

基于密度影响函数的聚类算法构造蓝牙分散网^①

A Clustering Algorithm of Density - Based Influence Function to Construct Bluetooth Scatternet

张波涛 张莉萍 (武汉科技大学计算机学院 湖北武汉 430065)

摘要: 针对蓝牙分散网本身具有的特殊限制和特性,文章提出了一种基于密度影响函数的聚类算法构造蓝牙分散网的方法。首先通过密度影响函数对所有结点进行分区后构建匹克网,然后利用聚类算法形成最终的分散网。经数学证明,该方法为有效构建蓝牙分散网节省了时间。

关键词: 蓝牙 分散网 匹克网 异步 密度影响函数

1 引言

蓝牙技术是一种短距离无线通信的开放性标准。蓝牙工作在 2.4GHz 的 ISM 频段,采用电路交换和分组交换技术,其目的是用单芯片、低功耗、低成本、组网容易的无线模块来代替有线电缆连接,实现短距离无线语音和数据通信。

基于蓝牙的自组网可以通过互连匹克网成为分散网的方式构建。在发布的蓝牙协议规范中,只给出了匹克网的建立过程,对于如何构造分散网的问题并没有相关描述,但目前已有许多相关的研究。文献^[2]中提出的 TPSF 算法能处理拓扑的动态改变,无论何时一个结点想要与另一个结点开始初始通信,可以按需创建一个蓝牙分散网,从而在获得较高系统吞吐量的同时增加了连接建立延迟。文献^[3]提出的算法由单个设备决定拓扑结构,要求所有结点都在彼此通信范围内,使得算法的可靠性和扩展性较差。文献^[4]中 Bluenet 产生的分散网在相邻匹克网对之间建立多个连接路由,算法不能保证最终形成的网络的连通性。

文章提出的蓝牙分散网的构造是异步的完全分布式算法,算法的实现过程是基于密度影响函数的划分后再进行局部随机聚类,其中与文献^[5]提出的基于簇的分散网的创建思路类似,但是本文所描述的算法区别在于没有盲目地在所有区域对所有结点进行聚类随机形成匹克网,而是先引进基于密度的影响函数对所

有结点分区,高密度区进行局部随机聚类,低密度区可视结点为独立点。

2 基于密度影响函数^[6]对初始结点作初步划分

文章将区域的所有移动设备都看作独立的点,相应的区域似为数据空间。该方法主要基于以下的思想:每个数据点对数据空间的影响可以用一个数学函数来形式化地模拟,它描述了一个数据点在邻域内的影响,称为影响函数;数据空间的整体密度为所有数据点的影响函数的总和;可以通过确定密度吸引点来得到聚类,所谓的密度吸引点是全局密度函数的局部最大。其中影响函数可以是一个任意的函数,它由某个邻域内的两个对象之间的距离来决定。

假设 x 和 y 是 d 维特征空间 F_d 中的对象。数据对象 y 和 x 的影响函数是一个函数 $f_{yB} : F^p \rightarrow R^+ \cup 0$,它是根据一个基本的影响函数 $f'_B(x) = f_B(x, y)$ 来定义的。

在一个对象 $x(x \in F^p)$ 上的密度函数被定义为所有数据点的影响函数的和。给定 n 个数据对象, $D = \{x_1, x_2, x_3, \dots, x_n\} \subset F^p$, 在 x 上的密度函数定义如下: $f^p(x) = \sum_{i=1}^n f'_B(x)$ 根据密度函数,能够定义该函数的梯度和密度吸引点。一个点 x 是被一个密度吸引点 x^* 密度吸引的,如果存在一组点 $x_0, x_1, x_2, \dots, x_k, x_0 = x, x_k$

① 基金项目:国家 973 基金项目(2003CB314806); 863 基金项目(NO. 2006AA10Z253)

$=x^*$, 对 $0 < i < k, x_{i-1}$ 的梯度是在 x_i 的方向上。对一个连续的和可微的影响函数, 一个用梯度指导的爬山算法能用来计算一组数据点的密度吸引点。在此对爬山算法不作描述。

在此假设初始阶段有 N 个移动设备, 利用矩阵记录各结点之间的影响函数的信息。

$$M_{m \times n} = \begin{bmatrix} f_{11}^0 & f_{12}^0 \cdots & f_{1n}^0 \\ f_{21}^0 & f_{22}^0 \cdots & f_{2n}^0 \\ \vdots & \vdots & \vdots \\ f_{m1}^0 & f_{m2}^0 \cdots & f_{mn}^0 \end{bmatrix}$$

其中 f_{ij}^0 表示结点 i 与 j 之间的密度影响函数的值, 每一行向量 $M[i, \cdot]$ 表示所有结点到某个结点的影响函数的值, 每一列向量 $M[\cdot, j]$ 表示某个结点到所有结点的影响函数的值。设定一个合适的区间范围 ι , 以及一个特定的阈值 ξ , 将每列的值按照升序将结点作初步化分为高密度区域和孤立点区域, 其中 ι 和 ξ 的取值最为关键。

利用密度影响函数对结点作了初步划分后, 孤立结点作为独立的微型分散网结构拥有分散网的所有信息; 下一步将高密度区的结点作进一步的局部聚类。

3 局部随机聚类的算法描述

文章提出的是运用随机聚类算法在密度区将一系列分布的、自主的结点构成相连的拓扑网络。为了算法有效进行, 应该满足如下几点:

(1) 结点应该形成星型簇, 每个簇都有一个簇头。主结点位于簇中心, 其它结点成为主结点的从结点。

(2) 每个簇的从结点数最多为 S (设定的值), 簇的大小要保证为最大值, 即簇要满。

(3) 为减少开销, 簇的主从结点确定后应该避免簇间结点转移和主从角色的互换。

(4) 算法循环调用完后, 每个结点应该明确自己的主从角色, 若是从结点应当知道自己所在簇的主结点, 且簇内是相连的没有孤立点。

(5) 算法调用结束后, 未加入簇的孤立点应该包含所有簇的信息。

3.1 高密度区随机聚类的模型介绍

网络中的任两个结点在彼此的频率范围内, 若形成一条连接彼此就是主从关系, 一个若是主结点, 另一

个必是从结点。其中任何结点都有同等机率成为主结点或从结点, 或与任何一个结点建立连接。

假设结点的总数为 N , 每个结点有唯一标识自己的 ID 值, 簇的最大结点数为 S (其中包括簇的头结点)。该网络是异步的, 每个结点都保持自己的时钟频率, 所以结点之间使用固定的时钟频率进行通信。当结点持续发送广播搜索信息时, 在广播时序内传输结点持续监听并作相应回复。若广播结点的频率与监听结点的频率相匹配时, 监听结点才会成功地获取消息, 接着广播一个回复。为了避免结点之间的冲突, 在广播响应时需要使用退让机制。其中用于搜索其它结点的广播消息内不包含传播结点的 ID 值, 因此回复响应的结点也不知自己在回复哪个结点, 这种随机的模式正是与其它文献所提及模式的不同之处。首先对结点所涉及的活动作个简单定义, 如表 1 所示。

表 1 结点各状态的定义

结点状态	结点状态的定义
寻呼	一个结点发送请求包, 该请求包不包含发送方的 ID 或任何其它信息。
寻呼扫描	一个结点监听请求包, 然后广播请求响应包, 该请求响应包包含发送方的 ID 值和时钟频率。
匹克网的形成	一个结点通过发送匹克包消息, 试着去连接它已知道的 ID 值和时钟频率的结点。因为匹克包消息中包含有目的结点的 ID 值。如果连接成功该结点自动成为主结点。
匹克网的扫描	一个监听匹克包的结点收到匹克包消息, 接着与发送匹克包的结点建立连接, 连接成功后该监听结点自动加入该匹克网。

在这些状态中, 寻呼和寻呼扫描或匹克网的形成和扫描都是随机的, 很明显两个结点只有在自反的情况下才能彼此搜索到对方。寻呼和寻呼扫描阶段与搜索结点的连接安装是保持一致的, 通过发送请求包搜索结点与之连接的也是同步发生的, 同样道理, 当一个成功的请求执行时, 匹克网的形成与匹克网响应包也成功投递。

3.2 随机聚类算法的构造

算法分两步将高密度区的结点快速聚类, 把分散

的结点连接成星型拓扑图。为了提高接收其它结点发送的消息的可靠性,在此算法中使结点持续广播或监听。

3.2.1 算法第一步

该阶段目的是通过 *Berboulli* 实验获得候选主结点或候选从结点。假设共有 N 个结点,簇的最大结点数为 S 个,主结点的理想数目为 k 个,其中 $k = [(N - 1) / s] + 1$,具体描述如下。

N 个结点进行 T 次 *Berboulli* 实验成功的概率为 P [*success*] = p , 试验成功的结点为候选主结点,未成功的结点为候选从结点。设在 *Berboulli* 实验中有 X 个结点成为候选主结点,设 X_i 是每次 *Berboulli* 实验中成功的结点数,其中 $P[X_i > m]$ 非常小,取近似值 $m = 1$,由文献^[6,7]

$$P[X_i \geq m] \leq \left[\frac{P}{m/N} \right]^m \left[1 - \frac{P}{m/N} \right]^{N-m} \quad (1)$$

为了在每次试验中确保取得最少的候选主结点,通过评估等式

$$P^m (1 - P)^{N-m} = \eta \left[\frac{m}{N} \right]^m \left[1 - \frac{m}{N} \right]^{N-m} \quad (2)$$

假定 = 0.001, 当 $p = m/N$ 时, $LHS > RHS$, 当 = 1 时, $LHS < RHS$ 。多次进行 *Berboulli* 实验的目的是为了控制候选主结点的数目,得到合适的簇的个数。为此运用 *Chernoff bound* 公式文献^[8]

$$P[Z \geq (1 + \varepsilon)]E[Z] \leq e^{-\frac{\varepsilon^2}{3}E[Z]} \quad (3)$$

$$P[Z \leq (1 - \varepsilon)]E[Z] \leq e^{-\frac{\varepsilon^2}{2}E[Z]} \quad (4)$$

其中 Z 是任意的值, $0 \leq \varepsilon \leq 1$, 显然

$$E[X] = N(1 - (1 - P)^T) \quad (5)$$

取 $T = \log(1 - k/N) / \log(1 - p)$, $E[X] = k$, 其中 $P[X \geq 2k] = P[Z \geq (1 + 1)k] \leq e^{-2k/3} < e^{-k/4}$ 与 $P[X \geq 3k/4] = P[X \leq (1 - 1/4)k] \leq e^{-9k/32} < e^{-k/4}$ 所以 *Berboulli* 实验中成功的候选主结点数 $3k/4 < X < 2k$, 概率近似 $p = 1 - 2e^{-k/4}$ 。

3.2.2 算法第二步

该阶段利用竞争机制的算法确定选择一个超级结点,然后确定最终的主从结点,计算主结点的实际个数并连接所有结点的信息。超级结点的选举与簇的形成交叉进行,然后超级结点返回算法形成一个预期的拓

扑网络。首先对各结点状态作如下定义:见表 2。

表 2 各结点状态定义

结点类型	结点类型定义
候选从结点 (slave - designate)	在 <i>Berboulli</i> 实验中没有成功加入到任何簇的结点
从结点 (slave)	在实验中成功加入到簇中的结点
候选主结点 (master - designate)	在实验中成功地成为了候选主结点,但是还没有接收到足够的候选从结点的请求响应
主结点 (master)	候选主结点收到了 s 个候选从结点的请求响应
候选超级结点 (super - master - designate)	主结点已经接收到部分簇的响应
超级结点 (super - master)	主结点已经接收到所有簇的响应
代理从结点 (proxy - slave)	主结点指定自己的任一从结点作为代表参与超级主结点的选举

算法的第一步已经确定了候选主结点和候选从结点。每个初始结点都处于查询扫描状态,等待其他设备搜索。算法的第二步创建过程如下:

①初始候选主结点搜索附近结点,创建初始匹克网:该结点处于寻呼状态,搜索周围一定数目的结点。被搜索到的结点处于寻呼扫描状态,等待该结点分派角色。在匹克网未满时,候选主结点将搜索到的结点连接到匹克网并任命为从结点。若该结点是候选主结点的第一个从结点,候选主结点任命这个从结点为自己的代理从结点;若匹克网满时,候选主结点宣布自己为该匹克网的主结点,此时对其它候选从结点忽略不计。因为超级主结点的选举和匹克网的形成是交叉进行的,在匹克网的形成周期内,若主结点或候选主结点接收到 k 个匹克网的代理从结点的响应时,该结点自己为超级主结点;若没有收到任何响应,该结点退成候选从结点,处于寻呼扫描状态试图被搜索连入其它匹克网。

②初始候选从结点持续发送请求加入匹克网:若候选从结点发送的请求没有得到任何搜索结点的响应,接着返回到寻呼扫描;若被搜索结点响应连接即停止寻呼扫描;若如①被任命为代理从结点,还需持续寻呼扫描参与超级主结点的选举。

③匹克网间的连接:代理从结点不停地在寻呼扫

描和连接状态之间转化,连接状态远远要比寻呼扫描状态花的时间代价要小。若候选超级主结点发出请求时,代理从结点充当临时的桥结点为匹克网间传送消息。一旦超级主结点被选举出,代理从结点作为本匹克网与超级主结点的匹克网的桥结点,实现匹克网间的通信。

设在高密度区通过 Berboulli 实验得到的实际候选主结点的个数为 X , 然而理想的候选主结点数为 $k = \lfloor (N-1)/S \rfloor + 1$ 。算法的第二步具体实现过程如下:

```

{ if ( $x \leq k$ )
    候选超级主结点搜索所有代理从结点的消息;
    /* 如图 1 所示 */
if ( $x < k$ )
{ 候选主结点主动给代理从结点发送消息;
  /* 超级主结点选举周期到达 */
  推举代理从结点的主结点为超级主结点;
  /* 此时超级主结点还未将匹克网的信息传送到所有结点 */
  超级主结点进一步寻呼扫描剩余结点;
if ( $0 < k - x < s$ )
  /* 剩余结点数小于匹克网的规模,即剩余结点不足以形成一个匹克网 */
  { 超级主结点任命  $k - x$  个结点为独立候选主结点; }
  /* 每个候选主结点是独立的包含所有匹克网的信息,如图 2 所示 */
  else if ( $k - x > s$ )
    /* 剩余结点数大于匹克网的规模,即剩余结点可以继续形成匹克网 */
    { 超级主结点随机任命  $k - x$  中的  $s$  个结点为候选主结点;
      重复①②; /* 加入匹克网 */
      任命剩下的  $k - x - s$  个结点为独立候选主结点;
    }
}
}

```

```

else if ( $x > k$ )
  /* 候选超级主结点获知至少有一个匹克网未满足,如图 3 所示 */
  { 候选主结点请求连接所有匹克网的消息;
    推举代理从结点的最高成的主结点为超级主结点;
  }
}

```

```

/* 所有结点的信息都被添加到匹克网中 */
所有主结点将匹克网的信息传给超级主结点;
将未满足的匹克网拆开平均分配结点到  $k$  个匹克网中,向超级主结点返回信息;
}
}

```

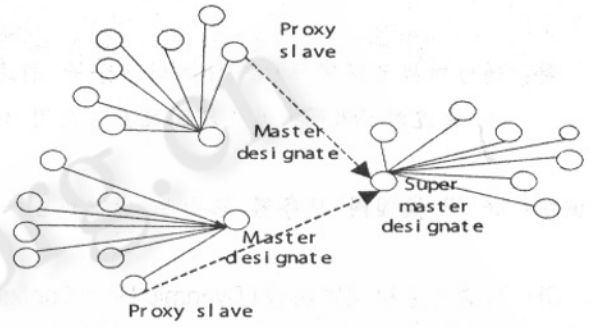


图 1 $x = k$

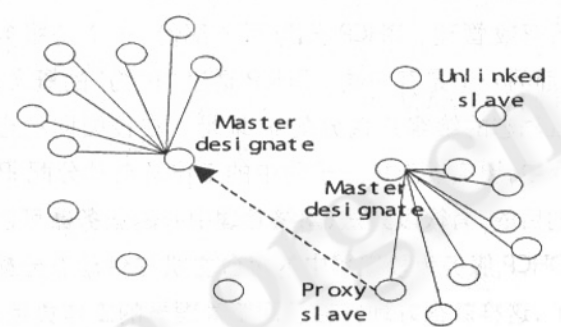


图 2 $x < k$

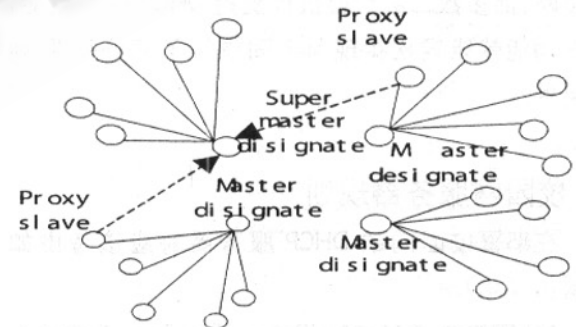


图 3 $x > k$

4 结束语

本文设计和实现了一种蓝牙分散网的形成算法。
(下转第 72 页)

在结点数未知且高密度的局部通信范围内,算法不是盲目构建分散网,而是有目的且有选择性的分区构建,这样为构建分散网节省了时间。为了创建一个更加有效的分散网,今后的研究方向该考虑三维空间中蓝牙设备的分散网进一步组建的问题。

参考文献

- 1 金纯,许光辰,孙睿. 蓝牙技术. 北京:电子工业出版社,2001.
- 2 T. Salonidis, P. Bhagwat, L. Tassiulas, et al. Distributed topology construction of Bluetooth personal area networks. In: Proc. IEEE INFOCOM '2001. Los Alamitos. CA:IEEE Computer Society Press. 2001. 1577 - 1586.
- 3 R. Lakshmi, K. Manika, S. Abhinanda. et al. Clustering algorithms for wireless ad hoc networks. In: Proc. DIAL. M 2000. New York: ACM Press, 2000. 54 - 63.
- 4 Z. F. Wang, R. J. Thomas, Z. Hams. Bluenet - A new scatternet formation scheme. In: Proe. HICSS - 35 '02. LosAlmitos, CA: IEEE Computer Society Press, 2002. 61 - 69.
- 5 S. Basagni, C. Petrololi. Multihop scatternet formation for Bluetooth network. In: Proe. IEEE VTC'2002. Los Alamitos. CA: IEEE Computer Society Press, 2002. 424 - 428.
- 6 Jiawei Han, Micheline Kamber. 数据挖掘概念与技术. 机械工业出版社, 2001, 8.
- 7 G. S. Fishman, Monte Carlo. Concepts, Algorithms and Applications. Springer - Verlag, 1996.
- 8 H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on sums of observations. Ann. Math Statist, 23:493 - 507.