

认知 Ad Hoc 网络中基于信道相似度的分簇算法研究

徐 宁,张沪寅,王 晶,徐 方,汪志勇
(武汉大学计算机学院,湖北武汉 430072)

摘 要: 针对传统分簇算法无法适用于信道动态变化的认知 Ad Hoc 网络,提出了一种基于信道相似度的分布式分簇算法. 首先计算节点间的信道相似度,利用改进的 EM 算法估计节点属于不同簇的概率,再结合图的最小割算法取得最优的分簇结果. 算法既最大化簇内相似度,也最小化簇间相似度. 最后,提出了一个协调机制,可以同步全局的分簇信息. 整个过程完全分布式运行,并且无需依赖公共控制信道. 仿真结果表明,算法能够根据信道变化,动态地调整分簇结构,提高簇内公共信道数量. 与此同时,算法还能有效减少簇间公共信道,降低簇间通信干扰.

关键词: 认知 Ad Hoc 网络; 分簇算法; 信道相似度

中图分类号: TN929.5 **文献标识码:** A **文章编号:** 0372-2112 (2016)10-2323-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2016.10.006

Channel Similarity Based Clustering Algorithm in Cognitive Ad Hoc Network

XU Ning, ZHANG Hu-yin, WANG Jing, XU Fang, WANG Zhi-yong
(School of Computer, Wuhan University, Wuhan, Hubei 430072, China)

Abstract: As the traditional clustering algorithm cannot be applied to the cognitive ad hoc network for dynamic channels, a distributed clustering algorithm based on the similarity of channels has been proposed. Firstly the channel similarity between nodes will be calculated and the probability of a node within the cluster will be estimated using an adapted EM algorithm. Then by using minimum cut algorithm in graph theory, the optimal clustering results will be obtained with maximum similarity within a cluster and minimum similarity between clusters. Finally, a coordination mechanism to synchronize the global clustering information has been proposed. Throughout, these processes are evenly distributed, without relying on a common control channel. The simulation results show that the proposed algorithm can change the cluster structure according to the dynamic nature of channels, increase the intra-cluster common channels, and effectively reduce inter-cluster common channels to lower the interference.

Key words: cognitive Ad Hoc network; clustering; channel similarity

1 引言

近年来,随着无线设备的急速增加,ISM 开放频段已经变得拥挤不堪,然而许多已分配的授权频段却没有被充分利用^[1]. 认知无线电技术是提高频谱效率的有效技术手段^[2]. 配备认知无线电(Cognitive Radio)的次用户节点(Secondary User, SU)能够感知环境中的频谱资源,重新配置(Reconfigure)自己的射频参数,在不干扰主用户(Primary User, PU)的前提下,伺机使用 PU

的授权频段进行通信,从而提高授权频段的使用效率. 认知 Ad Hoc 网络(Cognitive Radio Ad Hoc Network, CRAHN)因为具有较高的灵活性和拓展性,成为目前研究关注的焦点^[2].

分簇技术是提高 Ad Hoc 网络效率的有效手段,它可以有效地降低网络复杂度、减小网络协调开销、提高网络拓展性、延长网络时间^[3]. 许多上层协议,如路由协议^[4,5]、定位算法^[6],都需要借助分簇技术才能够适用于 Ad Hoc 网络环境.

传统的 Ad Hoc 网络分簇算法需要通过公共控制信道 (Common Control Channel) 来交换控制信息. 但是在 CRAHN 中, 不同 SU 通过认知无线电技术感知的可用信道也不同, 这就会导致 SU 之间因为没有公共信道而无法创建交换控制信息. 另一方面, 即使 SU 之间能够找到公共控制信道, 但是 SU 的感知信道是随时间动态变化的, 因此, 没有办法保证预设的公共控制信道在任何时刻都是可用的. 例如, 在图 1 中, 有 3 个主用户 PU1、PU2 和 PU3, 它们占用的授权频段分别为 1 号信道、5 号信道和 2 号信道. SU1-SU8 表示的是八个次用户, 括号内是他们各自感知的可用信道集合. 从图中可以看到, 对于全体次用户 SU1-SU8 而言, 它们的可用信道集合交集为空, 即不存在公共信道.

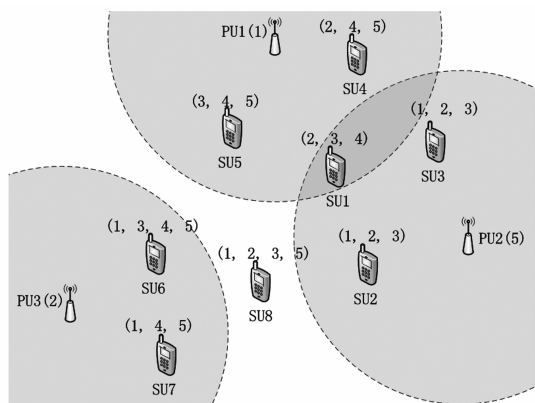


图1 认知Ad Hoc网络拓扑

为了解决 CRAHN 中的分簇问题, 我们提出了一个基于信道相似度的分簇算法. 首先, 通过邻居发现过程获取局部的网络拓扑. 然后, 根据这些局部信息, 将网络节点的信道相似度信息按照高斯混合模型 (Gaussian Mixture Model, GMM) 建模, 用最大似然法估计每个节点属于不同簇的概率. 接着, 把网络分簇问题形式化为图割问题, 通过计算最小割取得最优的分簇结果. 算法不仅最大化簇内信道相似度, 同时也最小化簇间信道相似度. 最后, 通过一个网络协调机制, 将不同节点独立计算的分簇结果在网络中进行同步, 使得所有节点拥有一致的网络分簇信息.

本文的贡献主要体现在以下几个方面:

(1) 在无公共控制信道的 CRAHN 中实现控制报文的交换;

(2) 给出了信道相似度的定义, 并提出了一个基于信道相似度的分簇算法;

(3) 设计了一个网络协调机制, 仅通过一次广播就能够同步所有节点的分簇结果.

2 系统模型

我们假设主用户和次用户均匀地分布在同一区

域, 采用分时隙的无线网络系统. 每个时隙内, 节点采用 TDMA 或 CSMA/CD 协议访问信道. 该区域的频谱资源被划分为若干无重叠的正交信道, 并且每条信道都有唯一的编号标识, 记为 $C = \{1, 2, \dots, M\}$, 其中 C 表示信道集合, $1, 2, \dots, M$ 对应 M 条互不重叠的信道. 主用户随机地占用一条或多条信道. 所有次用户都配备了一个半双工的收发器, 即在单位时间内, 次用户只能在同一条信道上接收或发送数据. 所有次用户都感知到相同的信道视图, 也就是说, 对于某一空闲频段, 所有感知到此频段的次用户都会将其映射为相同信道编号, 不会出现同一频段对应多个信道编号的情况. 对于次用户 SU_i , 它的可用信道集合用向量 $\mathbf{c}_i = (c_i^1, c_i^2, \dots, c_i^M)$ 表示, 其中 $c_i^j \in \{0, 1\}$ 表示 SU_i 的 j 号信道是否可用. 如果可用, $c_i^j = 1$, 否则 $c_i^j = 0$.

我们将 CRAHN 看作一个带权无向图 $G(V, E)$, 其中 V 代表节点集合, 它代表所有的次用户, 次用户数量为 $N = |V|$; E 代表所有边的集合, 它代表次用户间的单跳链路. 边的权值 w_{ij} 表示节点间的信道相似度, 其中 i, j 分别对应边的两个端点.

3 邻居发现

CRAHN 是一个自组织的网络, 节点之间都是对等关系. 在 CRAHN 中, 由于没有预先分配的公共控制信道, 所以简单的广播无法适用. 为了解决这个问题, 我们采用下面的邻居发现协议.

(1) SU_i 更新可用信道集合 $\mathbf{c}_i = (c_i^1, c_i^2, \dots, c_i^M)$;

(2) 在时隙 t 内, 如果第 t 条信道可用, 则 SU_i 广播自己的邻居集合 Ngb_i , 并将收到的 SU_j 的邻居集合 Ngb_j 加入到自身的 Ngb_i 中; 如果信道不可用, 则在该时隙内休眠;

(3) 直至遍历 C 中所有信道.

需要注意的是, 上述邻居发现协议还依赖次用户之间的时隙同步. 通常, 我们可以使用信号发生器, 在多条信道同时发射引导信号, 次用户一旦接收到引导信号便启动邻居发现过程. 但是, 次用户之间仍会存在细小的时隙漂移. 因此, 为了取得更好的同步效果, 应当适当延长网络的时隙长度. 假设网络的最大广播时延为 δ_{\max} , 将时隙长度设置为 $2\delta_{\max}$ 就能保证有充分的交互时间. 需要注意的是, 在整个分簇过程中, 我们假设主用户的状态不发生改变, 即次用户的可用信道集合不发生变化.

4 分簇算法

本节我们提出一个基于信道相似度的分布式分簇算法. 一方面, 通过最大化簇内节点相似度来取得更多的簇内公共信道; 另一方面, 通过最小化簇间信道相似

度来减小簇与簇之间的干扰.

4.1 信道相似度的定义

首先,我们给出节点间信道相似度的定义.

定义 节点 i 与节点 j 的信道相似度为

$$s(i, j) = \frac{\text{sum}(c_i \& c_j)}{M} \cdot (1 - 10^{\frac{RSSI_{\min} - RSSI_{ij}}{10\alpha}}) \cdot \frac{1}{e^{3(h-2.5)} + 1}$$

其中 c_i, c_j 是节点 i, j 的信道向量, $RSSI_{ij}$ 是节点间的平均接收信号强度, $RSSI_{\min}$ 是最小信号接收强度, α 是路径损耗系数, h 是节点跳数.

通过相似度的计算,我们就将邻居节点的信道向量映射为一个 $(0, 1)$ 区间上的实数.

4.2 图的分割

假设执行分簇算法的节点是 SU_i . 分簇算法实际上就是要找到一个最优的图割,使得分割后的两个子图之间相似度最小,但它并不能保证子图内部具有较高的相似度.为了解决这个问题,我们需要对图 G 进行必要的修改.

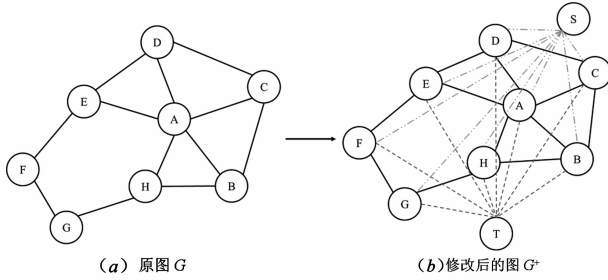


图2 认知Ad Hoc网络拓扑对应的图结构

如图2所示,在原图 G 中,加入两个辅助节点 S 和 T ,并将所有的节点与 S, T 连接.节点 p 与 S 连线的权值为 p, S 被划分到同一子图的概率,即 $w_{ps} = \Pr(p \in S)$;节点 p 与 T 连线的权值为 p, T 被划分到同一子图的概率,即 $w_{pt} = \Pr(p \in T)$.它们的具体计算方法将在下文介绍.

引入辅助边后,图割 Q 被分为了两类,一类是属于原图 G 中边集 E 的割边,它们的权值之和记为 W_E ;另一类属于辅助边集合 F 的割边,它们的权值之和记为 W_F .可以看出, W_E 越小,说明相似度权值低的边被割断,子图间的相似度也就越小;而 W_F 越小,说明隶属概率较低的节点被剔除,节点属于对应子图的概率就越高.因此, G^+ 的最小割既能保证分割后的两个子图之间相似度最低,也能保证子图内部相似度最高.这里,我们采用 Boykov、Kolmogorov 的 Min-Cut/Max-Flow 算法^[19],因为它能快速求解图的最小割.分割后, SU_i 所在的子图就是初始的分簇结果.

下面介绍辅助边的权值计算,即估计节点属于簇内或簇外的概率.

4.3 概率估计

首先, SU 计算自己与所有邻居节点的相似度,得到

一个相似度向量 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$,其中 x_i 表示 SU 与节点 i 的信道相似度, n 是 SU 邻居节点数量.下面我们用高斯混合模型(GMM)对 \mathbf{x} 进行建模.假设 \mathbf{x} 是由 2 个高斯分布混合而成的,分别对应簇内数据和簇外数据,那么 \mathbf{x} 的概率密度函数为

$$p(\mathbf{x}) = \pi_0 N_0(\mathbf{x} | \mu_0, \sigma_0) + \pi_1 N_1(\mathbf{x} | \mu_1, \sigma_1)$$

其中 $N_k(\mathbf{x} | \mu_k, \sigma_k)$ ($k = 0, 1$) 表示期望为 μ_k 、标准差为 σ_k 的高斯概率密度函数, π_k 表示高斯分量在模型中所占的权重.

令 $r_i = \Pr(x_i \in N_0) (x_i \in \mathbf{x})$,表示 x_i 由簇内分量 N_0 生成的概率,即节点 i 属于簇内的概率.那么,每个 x_i 可以看作是由簇内高斯分量 N_0 和簇外高斯分量 N_1 混合而成.因此, r_i 可以由式(1)计算.

$$r_i = \frac{\pi_0 N_0(x_i | \mu_0, \sigma_0)}{\pi_0 N_0(x_i | \mu_0, \sigma_0) + \pi_1 N_1(x_i | \mu_1, \sigma_1)} \quad (1)$$

现在,问题就转化为根据已知数据 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 估计 GMM 的参数.最大似然法是常用的参数估计方法,我们定义似然函数为

$$L = \sum_{i=1}^n \log \{ \pi_0 N(x_i | \mu_0, \sigma_0) + \pi_1 N(x_i | \mu_1, \sigma_1) \} \quad (2)$$

由于式(2)无法通过求导直接计算极值,我们采用 EM 算法(Expectation Maximization Algorithm),以迭代的方式来获取参数的最大似然估计.算法主要包含四个步骤:

(1) 初始化设置.用 Min-Cut/Max-Flow 算法对原始图 G 进行分割,若节点 i 与 SU 同簇,则令 $r_i = 1$,否则 $r_i = 0$.

(2) 参数估计.假设 r_i 是正确的,那么可以根据式(3)计算 GMM 的参数.

$$\begin{cases} n_0 = \sum_{i=1}^n r_i, n_1 = \sum_{i=1}^n (1 - r_i) \\ \pi_0 = \frac{n_0}{n}, \mu_0 = \frac{1}{n_0} \sum_{i=1}^n r_i x_i, \sigma_0 = \frac{1}{n_0} \sum_{i=1}^n r_i (\mu_0 - x_i)^2 \\ \pi_1 = \frac{n_1}{n}, \mu_1 = \frac{1}{n_1} \sum_{i=1}^n (1 - r_i) x_i, \sigma_1 = \frac{1}{n_1} \sum_{i=1}^n (1 - r_i) (\mu_1 - x_i)^2 \end{cases} \quad (3)$$

其中, n_0 为 N_0 在所有 x_i 中所占比重的总和, n_1 为 N_1 在所有 x_i 中所占比重的总和.

(3) 更新 r_i .将步骤(2)中计算的参数代入式(1),获取新的 r_i 值.

(4) 重复步骤(2)(3),直到参数收敛,最终的 r_i 即为所求.

5 簇的一致性协调与维护机制

5.1 一致性协调

在邻居发现阶段中,不同节点有不同的邻居,因此

它们的分簇结果也会存在差异. 下面, 我们提出了一套协调机制来消除这种不一致性.

将次用户 SU_i 的分簇结果记为 $D_i(X_i, W_i)$. D_i 是一个二元组, X_i 表示簇内次用户集合; W_i 表示簇的效用值, 由式(4)定义, 其中 $\langle X_i, X_i \rangle$ 表示簇内所有非辅助边的集合.

$$W_i = \sum_{e \in \langle X_i, X_i \rangle} w_e \quad (4)$$

首先, SU_i 将自己的分簇结果 D_i 通过邻居发现协议进行广播, 在 4-跳邻居范围内交换分簇结果. 将收集到的分簇结果集合记为 $DS_i = \{D_1, D_2, \dots, D_n\}$, 其中 n 为 SU_i 的邻居节点数量. 接下来, SU_i 顺序执行下面的两个操作.

操作 1: SU_i 遍历 DS_i 中所有 2-跳邻居的分簇结果, 选择效用值最大并且包含自己的簇, 并将它作为自己新的分簇, 记为 $D_i^{(1)}$.

注意, 这里并不能保证 $D_i^{(1)}$ 中包含的次用户都有相同的分簇信息. 为了消除类似的重复节点, 还需要进行下面的操作.

操作 2: 对于 SU_i , 遍历新分簇结果 $D_i^{(1)}$ 中的每一个次用户 SU_j , 因为 SU_j 也在 DS_i 中, 用操作 1 计算 SU_j 对应的 $D_j^{(1)}$. 如果 $D_i^{(1)} \neq D_j^{(1)}$, 则将 SU_j 从 $D_i^{(1)}$ 中删除. 最终得到 SU_i 的分簇结果 $D_i^{(2)}$.

5.2 簇的维护机制

当网络簇结构建立之后, 一旦 PU 开始活动, 那么将会影响簇内的公共可用信道资源. 我们可以利用多条备用信道来保持簇结构的稳定性, 具体的流程如图 3 所示.

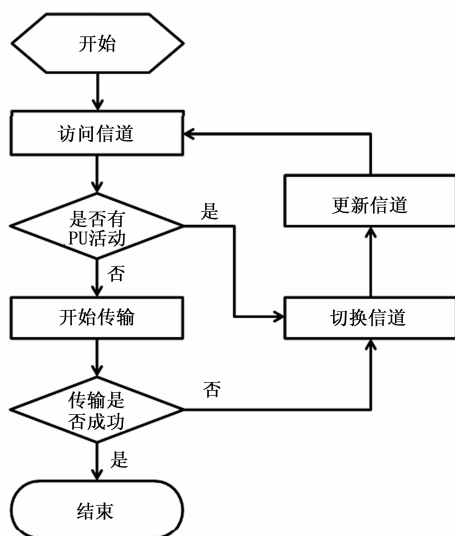


图3 分簇的维护流程

当分簇完成之后, 簇内节点就根据可用信道列表协商出一个备用信道顺序. 当检测到有 PU 活动时, 簇

内节点根据事先协商的顺序, 在备用信道中进行切换操作, 同时更新可用信道列表. 一旦切换成功, 再次进行备用信道列表的协商. 这样就能延长分簇的有效时间, 从而保持簇结构的稳定状态.

6 性能评估

本节我们将通过仿真实验来评估分簇算法在认知 Ad Hoc 网络动态环境下的性能. 在 100×100 的区域内, 随机地部署一定数量的次用户, 次用户的通信范围设置为 10. 我们通过调整次用户数量和信道总数来评估算法的各项性能指标, 所有数据都是 100 次实验的平均值.

作为参考, 我们对比了另外三种协议, 他们分别是: (a) CogMesh^[14,15]; (b) SOC^[18]; (c) 文献[11]中提出的算法. 为了方便下文引用, 我们将^[11]中的算法称为 VBC (Virtual Backbone Construction), 将本文的算法称为 CAC (Channel-Aware Clustering).

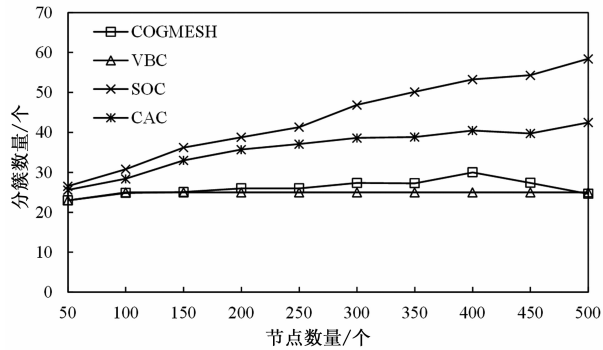
6.1 不同的节点数量

在第一组实验中, 我们固定信道总数 ($M = 20$), 不断增大网络节点数量, 考察其对算法的影响.

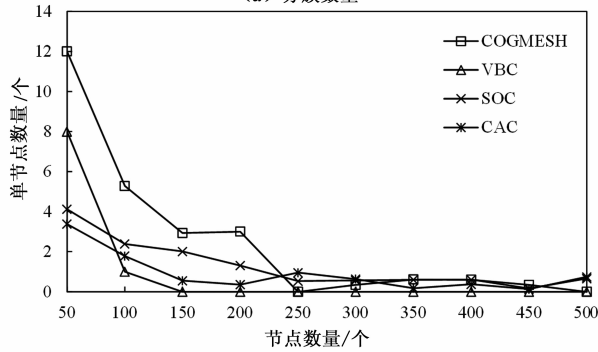
图 4(a) 展示了网络中分簇数量与次用户节点数量之间的关系. 可以看出, 一类是随着节点数量的增加, CogMesh、VBC 的分簇数量基本保持不变, 如; CogMesh 算法中, 每个节点的通信范围内只存在一个簇首, 因此分簇数量可以近似地表示为区域面积与节点的通信覆盖面积之比. 因此, 在节点通信范围一定的情况下, CogMesh 算法的分簇数量是一个常数. VBC 是一个基于地理位置的分簇算法, 它先将物理区域划分为若干子区域, 然后再子区域内选择簇首, 分簇数量只与子区域的数量有关, 而与网络节点数量无关. SOC 和 CAC 是根据节点间的信道相似度分簇, 在主用户数量保持不变的情况下, SOC、CAC 的分簇数量与次用户数量有关. 图 4(a) 中, CAC 的曲线位置要明显低于 SOC, 因此 CAC 的簇间协调开销要小于 SOC.

图 4(b) 展示了单节点簇的分布情况. 单节点簇的形成是由于节点孤立于其它节点导致的, 主要受网络拓扑和射频范围的影响. 从图中可以看到, 当节点数量大于 200 时, 单节点簇基本消失. 这是因为网络节点密度增大后, 节点分布更加紧凑, 出现孤立节点的概率降低. 当节点数量小于 200 时, CAC 算法能够很好的控制单节点簇的产生. 因为算法对最小割进行了改进, 降低了分割单个节点的概率.

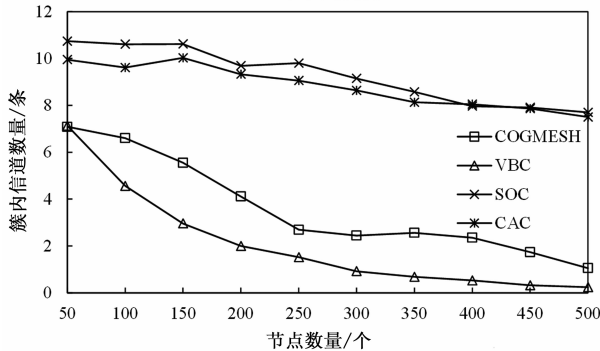
图 4(c) 展示了簇内公共信道数与节点数量的关系. 可以看到, 由于 CogMesh、VBC 算法都没有考虑节点信道的动态特性, CAC 曲线的下降速率要略大于 SOC, 这是由于 SOC 采用最大二分图算法进行分簇, 可以取得最大的簇内公共信道, 但是并没有考虑簇间的干扰问题.



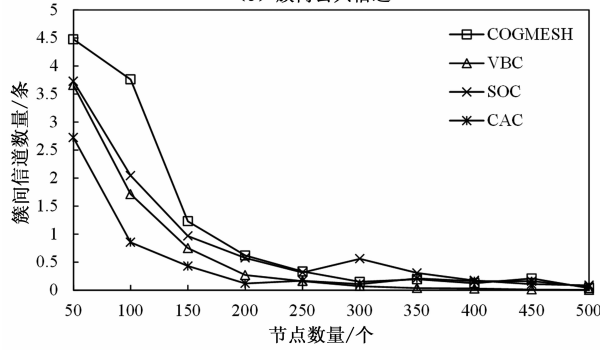
(a) 分簇数量



(b) 单节点簇数量



(c) 簇内公共信道

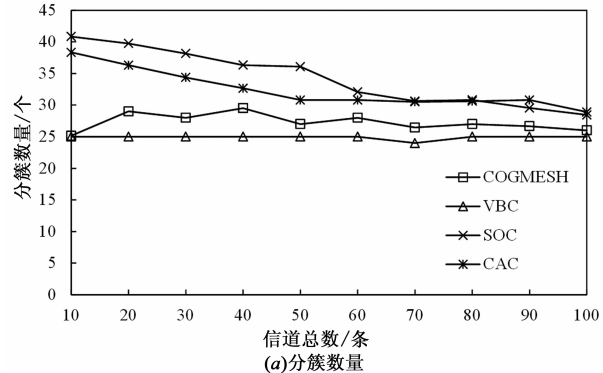


(d) 簇间公共信道

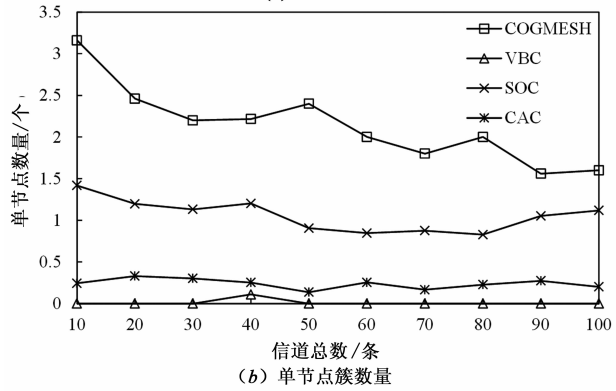
图4 固定信道总数 ($M=20$), 改变节点数量的实验结果

而 CAC 则采用了改进的图最小割算法,不但考虑簇内公共信道,还引入了最小化簇间公共信道的优化目标,这种权衡导致了 CAC 在簇内公共信道数量上稍低于 SOC.

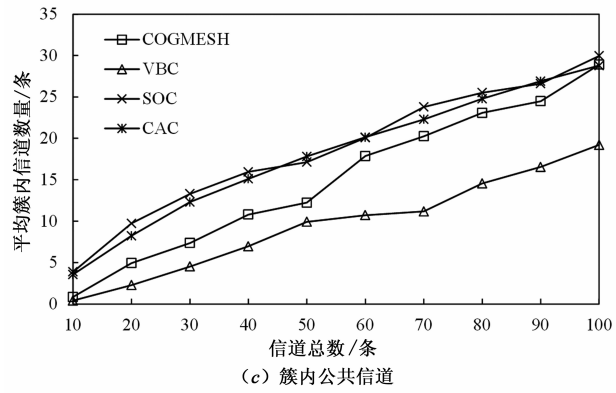
图 4(d) 展示了簇间公共信道数与节点数量的关系. 虽然当节点数大于 200 时,簇间信道数量趋近于 0,



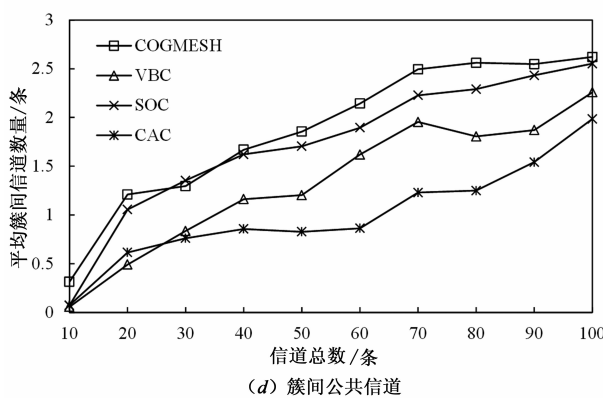
(a) 分簇数量



(b) 单节点簇数量



(c) 簇内公共信道



(d) 簇间公共信道

图5 固定节点数量 ($N=200$), 改变信道总数的实验结果

但节点数量在 200 以内时,算法之间仍有比较明显的差异. 从图中可以看到, CAC 的簇间信道数量最少.

6.2 不同的信道总数

在第二组实验中,我们固定节点数量不变 ($N =$

200),考察信道总数对算法的影响。

从图 5(a)可以看出, CogMesh、VBC 算法由于对信道变化不敏感,所以它们的分簇数量基本保持不变。而 SOC、CAC 算法的分簇数量,随着信道数量地增加会缓缓降低,这是因为信道总数的增加会导致公共信道的增加。在保证簇内公共信道数量稳定的前提下,单个簇能够容纳更多的节点,所以分簇数量就会减少。CAC 的分簇数量少于 SOC,因此在簇间开销上存在优势。

图 5(b)展示了单节点簇的数量, VBC、CAC 算法要远低于其它两种算法。VBC 单节点簇低的原因是子区域出现孤立节点的概率降低,而 CAC 通过最大化簇内相似度,降低出现单节点的图割的概率。

图 5(c)展示了簇内公共信道数量随信道总数的变化情况。所有曲线都呈逐步增长的趋势, SOC、CAC 仍旧在公共信道上取得较大的优势,但随着信道总数的增加,这种优势将会逐步的缩小。而事实上,信道资源通常十分有限,不会出现大量的可用信道。因此, SOC、CAC 更适合认知 Ad Hoc 网络。

图 5(d)展示了簇间公共信道数量随信道总数的变化情况。CogMesh、SOC 的簇间公共信道数量随着信道总数而快速增长。CAC 的簇间公共信道数最小,这有利于降低簇间干扰。

7 总结与展望

本文研究了认知 Ad Hoc 网络的分簇问题。由于认知 Ad Hoc 网络中信道随时间动态变化,而且不存在预先分配的公共控制信道,因此传统的分簇算法很难满足认知 Ad Hoc 网络在性能、效率上的需求。本文提出了一个基于信道相似度的分布式分簇算法,将认知网络的分簇问题形式化为图论中的最小图割问题,既最大化簇内节点的相似度,又最小化簇间节点的相似度。该算法能够完全分布式地运行,并且以较低的广播开销完成整个分簇过程,可以应用于实际的无线网络环境中。

但是,该协议目前只适用于拟静态的网络,我们的下一步工作是研究如何将其应用于移动环境中。首先,利用分簇算法执行过程中获得的局部网络状态,认知节点可以维护一个邻簇信息列表。当节点即将脱离分簇时,它可以在邻簇列表中检测自己应该加入的分簇,然后调整射频参数执行切换操作,最终加入新的分簇。这样,网络簇结构就得到了相对稳定。此外,还可以将节点的移动特性融入到分簇问题的约束当中,求解新的最优化问题,从而构造更有利于节点进行移动、切换的簇结构。

参考文献

- [1] Akyildiz I F, Lee W-Y, Chowdhury K R. CRAHNs: cognitive radio ad hoc networks [J]. *Ad Hoc Networks*, 2009, 7 (5): 810 – 836.

- [2] Akyildiz I F, Lee W-Y, Vuran M C, et al. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey [J]. *Computer Networks*, 2006, 50 (13): 2127 – 2159.
- [3] Younis O, Fahmy S. Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach [A]. *Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies* [C]. Hong Kong: IEEE, 2004. 629 – 640.
- [4] Lin C R, Gerla M. Adaptive clustering for mobile wireless networks [J]. *IEEE Journal on Selected Areas in Communications*, 1997, 15 (7): 1265 – 1275.
- [5] Banerjee S, Khuller S. A clustering scheme for hierarchical control in multi-hop wireless networks [A]. *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies* [C]. Alaska: IEEE, 2001. 1028 – 1037.
- [6] Estrin D, Govindan R, Heidemann J, et al. Next century challenges: scalable coordination in sensor networks [A]. *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking* [C]. Seattle: ACM, 1999. 263 – 270.
- [7] Zhao J, Zheng H, Yang G-H. Distributed coordination in dynamic spectrum allocation networks [A]. *2005 First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks* [C]. Baltimore: IEEE, 2005. 259 – 268.
- [8] Cabric D, Mishra S M, Willkomm D, et al. A cognitive radio approach for usage of virtual unlicensed spectrum [A]. *Proc of 14th IST Mobile Wireless Communications Summit* [C]. Dresden: IEEE, 2005. 1 – 4.
- [9] Cabric D, Mishra S M, Brodersen R W. Implementation issues in spectrum sensing for cognitive radios [A]. *Conference Record of the Thirty-eighth Asilomar Conference on Signals, Systems and Computers* [C]. Monterey: IEEE, 2004. 772 – 776.
- [10] Chowdhury K R, Akyildiz I F. OFDM-based common control channel design for cognitive radio ad hoc networks [J]. *IEEE Transactions on Mobile Computing*, 2011, 10 (2): 228 – 238.
- [11] Bian K, Park J-M, Chen R. Control channel establishment in cognitive radio networks using channel hopping [J]. *IEEE Journal on Selected Areas in Communications*, 2011, 29 (4): 689 – 703.
- [12] Theis N C, Thomas R W, DaSilva L A. Rendezvous for cognitive radios [J]. *IEEE Transactions on Mobile Computing*, 2011, 10 (2): 216 – 227.
- [13] Lin Z, Liu H, Chu X, et al. Jump-stay based channel-

- hopping algorithm with guaranteed rendezvous for cognitive radio networks[A]. Proceedings of 2011 IEEE INFOCOM[C]. Shanghai; IEEE, 2011. 2444 – 2452.
- [14] Chen T, Zhang H, Maggio G M, et al. CogMesh: a cluster-based cognitive radio network[A]. 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks[C]. Dublin; IEEE, 2007. 168 – 178.
- [15] Chen T, Zhang H, Maggio G M, et al. Topology management in CogMesh: a cluster-based cognitive radio mesh network[A]. IEEE International Conference on Communications 2007[C]. Glasgow; IEEE, 2007. 6516 – 6521.
- [16] Dai Y, Wu J, Xin C. Virtual backbone construction for cognitive radio networks without common control channel [A]. Proceedings of IEEE INFOCOM 2013[C]. Turin; IEEE, 2013. 1456 – 1464.
- [17] Zhao J, Zheng H, Yang G-H. Spectrum sharing through distributed coordination in dynamic spectrum access networks[J]. Wireless Communications and Mobile Computing, 2007, 7(9): 1061 – 1075.
- [18] Lazos L, Liu S, Krunz M. Spectrum opportunity-based control channel assignment in cognitive radio networks [A]. 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks 2009[C]. New Orleans; IEEE, 2009. 1 – 9.
- [19] Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(9): 1124 – 1137.

作者简介



徐 宁 男, 1989 年生于湖北十堰, 博士生, 主要研究方向为移动无线网络、认知 Ad Hoc 网络、上下文认知计算。

E-mail: davidxn@whu.edu.cn



张沪寅(通信作者) 男, 1962 年生于江苏苏州, 教授、博士、博士生导师, 主要研究方向为网络 QoS、计算机网络、新一代互联网体系结构。

E-mail: zhy2536@whu.edu.cn