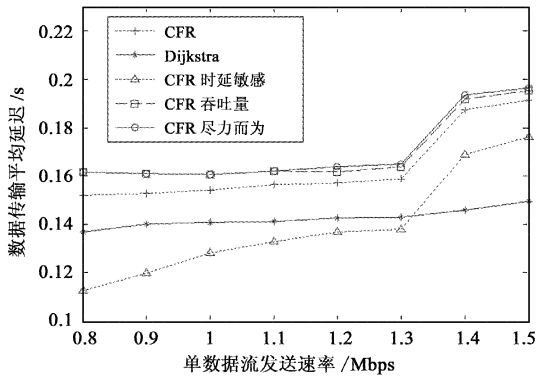


图 3 数据传输平均延迟

Fig. 3 Average transmission delay

算法数据传输延迟一直较小,特别是当发送速率增加到 1.3Mbps 之后,较重的数据负载情况下,最短路径路由算法仍然能够获得最短的数据传输延迟,但是这是以较高的丢包率作为代价换取的。路由算法 CFR,发送速率增加到 1.3Mbps 后,由于负载加重,数据分流,其中也包括延迟敏感数据,所以数据传输延迟明显增加。图 4 显示使用各种不同路由机制所得到的平均数据吞吐量。此处数据吞吐量是指在拥塞控制算法 VCP^[11]作用下,发送方实际发送数据速率。平均数据吞吐量计算公式如下:

$$\text{avg_through} = \frac{\sum_{i=1}^N \text{data_amount}_i}{N \cdot \text{time}} \quad (18)$$

其中, data_amount_i 是第 i 条数据流时间 time 内发送

数据总量, N 为使用同一路由机制数据流个数。链路丢包率作为路由测度的平均数据吞吐量较其他机制能够获得最高的吞吐量。使用延迟作为路由测度的数据流,由于需要满足时延要求,所以比仅能获得尽力而为服务的数据吞吐量大。最短路径 Dijkstra 路由算法由于没有负载均衡机制,使得节点链路负载较重,拥塞控制算法 VCP 降低拥塞窗口,使得实际发送数据量降低,数据流平均吞吐量减小。而将 ELB 应用在最短路径 Dijkstra 算法上,负载得到均衡,链路负载较轻,数据吞吐量比最短路径 Dijkstra 算法高。

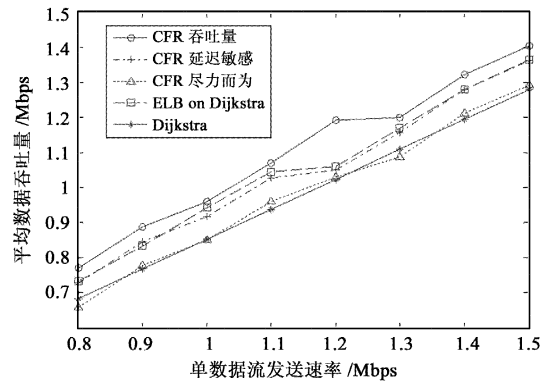


图 4 平均数据吞吐量

Fig. 4 Average throughput

上述两个仿真结果表明 CFR 路由算法由于采用了不同路由测度,以及有效的负载均衡机制,能够满足各种数据流的 QoS 需求。

表 4 终端用户全球分布情况

Table 4 Distribution of end users over the world

源端	目的端					
	北美洲 (%)	南美洲 (%)	欧洲 (%)	非洲 (%)	亚洲 (%)	大洋洲 (%)
北美洲 (%)	60	10	15	2	10	3
南美洲 (%)	35	40	12	2	8	3
欧洲 (%)	40	5	40	2	10	3
非洲 (%)	40	2	30	20	5	3
亚洲 (%)	20	2	10	2	50	6
大洋洲 (%)	40	2	10	2	12	34

2.2.3 负载均衡机制

使用 CFR、基于快照路由算法、最短路径 Dijkstra 算法,以及在最短路径 Dijkstra 算法基础上实现 ELB 进行网络负载均衡比较。其中,CFR 链路负载门限 $\text{threshold} = 0.95$ 。网络负载均衡系数计算公式如下^[9]:

$$f = \frac{\left(\sum_{i=1}^n x_i\right)^2}{n \sum_{i=1}^n x_i^2} \quad (19)$$

其中 n 为卫星网络链路总条数,包括卫星之间链路以及卫星与地面站间链路; x_i 表示第 i 条链路实际发

送数据包个数。系数变化范围是 0 到 1, 系数越接近于 1 表示网络负载越均衡。

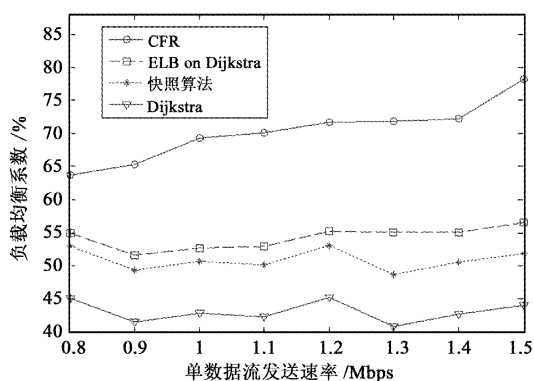


图 5 负载均衡系数

Fig. 5 Traffic distribution index

仿真结果如图 5 所示, 由于最短路径 Dijkstra 算法没有考虑到负载均衡问题, 负载均衡系数较低。将 ELB 引入最短路径 Dijkstra 算法, 网络负载均衡系数明显上升, 但是负载均衡系数仍然较小。CFR 根据不同数据流特性选择不同的传输路径, 尽力满足各种数据服务质量, 保证负载均衡, 可以达到较好的负载均衡性。快照算法根据全网拓扑信息计算路由, 所以能够获得较高的负载均衡系数。

3 结束语

本文提出了一种可用于多层卫星网络的基于地理位置信息的路由协议 CFR。该协议根据卫星以及地面节点的位置信息进行路由选择。为满足不同数据流的 QoS 需求, 本文将数据流分为时间敏感、吞吐量敏感以及尽力而为三种类型, 并使用链路延迟和链路丢包率作为路由测度。在此基础上, 为均衡网络负载, 根据数据传输路径上负载情况对网络数据进行分流。实验结果表明 CFR 路由算法根据数据流服务类型采用不同路由测度, 能够满足各种数据流的 QoS 需求, 并有效均衡了网络负载。

参 考 文 献

- [1] Ekici E, Akyildiz I F, Bender M D. A distributed routing algorithm for datagram traffic in LEO satellite networks [J]. IEEE/ACM Transactions on Networking, 2001, 9(2):137 - 147.
- [2] Henderson T R, Katz R H. On distributed, geographic-based packet routing for LEO satellite networks [C]. Proceedings of IEEE GLOBECOM, San Francisco, CA, USA, 2000.
- [3] Akyildiz I F, Ekici E, Bender M D. MLSR: A novel routing

algorithm for multilayered satellite IP networks [J]. IEEE/ACM Transactions on Networking, 2002, 10(3): 411 - 424.

- [4] Chen C, Ekici E. A routing protocol for hierarchical LEO/MEO satellite IP networks [J]. Wireless Networks, 2005, 11(4):507 - 521.
- [5] Lee J, Kang S. Satellite over satellite (SOS) networks: a novel architecture for satellite network [C]. Proceedings of IEEE INFOCOM, Tel-Aviv, Israel, 2000.
- [6] Chen C, Chen Z S. Towards a routing framework in ad hoc space network [J]. International Journal of Ad Hoc and Ubiquitous Computing, 2010, 5(1):44 - 55.
- [7] Wood L. Internetworking with satellite constellations, School of Electronics [D]. Computing and Mathematics, University of Surrey, Guildford, 2001.
- [8] Bai J J, Lu X C, Lu Z X. Compact explicit multipath routing for LEO satellite networks [C]. Proceedings of IEEE Workshop on High Performance Switching and Routing, Hongkong, China, 2005.
- [9] Taleb T, Mashimo D, Jamalipour A, et al. Explicit load balancing technique for NGE satellite IP networks with on-board processing capabilities [J]. IEEE/ACM Transactions on Networking, 2009, 17(1):281 - 291.
- [10] Couto D S J D, Aguayo D, Bicket J, et al. A high-throughput path metric for multi-hop wireless routing [J]. Wireless Networks, 2005, 11(4):419 - 434.
- [11] Xia Y, Subramanian L, Stoica I. One more is enough [J]. IEEE/ACM Transactions on Networking, 2008, 16(6):1281 - 1294.
- [12] Varga A. OMNeT++ discrete event simulation system. <http://www.omnetpp.org>.
- [13] Gounder V V, Prakash R, Abu-Amara H. Routing in LEO-based satellite networks [C]. Proceedings of Emerging Technologies Symposium on Wireless Communications and Systems, Richardson, TX, USA, 1999.
- [14] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance [J]. IEEE/ACM Transactions on Networking, 1993, 1(4):397 - 413.
- [15] Mohoric M, Werner M, Svigelj A, et al. Adaptive routing for packet-oriented intersatellite link networks: performance in various traffic scenarios [J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 808 - 818.

作者简介:王路(1984 -),男,博士生,中国科学院软件研究所,主要研究方向为卫星网络通信,计算机网络体系结构。通信地址:中科院软件所综合信息系统技术国家级重点实验室(100190)
E-mail:wlu810@gmail.com

(编辑:余 未)