

适于低轨卫星 IP 网络的特定源组播算法

程连贞, 刘 凯, 张 军, 宋学贵

(北京航空航天大学电子信息工程学院, 北京 100083)

摘 要: 为了解决低轨卫星 IP 网络中现有特定源组播算法的信道资源浪费问题, 本文提出了一套新的特定源组播算法, 即基于核心群的特定源组播算法 (CSSM) 和加权的 CSSM 算法 (w -CSSM)。CSSM 算法以源节点作为初始核心群, 通过核心群和剩余组成员的最短路径方法逐步扩展直至整棵组播树构建完成, 所得的树代价最小, 从而大大提高了网络的带宽利用率和传输效率。在 w -CSSM 算法中, 加权因子可以自适应调整以适度增大树代价, 降低端到端传播时延, 以支持某些有严格端到端时延要求的实时组播业务。通过与低轨卫星 IP 网络中典型特定源组播算法 MRA 的仿真比较, 可以看出 CSSM 和 w -CSSM 算法的树代价性能比 MRA 有较大改善, 不过端到端传播时延略高。

关键词: 卫星 IP 网络; 低轨; 路由; 组播; 特定源组播

中图分类号: TN927. + 3

文献标识码: A

文章编号: 0372-2112 (2007) 06-1050-06

Source-Specific Multicast Algorithms for LEO Satellite IP Networks

CHENG Lian-zhen, LIU Kai, ZHANG Jun, SONG Xue-gui

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

Abstract: To resolve the channel resources waste problem of current source-specific multicast (SSM) routing algorithm in low earth orbit (LEO) satellite IP networks, a new SSM algorithm called core-cluster-based source-specific multicast (CSSM) algorithm and its improved one (the weighted CSSM algorithm) are presented. The CSSM algorithm takes source node as initial core-cluster, and extends it step by step to construct entire multicast tree with the lowest tree cost by a shortest path scheme between newly-generated core-cluster and surplus group member nodes, which can greatly improve transport bandwidth utilization and transport efficiency. In the w -CSSM algorithm, end-to-end propagation delay can be decreased and meanwhile tree cost will be increased a bit by adaptively adjusting the weighted coefficient to meet strict end-to-end delay requirements of some real-time multicast applications. Finally, performance comparison of the two algorithms with the typical SSM algorithm (i. e. MRA) in LEO satellite networks is provided. Simulation results show that their tree cost performance is greatly better than that of MRA at the expense of a bit higher end-to-end propagation delay.

Key words: satellite IP networks; LEO; routing; multicast; source-specific multicast (SSM)

1 引言

组播技术有效地解决了单点发送多点多跳接收和多点发送多点多跳接收的问题, 源节点尽可能地沿公共共享路径向多个目的节点同时发送同一分组, 分组只在分支节点处被复制, 在每条链路上只转发一次, 从而最大程度地利用了网络资源^[1]。

无线信道广播和覆盖范围广的特性使得卫星网络非常适合组播通信。相比高轨 (GEO) 和中轨 (MEO) 卫星网络, 低轨 (LEO) 卫星网络的时延小, 可以支持实时多媒体组播业务如视频会议等^[2~4]。不过, 星上资源稀缺的特点给它们的应用带来了一定的局限性, 从而节约资

源的组播技术在卫星网络中具有广阔的应用前景。

一般地, 组播算法分为源 (source-based) 和共享树 (shared tree) 两类, 其中源组播又可分为任意源组播 (ASM) 和特定源组播 (SSM)^[5]。SSM 实际上是 ASM 的一个分支, 同时又简化了 ASM 的组播模型, 它使用源地址 S 和组地址 G 即 (S, G) 共同标识组播会话。

地面网络中传统的基于源的组播算法 (如 DVMRP、MOSPF 等) 和共享树算法 (如 CBT、PIM-SM 等) 只适用于固定或者拓扑缓慢变化的网络, 在高动态网络中, 节点的快速移动会产生很大的交换和控制开销。并且, 卫星运动的可见性、周期性和规则性以及星上资源稀缺的特点, 都使得卫星网络需要专用、简单、高效的组播算法。

收稿日期: 2006-01-04; 修回日期: 2006-10-30

基金项目: 国家自然科学基金 (No. 60532030, 10377005, 10577005); 航天科技创新基金

目前,已提出的专用于低轨卫星 IP 网络的组播算法主要有 MRA (multicast routing algorithm)^[3]、RST (rectilinear steiner trees)^[4]等.其中 RST 算法使用整数线性规划方法求解最小 RST 树,使组播路由的带宽使用最少,适合非实时的组播应用,但计算过程较为复杂;在 MRA 中,源和中间卫星节点使用数据报路由算法(DRA)^[6]计算每个目的卫星节点的下一跳路由方向,得到的端到端传播时延较小,适合实时多媒体业务;不过,由于仅利用当前树上节点到目的节点的单跳(one-hop)局部信息(即下一跳路由方向)合并路径,得到的组播树树代价很高.而组播的基本问题是找到一棵代价低的组播树,即如何使树上节点(on-tree nodes)尽量共享传输路径.定义树代价为组播树的根(源或核)到目的节点的所有链路代价的总和,它反映了对网络信道资源(即传输带宽)的占用情况,如果每条链路的代价是 1,树代价即为组播树上的总链路数.图 1 是一个颇具代表性的 MRA 组播树实例,采用 12 × 12 的栅格结构(总计有 144 颗卫星节点、264 条链路)表示简化的低轨卫星 IP 网络的拓扑,星际链路(ISLs)的长度依据文献[3]和[6]设置.从图中可以看出,MRA 组播树的链路共享性很差,一个 30 节点的组播组需要 92 条链路、63 个中间节点转发分组,对网络资源的耗费很大.

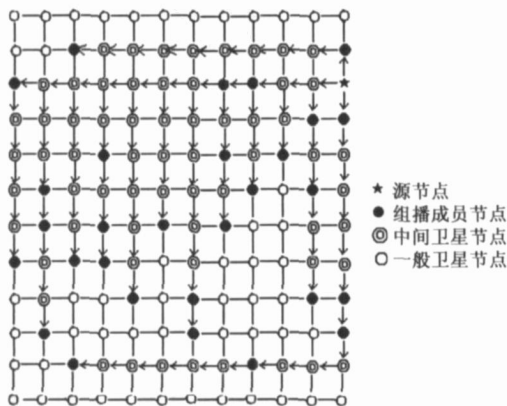


图 1 MRA 特定组播树实例

为了解决 MRA 的信道资源浪费问题,又保留其简单性,本文提出基于核心群的特定源组播(core-cluster-based source-specific multicast, CSSM)算法,以源节点作为初始核心群,通过核心群和剩余组成员的最短路径方法逐步扩展直至整棵组播树构建完成,旨在使树代价最小,提高网络的信道资源利用率和传输效率.并且,为了支持有严格端到端时延要求的实时组播业务,本文还设计了加权的 CSSM(简称 w-CSSM)算法,所提出的加权因子可以自适应地在组播树的树代价和端到端传播时延之间作折衷.

本文其他部分的安排如下:第二部分介绍了低轨卫星网络的星座模型和相关假设;第三部分详细描述

了 CSSM 和 w-CSSM 算法,提出了它们之间的自适应转换方法和加权因子的自适应调整策略,并介绍了动态组成员关系维护机制和组播树更新方法,分析了算法的复杂度和适用性;第四部分对算法性能进行了仿真评估;最后总结全文.

2 低轨卫星网络的星座模型和相关假设

假设低轨卫星网络由 P 个轨道组成,每个轨道上有 S 颗卫星,是 $P \times S/P/0$ 的极轨道星座类型,并且每颗卫星与相邻卫星之间有 4 条 ISLs 相互连接,其中 2 条是轨间 ISLs,另外 2 条是轨内 ISLs.轨内 ISLs 的长度相等且固定不变,轨间 ISLs 的长度随卫星地理位置的改变而变化.另外,由于极轨道星座在两极附近的卫星过于密集,常常关闭一些轨间 ISLs,假定所连接的两颗卫星的星下点纬度都大于 75°时,关闭该轨间 ISLs.

CSSM 和 w-CSSM 算法基于逻辑位置概念^[3,6]构建组播树.假设星座内不同轨道上的卫星是初始对齐的,分别将初始时刻的卫星轨道和轨道上的卫星按照特定的次序编号,使用 p 标识轨道编号($0 \leq p \leq P-1$)、 s 标识卫星编号($0 \leq s \leq S-1$),则 (p, s) 代表卫星节点.根据卫星的覆盖范围,将星座覆盖区划分为与卫星个数相等的逻辑位置,逻辑位置与卫星节点一一对应,卫星的星下点位于对应的逻辑位置中心,逻辑位置形成的虚拟静态、结构规则的低轨卫星网络拓扑如图 2 所示.在路由计算时,将逻辑位置作为跳,则星上路由表只与逻辑位置相关,屏蔽了卫星的移动性,简化了星上路由.卫星移出逻辑位置时必须移交路由表,移出的卫星发送它的路由表到轨道内移进的后续卫星,并从轨道内前序卫星接收新的路由表.

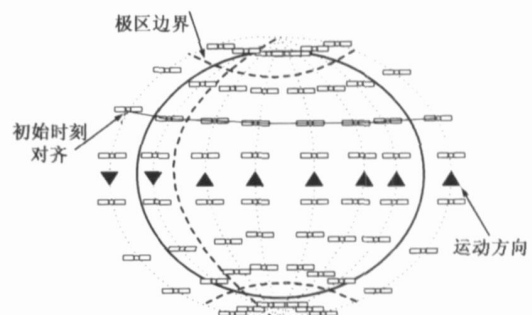


图 2 低轨卫星网络的星座模型

假设低轨卫星 IP 网络中的组播组 G 共有 N 个成员节点, $G = \{g_s, g_1, g_2, \dots, g_i, \dots, g_{N-1}\}$,其中 g_s 是源节点, $g_i (i = 1, \dots, N-1)$ 是非源成员节点,定义 $G_0 = \{g_1, g_2, \dots, g_{N-1}\}$.

3 CSSM 和 w-CSSM 算法

CSSM 和 w-CSSM 算法基于组播组成员的逻辑位置信息计算连接全部成员的低树代价组播树.

3.1 相关概念和规则

(1) 最短路径

定义两节点间的各条路径中所经跳数最少的路径为最短路径。

(2) 最近距离路径

定义两节点间的各条路径中长度最短的路径为最近距离路径。

(3) 最近距离最短路径

如果计算得到的最短路径不止一个,则将链路长度最短的最短路径称为最近距离最短路径。

(4) 组播路径决策方法

对于网络中点到多点的组播通信,需要构建某节点 n_0 到某节点集 H 内各节点的组播路径,首先计算 n_0 与 H 中各节点之间的最短路径并比较,选择其中的最近距离最短路径作为连接 n_0 与 H 的组播路径;然后将组播路径上的全部节点构成节点集 K , H 除去 K 中所有节点后剩余的节点构成节点集 H' , K 与 H' 之间的组播路径构建方法如下:计算 K 中每个节点与 H' 各节点之间的最短路径并比较,选择最近距离最短路径作为 K 与 H' 之间的组播路径。

(5) 核心群(core-cluster)

在构建组播路径的过程中,将源节点和已经连接的成员节点以及连接路径上的中间节点构成的节点集称为核心群。核心群内节点数随组播路径的扩展而增加,初始核心群 C_0 只包括源节点。

3.2 组播树构建过程

下面介绍核心群合并方法构建组播树的过程:

(1) 使用 DRA 计算^[6]得到集合 G_0 与初始核心群 C_0 之间的最近距离最短路径和其上属于 G_0 的目的成员节点 g_1 以及相应中间节点(记为 $g_{1a}', g_{1b}', \dots$),以它作为新的组播路径,将 g_1 、中间节点与 C_0 合并得到一个新的核心群 $C_1 = \{g_s, g_1, g_{1a}', g_{1b}', \dots\}$ 。令 $C_0' = \{g_1\}$, $i = 1$ 。

(2) $G_i = G_{i-1} - C_{i-1}'$, C_i 为 G_i 与 C_i 的交集。 C_{i+1} 的具体产生方法是:计算 G_i 与 C_i 之间的最近距离最短路径和相应属于 G_i 的目的成员节点 g_{i+1} 以及中间节点(记为 $g_{(i+1)a}', g_{(i+1)b}', \dots$),连接它作为新的组播路径,将 g_{i+1} 、中间节点与 C_i 合并后,得到新核心群 $C_{i+1} = \{g_s, g_1, g_{1a}', g_{1b}', \dots, g_2, g_{2a}', g_{2b}', \dots, g_{i+1}, g_{(i+1)a}', g_{(i+1)b}', \dots\}$ 。

(3) 令 $i = i + 1$,继续步骤(2)直到 $G_i = \Phi$ 即成员节点全部合并到核心群内为止,此时整棵组播树全部构成。

为了便于理解,图3示例了一个15节点组播组的组播树构建方法和步骤(以数字序号标识构建步骤), $G = \{g_s, g_1, g_2, \dots, g_{14}\}$, $g_s = (1, 5)$ 。假设星座包括12个

轨道、每个轨道上有24颗卫星,组播组成员均匀分布在东半球。轨道编号 p 按照从西到东的顺序标记,卫星编号 s 按照从南到北的顺序标记。假设轨道高度是

1375km,依据公式 $L_v = \sqrt{2} R \sqrt{1 - \cos\left(\frac{360^\circ}{S}\right)}$ 和 $L_h = \alpha \cos(lat)$ 分别计算轨内和轨间 ISLs 的长度,其中 $\alpha = \sqrt{2} R \sqrt{1 - \cos\left(\frac{360^\circ}{2 \times p}\right)}$, lat 是卫星节点的当前纬度。下面详细介绍构建步骤:

$G_0 = \{g_1, g_2, \dots, g_{14}\}$, 计算 $C_0 = \{g_s\}$ 到 G_0 中各节点的最短路径并比较它们的跳数,得到 g_1, g_2, g_5 与 g_s 之间的最短路径的跳数最少;分别计算它们的链路长度,得知 g_1, g_2 与 g_s 之间的最短路径均是最近距离最短路径。随机选取其中之一(假设是 g_1 与 g_s 之间的最短路径)作为组播路径,并将 g_1 、中间节点(1,4)以及 C_0 合并,得到 $C_1 = \{g_s, g_1, (1, 4)\}$, $C_0' = \{g_1\}$, $i = 1$ 。

$G_1 = G_0 - C_0' = \{g_2, g_3, \dots, g_{14}\}$, 计算 G_1 中的节点与 C_1 中各节点之间的最短路径并比较它们的跳数,得到 g_2 距离中间节点(1,4)的跳数最少,连接 g_2 和(1,4)作为新的组播路径,得到 $C_2 = \{g_s, g_1, (1, 4), g_2\}$, $C_1' = \{g_1, g_2\}$ 。

$i = 2$, $G_2 = G_1 - C_1' = \{g_3, g_4, \dots, g_{14}\}$, 计算 G_2 中的节点与 C_2 中各节点之间的最短路径并比较它们的跳数,得到 g_5 距 g_s 和 g_2 均为2跳,再比较它们的链路长度,得到 g_5 距 g_s 更近,连接这条最近距离最短路径作为组播路径,并将 g_5 和中间节点(2,5)与 C_2 合并,得到 $C_3 = \{g_s, g_1, (1, 4), g_2, g_5, (2, 5)\}$, $C_2' = \{g_1, g_2, g_5\}$,

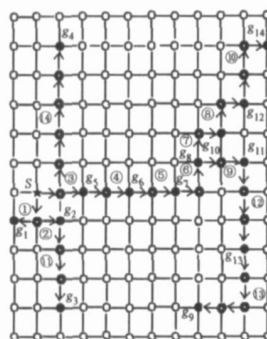


图3 CSSM组播树构建示例

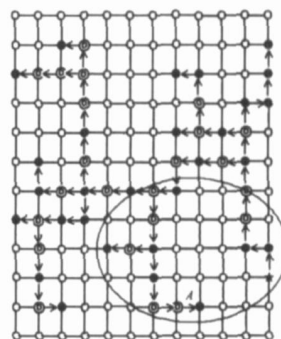


图4 “组播路径绕远”现象

$i = 13$, $G_{13} = \{g_4\}$, 计算 g_4 与 C_{13} 中各节点之间的最短路径并比较它们的跳数,得到 g_4 与(2,5)之间的最短路径是最近距离最短路径,以它作为组播路径,并将 g_4 和中间节点(2,6)、(2,7)、(2,8)、(2,9)与 C_{13} 合并,得到 $C_{14} = \{g_s, g_1, (1, 4), g_2, g_5, (2, 5), g_6, (4, 5), g_7, (6, 5), g_8, (8, 5), g_{10}, (8, 7), g_{12}, (9, 8), g_{11}, (9, 6), g_{14}, (9, 10), (10, 10), g_3, (2, 2), (2, 3), g_{13}, (10, 5), (10,$

4), $g_9, (10, 2), (10, 1), (9, 1), g_4, (2, 6), (2, 7), (2, 8), (2, 9)\}$.

(14) $i = 14, G_{14} = \Phi$, 此时组播树构建完成.

3.3 组播树性能均衡措施

由于核心群合并方法仅注重使树上节点最大可能地共用传输路径, 没有充分考虑路径长度对时延性能的影响, 致使源和附近某些节点之间的组播路径可能需要绕过多个公共中间节点, 致使分组的端到端传播时延增大. 图 4 是典型的 CSSM 组播树实例图, 椭圆内距源较近的部分节点 (如 A) 为了共用组播路径多绕了多个中间节点, 产生“组播路径绕远”现象. 为了减轻乃至避免这种现象, w -CSSM 算法将加权的方法引入核心群合并机制, 即使用加权因子 $w (0 < w < 1)$ 获得折衷.

已知集合 G_i 中某节点 g_i , 假设 g_i 到核心群 C_i 的最近距离最短路径的链路数是 l_i , g_i 到源节点 g_s 的组播路径链路数是 l_i' , 可知 l_i 反映了构建新的组播路径时树代价的增长情况, l_i 反映了源节点与目的节点之间的端到端传播时延性能. 令 $W_i = l_i \times w + l_i' \times (1 - w)$, 即 W_i 是 l_i 和 l_i' 以 w 加权的加权和.

已知 C_0 和 G_0 , 令 $i = 0$. 下面说明详细步骤:

(1) 令 $j = 0$, 从 G_i 中随机选取 1 个成员节点记为 $s_1, G_j' = G_i - \{s_1\}$.

(2) 从 G_j 中随机选取 1 个成员节点记为 $s_2, G_{j+1}' = G_j - \{s_2\}$, 计算将节点 s_1 和 s_2 分别以最近距离最短路径连接到 C_i 需增加的链路数 l_1 和 l_2 , 然后计算沿已有组播路径到 g_s 的链路数 l_1' 和 l_2' , 得到下式:

$$W_1 = l_1 \times w + l_1' \times (1 - w) \quad (1)$$

$$W_2 = l_2 \times w + l_2' \times (1 - w) \quad (2)$$

(3) 比较 W_1 和 W_2 , 若 $W_1 > W_2$, 令 $s_1 = s_2$; 否则不作处理.

(4) 判断 G_{j+1}' 是否为空集, 如果不为空则令 $j = j + 1$, 继续执行步骤 (2) 和 (3); 否则将最终得到的节点 s_1 (记为 g_{i+1}') 以最近距离最短路径连接到核心群 C_i , 并将其上的中间节点 (记为 $g_{(i+1)a}', g_{(i+1)b}', \dots$) 和 s_1 与 C_i 合并为新核心群 C_{i+1} . $G_{i+1} = G_i - C_i', C_i'$ 为核心群 C_{i+1} 中属于 G_0 的成员节点集合, 即 $C_i' = \{g_1', g_2', \dots, g_{(i+1)'}\}$.

(5) 令 $i = i + 1$, 依次执行步骤 (1)、(2)、(3)、(4) 直到集合 $G_i = \Phi$ 为止, 此时整棵组播树全部构成.

3.4 CSSM 和 w -CSSM 算法的自适应转换

CSSM 和 w -CSSM 算法之间的自适应转换依据下述原则: 如果组播会话对端到端传播时延有一定限制, 首先判断 CSSM 组播树是否满足要求: 是则使用 CSSM 组播树; 否则构建 w -CSSM 组播树, 以通过树代价的适度增加换取较低的端到端传播时延.

另外, 加权因子 w 可依据下述策略自适应选取: 设定步长, 递减计算 w 值并计算相应的组播树, 直到得到的组播树满足要求, 即当前的 w -CSSM 组播树可用. 举例如下: 设步长为 0.2, 则 w 值依次是 0.8、0.6、0.4、0.2, 首先计算 0.8 对应的组播树是否满足要求, 如否则计算 w 为 0.6 时是否满足要求, 依次计算直到选定一棵符合要求的组播树为止. 加权因子的步长可根据实际情况灵活设置, 步长越小计算量越大, 步长大时得到的组播树可能非最优.

为了评估加权核心群合并方法的性能, 分别取 w 为 0.8 和 0.4, 记相应的 w -CSSM 算法为 w_1 -CSSM 和 w_2 -CSSM, 图 5 与图 1 的组成员分布和源节点相同. 从图中可以看出, 图 5(a) 中组播树的路径分布非常紧凑, 树代价较图 1 中的 MRA 组播树有很大改善, 不过在节点 B 和 C 处仍出现了“组播路径绕远”现象; 图 5(b) 中组播树的路径分布紧凑, 树代价比图 1 中的 MRA 组播树有较大改善, 比图 5(a) 中的组播树稍差, 基本上没有出现“组播路径绕远”现象. 综上可知 w 值大时树代价性能更好.

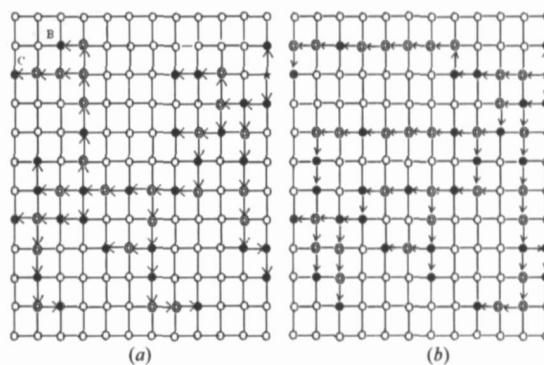


图 5 w -CSSM 组播树实例

3.5 动态组成员关系的维护和管理

为了支持组成员的随机加入和退出, 需要维护和管理动态组成员关系, 包括下述 3 种操作.

成员加入: 要加入组播组的成员发送“加入请求”到源节点, 源回复“加入确认”, 并以现有组播树上所有节点组成的集合作为核心群, 计算加入节点到核心群的最近距离最短路径并连接到组播树上.

成员退出: 如果有成员节点想退出, 则向源节点发送“退出请求”报文, 在收到源的“退出确认”报文后成功退出. 如果退出成员之前同时也是中间节点, 则仍继续转发分组; 如果之前是叶子节点, 则源节点直接将其从成员列表中删除, 并删除冗余路径.

组播树更新: 当成员加入和/或退出的次数超过门限值 δ 时, 启动组播树更新操作, 即重新构建组播树.

3.6 计算复杂度和适用性

CSSM 和 w -CSSM 算法的计算复杂度为 $O(N^3)$, MRA 的计算复杂度为 $O(LN)$, 其中 L 为所构建组播树

上的节点总数 ($L > N$), 所以 (w)-CSSM 算法的复杂度稍高于 MRA.

在移动自组织网络中, 基于位置信息的组播协议可以自适应网络拓扑变化, 提高传输可靠性, 不需维护整个网络的路由表, 具有良好的可扩展性, 所以成为未来路由协议发展的一个方向^[7]. CSSM 和 (w)-CSSM 算法基于逻辑位置构成的虚拟、静态的网络拓扑构建组播树, 与移动自组网中基于位置信息的路由机制相似, 如 GRID 协议^[8]将网络覆盖的地理区域划分成多个虚拟网格 (grid), 从网格内选取出一个节点作为网关, 负责向邻居网关节点和本地节点转发分组. 由于 (w)-CSSM 算法和 GRID 协议均使用地理位置划分的方法构建虚拟的网络拓扑, 并且逻辑位置与网格是可以对应的, 所以 (w)-CSSM 算法可用于移动自组网环境中作为骨干组播路径的构建方案. 不过, 由于移动自组网的节点运动通常不像卫星节点具有一定的规律性, 因此它们需要不同的移交策略, 比如当某个网络中的网关节点移出或失效时, 可以立刻选取该网格中的另一节点作为网关并移交路由表.

4 性能仿真和评估

假设低轨卫星星座包括 12 个轨道、每个轨道上有 24 颗卫星, 组播组成员均匀分布在东半球, 仿真的组播组成员数目在 8 到 30 之间.

为了评估算法性能, 本文仿真了下述 4 种组播组成员分布情况. 场景 1 是远端分布, 源节点位于半球内两个边缘轨道相交的顶点上, 组成员均匀分布在源节点对角的半边网络内; 场景 2 是近端分布, 源节点位于半球内两个边缘轨道相交的顶点上, 组成员均匀分布于源节点所在的半边网络; 场景 3 是均匀分布, 以半球网络内某个指定的中心节点为源, 组成员均匀分布在整个半球网络; 场景 4 是中心分布, 以某个指定的中心节点为源, 组成员均匀分布在网络内距源节点半径为 3 跳之外的 4 个角落区域. 图 6 中示例了上述 4 种场景, 场景 3 和 4 中小圆内的 4 个节点均可作为中心节点, 另外, 场景 4 中的矩形是以小圆内左上角中心节点作为源时目的成员节点分布的边界.

定义以下 4 种归一化的组播树性能参数以有效评估算法: 即 MRA 与 (w)-CSSM 组播树的树代价的比值, 记为 t_c ; MRA 与 (w)-CSSM 组播树树长 (即组播树上所有链路长度的总和) 的比值, 记为 t_l ; MRA 与 (w)-CSSM 组播树的中间节点数的比值, 记为 n ; 所有成员节点均可作源时, MRA 与 (w)-CSSM 组播树中的源和目的节点对的平均端到端传播时延平均值的比值, 记为 d .

图 7~10 是上述场景的性能仿真结果. 从图 7 可以看出, 树代价性能最好的是远端分布场景时的 CSSM 算法,

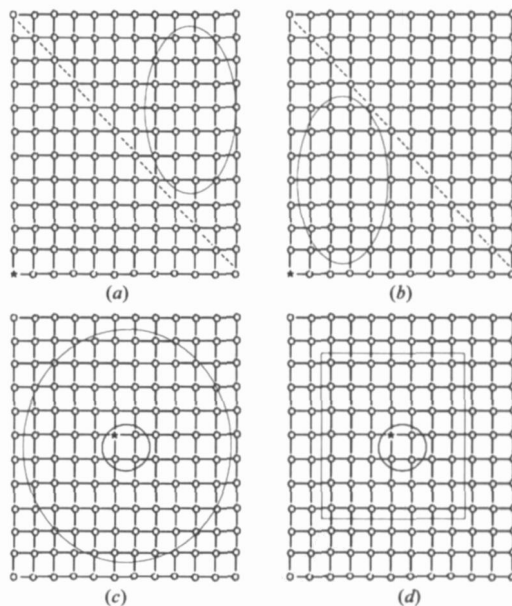


图 6 仿真的各种分布场景

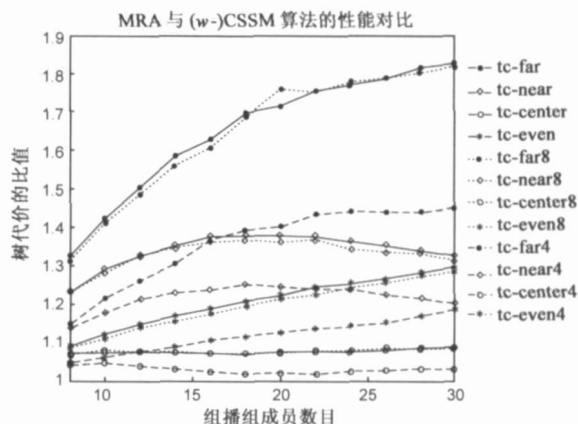


图 7 树代价性能对比

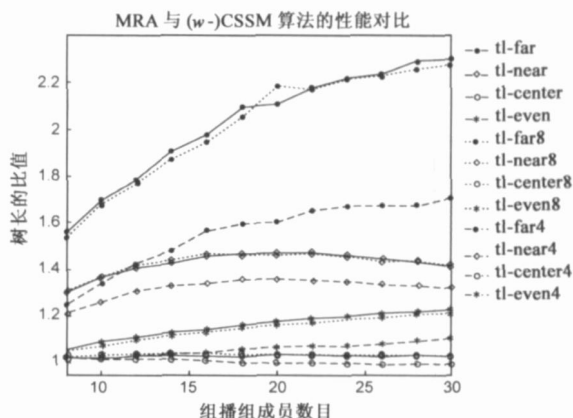


图 8 树长性能对比

MRA 与 CSSM 组播树的树代价之比最高达 1.8 多. 从图 8 可以看出, 尽管低轨卫星网络轨间 ISLs 的长度不同, 但组播树树长与树代价的性能总体上相吻合. 图 9 说明 MRA 与 CSSM 组播树的平均中间节点数比值在远端和近端场

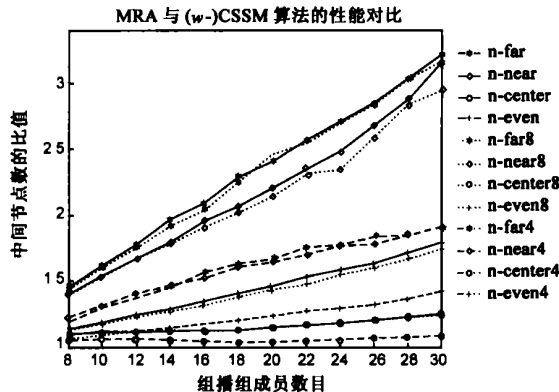


图 9 中间节点占用情况对比图

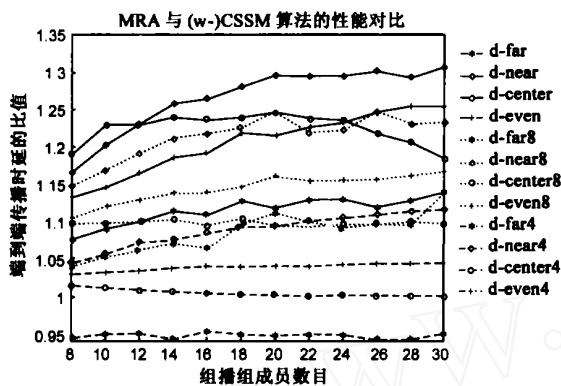


图 10 端到端传播时延性能对比

景时最高已经超过 3,与 w_1 -CSSM 组播树的中间节点数比值最高也超过 3;而且,在仿真的组播组规模范围内,随着成员数目的增加, MRA 与 (w_-) CSSM 组播树的中间节点数比值呈近似线性增长,说明所提算法在组成员密度较大时更具优越性.图 10 说明 CSSM 算法的端到端传播时延高于 MRA,尤其在近端分布场景中,因为此时源与目的节点之间非常靠近,共用的链路在端到端组播路径中所占比例稍大,所以平均端到端传播时延增加的幅度有少许提升.综上所述可知, CSSM 算法的树代价性能是以端到端传播时延性能为代价换取的.

另外,从上述仿真结果可以看出, w_1 -CSSM 算法的性能比较接近于 CSSM 算法,另外,其树代价、树长和中间节点数性能皆好于 w_2 -CSSM 算法,端到端传播时延性能不如 w_2 -CSSM 算法,说明使用加权因子可以在组播树的树代价与端到端传播时延之间作折衷.此外,远端分布场景时 (w_-) CSSM 算法的综合性能最好,中心分布场景时 (w_-) CSSM 尤其 w_2 -CSSM 算法的性能与 MRA 较为接近,这主要源于目的节点散布在网络边缘并且比较集中,所以共用的组播路径绝大部分在网络中心区域,这种情况下生成的 (w_-) CSSM 组播树与 MRA 组播树比较相似.

5 结论

无线信道广播和覆盖范围广的特性使得卫星非常

适合传输组播业务.低轨卫星网络的时延较小,适合实时多媒体组播通信,不过星上资源稀缺的特点给它们的广泛应用带来了巨大挑战,高效利用资源的组播技术可以缓解这个矛盾.针对现有低轨卫星 IP 网络中典型源组播算法的问题,本文提出了节约网络资源的 CSSM 算法,所得组播树的树代价最小.此外,为了支持某些有严格端到端传播时延要求的组播业务,提出了改进的 w -CSSM 算法.仿真结果表明,与 MRA 相比, CSSM 和 w -CSSM 算法构建的组播树的树代价和树长性能有较大改善,节省了大量的中间节点,从而高效利用了网络传输带宽资源,提高了传输效率.

参考文献:

- [1] L H Sahasrabudhe, B Mukherjee. Multicast routing algorithms and protocols: a tutorial [J]. IEEE Network, 2000, 14 (1) : 90 - 104.
- [2] Yurong Hu, Li V O K. Satellite-based internet: A tutorial [J]. IEEE Communications Magazine, 2001, 39 (3) : 154 - 162.
- [3] E Ekici, I F Akyildiz, et al. A multicast routing algorithm for LEO satellite IP networks [J]. IEEE/ACM Transactions on Networking, Apr. 2002, 10 (2) : 183 - 192.
- [4] De Nian Yang, Wanjiun Liao. On multicast routing using rectilinear Steiner trees for LEO satellite networks [A]. in Proc. IEEE Globecom 2004 [C]. Hyatt Regency Dalls, TX: IEEE, Apr. 2004, 2712 - 2716.
- [5] H Holbrook, B Cain. Source-Specific Multicast for IP draft-ietf-ssm-arch06.txt [Z]. Internet Engineering Task Force, 2004-02.
- [6] E Ekici, I F Akyildiz, et al. A distributed routing algorithm for datagram traffic in LEO satellite networks [J]. IEEE/ACM Trans Networking, 2001, 9 (2) : 137 - 147.
- [7] 张帆, 李德敏, 陶莉. 基于位置信息的 MANET 路由协议综述 [J]. 计算机工程与应用, 2005, 41 (20) : 120 - 123. Zhang Fan, Li Demin, Tao Li. A review of location-based MANET routing protocols [J]. Computer Engineering and Applications, 2005, 41 (20) : 120 - 123. (in Chinese)
- [8] Werr Hwa Liao, Jang-Ping Shen, Yur-Chee Tseng. GRID: a fully location-aware routing protocol for mobile ad hoc networks [J]. Telecommunications Systems, 2001, 18 (1 - 3) : 37 - 60.

作者简介:



程连贞 女, 1978 年 5 月生于山东菏泽, 北京航空航天大学博士生, 主要研究领域为移动通信网络的路由协议和信道接入协议.
E-mail: clz.tea@sina.com