

基于卫星时变网络的时延受限最小费用路由算法

张 涛, 张 军, 柳重堪

(北京航空航天大学电子信息工程学院, 北京 100083)

摘 要: 卫星网络是一种特殊的时变网络, 其 QoS 路由问题比地面网络 QoS 路由问题更为复杂. 本文利用卫星拓扑的可预测性, 通过引入业务的中断概率和费用模型来描述链路切换对业务 QoS 以及网络的影响, 并给出时延受限最小费用路由 (SDCMCR) 算法来解决在卫星链路时延延长的情况下寻找满足时延限制条件且受切换影响最小的路径的问题, 该算法可兼顾网络业务中断率和业务阻塞率等性能, 且计算复杂度低, 可适应星上设备处理能力的要求. 仿真表明算法在计算复杂性、业务请求被阻塞的概率等方面都优于同类算法.

关键词: 卫星通信网络; 服务质量路由; 时延受限; 路由算法

中图分类号: TN927+ . 23 **文献标识码:** A **文章编号:** 0372-2112 (2006) 09-1584-06

A Delay Constraint Minimum Cost Routing Algorithm for Satellite Time-Varying Network

ZHANG Tao, ZHANG Jun, LIU Zhong-kun

(School of Electronics and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China)

Abstract Mobile satellite network is a special time-varying network, and its QoS Routing problem is more complicated than the QoS Routing problem of current terrestrial networks. By exploiting the predictive nature of the mobile satellite topology and introducing the interrupting probability and the cost model of service, the influence on the QoS and network performance caused by the handover is described. Besides, a novel satellite network delay constraint minimum cost routing (SDCMCR) algorithm, which can obtain a path to satisfy the delay constraint as well as the minimization of influence caused by the handover, is proposed. This algorithm gives attention to the network performance of service interrupting and blocking. And at the same time, it has a low computing complexity to adapt to the restricted computing capability of satellite equipment. Correlative simulation indicates that this novel algorithm is superior to other current algorithm in respect of computing complexity, service blocking probability, etc.

Key words satellite communication network; QoS Routing; delay constrained routing algorithm

1 引言

QoS 路由技术是未来高速卫星通信网的关键技术之一, 是实现卫星网络多媒体通信和流量工程的基础^[1]. 但由于移动卫星网的特殊时变性, 网络中频繁出现链路切换, 使得针对卫星网络的 QoS 路由技术变得十分复杂. 为此, 许多学者提出将时变拓扑条件下的路由问题规约到静态拓扑下的路由问题, 来降低网络拓扑频繁变化对路由性能的影响. 如 GALPEDA (genetic algorithm, linear programming and extended Dijkstra shortest path algorithm) 算法^[2]将卫星运行周期分为很多个有限的时间区间, 并假设在该区间内卫星网络是静态的, 利用地面固定网络成熟的 GA 和

LP 算法对网络中优先级高的业务实行带宽预留来保证网络的 QoS 快照系列^[3] 算法通过定义多个“快照”, 将卫星网络表示成一系列“快照”的循环, 并利用 K 路由来实现网络的 QoS. 这类方法没有考虑所得最优路径可存在的时间, 该路径可能会很快随着卫星的切换而中断, 无法保证对许多业务 (如语音) 的 QoS.

解决卫星网络 QoS 路由问题的另一个思路是利用卫星运行的规律性, 通过对网络拓扑变化的预测来得到最优的 QoS 路径. 如基于概率的路由 (PRP)^[4], 该方法为每个业务呼叫尽可能的选取一条在呼叫持续时间内不会发生链路切换的路径; 文献 [5] 中通过定义链路的生成时间来选取一条持续时间最长的路径, 以保证业务不会因链路切

换而中断. 这些算法可降低链路切换对数据传输的影响, 减少网络的重路由率, 但算法同时牺牲了网络的阻塞率, 由于网络中许多路径因呼叫持续时间不够而被删除, 整个网络流量将主要集中在某一些链路上, 使得系统的阻塞率大大上升, 同时还会带来网络流量分布不均, 网络整体利用率低等缺点, 这种现象在网络负荷率高时表现尤为明显.

目前地面网络中一般将 QoS 路由问题规约为多约束最优寻路 (MCOP) 问题来解决, 在满足业务 QoS 的同时兼顾整个网络流量均衡, 这种方法可很好的克服 PRP 算法中因流量过于集中而带来的种种问题, 但现有的针对 MCOP 问题的算法, 如 C-N 算法^[6]、HMCOP (Heuristic for MCOP) 算法^[7]等, 大多只能得到次优解, 且计算复杂度大, 无法很好的适用于计算资源相对有限的卫星节点上.

卫星时变网络 QoS 路由问题需同时考虑链路时延长和拓扑时变等特性, 而拓扑的时变性可最终影响网络路径费用的大小, 这是因为链路切换会给网络带来的巨大额外开销 (如重路由带来的路由和信令开销等). 基于这种思路, 本文重点解决卫星时变网络中的时延受限最小费用路由问题, 通过将网络切换引入路径费用中, 来寻找一条既满足业务时延要求又可尽量降低链路切换对网络性能影响的最优 QoS 路径, 给出一种适用于卫星节点的时延受限最小费用路由 (SDCMCR) 算法. 该算法具有: (1) 可得到满足约束条件的最优无环解, 且计算复杂度小, 接近无约束的 Dijkstra 算法; (2) 可有效降低链路切换对 QoS 以及网络性能的影响; (3) 采用自适应动态调整技术, 算法适应性好等优点.

2 问题描述及符号说明

定义 1 卫星时变网络是一个各边权值都是以时间为变量的周期函数的无向图 (若两节点不相邻, 则权值为 ∞), 其模型为 $G(V, E, D(t))$, 其中 V 是有限节点集; E 是有限边集, $E \subseteq V \times V$; $D(t) = \{d_{ij}(t)\}$, $d_{ij}(t)$ 是以时间为变量, T 为周期的函数, 表示节点 i 与 j 间的时延, T 为网络 G 的周期.

$d_{ij}(t)$ 还可表示节点 i 与 j 间的连接关系, 反映网络的拓扑时变性. 这里引入一个新的概念: 生成时间来表示两节点间可持续通信的时间, 记为 $TL_{ij}(t)$, 且有如下关系存在: 在周期 T 内若节点 i 与 j 一直相连 (如轨内星间链路), 则 $d_{ij}(t)$ 为一连续函数, $TL_{ij}(t) = \infty$; 若节点 i 与 j 会发生切换 (如轨间星间链路), $d_{ij}(t)$ 为以切换时间为分段点的分段函数, 设分段点为 t_1, t_2, \dots , 若 $t \in [t_1, t_2]$ 且 $d_{ij}(t) \neq \infty$, 则 $TL_{ij}(t) = t_2 - t$

t 时刻, 若 η 为任意边 $e(i, j)$ 上的某一业务流, 且该业务还需要持续的时间为 x , 如 $x > TL_{ij}(t)$, 则该业务将会因网络的切换而中断, 将这种中断的概率定义为业务 η 的中断概率, 记为 $P_\eta(t)$, 其大小由 t 时刻该业务的持续时间分布模型 ξ 和 $TL_{ij}(t)$ 决定, 可表示为: $P_\eta(t) = f(\xi, TL_{ij}(t))$,

这里, ξ 由不同的业务决定, 如 Poisson 模型 (语音通信) 等.

为了保证链路中断后业务的 QoS 需要采取重路由或备份路径等方法, 这都会给网络带来额外开销. 因此可将这种链路切换对业务 QoS 的影响折合为该链路针对业务 η 的额外费用. 当然, 这种影响不仅与 $P_\eta(t)$ 有关, 还与当前网络的负荷情况以及业务对路径可靠性的敏感程度等因素有关.

根据上述分析, 则考虑切换影响后链路 $e(i, j)$ 针对业务 η 的费用模型 $c_\eta(t)$ 可以表示为式 (1). 该模型基于如下五个假设: (1) $e(i, j)$ 因切换而带来的额外费用 $c_\eta(t)$ 是以网络的负荷率、 $P_\eta(t)$ 以及业务对路径可靠性的敏感程度为变量的函数, 其中, $P_\eta(t)$ 是影响 $c_\eta(t)$ 大小的主要因素; (2) $c_\eta(t)$ 应满足如下边界条件: 当 $P_\eta(t) = 0$ 时, $c_\eta(t)$ 等于 $e(i, j)$ 上业务 η 的正常费用 (即未考虑切换时的费用 $c(t)$); 当 $P_\eta(t) \rightarrow 1$, $c_\eta(t) \rightarrow \infty$; (3) $c_\eta(t)$ 的大小与网络目前的负荷情况有关, 当网络的负荷率比较高的时候, 链路切换时无论是重新申请带宽还是提前预留带宽都比较困难, 这时切换对业务 QoS 和网络性能的影响相对较大, 即此时 $c_\eta(t)$ 较大; 反之, $c_\eta(t)$ 较小; (4) $c_\eta(t)$ 的大小还与具体的业务 η 有关, 对于一些对切换不“敏感” (即受切换影响小) 的业务, 如数据业务, $c_\eta(t)$ 值较小, 否则, $c_\eta(t)$ 值较大, 这里引入一个参数 γ 来描述不同业务对 $c_\eta(t)$ 的影响, γ 的具体取值参见图 5; (5) $P_\eta(t)$ 等对业务 QoS 的影响不是呈线性关系, 当 $P_\eta(t)$ 或网络负荷率越接近 1, 这种影响将越明显, 这里假设这种关系为指数关系.

$$c_\eta(t) = \frac{e^{\gamma \times \alpha \times P_\eta(t)}}{1 - P_\eta(t)} \times c(t) \quad (1)$$

其中 $c(t)$ 为 $e(i, j)$ 上业务 η 的正常费用 (未考虑切换时), 当 $P_\eta(t) = 0$ 时, $c_\eta(t) = c(t)$; γ 为业务因子, 表示业务对路径可靠性的敏感程度, $\gamma > 0$; α 为网络负荷率, $\alpha \in [0, 1]$.

为了便于研究和描述, 本文假设: 给定时刻 t , $G(V, E, D(t))$ 和业务 η 则与时间 t 有关的函数都可用一个常数来表示 (如 $P_\eta(t)$ 可表示为 P_η), 算法所需的其他符号说明如下:

- s, t : 源节点、目的节点;
- $U(i, j)$: 节点 i 到 j 间的一条路;
- $D(U(i, j))$: $U(i, j)$ 的时延值, 其值等于该路各边 d 之和;
- $C(U(i, j))$: $U(i, j)$ 的费用值, 其值等于该路各边 c_η 之和;
- D_r : 路径的时延约束值;
- $D_{\min}(s, t)$: s 到 t 的最短时延路的时延值;
- $C_{\min}(s, t)$: s 到 t 的最小费用路费用值;
- $D_{\text{amin}}(s, t)$: s 到 t 的最小费用路的时延值;
- $C_{\text{amin}}(s, t)$: s 到 t 的最短时延路费用值;
- Path_Set 算法的数据结构, 用于记录算法中的临时路;

$V_p(y)$: 算法的数据结构, 用于记录已检查过的从源节点到 y 的有效路。

定义 2 若 G 中任意边 $e(i, j) \in E$, 存在非负的 d_{ij}, P_{ij} 和 c_{ij} , 则卫星时变网络中的时延受限最小费用路由 (SDCMCR) 问题可描述为: 在图 G 中, 寻找一条从源节点 s 到目的节点 t 的路径 $U(s, t)$, 满足: $D(U(s, t)) \leq D_r$, 且 $\min(C(U(s, t)))$ 。

3 SDCMCR 算法设计

3.1 算法设计思想

对于卫星网络, 一方面, 由于卫星资源的昂贵, 需要算法找到满足条件的最小费用的解; 另一方面, 星上计算能力和储存能力的限制需要算法具有很好的运行效率。所以要求 SDCMCR 算法在不牺牲寻路性能的基础上具有尽可能低的计算复杂度。

由于 SDCMCR 问题同时涉及时延和费用两个度量, 所以针对该问题的算法在搜索过程中每个节点需针对约束条件保留若干个可能的路径, 导致问题的复杂度为 NPC。解决的思路是尽量减少算法需要搜索的可行路径, 例如 HM COP 算法采用仅保留满足约束条件的解的方法来降低算法的搜索空间。相关分析如下:

命题 1 对于图 G 中的任意节点 s, t 和 m , 设路 $U(s, m)$ 的时延为 x , t 到 m 的最小时延为 y ; 若 $x + y > D_r$, 则经过路 $U(s, m)$ 的所有路 $U(s, t)$ 都不满足条件 $D(U(s, t)) \leq D_r$ 。

证明是显然的, 略。

由命题 1 HM COP 算法的约束条件为 $x + y > D_r$, D_r 越小, 则被保留的可行路径越少, 算法效率越高。如图 1 所示, 其可行解区间为 $ab-gh$ 且当 D_r 逼近 D_{min} 时, 可行解区间很小, 算法效率很高,

随着 D_r 与 D_{min} 差距拉大, 算法效率下降很快, 所以该算法设计者提出此时还须强制限制每个节点保留的最大路径数来进一步降低算法复杂度, 但这样会牺牲算法的准确性和实用性。

由上面的分析可知, 可行解区间的大小将直接影响算法的运行效率, 所以可以通过进一步限制其大小来提高算法效率, 根据这个思路, 本文给出的 SDCMCR 算法作了下面两方面的改进:

(1) 在解空间内找到路 U_i 使其 $D \leq D_r$, C 尽量小。令 C 值为 C_m , 并以 C_m 为约束条件来限制可行解区间的大小。

该改进的判断规则为: 设当前路径为 $u_{s,x}$, 若经过路 $u_{s,x}$ 到目的节点 t 基于费用最小的路径为 $u_{s,t}$, 则判断 $u_{s,x}$ 是否满

足约束条件 C_m 来决定是否保留当前路径。

命题 2 对于图 G 中的任意节点 s, t 和 m , 存在一条费用为 C_m 路 $U_m(s, t)$, 设路 $u(s, m)$ 的费用为 x , t 到 m 的最小费用为 y ; 若 $x + y > C_m$, 则经过路 $u(s, m)$ 的所有路 $u(s, t)$ 都不会比路 $U_m(s, t)$ 优。

证明是显然的, 略。

由命题 1 命题 2 可保证 SDCMCR 问题的全局最优值在区间 $abef$ 内 (如图 1 所示), 改进后的算法搜索空间将大大缩小, 算法效率进一步提高。这里 C_m 的获取是算法实现的一个关键, 具体来说, 若令 C_0 为某 SDCMCR 问题的最优路径对应的费用值, 则 C_m 的值越接近 C_0 , 算法效率越高。算法在初始 C_m 值的获取上, 重点考虑如下两方面:

- 对于某一个 SDCMCR 问题, 若问题有解, 则一定有 $D_{min} \leq D_r$ 存在, 所以可令 $C_m = C_{d_{min}}$

- 若 $D_{min} > D_r$, 则可以通过修改最小费用路 p_1 (见图 1) 来获取比 $C_{d_{min}}$ 更优的 C_m 。

具体的实现参见下一节步骤 2

(2) 在算法运行过程中, 采用动态调整 C_m 的方法, 使可行解区间进一步优化。这种动态调整使算法具有很好的自适应性, 可以克服诸如 HMCOP 算法效率受 D_r 和 D_{min} 相对位置关系影响等缺点, 算法适应性大大加强。

3.2 算法描述

SDCMCR 算法的主体模块包括以下三个步骤:

第 1 步: 算法初始化。首先根据相应的业务 η 和业务到达时间 t 计算出卫星网络 G 各边的时延 d_i , 中断概率 P_{ij} 和费用 c_{ij} 。其次, 分别求 G 中任一节点 y 到目的节点 t 基于费用和时延的最优路; 判断 $D_{min}(s, t)$ 是否大于约束条件, 若大于则算法停止; 判断 $D_{min}(s, t)$ 是否满足约束条件, 若满足则算法停止; 记录 $D_{min}(y, t)$, $C_{min}(y, t)$ 和 $C_{d_{min}}(y, t)$, 以备主约束计算用。

第 2 步: 调用 C_m 求解模块, 求得算法初始的费用限定条件 C_m 。

第 3 步: 调用主约束计算模块, 计算从源节点 s 到目的节点 t 的最优路径。

C_m 求解模块采用修改最小费用路的方法获得, 模块从路的目的节点 t 开始, 依次检查每一个节点。具体实现为: 对于已检查的节点 x , 首先获得 x 的下一个节点 y ; 然后判断路径 $U(s, y)$ 的时延值与 $D_{min}(y, t)$ 之和是否满足约束条件, 若满足则判断其费用值与 $C_{d_{min}}(y, t)$ 之和是否小于 C_m , 如小于, 则替代之。具体伪码实现如下。

BEGIN

$$C_m = C_{d_{min}}(s, t)$$

$$y = t$$

WHILE ($y \neq s$) DO {

$$y = \text{NEXT_NOTE}(U(s, t), y)$$

$$\text{IF } D(U(s, y)) + D_{min}(y, t) \leq D_r$$

$$\text{AND } C(U(s, y)) + C_{d_{min}}(y, t) < C_m$$

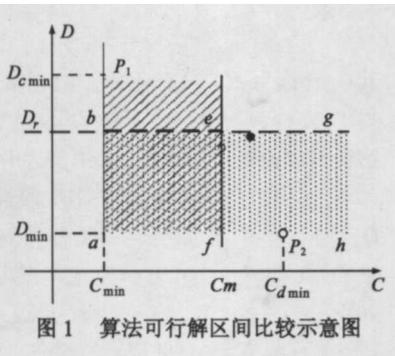


图 1 算法可行解区间比较示意图

$$C_m = C(U(s, y)) + C_{min}(y, t) \quad \}$$

END

主约束计算模块采用类似于 Dijkstra 算法的方法来进行求解. 该模块在计算中需维护一个临时路集合 Path_Set 将当前满足条件的路 (可行解) 加入该集合, 具体实现如下:

设当前的最优路为 $U(s, x)$, 对于 x 的邻接节点 y , 则 (1)检查 y 是否已在路 $U(s, x)$ 中, 若在则表示路径 $U(s, y)$ 存在环路, 该路无效; (2)计算 $U(s, y)$ 的时延值与 $D_{min}(y, t)$ 之和, 若约束超限, 则 $U(s, y)$ 不是可行解; (3)计算 $U(s, y)$ 的费用值与 $C_{min}(y, t)$ 之和, 若约束超限, 则 $U(s, y)$ 不是可行解; (4)动态调整 C_m , 对于可行解 $U(s, y)$, 计算其费用值与 $C_{min}(y, t)$ 之和, 如小于 C_m , 则替代之; (5)将 $U(s, y)$ 的时延和费用值与节点 y 已记录的时延和费用值相比, 若不都大于或等于则将路加入 Path_Set 和 $V_p(y)$.

模块确定当前最优路径的原则是: Path_Set 中费用最小的路, 若同时有几条存在, 则判断这几条路的时延与该路最外一点 y 到 t 的最小时延值之和, 取最小的那条作为当前最优的路径. (模块详细伪码见图 2)

```
Main_Calculation(G(V, E), D, C, s, t, D_r)
```

```
1 BEGIN
2 Initialize(Path_Set, V_p(i), i ∈ V)
3 WHILE (Path_Set ≠ ∅) DO
4 U(s, x) = Remove_Min_Path(Path_Set)
5 IF (x = t) THEN RETURN U(s, x)
6 FOR y ∈ Γ_x DO
7 IF Loop_Det(U(s, x), y) FAIL THEN CONTINUE
8 IF D(e(x, y)) + D(U(s, x)) + D_min(y, t) > D_r THEN CONTINUE
9 IF C(e(x, y)) + C(U(s, x)) + C_min(y, t) > C_m THEN CONTINUE
10 IF C(e(x, y)) + C(U(s, x)) + C_min(y, t) < C_m THEN
11 C_m = C(e(x, y)) + C(U(s, x)) + C_min(y, t)
12 IF Check_Path_Vp(V_p(y), U(s, y)) FAIL THEN
13 Inset_Path(Path_Set, U(s, y))
14 Inset_Path_Vp(V_p(y), U(s, y))
15 ENDFOR
16 ENDDO
17 END
```

图 2 主约束计算模块伪码

3.3 算法性能分析

算法的计算复杂性主要体现在第 1 步和第 3 步, 其中第 1 步需要用两次 Dijkstra 算法, 其计算复杂度为 $O(|V|^2)$. 第 3 步的计算复杂度与算法的搜索空间有关, 假设该空间的节点数目为 n , 节点的最大邻居数目为 E_m , 每个节点保留的最大路径数目为 L_m , 则主约束计算模块最多进行 $(n - 2)L_m$ 次主循环. 每次主循环需要最多 E_m 次副循环, 每次副循环中最耗时的是图 2 的第 13 行, 需要最多 $(n - 2)L_m$ 次比较, 所以该模块的计算复杂度为 $O(E_m n^2 L_m^2)$.

整个算法的计算复杂度为 $O(|V|^2 + E_m n^2 L_m^2)$.

由上面的分析可知, 主约束计算模块的计算复杂度主要与 n 和 L_m 有关. 由于算法在保证最优解的同时有效的降低需要搜索的空间, 此时的 n 和 L_m 一般很小, 表 1 中给出了在节点数 $|V|$ 为 100~1000 时不同规模的网络, 本算法和 HMCOP 算法的主约束计算模块在最坏情况下所需要的最多比较次数, 取得该数据的测试条件为: 利用 Transit-Stub 模型^[8]随机生成不同规模的网络, 每次测试针对十个不同的拓扑, 10000 次不同的请求, 最终结果为得到的 10000 个数据中的最大一个.

表 1 两种算法主约束计算模块所需最多比较次数表 (万次)

$ V $ 个	100	200	400	500	600	700	800	1000
HMCOP	7.5	34.7	153	233	358	432	636	1040
SDCOP	0.9	3.4	10.5	13.2	14.5	17.1	20.4	23

上表可看出在最坏的情况下, 本算法主约束计算模块所需的最多比较次数约等于节点数的平方, 与经典的 Dijkstra 算法计算复杂性大体相当, 算法在计算复杂度方面比 HMCOP 算法改进了近一个数量级. 详细仿真结果见下节.

4 仿真分析

本节给出了 SDCMCR 在算法运行效率和性能方面的仿真测试, 并对比多种同类型的已有算法. 其设计如下:

(1)实验选用美国 Teledesic 移动卫星网轨道参数, 共 288 颗卫星均匀分布在 12 个轨道面, 轨高 1375km, 倾角 84.7°, 每颗星周围有 8 颗邻居, 为十字架状分布; (2)业务 η 为简单流, 其平均业务持续时间 $\mu = 3$ 分钟, 则 G 中边 $e(i, j)$ 的中断概率可表示为: $P_{\eta} = e^{-(TL_{\eta}^A)}$; (3)网络节点的排队过程为最简单流损失制模式, 业务报平均服务时间为 $\ln s$ 则两节点间的时延可近似表示为: $(x/c + 1/(1 - \alpha))m s$ 其中 x 为节点间距离, c 为光速; (4)针对该卫星网每次实验都随机产生 10000 次不同的请求, 每次请求包括: 请求发生的时间 t_s 源宿节点对、时延约束值和费用 c 其中源宿节点是在所有节点中随机选取, 而 t 在 $[0, T]$ 中随机选取, c 在 $[5, 10]$ 中随机产生. 每组实验针对不同的测量内容可得到 10000 个数据, 最终的结果取这些数据的均值. 仿真测试分为如下三组.

4.1 算法运行效率和性能比较

本组实验给出了本算法和 HMCOP 算法在时间、空间复杂度和算法性能三方面的对比结果. 考虑到在 $D_{min} > D_r$ 和 $D_{min} \leq D_r$ 时, 两种算法都只执行了第一步, 在耗时和性能方面没有区别, 实验中的时延约束值在 $[D_{min}, D_{max}]$ 内随机取得. 由于 HMCOP 在运行中每个节点保留的最大路径数 V_{pr} 将直接影响算法的效率和性能, 实验分别给出 $V_{pr} = 1, \dots, 20$ 时的结果.

图 3 中 T_r 表示算法的耗时比, $T_r = (\text{HMCOP 主约束计算模块耗时}) / (\text{SDCMCR 主约束计算模块耗时})$, 主要考虑

算法主约束计算模块的耗时情况,其大小等于每次调用相应模块时,调用前后的计算机时间差。 M_r 表示两种算法所需的储存空间比,主要考虑算法在计算过程中所需要的临时队列的大小, $M_r = (\text{HMCOP 临时队列大小}) / (\text{SDCMCR Path. Set} + V_p \text{大小})$ 。 M_c 表示两种算法寻路性能比,比较两种算法得到的路径的费用,费用越小的得到的路径越优, $M_c = (\text{SDCMCR 得到费用小于 HMCOP 的次数}) / 10000$

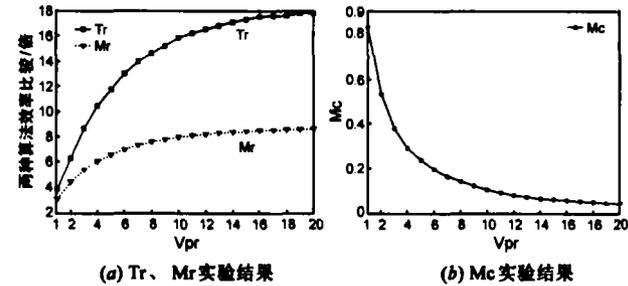


图 3

由图 3 可得当 $V_{pr} = 1$ 时, HMCOP 的效率接近 SDCMCR ($Tr = 3.8$), 但此时 SDCMCR 有 83% 的解优于 HMCOP, 随着 V_{pr} 的变大, 存在两种相反的变化趋势: Tr 迅速上升 (图 3(a)) 而 M_c 迅速下降 (图 3(b)), 在 $V_{pr} = 12$ 时 Tr 在 17 附近趋于平稳, 而 M_c 也趋于零。结果表明 SDCMCR 在算法计算复杂度方面有明显改善, 同时在空间复杂度方面也得到了很好改善 ($M_r = 8$ 左右), 所以, SDCMCR 比 HMCOP 更能适应星上设备处理能力有限的实际情况, 可以用于星上在线 QoS 路由计算。

4.2 业务阻塞率以及中断率比较

本组实验给出了本算法和 PRP 算法、未考虑链路切换的路由算法 (如 GALPEDA 算法) 在不同网络负荷率下, 业务因切换而被中断的概率以及业务请求被阻塞情况的比较结果。实验中的时延约束值 $D_r = 100ms$ $\gamma = 5$ PRP 算法的路径中断概率门限值为 0.1

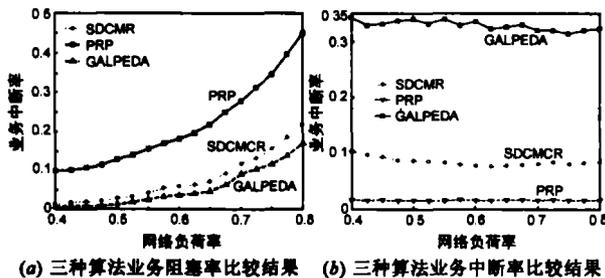


图 4

由图 4(a) 可得随着网络负荷率的增加, 三种算法业务请求的阻塞概率都呈上升趋势, 不过很明显, SDCMCR 在这方面的性能要明显好于 PRP, 与 GALPEDA 大体相当。这是因为 PRP 在计算时会剪切掉中断概率高的链路, 使得业务过于集中于某些链路而造成较高的阻塞概率, 同时这也会带来网络流量分布不均, 网络整体利用率低等缺点, 而 SDCMCR 算法则可克服这些缺点, 且该算法中业务因网络

切换而中断的概率要远远小于没有考虑未来拓扑切换情况的 GALPEDA (图 4(b)), 因而可有效降低切换对业务 QoS 性能的影响。结果表明 SDCMCR 算法能很好的兼顾网络业务阻塞率和业务中断率两方面的性能, 尤其适合于切换频繁的移动卫星网络。

图 4(b) 还可看出随着负荷率 α 的增加, 业务中断率呈下降趋势, 这也有利于算法在较高网络负荷时的使用。

4.3 γ 值的选取对业务阻塞率和中断率的影响

γ 表示业务对路径可靠性的敏感程度, 它与具体的业务有关。本组实验给出了 γ 值与业务阻塞率和中断率的关系, 由此可得出不同的业务 γ 值可能的取值。实验中的时延约束值 $D_r = 100ms$ $\alpha = 0.5$

由图 5 可得

随着 γ 值的增加, 算法中业务因网络切换而中断的概率将逐渐下降, 当 $\gamma = 9$ 后, 中断率逐渐稳定在 0.04 左右, 接近 PRP 的中断率, 与此同时, 业务请求

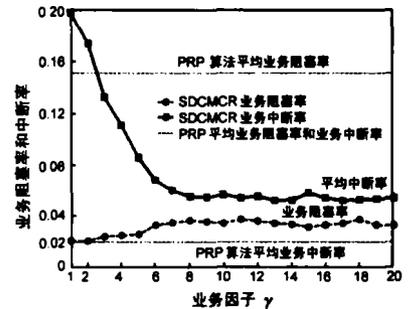


图 5 γ 值与业务中断率和阻塞率的关系

的阻塞概率也略有上升, 但仍远远好于 PRP。实验结果表明, 对于某些业务 (如数据业务), 网络可以通过灵活的路由策略来降低切换对业务的影响, 此时算法的 γ 可取较小的值 (如 2 或 3), 以牺牲路径的稳定性来获取其他方面的利益 (如阻塞率、网络流量更均衡等); 而对于某些业务 (如语音业务), 链路切换需要网络采用信道预留等高代价方式来平滑, 此时算法的 γ 可取较大的值 (如 6 或 7), 以尽量降低业务的中断概率。

5 结论

QoS 路由技术在宽带卫星通信网络中具有十分重要的意义, 一直以来都没能得到很好的解决。本文给出了 SDCMCR 算法来解决在卫星通信网络链路时延延长的情况下寻找满足时延限制要求的最优路径问题。

SDCMCR 算法充分考虑了卫星网络的拓扑时变特性, 给出了业务的费用模型, 该模型将链路切换对业务的 QoS 以及网络的影响折合为该业务在链路上的额外费用, 并通过引入了业务的中断概率来预测卫星网络的动态性。利用该模型, 算法通过寻找满足时延限制要求的最小费用路径来有效降低链路切换对业务的 QoS 以及网络流量分布的影响。该算法采用自适应参数设计、有效减小搜索空间的方法来有效降低算法的复杂度, 以适应算法对星上计算和储存能力的要求, 仿真表明该算法不仅在计算复杂度方面优于 HMCOP 算法, 且能很好的兼顾网络业务阻塞率和业务中断率两方面的性能。一方面算法中业务因切换而中断

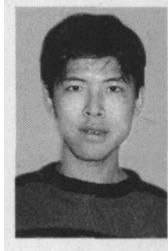
的概率大大下降, 另一方面在业务请求被阻塞的概率和网络流量均衡性等方面本算法也优于同类型的 PRP 算法。

参考文献:

- [1] Ibnkahk M, Rahman Q M, Sulym an A I, etc High-speed satellite mobile communications Technologies and challenges [J]. Proceedings of the IEEE, 2004, 92(2): 312- 339.
- [2] Monahan G, Septiawan R, Sugden S A multiservice traffic allocation model for LEO satellite communication networks [J]. IEEE Journal on Selected Areas in Communications, 2004, 22(3): 501- 507
- [3] Gounder V V, Prakash R, Abu-Amara H. Routing in LEO-based satellite networks[A]. Wireless Communications and Systems 1999 Emerging Technologies Symposium [C]. Richardson: IEEE, 1999, 22: 1- 22, 6
- [4] Uzunaloglu H. Probabilistic routing protocol for low earth orbit satellite networks [A]. ICC' 98 [C]. Atlanta: IEEE, 1998, 89- 93.
- [5] Jukan A, Hoang Nam Nguyen, Franzl G. QoS-based routing methods for multihop LEO satellite networks[A]. ICON 2000[C]. Singapore: ICON, 2000, 399- 405.
- [6] Cheng S, Nahstedt K. On finding multi-constrained paths [A]. ICC' 98[C]. Atlanta: IEEE, 1998, 874- 879.
- [7] Kokmaz T, Kunz M. Multi-constrained optimal path selection [A]. Proc of the INFOCOM 2001 [C]. Anchorage: IEEE, 2001, 834- 843.

- [8] Calvert K I, Doar M B, Zegura E W. Modeling internet topology [J]. IEEE Communications Magazine, 1997, 35(6): 160- 163.

作者简介:



张 涛 男, 1973 年生于山东文登市, 现为北京航空航天大学民航数据通信及新航行系统重点实验室博士研究生, 主要研究方向为移动通信, 卫星网络通信. E-mail: zhtao73@163.com



张 军 男, 1965 年生于安徽合肥, 博士学位, 现为北京航空航天大学电子信息工程学院院长, 博士生导师, 研究方向为天空地一体化网络通信。



柳重堪 男, 1941 年生, 博士生导师, 研究方向为信息传输与处理, 信号与系统中的反问题。