

适于低轨卫星 IP 网络的核心群合并共享树组播算法

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 网络中现有典型源组播算法的信道资源浪费问题, 该文提出了一种低树代价的组播算法, 即核心群合并共享树(CCST)算法, 包括动态近似中心(DAC)选核方法和核心群合并组播路径构建方法。DAC 方法基于逻辑位置形成的虚拟静态、结构规则的网络拓扑选择核节点。在核心群合并方法中, 以核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, 从而使得组播树的树代价最小, 大大提高了网络的传输带宽利用率和组播传输效率。最后, 与低轨卫星 IP 网络中的其他几种典型算法进行了性能对比, 仿真结果说明, CCST 算法的树代价性能比其它算法有较大改善, 而端到端传播时延略高。

关键词: 卫星 IP 网络; 低轨; 组播; 共享树; 选核

中图分类号: TN927+.2

文献标识码: A

文章编号: 1009-5896(2007)11-2632-05

The Core-Cluster Combination-Based Shared Tree Algorithm 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 the typical source-specific multicast routing algorithm in Low Earth Orbit (LEO) satellite IP networks, a new core-based shared tree algorithm named Core-cluster Combination-based Shared Tree (CCST) algorithm is proposed in this paper. It includes a core selection method named Dynamic Approximate Center (DAC) and a multicast route construction scheme named core-cluster combination. The DAC method selects core node based on virtually static and regular network topology formed by logical locations. The core-cluster combination scheme takes core node as initial core-cluster, and extends it to construct entire multicast tree with the lowest tree cost step by step by a shortest path scheme between newly-generated core-cluster and surplus group members, which can greatly improve transport bandwidth utilization and multicast transport efficiency. Finally, the CCST algorithm is compared with several other typical algorithms in LEO satellite IP networks, and simulation results show that its tree cost performance is greatly better than the others at the expense of a bit higher end-to-end propagation delay.

Key words: satellite IP networks; LEO (Low Earth Orbit); multicast; shared tree; core-selection

1 引言

覆盖范围广和无线广播的特性使得卫星网络非常适合转发组播业务。与高轨(GEO)和中轨(MEO)卫星网络相比, 低轨(LEO)卫星网络的信息传输时延很小, 可以支持基于 IP 组播的实时多媒体业务如视频会议等^[1-3]。不过, 星上资源稀缺的特点使得它们的应用具有一定的局限性, 所以研究节约资源的低轨卫星网络的星间组播技术具有重大的实际意义和应用价值。

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, 即源节点如何尽最大可能沿公共共享路径向多个目的节点同时发送相同的分组, 这些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。组播的基本问题是尽量共享传输路径即找到一棵代价低的组播树。一般地, 组播算法分为两类: 一类是源(source-based)组播算法, 为每个源节点生成一棵以源为根的最短路径树, 时延性能较好; 另一类是共享树(shared tree)组播算法, 以组播组中某(几)个成员节点为根(称为核), 生成由组播组中所有成员共享的一棵组播树, 这种算法的代价性能和可扩展性较好, 但时延性能比源组播算法差^[4]。

传统的地面网络组播算法中, 基于源的方法如距离向量组播路由协议(DVMRP)、开放式组播最短路径优先

2006-3-27 收到, 2006-09-13 改回

国家自然科学基金(60532030, 10577005)和航天科技创新基金资助课题

(MOSPF)等只适用于固定或者拓扑缓慢变化的网络中; 而共享树方法如核基树(CBT)、稀疏模式的协议无关组播(PIM-SM)等在拓扑动态变化的网络中因为移动导致频繁的信息交换, 从而产生很大的开销。鉴于卫星网络的节点具有高速移动性并且星上资源稀缺, 不适合直接使用这些算法, 需要设计专用、简单、高效的组播算法。

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)^[1]、RST (Rectilinear Steiner Trees)^[2]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, 但计算过程比较复杂; 在 MRA 中, 源和中间卫星节点使用数据报路由算法(DRA)^[5]为每个目的卫星节点确定下一跳路由方向, 生成的源组播树的端到端传播时延较小, 适合实时多媒体业务。但是, 由于仅利用当前树上节点(on-tree nodes)到目的节点的单跳(one-hop)局部信息(下一跳路由方向)合并路径, 创建的组播树的树代价很高, 耗费大量的网络传输资源。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 的栅格结构即 144 颗卫星节点、264 条链路表示简化的低轨卫星 IP 网络拓扑, 星际链路(ISLs)的长度依据文献[1]和[5]设置, 一个 30 节点的组播组均匀分布在网络中。从图中可以看出, 组播树上有 77 个中间节点、106 条链路, 占用了网络中大部分的中间节点和链路, 很多链路并没有被树上节点共享。

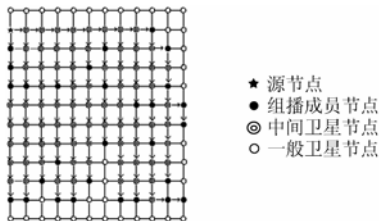


图 1 MRA 组播树实例

为了解决 MRA 的资源浪费问题, 本文提出了核心群合并共享树(Core-cluster Combination-based Shared Tree, CCST)算法, 包括动态近似中心(Dynamic Approximate Center, DAC)选核和核心群(core-cluster)合并的组播路径构建方法。

本文其它部分的安排如下: 第 2 节介绍了低轨卫星网络的星座模型, 第 3 节详细描述了 CCST 算法, 主要介绍了动态核位置选择方法和组播路径构建机制, 第 4 节仿真评估了算法性能; 最后总结全文并给出了相关结论。

2 低轨卫星网络的星座模型

假设低轨卫星星座由 P 个轨道组成, 每个轨道上有 S

颗卫星, 是 $P \times S/P/0^\circ$ 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, 所有轨内 ISLs 的长度相等且固定, 轨间 ISLs 的长度随卫星地理位置变化。另外, 由于极轨道星座在两极附近区域的卫星过于密集, 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。

CCST 算法基于逻辑位置概念^[5]构建组播树。假设网络内的卫星是初始对齐的, 在初始时刻将整个星座的轨道按照特定的次序编号, 使用 p 标识轨道编号 ($0 \leq p \leq P-1$); 并将轨道内的卫星依次编号, 使用 s 标识卫星编号 ($0 \leq s \leq S-1$), 则 (p, s) 可以表示一个卫星节点。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, 逻辑位置与卫星一一对应, 卫星的星下点位于对应的逻辑位置的中心, 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, 则星上路由表只与静态的逻辑位置相关, 不再涉及由于卫星高速运动给路由计算带来的复杂性, 即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。

假设低轨卫星 IP 网络中某组播组 $G = \{g_0, g_1, g_2, \dots, g_{N-1}\}$, 共有 N 个成员节点, 其中 g_i ($i=0, \dots, N-1$) 代表卫星成员节点的逻辑位置, $g_i = (p_i, s_i)$ 。



图 2 低轨卫星网络的星座模型

3 CCST 组播算法

3.1 核位置选择

核位置对组播树的代价和时延性能有重要影响, 因此核位置选择是核共享树组播算法中的关键问题之一^[6-8]。随着组成员的动态加入和退出, 原本最优的核共享树可能会退化为随机核树, 所以需要进行核移植^[9,10], 即重新选核并重建组播路径。另外, 在移动网络中节点位置的动态变化也会导致核移植, 而核移植将带来额外的网络开销、较长的等待时延甚至会丢包。

3.1.1 相关工作 目前已有面向地面 IP 网络的选核方法^[6-9,11]。文献[7]将选核方法分为任意(arbitrary)、随机(random)、基于拓扑(topology-based)和基于组(group-based)的 4 类。随机方案从网络中随机选取一个节点作为核, 相比

其他方法, 具有相当大的平均时延; 基于组的方案基于网络拓扑和组成员位置选核; 基于拓扑的方案基于大致的网络拓扑特征选核, 重算拓扑的时间粒度(time scale)相对较粗, 很难获得高速移动网络当前精确的拓扑结构; 基于性能的方案从一个节点(接收节点、源或者所有节点)集合中选择以某些性能度量综合加权后具有最优值的节点为核, 这种方法过于复杂不适合资源稀少的网络节点使用。综上可知, 基于组的方案最适合低轨卫星网络。

文献[11]提出了几种基于组的选核方法, 最大中心树(MCT)的选核原则是核到其它成员节点的最大距离最小, 平均中心树(ACT)按照核到所有组成员的平均距离最小的原则选核, 直径中心树(DCT)以每个成员节点到距它最远的两个成员节点的距离和为直径, 最小直径的中点作为核节点。然而, 它们均使用成员节点到其他成员的链路距离作为选路准则, 在拓扑快速变化的卫星网络中, 频繁变化的节点连接关系将导致复杂、繁琐的星上计算, 所以不适合用于星上资源稀少的卫星网络中。

3.1.2 DAC 方法选核 DAC 方法选择核位置的详细过程如下: 设组 G 的中心位置的逻辑位置是组 G 中成员逻辑位置的平均值, 记为 g_a , $g_a=(p_a, s_a)$, 其中 p_a 和 s_a 可由下式求得:

$$p_a = \frac{1}{N} \left(\sum_{i=0}^{N-1} p_i \right) \quad (1)$$

$$s_a = \frac{1}{N} \left(\sum_{i=0}^{N-1} s_i \right) \quad (2)$$

g_a 与组 G 中成员节点之间距离的最小值记为 D_{\min} , 可从式(3)求得, 其中 $i=1, \dots, N-1$ 。将与 g_a 距离最近的逻辑位置记为 $g_u=(p_u, s_u)$, 其中 p_u, s_u 满足式(4)。若式(4)有唯一解, 则以它作为核位置, 即 $g_c=g_u$; 若有多个解, 则随机选取其中之一作为核位置。将核位置记为 $g_c, g_c=(p_c, s_c)$ 。

$$D_{\min} = \min\{D_i | D_i = \sqrt{(p_i - p_a)^2 + (s_i - s_a)^2}\} \quad (3)$$

$$d_c = d_{\min} = \sqrt{(p_u - p_a)^2 + (s_u - s_a)^2} \quad (4)$$

DAC 选择方法简单、高效, 在高动态的网络中也不需要非常复杂的星上计算, 适合低轨卫星网络, 并且可以根据组播组在网络中的分布情况自适应选择最优的核节点。另外, 由于采用了逻辑位置形成的虚拟静态的拓扑结构, 所以只因组成员关系的动态变化触发核移植, 降低了核移植频率, 节省了宝贵的星上资源。

3.2 核心群合并方法构建组播树

在核心群合并方法中, 以核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, 提高了网络带宽利用率和组播传输效率。

假设已使用 DAC 方法为组播组 G 选出核节点 g_c , $G=\{g_0, g_1, g_2, \dots, g_{N-1}\}$, 为了易于表达后续内容, 假设 g_0 被选作核 g_c 。定义非核成员节点集 $G_0=\{g_1, g_2, \dots, g_{N-1}\}$ 。

3.2.1 相关概念和规则

(1)最短路径: 定义两节点间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: 如果最短路径不止一个, 则将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: 对于点到多点的组播通信, 需要构建网络中某节点 n_0 到节点集 H 的组播路径, 首先计算 n_0 与 H 中各节点之间的最近距离最短路径并比较, 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 到 H' 的组播路径构建方法如下: 计算 K 中每个节点与 H' 各节点之间的最近距离最短路径并比较, 选择其中的最近距离最短路径作为组播路径。

(5)核心群: 在构建组播路径的过程中, 将核节点和已经与其建立组播路径的成员节点以及这些已生成的组播路径上的中间节点共同组成的节点集合称为核心群, 核心群内的节点随着组播路径的扩展不断增多, 初始核心群只包括核节点 g_c , 即 $C_0=\{g_c\}$ 。

3.2.2 核心群合并方法 图 3 给出了核心群合并方法构建组播树的流程图, 下面详细介绍构建过程:

(1)使用 DRA 计算集合 G_0 与初始核心群 C_0 之间的最近距离最短路径, 得到其上属于 G_0 的目的成员节点 g'_1 以及中间节点(记为 g'_{1a}, g'_{1b}, \dots), 使用最近距离最短路径连接 g_c 和 g'_1 作为新的组播路径, 将 g'_1 和中间节点以及 C_0 合并得到一个新的核心群 $C_1=\{g_c, g'_1, g'_{1a}, g'_{1b}, \dots\}$ 。令 $C'_0=\{g_c, g'_1\}$, $i=1$ 。

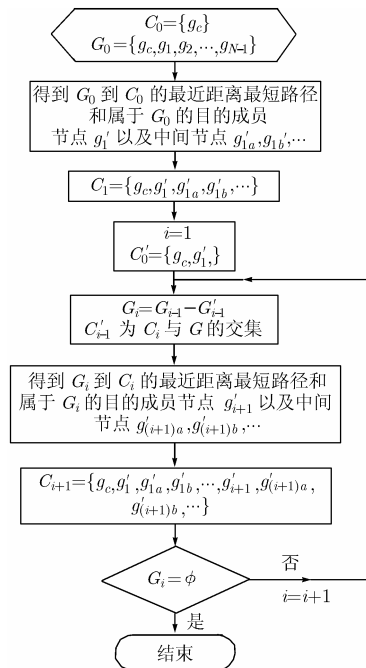


图 3 核心群合并方法构建组播树流程图

(2) $G_i = G_{i-1} - C'_{i-1}$, C'_{i-1} 为 C_i 与 G 的交集, G_i 包括的是核心群合并后 G_{i-1} 中剩余的成员节点。下一个核心群 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, 得到其上属于 G_i 的目的成员节点 g'_{i+1} 和中间节点(记为 $g'_{(i+1)a}, g'_{(i+1)b}, \dots$), 连接这条最近距离最短路径作为新的组播路径, 将 g'_{i+1} , 中间节点与核心群 C_i 合并后, 得到 $C_{i+1} = \{g_c, g'_1, g'_{1a}, g'_{1b}, \dots, g'_{i+1}, g'_{(i+1)a}, g'_{(i+1)b}, \dots\}$ 。

(3) 令 $i=i+1$, 继续步骤(2)直到 $G_i = \phi$ 即组播树 G 内的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建完成。

3.3 算法的复杂度分析

MRA 构建组播路径的计算复杂度为 $O(LN)$, 其中 L 为所构建组播树上的节点总数 ($L > N$), CCST 算法构建组播路径的计算复杂度为 $O(N^3)$, 稍高于 MRA。

3.4 动态组成员关系维护

为了支持组成员随机加入和退出, 需要维护动态组成员关系, 包括下面 3 种操作。

成员加入: 要加入某个组播组的卫星节点向邻节点广播包含了组播会话 ID 的“加入请求”报文, 非树上节点的邻节点继续向其邻节点广播这个报文, 直到它到达某个树上节点, 然后沿组播树发送到核节点。核节点回复“加入确认”报文, 并以现有组播树上的节点集作为核心群, 使用核心群合并方法计算新加入节点到核心群的最近距离最短路径并建立连接作为组播路径。

成员退出: 如果某个成员节点要退出组播会话, 它直接向核节点发送“退出请求”报文。核节点回复以“退出确认”报文。并且, 如果它是组播树上的叶子节点(leaf node), 和上游分支点之间的冗余路径及其相关信息将从相关节点中删除。如果是分支点, 它的信息将从成员列表中删除, 但它仍需维护原有组播路径。若核节点要退出并且组播会话对安全的需求较高, 则需要核移植操作; 否则, 它仍继续作为核节点, 直到组成员的动态加入/退出次数达到门限值 δ 后触发核移植操作。

组播树更新: 当核节点退出安全级别较高的组播会话或者成员加入/退出次数超过门限值 δ 时, 启动核移植, 即重新使用 DAC 方法选核并移交组播信息到新核, 然后使用核心群合并方法重新构建组播树。

4 性能评估

4.1 组播树实例和分析

为了对比组播路径构建方法, 本文定义了一种新的扩展组播算法称为有核 MRA(Core-based MRA, CMRA), 它使用 DAC 选核方法和 MRA 组播路径构建机制, 以在组分布和核位置均相同的情况下对比算法的性能。

图 4 和图 5 分别是组分布相同的情况下 CMRA 和 CCST 组播树实例。可以看出, CCST 组播树的中间节点数和树代

价比 CMRA 尤其是图 1 中的 MRA 组播树显著减少, 说明 CCST 算法可以大大节约网络信道资源; CMRA 组播树比 CCST 组播树的中间节点数和树代价有一定的增长, 但明显低于 MRA 组播树; 合并路径效果最差即树代价最高的是 MRA 组播树。由此可知, 在这些算法中, CCST 算法所得的树代价最少, 即资源利用率最高(见表 1)。

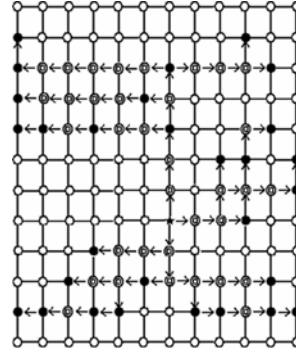


图 4 CMRA 组播树实例图

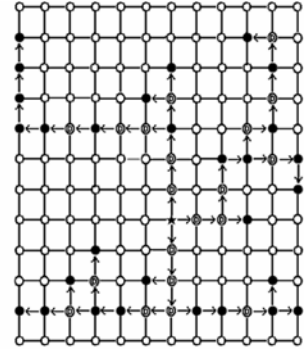


图 5 CCST 组播树实例图

表 1 算法的组播树实例对比

算法	选核方法	构路方法	图	中间节点数	树代价
CCST	DAC	核心群合并	图 5	20	49
CMRA	DAC	MRA	图 4	35	64
MRA	无	MRA	图 1	77	106

4.2 仿真环境与性能参数

假设 $P=12$, $S=24$, 组播组成员均匀分布在低轨卫星网络中, 仿真的组成员数目从 8 增加到 30, 通过仿真分别与 MRA 和 CMRA 算法作对比, 以有效评估 CCST 算法的性能。

定义下面 4 种归一化性能参数: 平均树代价的比值、平均树长(组播树上所有组播链路长度的总和)比值、组播树的平均中间节点数的比值以及组播组中每个成员节点作为源时源与所有目的节点之间的平均端到端传播时延的平均值的比值, 分别记为 h , l , n 和 d , 其中 $d = \frac{1}{N} \sum_{i=0}^{N-1} d_i$, d_i 等于第 i 个源到所有目的成员节点的端到端传播时延的平均值。

4.3 性能仿真

图 6 给出了 CCST 算法与 MRA 的性能参数对比曲线。从图中可以看出, 随着组播组成员数目的增加, 所有参数的曲线均呈增长趋势。其中, CCST 算法的树长和中间节点数性能均显著好于 MRA, 尤其中间节点数性能, MRA 与 CCST 组播树的平均中间节点数比值最大达到了 2.25; 不过, MRA 的时延性能好于 CCST 算法。另外, 从图中还可以看出, CCST 组播树的树代价与树长性能总体上大致相当, 均

反映了通信链路的使用情况。

图 7 是 CCST 算法与 CMRA 的对比结果,用于比较核心群合并和 MRA 两种组播路径构建方法的性能。可以看出,图中性能曲线的总体趋势与图 6 大致相符,均随着组成员数目的增加呈上升趋势,不过与 MRA 相比,CMRA 与 CCST 算法之间的性能比较接近,因为它们均是共享树算法,只是使用的组播路径构建方法不同,CMRA 的时延性能较好但浪费信道资源。因此,CCST 算法的树代价、树长和中间节点数性能最好,MRA 最差,平均端到端传播时延性能与此相反。

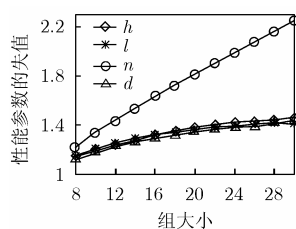


图 6 MRA 与 CCST 算法的性能对比

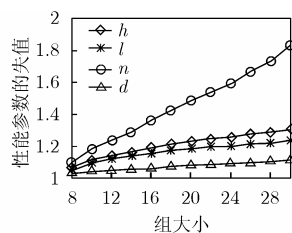


图 7 CMRA 与 CCST 算法的性能对比

5 结束语

由于卫星网络具有无线信道广播和覆盖范围广的特性,非常适合传输组播业务,并且低轨卫星网络的时延很小,所以利用低轨卫星 IP 网络进行多媒体组播业务的转发具有较好的应用前景。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,所以研究节约资源的低轨卫星网络的星间组播技术具有重大的实际意义和应用价值。针对星上资源稀缺的实际情况和现有典型源组播算法的信道资源浪费问题,本文提出了一种单核共享树组播算法即 CCST 算法,可以使所得树代价最小。仿真结果说明,CCST 算法的树代价性能较已有算法得到了显著改善,可以大大节省网络传输带宽资源,提高组播传输效率,不足的是端到端传播时延略高。

参考文献

- [1] Ekici E., Akyildiz I F, and Bender M D. A multicast routing algorithm for LEO satellite IP networks. *IEEE/ACM Trans. on Networking*, 2002, 10(2): 183-192.
- [2] Yang De Nian and Liao Wanjiun. On multicast routing using rectilinear Steiner trees for LEO satellite networks. In Proc.

- IEEE Globecom, Hyatt Regency Dalls, TX, 2004: 2712-2716.
- [3] Hu Yurong and Li Victor O K. Satellite-based internet: A tutorial. *IEEE Communications Magazine*, 2001, 39(3): 154-162.
- [4] Sahasrabudde L H and Mukherjee B. Multicast routing algorithms and protocols: A tutorial. *IEEE Network*, 2000, 14(1): 90-104.
- [5] Ekici E, Akyildiz I F, and Bender M D. A distributed routing algorithm for datagram traffic in LEO satellite networks. *IEEE/ACM Trans. on Networking*, 2001, 9(2): 137-147.
- [6] Thaler D and Ravishankar C V. Distributed center location algorithms: proposals and comparisons. In Proc. INFOCOM'96, San Francisco, CA, 1996: 75-84.
- [7] Calvert K L, Zegura E W, and Donahoo M J. Core selection methods for multicast routing. In International Conference on Computer Communications Networks, Las Vegas, Nevada, 1995: 638-642.
- [8] Raghavendra A D, Rai S, and Iyengar S S. Multicast routing in internetworks using dynamic core based trees. In Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference, 1996: 232-238.
- [9] Ali S and Khokhar A. Distributed center location algorithm. *IEEE Journal on Selected Areas in Communications*, 1997, 15(3): 291-303.
- [10] Donahoo M J and Zegura E W. Core migration for dynamic multicast routing. In International Conference on Computer Communication Networks, Columbus, OH, 1996: 92-98.
- [11] Wall D W. Mechanisms for broadcast and selective broadcast [Ph.D. Thesis]. Stanford University, June 1980.

程连贞: 女, 1978 年生, 博士生, 研究方向为无线移动网络的路由和信道接入。

刘 凯: 男, 1973 年生, 博士, 副教授, 主要研究领域为航空电信网、移动 Ad hoc 网络、无线接入、无线传感器网络和空地一体化网络。

张 军: 男, 1965 年生, 博士, 信息工程学院院长, 教授, 博士生导师, 主要研究领域为航空电信网、空地一体化网络、空中交通管理和卫星通信导航等。

宋学贵: 男, 1980 年生, 博士生, 研究方向为无线移动自组织网络的分群算法。