注：还没有图名、表名、公式名，后续研究还在进行中。求最大延迟的算法会遇到参数组合爆炸的问题，一开始运行很慢很慢动不动就几个小时，最后优化到了半小时内得到99\*99=9801种情况的最大延迟。由于寒假没能好好控制自己，回家学习是骗人的，所以现在进度好像慢了，但宁缺毋滥，期待老师的批评指导。哦对了前面相关工作很多是参考的别人论文，从基本激活策略那里开始是自己的“产品”，我想先把这个不成熟版本给老师看一下，思路是否正确，接下来有何建议，最后当然是谢谢老师啦。

# 引言：

传感器网络是由大量低成本、低能耗、小体积的传感器节点构成的自组织网络。在恶劣环境、无人值守的区域有着广泛应用。由于传感器的能量有限，为延长使用寿命经常在低占空比的情况下工作，但由此产生了较大的发现延迟。

以往的研究主要注重于在较小占空比下尽可能缩短发现延迟。由于同步方法耗时耗能，因此网络主要采用异步方法进行通讯。目前异步方法是的基本思想是将连续时间离散化为一个个时隙，两个节点在激活槽重叠时能够发现彼此。

目前异步邻居发现算法主要分为两类：概率性算法和确定性算法。概率性算法追求在一定时间内，能够较大概率地发现邻居节点；确定性算法追求的是能够在一定时间内100%发现邻居节点。

本文首先对比了目前邻居发现的几种方法及其性能指标，然后提出一种新的激活策略。考虑到实际应用中部分节点的能量剩余问题，本文进一步提出了基于剩余能量的激活策略，充分利用能量以提高网络性能。

# 问题与挑战

## 问题说明

无线传感器网络是由多个传感器构成的自组织网络，每一个传感器都是网络中的一个节点，各节点通过无线收发机交换信息，处于无线信号范围之内的节点称为邻居。初始时刻邻居节点是未知的，互为邻居的节点只有在双方通信、交换信息之后，才能发现彼此，这个过程叫做邻居发现。

## 问题挑战

在节点拥有足够的电源供给的网络中，只要接收方一直保持监听状态，发送方周期性地广播邻居通告消息，就可以保证的邻居节点能够发现它。不同于传统网络，无线传感网络中的节点供电有限，在邻居发现协议的设计之中存在如下挑战：

1. 能量有效性

传感器网络本身固有的特性决定了传感器节点只能使用电池供电。而电池容量往往是有限的，所以每个节点应该节约用电，以延长生存时间。研究表明，节点处于发送和空闲监听状态的时间，主导了节点的能量消耗。因此，邻居发现协议，应该尽量减少节点处于开启状态的时间。

1. 实时性

传感器网络对于邻居发现了延迟有一定的要求，过长的延迟会大大降低用户体验。在能量消耗相当的条件下，延迟越小的协议性能越好。

1. 网络动态性

网络动态性包括节点的加入，退出和位置移动等。由于网络的动态性，已发现的节点可能随时失效；同时，新的节点可能随时加入。

## 问题模型

2.3.1调度模式：

传感器网络中的节点使用睡眠-唤醒调度的模式周期性地打开和关闭无线收发机。节点m的无线收发机的工作模式可以表示二值函数，当时，节点m在t时刻处于睡眠状态，当时节点m在t时刻处于邻居发现状态。

2.3.2邻居发现：

节点m与节点n互相发现的充要条件是：节点m与节点n处于彼此的无线信号范围内，存在一个时刻t使得。称节点m与节点n在t时刻发现彼此。

邻居发现问题是指：对于k个节点，如果任意两个节点都处于彼此的无线信号范围内，那么在时间L范围内总能发现彼此，L称为邻居发现的最大延迟。其数学表述如下：

2.3.3时隙模型：

将连续的时间离散为一个个时隙，当两节点A、B的唤醒时隙重合时便能够发现彼此：



## 评价标准

### 发现延迟

邻居发现延迟是指从两个节点进入彼此通信范围内的时刻起，到彼此接收到对方的邻居通告消息的时刻为止的时间间隔长度．发现延迟通常由具体应用指定，实时性要求较高的应用倾向于指定较小的发现延迟，以保证在指定时间内发现所有邻居．

发现延迟进一步可以细分为最坏发现延迟和平均发现延迟．最坏发现延迟是指发现最后一个邻居所需的时间，用于评价算法的理论边界．平均发现延迟是所有邻居发现时间的期望，用于评价算法的实际性能．

在最坏发现延迟相同的情况下，两种不同的邻居发现协议在不同的时间段发现的邻居数量也存在着差异．一般认为在初始阶段能更快地发现邻居的协议较优，这在发现延迟累积分布图中得以体现．如图【】所示，横坐标为时间轴，纵坐标为已发现邻居数占总邻居数的比率，可以看到虽然A、B两种协议发所有邻居所需要的时间相同，但是协议A的发现速度明显快于协议B，因此协议A的平均发现延迟小于协议B。



### 能耗

传感器网络中能耗决定了节点的生存时间，邻居发现协议应尽可能减少能耗以延长节点的寿命．节点的实际能耗往往难以直接计算，由于无线收发机唤醒时的能耗远高于节点的其他能耗，因此一般使用无线收发机的唤醒时间与总时间的比值作为评价节点能耗的标准，占空比越小，说明节点处于唤醒状态的时间越短，能耗越低。

### Power-Latency乘积

能耗和发现延迟是邻居发现协议设计中一对互相制约的因素．一般而言，发现延迟越短则这种邻居发现协议的效率越高，但往往它的能耗也越大，反之亦然．邻居发现协议设计的难点在于权衡这两个因素，在发现延迟能满足应用需求的前提下保证能耗尽可能低．为了综合分析发现延迟和能耗这两方面因素以比较不同邻居发现协议的优劣，文献【】提出了将两者的乘积作为综合衡量邻居发现协议性能的评价标准．设邻居发现调度模式的周期为T，最坏发现延迟为L，一个周期内的平均能耗为P，即

从而能耗与发现延迟的PL乘积为

采用时隙模型时：

# 相关工作

按照调度模式的确定性，邻居发现协议可以分为概率性协议和确定性协议。根据调度模式是否相同，邻居发现协议可以分为对称性和非对称性协议。

## 3.1概率性邻居发现

McGlynn等人提出Birthday协议**[1]**，每个节点在每个时隙上分别以概率、、进入发送、收听与睡眠状态。节点A发现节点B的条件是A在B的无线信号范围内，在t时刻节点A处于收听状态而节点B处于发送状态，那么在n个时隙内节点A发现B的概率为:

## 3.2确定性邻居发现：

### 3.2.1 对称邻居发现

在对称邻居发现中，每个节点采用的唤醒调度模式相同，但由于各个节点的启动时间不同或受时钟漂移的影响，节点间存在时间相位差．在实际应用中，使用对称邻居发现的前提是网络中所有节点的能耗需求比较类似，因此可以采用相同的唤醒调度模式。

对称邻居发现的数学描述是：对于任意节点对，存在时刻t使得，其中为节点i与节点j的相位差。

### 3.2.2 非对称邻居发现

1. Disco协议

Dutta等人提出Disco协议[2]。该协议基于中国剩余定理：当两个节点的唤醒调度周期互质时，其乘积范围内必然有重叠时隙。



Disco算法采用两个互素的数a、b作为节点A、B的激活参数。时隙编号从0开始，当时隙编号是a的整数倍时，A节点在此刻被激活，B节点同理。考虑到a=b时，两节点可能永远发现不了彼此，提出了以素数对、作为时序布置的标准。在保证100%发现彼此的前提下，平均发现延时为,占空比为。

1. Todis协议

Disco协议使用质数保证互质性，但10000里面只有1264个质数，参数的选择范围受到极大限制。Chen等人[3]针对Disco算法的这一不足提出了Todis算法，选择连续奇数的三元组作为唤醒调度参数，在一定条件下，保证互质性的同时增加了 参数的可选择空间。作者通过计算证实了最小不满足互质性的连续三元组为和，但这两组参数对应的占空比远小于实际占空比,基本用不上，故而Todis协议是有效的。

1. Searchlight

SearchLight协议[4]提出倍数方法，假设A节点的唤醒周期为T，B节点的唤醒周期为nT，那么A节点重复n次后调度周期和B节点等长，只需要在nT时间内能够保证邻居发现即可。基于此，作者提出在集合}中依据需求选择节点调度周期参数。只需要在较长周期中有连续覆盖T个时隙即可保证邻居发现。



1. Quorum协议

Quorum协议在的时隙矩阵中选取一行一列作为唤醒时隙

1. Hedis协议

Chen等人[5]提出Hedis协议，将调度周期设置为n(n-1)的矩阵，其唤醒调度定义如下：

若两节点选择参数m、n，只要m、n拥有相同的奇偶性就可以保证相互发现，平均发现延迟为。

1. Hello框架

Wei Sun等人将Quorum、Disco、U-Connect、Searchlight协议纳入一个统一的框架——Hello之中。作者定义的时隙矩阵，第一列为“锚”节点，全部激活，其余各列的激活节点称之为探测节点。

在对称情况下，只要探测节点完整遍历连续个 个连续位置，能够保证两节点互相发现，最大延迟上限为。



在非对称情况下，作者证明只要c为素数，便一定能保证两节点相互发现。最大延迟上限如下：

1. 各方法的对比

|  |  |  |
| --- | --- | --- |
| 方法名 | 平均延迟 | 支持的占空比 |
| Disco | , |  |
| Todis |  |  |
| Searchlight |  |  |
| Quorum |  |  |
| Hedis |  |  |
| Hello |  |  |

# 基本激活策略

**问题1**：是否存在一种策略，能够使用较小的占空比但能保证邻居发现？

**猜想1**：令A、B节点调度模式分别为：

当与的最大公倍数不超过b时(即), A、B一定能够发现彼此。

**实例**：



图1 基本可行策略(蓝色代表被激活)



**理论证明：**

如图[]所示(固定B节点，其偏移量为0)。在上文选择的激活策略下：

1. A节点一共有两类激活点:

d=0时：和;

d=1时：和。

1. B节点一共有三类激活点：

、、。

称图中激活点中的表达式值为类别号，那么,成立，，。



假设A、B在t时刻可以相互发现，A的相位差为d，则如下方程组有解：

该方程组有解的充要条件：

（1-3）

式1-3有解的充要条件：

(1-4)

而1-4恒成立，证明如下：

若存在,则b+1与有公共质因子，而与b有相同质因子，即b与b+1有公共质因子m，且m>1。显然不成立，即不存在这样的k。则。但不保证。满足j取值范围可行的一个充分条件是：序列恰好是的完全剩余系。

根据**推论1**可知：取，,。当且时满足条件。

**问题2**：基本激活策略下，邻居发现的最大延迟是多少呢？

概念说明：

* **相位差**：定义节点的第一个激活点为，以为参考（节点A相位差为0），的相位差可能是：。
* **局部最小延迟**：在给定相位差下，A、B两节点第k次相互发现的时刻记为,其中最小的即为局部最小延迟，记为。规定当A、B无法相互发现时，。
* **延迟上限**：给定A、B的激活调度策略，对于每一个相位差都会存在一个，延迟上限。
* **偏差**：在给定相位差下，激活点与最邻近的激活点的时间差称为与的偏差，且。

使用如下样例进行解释：



时：



时：



以此类推取遍集合。(如果以B0为参考，取遍集合)

* 与、、的关系

对于每一个:在原始相位差的基础上附加一个得到（），存在一个使得A、B在处第k次相互发现。函数表述为：

这样得到的集合为。，以A0为参考时。且预设为正无穷，根据最大延迟的定义得到：





* 伪代码

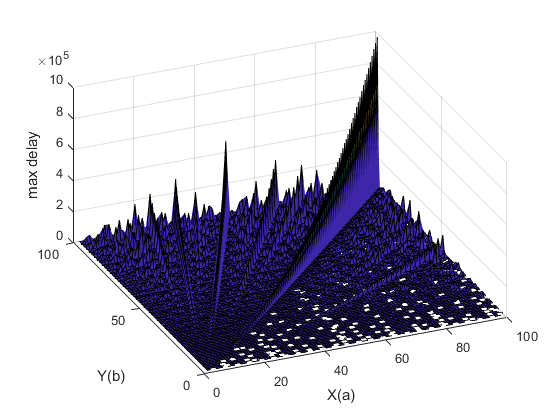
初始：series1为节点A调度策略的行向量表示，series2为节点B调度策略的行向量表示。

|  |
| --- |
| num1= length(series1);num2= length(series2);  if num1>num2:  all\_theta=set(range(0,num2-1));  delay=[];  for each ai in A’s activated node:  series2\_= move series2 ai units to left  theta=set(index(series2\_));  all\_theta-=theta;  delay+=[(ai,th) for th in theta];  if all\_theta is None:  return max([k[0] for k in delay]);  return inf;  end  else:  all\_theta=set(range(0,num2-1));  delay=[];  for each bi in B’s activated node:  series1\_= move series1 bi units to left  theta=set(index(series1\_));  all\_theta-=theta;  delay+=[(bi,th) for th in theta];  if all\_theta is None:  return max([k[0] for k in delay]);  return inf;  end  end |

* 结果分析：

将a、b的取值集合设为，支持的占空比范围为,所构成的组合共有个。经算法检验，共有193个组合不满足的条件，即有9608个可行组合。

计算并得到a-b-max delay的分布图如下：



由图可知，当时延迟较大，且最大延迟为970200，在处取得。至此我们已经知道了内的所有情况，对于更多的情况(比如)我们是否能得到呢？最简单的方法是穷举计算，但这个问题在多项式时间内难以解决，将出现组合爆炸的情况。

为预测在未知参数下的发现延迟， 我们采用曲线拟合进行合理外推：先使用,的数据进行拟合，然后利用余下数据进行检验。

由于数据存在规律性的波动：



所以可分和两类进行拟合：



拟合结果：

拟合效果：



拟合结果：

拟合效果：

拟合检验：使用的数据对拟合结果进行检验，对比图如下：



由图可知预测效果很好，在时可以使用如下公式估计最大延迟：

同理对于其它情况也能得到估计公式，见附录[]。

**问题3**：当节点有剩余能量时,如何改进调度策略使得发现延迟进一步减小？

针对此问题有两种解决方案：

1. 使用基本激活策略但改变节点参数。
2. 保持原有参数增加激活点。

**策略（a）**:

只需要减小节点参数a、b即可大大降低发现延迟，但应尽可能避免如下配置情况：

**策略（b）**：

首先对基本激活策略进行进一步分析。调度矩阵中第一列为锚点，使用字母G表示，其它列的激活点为探测点,使用字母P表示。

假设,在不改变b的情况下在B节点的调度矩阵中增加激活点以减小发现延迟：

当时：



类似一个追击场景，A的锚点以速度3“追击”B的锚点，当超越单位k时，若前方k处恰有探测点，则A、B相互发现。

这里我分析出来有点奇怪，如果就增加激活点，随机加入就行了，没有一个或几个点存在相对优势，大家在概率上是地位同等的。还不敢写。

# 参考文献

[1]McGlynn M J, Borbash S A. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks[C]//Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing. ACM, 2001: 137-145.

[2] Dutta P, Culler D. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications[C]//Proceedings of the 6th ACM conference on Embedded network sensor systems. ACM, 2008: 71-84.

[3] Chen, Lin, et al. "On heterogeneous duty cycles for neighbor discovery in wireless sensor networks." *Ad Hoc Networks* 77 (2018): 54-68.

[4] Bakht, Mehedi, Matt Trower, and Robin Hilary Kravets. "Searchlight: Won't you be my neighbor?." *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012.

[5] Chen, Lin, et al. "On heterogeneous neighbor discovery in wireless sensor networks." *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015.

[6] Sun, Wei, et al. "Hello: A generic flexible protocol for neighbor discovery." *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014.