

# 一种异步的认知无线网络跳频算法

李晓艳, 张海林, 郭超平, 张晓龙

(西安电子科技大学综合业务网理论及关键技术国家重点实验室, 710071, 西安)

**摘要:** 针对认知无线网络中次用户对之间的交会问题,即次用户对在同一时间接入到相同的信道,提出了一种异步的基于循环请求集的跳频算法(ACHA). 首先,次用户将可用的空闲信道按照循环请求集的结构生成跳频序列,然后再跟随跳频序列依次接入到可用信道中. 利用循环请求集系统的旋转相交特性,ACHA 算法能够保证任意 2 个次用户之间的交会必然发生在一个基本跳频序列周期之内. 仿真结果表明:与正交序列跳频算法和基于三角形数的跳频算法相比,ACHA 算法能将平均交会时间降低 20%和 40%

**关键词:** 认知无线电; 交会; 跳频序列; 循环请求集

**中图分类号:** TN929.5   **文献标志码:** A   **文章编号:** 0253-987X(2012)12-0030-06

## Asynchronous Channel Hopping Algorithm for Cognitive Radio Networks

LI Xiaoyan, ZHANG Hailin, GUO Chaoping, ZHANG Xiaolong

(State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China)

**Abstract:** An asynchronous channel hopping algorithm (ACHA) based on the cyclic quorum is proposed to realize rendezvous in cognitive radio networks (CRNs). Rendezvous means that secondary users (SUs) access the same channel at the same time. SU uses the structure of the cyclic quorum and generates the channel hopping sequences from its available idle channels. Then SUs tune themselves to their available channels following the channel hopping sequences. The ACHA makes use of the rotation closure property of the cyclic quorum system, and ensures the rendezvous between any two SUs to be realized within a basic channel hopping sequence period. Simulation results and comparisons with the generated orthogonal sequence-based algorithm and the triangular numbers-based algorithm show that the proposed algorithm can reduce the average time of rendezvous by 20% and 40%, respectively.

**Keywords:** cognitive radio; rendezvous; channel hopping sequence; cyclic quorum

在认知无线网络中,次用户以机会接入的方式使用未被主用户占用的空闲频段<sup>[1]</sup>. 根据网络的拓扑结构,认知网络可以分为集中式网络和分布式网络<sup>[2]</sup>. 在分布式网络中,由于缺少中心基站的支持,次用户间进行数据传输之前需要使用公共控制信道(common control channel, CCC)来交换彼此

的控制信息<sup>[3]</sup>. 如何设计和选择 CCC,是分布式认知网络 MAC 层的核心问题之一.

目前,CCC 的实现方法可以分为 3 类<sup>[4]</sup>:①全局固定的 CCC;②按区域使用的 CCC;③以跳频序列的方式实现动态的 CCC. 由于次用户只是授权频段的访问者而不是拥有者,当主用户出现时,次用户

必须退出正在使用的任何授权频段,因此在网络中布置一条全局范围内固定使用的 CCC 是比较困难的. 虽然全局固定的控制信道实施起来比较困难,但邻近区域的次用户间通常具有相似的可用信道,可以按照分簇算法将次用户划分为不同的簇,簇内共享一条 CCC. 这种方法需要很大的额外开销来建立和维持簇内的正常运作.

以跳频序列的方式实现动态的 CCC 是指次用户跟随一种随机或者特定的跳频序列接入到它的所有可用信道中. 即跳频序列决定了次用户访问它的可用信道的顺序. 当次用户对在相同的时间接入到相同的信道时,它们可以在此共同信道上交换彼此的控制信息,这即是交会. 跳频算法用以产生跳频序列,并且必须保证有共同可用信道的任意 2 个次用户之间能够实现交会.

目前已经提出了多种异步认知网络的跳频算法,文献[5]提出了一种正交序列跳频算法(GOS),该算法保证交会能在一个最大有限时间(maximum time to rendezvous, MTTR, 用  $t_{\text{MTTR}}$  表示)内达到,其  $t_{\text{MTTR}} = N(N+1)\Delta L$ ,  $N$  是认知网络中的总信道数量,  $\Delta L$  是一个时隙的长度. 文献[6]也提出了一种基于请求集的算法,其  $t_{\text{MTTR}} = N^2\Delta L$ , 该算法按照通信角色将次用户划分为发送者和接收者,但仅能保证发送者和接收者之间的交会,不能被应用于次用户间的邻居发现阶段,因为此时次用户希望能与它的所有邻居节点实现交会以交换必要的控制信息,从而获得对整个网络拓扑结构的了解,而不关心彼此之间的角色. 文献[7]提出了一种基于三角形数的跳频算法(TN),其  $t_{\text{MTTR}} = N(3N-1)\Delta L$ , 但该算法要求  $N$  必须是一个素数.

为了解决以上跳频算法存在的各种问题且获得一个有限的交会时间,本文利用循环请求集系统的旋转相交特性,设计了一种异步的跳频算法(ACHA). 在本文算法中,所有次用户独立地产生各自的跳频序列,并且能在异步环境下实现与其他次用户的交会,保证交会能在一个基本跳频序列周期中发生. 对比现有的跳频算法,ACHA 算法具有较少的系统假设和使用条件.

## 1 系统模型

考虑一个带有  $n=\{1, \dots, N\}$  个信道和  $m=\{1, \dots, M\}$  个次用户的认知无线网络,每个次用户装有一个半双工的收发器,在通信范围之内且在同一个信道上的次用户对可以相互通信. 在本文中,如不

作特别声明,节点与次用户 2 个名词通用. 假设系统时间被划分为相同长度的时隙  $\Delta L$ , 每个时隙足够长以至于可以交换多个数据包. 节点在网络内是不同步的,以  $C_i$  表示第  $i$  ( $i=1, \dots, M$ ) 个节点感知到的空闲频谱集合,本文假设所有节点有相同的空闲频谱集合<sup>[6]</sup>. 节点的跳频序列由一个基本跳频序列重复构成,基本跳频序列为  $X=\{x_0, x_1, \dots, x_{L-1}\}$ , 长度为  $L$ , 其中  $x_j$  表示节点在第  $j$  个时隙接入到  $x_j$  信道. 如果任何 2 个节点在某一个时隙同一个信道上相遇,这时交会会发生. 交会发生之后,节点间交换必要的控制信息,之后就切换到协商好的信道上进行数据传输.

## 2 请求集理论

下面简要介绍文献[8-9]中的请求集理论.

**定义 1** 给定一个普通集合  $U=\{0, \dots, K-1\}$ ,  $K$  为自然数,一个请求集系统  $Q$  是集合  $U$  下的一个非空子集,  $Q$  中的每个元素称为一个请求集,且请求集之间满足以下的相交特性

$$\forall p, q \in Q, \quad p \cap q \neq \emptyset \quad (1)$$

**定义 2** 给定一个非负的整数  $i$  和一个请求集  $p$ , 定义  $p$  的旋转请求集为

$$\begin{aligned} \text{rotate}(p, i) &= \{(a+i) \bmod K, \\ &a \in p, i \in [1, K-1]\} \end{aligned} \quad (2)$$

**定义 3** 请求集系统  $Q$  被称为具有旋转相交特性, 如果其中的请求集满足

$$\forall p, q \in Q, i \in [1, K-1], \text{rotate}(p, i) \cap q \neq \emptyset \quad (3)$$

例如,  $Q=\{\{0,1\}, \{2,0\}, \{1,2\}\}$  是一个在  $U=\{0,1,2\}$  下的请求集系统,它含有 3 个请求集,任取其中 2 个请求集  $\{0,1\}$  和  $\{1,2\}$ , 交集为  $\{1\}$ . 以请求集  $q=\{0,1\}$  为例,它带有 2 个元素,以标签  $S$  表示,图 1 示出了它的 2 个旋转请求集.

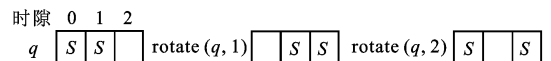


图 1 一个请求集和它的旋转请求集

从图 1 可以得出,  $q$  的 2 个旋转请求集为  $\text{rotate}(q, 1)=\{1,2\}$  和  $\text{rotate}(q, 2)=\{2,0\}$ . 同理可得出  $Q$  的其余 2 个请求集的旋转请求集, 这些请求集和旋转请求集满足定义 3, 则  $Q$  具有旋转相交性.

文献[8]提到了多种请求集系统, 这里仅介绍本文算法使用到的循环请求集系统的相关定义.

**定义 4** 设集合  $D=\{a_1, a_2, \dots, a_k \mid a_i \in [1, K-1]\}$

是一个有限集合,  $\forall d, d \in [1, K-1]$ , 都存在至少一个有序数对  $(a_i, a_j)$ ,  $a_i, a_j \in D$ , 使得  $d = (a_i - a_j) \bmod K$ , 则称  $D$  为  $(K, k)$  松弛循环差集.

**定义 5** 设  $D = \{a_1, a_2, \dots, a_k \mid a_i \in [1, K-1]\}$  是一个  $(K, k)$  松弛循环差集, 存在一个集合  $B$ ,  $B$  中元素为  $B_i = \{a_1 + i, a_2 + i, \dots, a_k + i\} \bmod K$ ,  $i \in [0, K-1]$ , 集合  $B$  被称为循环请求集系统, 其中元素被称为循环请求集.

文献[10]已经推导出  $K=3, \dots, 111$  下的松弛循环差集, 这个结果可以被直接用来构造循环请求集系统.

例如, 已知  $(3, 2)$  松弛循环差集  $D = \{1, 2\}$ , 按照定义 5 可得到循环请求集系统  $B = \{B_0 = \{1, 2\}, B_1 = \{2, 0\}, B_2 = \{0, 1\}\}$ .

文献[8]已经证明了循环请求集系统具有旋转相交性.

### 3 基于循环请求集的跳频算法

#### 3.1 算法的基本思想

首先考虑只有 2 个信道  $\{1, 2\}$  时的认知网络. 假设一个时隙的长度为  $\Delta L = 1$ . 根据循环请求集构造一个特殊的跳频序列  $u$ ,  $u$  中仅有信道  $\{1, 2\}$ . 为了得到最短的基本跳频序列长度, 选择如上节所示的  $K=3$  时的循环请求集  $B_0$  来构造序列  $u$ . 根据  $B_0$  的结构, 将信道 2 安排在时隙序号为  $\{1, 2\}$  的时隙中, 信道 1 安排在时隙序号为  $\{0\}$  的时隙中, 则  $u = \{1\ 2\ 2\}$ , 长度为  $L = |u| = 3$ . 跳频序列由序列  $u$  不断重复构成, 网络中所有节点都采用相同的跳频序列, 只是它们启动跳频序列的时间点是不一样的, 即彼此间存在时钟偏移. 图 2 为 3 个节点在不同时刻的跳频序列, 可以看出, 任意 2 个节点之间的时钟偏移刚好如同循环请求集  $B_0$  的旋转操作一样, 由于循环请求集系统  $B$  具有旋转相交特性, 所以在任意 2 个不同步的跳频序列中, 交会发生在信道 2 是能够保证的.

接下来对算法进行扩展, 使得本文算法可以适用于任意的  $N$  个信道. 在此之前, 先定义一种本文要用到的交会结构.

**定义 6** 对于一个给定的  $N, P$  是一个非素数且大于或等于  $N$ . 将这  $N$  个信道依次安排到一个形如二叉树的结构中, 称为交会结构  $R$ , 并且有  $P = 2^r$ , 其中  $r$  是  $R$  的行数. 例如, 给定  $N = 8$ , 则  $P = 2^3$ ,

$$R = \left\{ \begin{array}{cccccccc} & 1 & & & 2 & & & \\ 1 & & 2 & & 3 & & 4 & \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \right\}.$$

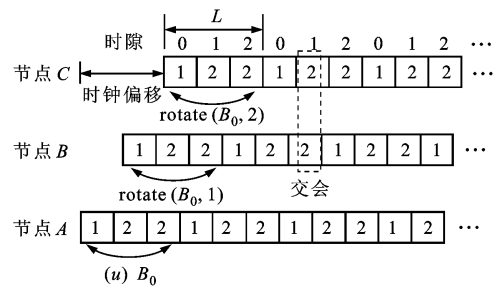


图 2 3 个节点在不同时刻的跳频序列

对于任意的  $N$ , 本文的异步跳频算法描述如下.

(1) 给定  $N$ , 根据定义 6 确定对应的交会结构  $R$ , 以及  $R$  的行数  $r$ .

(2) 应用循环请求集构造一个特殊的跳频序列  $u$ , 依据  $u$  的长度可确定基本跳频序列  $X$  的长度为  $L = |u|^r$ .

(3) 根据序列  $u$  的结构, 对于  $R$  中的每一行, 从左到右, 依次将每 2 个信道组合在一起, 构成一个子跳频序列 (简称子序列).

(4) 利用所有的子序列来构造基本跳频序列  $X$ , 规定对于  $R$  的每一行, 存在一个对应的暂时跳频序列. 考虑  $R$  的第 1 行, 由于只有信道  $\{1, 2\}$  存在, 此时的暂时跳频序列  $X_1$  就是序列  $u$ . 考虑  $R$  的第 2 行, 用此行的 2 个子序列分别替代第 1 行的信道 1 和 2, 则暂时跳频序列  $X_2$  的长度为  $L_2 = L_1 \times 3$ . 考虑第 3 行, 用此行的 4 个子序列分别替代第 2 行的 4 个信道, 则暂时跳频序列  $X_3$  的长度为  $L_3 = L_2 \times 3$ . 依次类推, 直到最后一行  $r$ , 则基本跳频序列  $X$  就是对应的  $r$  行的暂时跳频序列, 且有  $L = L_r$ .

举例说明本文的跳频算法. 给定  $N=4$  以及序列  $u = \{1\ 2\ 2\}$ , 则有  $P = N = 2^2, R = \left\{ \begin{array}{cc} 1 & 2 \\ 1 & 2\ 3\ 4 \end{array} \right\}$ . 考虑

$R$  的第 1 行, 序列  $X_1 = u, L_1 = 3$ . 考虑  $R$  的第 2 行, 存在 2 个子序列  $u_{21} = \{1\ 2\ 2\}$  和  $u_{22} = \{3\ 4\ 4\}$ , 用  $u_{21}$  和  $u_{22}$  分别替代  $X_1$  中的信道 1 和 2. 此时,  $L_2 = L_1 \times 3 = 9$ . 由于已经到达  $R$  的最后一行, 则节点的基本跳频序列  $X$  构造完成, 且有  $L = L_2$ . 基本跳频序列不断重复构成最终的跳频序列. 图 3 为 4 个节点在不同时刻启动各自的跳频序列, 图中用虚线表示出节点  $D$  在一个基本跳频序列内可以与其他 3 个节点发生交会. 其余节点间的交会情况类似.

#### 3.2 算法的正确性

下面证明本文跳频算法的正确性.

**定理 1** 对于任意 2 个时钟不同步的节点, 采用本

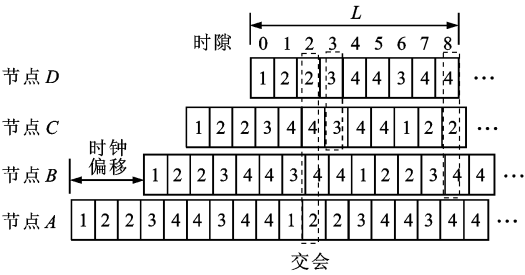


图 3 4 个节点在不同时刻的跳频序列

文的跳频算法生成的跳频序列能够使它们在一个基本跳频序列内实现交会。

**证明** 对于给定的  $N$ , 交会结构  $R$  和一个基本跳频序列  $X$  的长度  $L$  是确定的。由于序列  $X$  是由  $R$  中所有子序列构造的, 且信道之间存在替代关系, 为证明此定理, 每次同时考虑  $R$  中的 2 行。

在  $R$  的第 1 行, 只有 2 个信道  $\{1, 2\}$  且此时的暂时跳频序列  $X_1$  就是序列  $u$ 。在第 2 行, 共有 4 个信道和 2 个子序列 ( $u_{21}$  和  $u_{22}$ ), 并用  $u_{21}$ 、 $u_{22}$  分别替代  $X_1$  中的信道 1 和 2。

首先考虑  $X_1$  中的交会情况。由图 3 可知, 任意 2 个节点之间的交会均能够保证发生在信道 2 上。针对任意 2 个节点的跳频序列, 本文定义 2 种类型的时隙, 一种是交会时隙, 另一种是对角交会时隙。交会时隙是发生交会的时隙, 对角交会时隙是同一个信道在 2 个序列中相邻的西南方向的时隙。实际上从旋转相交性来看, 对角交会时隙在某个旋转尺度上就是交会时隙。2 种时隙的结构如图 4 所示。图 4 中针对节点 A 和 B, 信道 2 的交会时隙是 A 中的时隙 2 和 B 中的时隙 1, 对角交会时隙是节点 B 中的时隙 2 和 A 中的时隙 1, 而当时钟继续偏移时, 在节点 A 和 C 中, 信道 2 的交会时隙就是 A 中的时隙 1 和 C 中的时隙 2。

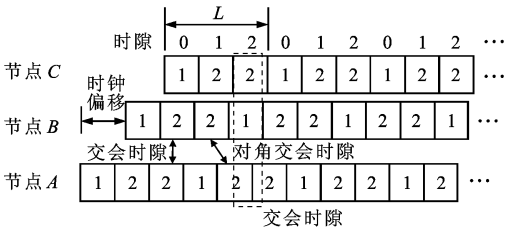


图 4 交会时隙与对角交会时隙

接着考虑  $X_2$  中的交会情况。当交会发生在  $X_1$  中时, 由于  $X_1$  和  $X_2$  的长度关系,  $L_2 = L_1 \times 3$ , 即 2 个  $X_2$  序列相差 3 个时隙以内, 仍然相当于 2 个  $X_1$

序列相差不足 1 个时隙。设  $k$  是任意 2 个  $X_2$  序列的时钟偏移量 ( $0 \leq k \leq 3$ ), 此时  $X_2$  中的子序列存在以下 2 种情况。

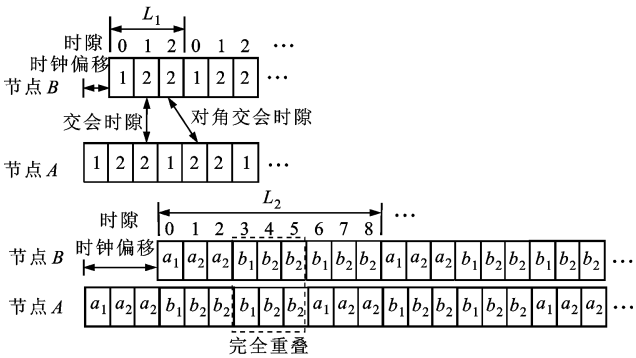
(1) 当  $k \pmod 3 = 0$  时, 替代  $X_1$  中交会时隙的子序列在  $X_2$  中完全重叠, 令  $u_{21} = \{a_1 \ a_2 \ a_2\}$ ,  $u_{22} = \{b_1 \ b_2 \ b_2\}$ ,  $X_2$  中子序列完全重叠的情况如图 5a 所示。此时, 交会必然发生。

(2) 当  $k \pmod 3 \neq 0$  时,  $X_2$  中的子序列是部分重叠的, 如图 5b 所示。此时, 替代交会时隙和对角交会时隙的 2 个部分重叠的子序列组合起来刚好满足请求集的旋转相交性,  $k \pmod 3 = 1$ , 组合的子序列刚好形如  $u_{22}$  和  $\text{rotate}(u_{22}, 1)$ , 则交会必然发生在  $u_{22}$  和  $\text{rotate}(u_{22}, 1)$  上。

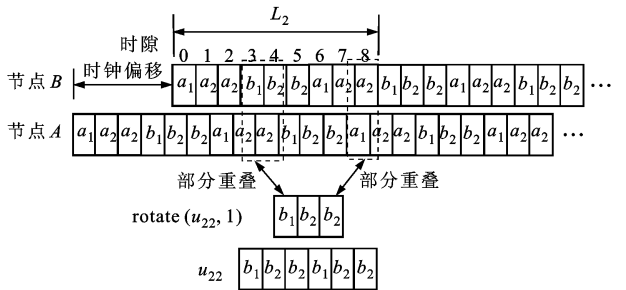
联合以上分析可以得到, 当交会发生在  $X_1$  上时,  $X_2$  中也同样保证了交会的发生。

由于  $R$  的第 2 行和第 3 行暂时跳频序列同样存在着与第 1 行和第 2 行暂时跳频序列一样的替代关系, 因此当交会发生在  $X_2$  上时,  $X_3$  中也同样保证了交会的发生。由此可以得出: 交会在每一行的暂时跳频序列中都是可以得到保证的。

通过以上分析证明, 本文的算法能够保证交会在一个基本跳频序列内发生。



(a)  $X_2$  中的子序列完全重叠



(b)  $X_2$  中的子序列部分重叠

图 5  $X_2$  中子序列的 2 种重叠情况

### 3.3 算法的期望交会时间

交会时间(time to rendezvous, TTR, 用  $t_{\text{TTR}}$  表示)是衡量跳频算法性能的一个重要指标<sup>[11]</sup>, 下面分析本文所提算法的期望交会时间  $E(t_{\text{TTR}})$ .

本文以每个信道为基本点考虑交会发生的情况, 算法的期望交会时间可表示为

$$E(t_{\text{TTR}}) = \sum_{i=1}^N p_i t_{\text{TTR}}(i) \quad (4)$$

式中:  $p_i$  为交会发生在信道  $i$  的概率;  $t_{\text{TTR}}(i)$  表示交会发生在信道  $i$  的时间;  $N$  为总的可用信道数量.

在本文算法中,  $p_i$  与信道  $i$  在一个基本跳频序列  $X$  内出现的总的时间有关, 出现的时间越长, 交会发生在该信道的可能性就越大. 因此,  $E(t_{\text{TTR}})$  也可以表示为

$$E(t_{\text{TTR}}) = \sum_{i=1}^N \frac{Y_i \Delta L}{L} t_{\text{TTR}}(i) \quad (5)$$

式中:  $Y_i$  为信道  $i$  在一个基本跳频序列  $X$  中所占的时隙数量;  $\Delta L$  为时隙的长度;  $Y_i \Delta L$  为信道  $i$  在  $X$  中出现的总时间;  $L$  为一个基本跳频序列  $X$  的时长.

为求  $Y_i$ , 分析交会结构和跳频序列的构成. 设存在一个交会结构(为便于理论分析, 以字母代替  $R$  中的信道)

$$R = \left\{ \begin{array}{cccc} & a & & b \\ aa & ab & ba & bb \\ aaa & aab & aba & abb & baa & bab & bba & bbb \end{array} \right\}$$

第1行暂时跳频序列  $X_1 = \{a b b\}$ ,  $Y_a = 1$ ,  $Y_b = 2$ . 用子序列  $\{aa ab ab\}$  和  $\{ba bb bb\}$  分别替代  $X_1$  中的  $\{a\}$  和  $\{b\}$ , 从而生成第2行暂时跳频序列  $X_2$ . 由于序列  $u$  的结构, 在  $X_1$  和  $X_2$  中, 相互替代的信道在所占的时隙数量上存在着2种倍数关系, 例如  $Y_{aa} = Y_a$ ,  $Y_{ab} = 2Y_a$ , 则有  $Y_{aa} = 1$ ,  $Y_{ab} = 2$ ,  $Y_{ba} = 2$ ,  $Y_{bb} = 4$ . 依次类推, 直到生成最终的跳频序列  $X = X_3$ . 由此可以得到每个信道在对应的暂时跳频序列中所占的时隙数量(用字母的上标数字来表示)

$$\left\{ \begin{array}{cccc} & a^1 & & b^2 \\ aa^1 & ab^2 & ba^2 & bb^4 \\ aaa^1 & aab^2 & aba^2 & abb^4 & baa^2 & bab^4 & bba^4 & bbb^8 \end{array} \right\}$$

归纳后可得

$$Y_i = \begin{cases} Y_1 = 1, & Y_2 = 2, & 1 \leq i \leq 2 \\ 2Y_{(i-2^m)}, & m = \lfloor \lg i \rfloor & 3 \leq i \leq N \end{cases} \quad (6)$$

再次分析本算法的跳频序列, 可以发现  $t_{\text{TTR}}(i)$  与信道  $i$  在一个基本跳频序列  $X$  内所占时隙的位置有关, 出现的位置越靠后, 所需要的交会时间就越长. 由于本文的跳频序列是依据信道在  $R$  中出现的

先后顺序依次安排的, 因此考虑某一信道的  $t_{\text{TTR}}$  的时候, 应将该信道所处时隙之前的所有时隙均考虑进去. 对于同一个信道的多个时隙位置, 取最靠后的时隙位置计算其交会时间

$$t_{\text{TTR}}(i) = \begin{cases} t_{\text{TTR}}(1) \leq \Delta L, & t_{\text{TTR}}(2) \leq 3\Delta L, & 1 \leq i \leq 2 \\ \left( Y_1 + \dots + Y_i + \frac{Y_{i+1}}{2} + \right. \\ \left. \frac{(Y_{i+2} + \dots + Y_N)}{2} \left\lfloor \frac{2i}{N+1} \right\rfloor \right) \Delta L, & 3 \leq i \leq N \end{cases} \quad (7)$$

以  $N=4$  为例(如图3所示),  $t_{\text{TTR}}(1) \leq \Delta L$ ,  $t_{\text{TTR}}(2) \leq 3\Delta L$ ,  $t_{\text{TTR}}(3) \leq 7\Delta L$ ,  $t_{\text{TTR}}(4) \leq 9\Delta L$ .

将式(6)、式(7)代入式(5), 即可得到期望交会时间  $E(t_{\text{TTR}})$ .

## 4 仿真实验

为了验证本文算法的正确性和有效性, 采用 Matlab 进行仿真分析. 在  $100 \text{ m} \times 100 \text{ m}$  的区域中随机放置 20 个节点, 节点的通信距离为 20 m, 所有的节点均在任一主用户的发射范围之内, 网络中共有  $N=10$  个信道, 节点间的时钟不同步. 设跳频序列的每个时隙长度为 10 ms, 共进行 1 000 次独立实验, 取实验的平均值作为最终结果. 本文选取 GOS 算法<sup>[5]</sup>和 TN 算法<sup>[7]</sup>作为参考进行性能比较.

图6给出了平均交会时间  $\bar{t}_{\text{TTR}}$  随主用户数量的变化情况. 可以看出, 3种算法的  $\bar{t}_{\text{TTR}}$  与网络中可用信道的数量成正比, 当网络中主用户数量增加, 这意味着可用的空闲信道数量减少,  $\bar{t}_{\text{TTR}}$  也随之减少. 当网络中可用信道数量是偶数的时候, TN 算法不能保证节点间的交会, 因此它提供的  $\bar{t}_{\text{TTR}}$  波动起伏较大. 本文的算法提供了比 GOS 和 TN 更低的平均交会时间.

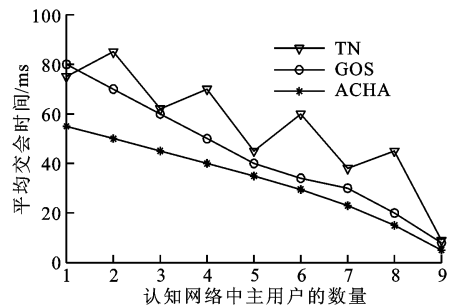


图6 平均交会时间随主用户数量变化的情况

图7给出了节点间发生交会时信道公平指数随主用户数量的变化情况. 本文使用 Jain 的公平指数  $F^{[12]}$  来表示网络中可用空闲信道的接入情况. 公平

指数定义为

$$F = (\sum H_f)^2 / (N \sum H_f^2) \quad (8)$$

式中:  $N$  是认知网络中可用的空闲信道总数量;  $H_f$  是某个信道  $f$  上交会发生的次数。

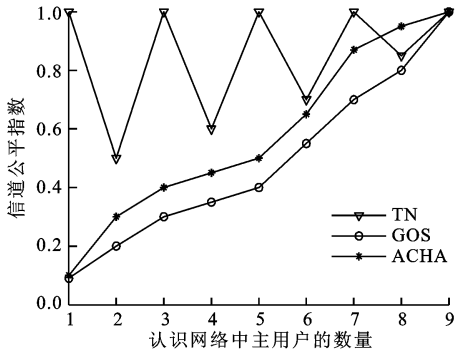


图 7 信道公平指数随主用户数量变化的情况

信道公平指数越高,反映跳频算法能够使得交会平均发生在各个信道,均衡了各个信道的利用率。当网络中可用信道数量为奇数时,TN 算法能够使得交会平均地发生在各个可用信道上,因此它提供了最高的公平指数,而 GOS 和本文算法均只保证了一个基本跳频序列周期内交会必然发生在一个信道上,因此得到的公平指数略低。当 2 个节点发生交会后,节点间可以交换必要的控制信息,之后就切换到协商好的信道上进行数据传输,此时所有的可用信道均有相同的概率被用来传输数据,信道的使用情况也不同于交会发生时的情形。

## 5 结 论

为了在异步认知网络中解决节点间的交会问题,本文利用循环请求集系统的旋转相交特性,设计了一种跳频算法。在本文算法中,所有次用户独立地产生各自的跳频序列并且能在异步环境下实现与其他次用户的交会。从理论上证明了本文算法可以提供一个有限的交会时间。仿真实验表明,与 GOS 和 TN 算法相比,本文算法可以得到最少的平均交会时间。当网络中的可用信道数量减少时,本算法可提供一个较好的信道公平指数。

## 参考文献:

[1] 张正浩,裴昌幸,陈南,等. 宽带认知无线网络分布式协作压缩频谱感知算法[J]. 西安交通大学学报, 2011, 45(4):67-72.  
ZHANG Zhenghao, PEI Changxing, CHEN Nan, et al. A sensing algorithm based on distributed coopera-

tive compressed spectrum for wideband cognitive radio networks[J]. Journal of Xi'an Jiaotong University, 2011, 45(4):67-72.

- [2] AKYILDIZ I F, LEE W Y, CHOWDHURY K R. CRAHNS: cognitive radio ad hoc networks[J]. Ad Hoc Networks, 2009,7(1):810-836.
- [3] 冯志勇,张平,郎保真,等. 认知无线网络理论与关键技术[M]. 北京:人民邮电出版社,2011.
- [4] 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.
- [5] THEIS N S, THOMAS R W, DASILVA L A. Rendezvous for cognitive radios [J]. IEEE Transactions on Mobile Computing, 2011, 10(2): 216-227.
- [6] BIAN K, PARK J M. Asynchronous channel hopping for establishing rendezvous in cognitive radio networks [C]// Proceedings of IEEE INFOCOM 2011. Piscataway, NJ, USA: IEEE, 2011: 236-240.
- [7] SHIN J, YANG D, KIM C. A channel rendezvous scheme for cognitive radio networks [J]. IEEE Communications Letters, 2010,14(10): 954-956.
- [8] IANG J R, TSENG Y C, HSU C S, et al. Quorum-based asynchronous power-saving protocols for IEEE 802.11 Ad Hoc networks [J]. ACM Mobile Networks and Applications, 2005,10(1):169-181.
- [9] 李美安,刘心松,王征. 一种基于松弛循环差集的高性能分布式互斥算法[J]. 电子学报, 2007, 35(1):58-63.  
LI Meian, LIU Xinsong, Wang Zheng. A high performance distributed mutual exclusion algorithm based on relaxed cyclic different set[J]. Acta Electronica Sinica, 2007, 35(1):58-63.
- [10] LUK W S, WONG T T. Two new quorum based algorithms for distributed mutual exclusion[C]// Proceedings of International Conference on Distributed Computing Systems 1997. Piscataway, NJ, USA: IEEE, 1997: 100-106.
- [11] LIN Zhiyong, LIU Hai, CHU Xiaowen, et al. Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio [C] // Proceedings of IEEE INFOCOM 2011. Piscataway, NJ, USA: IEEE, 2011: 2444-2452.
- [12] JAIN R. The art of computer system performance analysis: techniques for experimental design, measurement, simulation, and modeling [M]. New York, USA: John Wiley & Sons, 1991.

(编辑 刘杨)