移动低占空比无线传感网中低能耗的主动邻居发现算法

梁俊斌，周翔，李陶深

（广西大学计算机与电子信息学院广西多媒体通信与网络技术重点实验室，广西 南宁 530004）

摘 要：移动低占空比无线传感网（MLDC-WSN）是近年新兴的一种无线多跳网络，它由大量具有移动能力且会长时间进入睡眠状态的节点自组织而成，可以部署在恶劣环境中执行长期的监测任务，在国防、工业、农业等领域具有广泛的应用前景。但是，节点的移动和睡眠导致网络拓扑不断发生改变，使节点很难以较少的能耗快速发现其全部的邻居，导致节点无法获得最优的分布式决策结果，影响网络应用的效果。为了解决这个难题，提出一种新的主动式邻居发现算法。该算法使网络中的节点在苏醒时主动寻找自己的邻居，避免传统被动式邻居发现中长时间等待所产生的时延。此外，通过对邻居移动速度及距离的预测，快速确定未来下一时刻的邻居集合，在进一步减少时延的同时获得更准确的邻居发现结果。理论分析和实验结果表明，与已有算法相比，所提算法能够在 MLDC-WSN 中以更小的能耗、更低的时延发现全部的邻居。

关键词：移动低占空比无线传感网；低能耗；邻居发现

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000-436x.2018054

Energy saving proactive neighbor discovery algorithm in mobile low-duty-cycle wireless sensor network

LIANG Junbin, ZHOU Xiang, LI Taoshen

Guangxi Key Laboratory of Multimedia Communications and Network Technology,
School of Computer and Electronics Information, Guangxi University, Nanning 530004, China

Abstract: Mobile low-duty-cycle wireless sensor network is a new kind of wireless multi-hop network, which is self-organized by a large number of nodes that have mobile ability and are able to get into sleep for a long time. Such networks have wide application prospects in national defense, industry, agriculture and other fields that need long term monitoring in severe environments. However, the movement and the sleeping features of nodes lead to constantly change of network topology, which makes the nodes difficult to discover their neighbors quickly. Therefore, the nodes cannot achieve optimal distribution decisions. In order to solve this problem, a new proactive neighbor discovery algorithm was proposed. This algorithm made the nodes in the network take the initiative to find their neighbors when they woke up, and avoided the delay caused by long time waiting in the traditional passive neighbor discovery. In addition, by predicting the movement speed and distance of neighbors, the neighbor set at the next moment can be quickly determined, which can further reduce the delay and obtain more accurate neighbor discovery results. Theoretical analysis and experimental results show that compared with the existing algorithms, the algorithm can find all the neighbors in MLDC-WSN with less energy consumption and lower delay.

Key words: mobile low-duty-cycle wireless sensor network, energy saving, neighbor discovery

---

收稿日期：2017-09-05；**修回日期**：2018-03-16

基金项目：国家自然科学基金资助项目（No.61562005, No.61762010, No.61363067）；广西自然科学基金资助项目（No. 2015GXNSFAA139286）；广西高等学校千名中青年骨干教师培育计划基金资助项目（桂教人(2017) No.49）

Foundation Items: The National Natural Science Foundation of China (No.61562005, No.61762010, No.61363067), The Natural Science Foundation of Guangxi Zhuang Autonomous Region (No. 2015GXNSFAA139286), The Cultivation Plan for Thousands of Young and Middle-Aged Backbone Teachers in Guangxi Higher Education School (Guangxi Education People (2017) No.49)

1 引言

移动低占空比无线传感网（MLDC-WSN, mobile low-duty-cycle wireless sensor network）是近年出现的一种新型自组织网络。它与传统的无线传感器网络（WSN, wireless sensor network）一样，由大量能量、通信范围、存储容量和计算能力均有限的节点组成，主要部署在人类无法进入的恶劣环境中，从事长期的监测或跟踪任务[1,2]。但是，它与WSN有明显的区别：WSN中节点是静止且保持苏醒的，而MLDC-WSN中的节点会长时间进入睡眠状态以保存能量，仅在少量时刻苏醒并通过移动来执行任务[3]。因此，MLDC-WSN 的拓扑变化会更加频繁，导致节点邻居也不时地发生改变。

邻居发现是网络中的一个重要的操作，网络中的节点需要快速地相互发现，才能自组织形成网络。由于邻居发现在移动的网络中需要周期性多次执行，因此，以较小的能耗实现快速的邻居发现，是网络能正常开展工作及延长生命周期的重要保障[4,5]。在MLDC-WSN中，节点采用低占空比工作模式（即节点在每个工作周期中的大部分时间处于睡眠状态，仅在少量时刻苏醒），而节点间的邻居发现又要求节点同时处于苏醒状态，使节点等待被发现的邻居苏醒所需要的时间很长，造成巨大的时延。此外，节点苏醒时的移动会导致网络拓扑结构不断改变，使节点的邻居也会不时发生变化。因此，实现低时延、低能耗的邻居发现非常困难。

目前，已有的典型工作是采用节点主动苏醒的方式来进行邻居发现，但是这种方式要求节点多次主动苏醒，从而耗费较多的能耗。如何减少主动苏醒的次数、降低能耗，同时快速地完成邻居发现，仍然是一个亟待解决的难题。

本文提出了一种新的低能耗的选择性主动苏醒邻居发现（SPND, energy saving selectively proactive neighbor discovery）算法。在SPND算法中，节点能够结合网络中节点的移动模型，根据集合的划分，选择性地在邻居苏醒的时刻同时苏醒，与邻居节点进行确认。例如，节点 i 在某时刻已经拥有了自身的邻居集合，网络中节点经过一段时间的移动后，某些节点已经不再是节点 i 的邻居。在下一时刻，节点 i 只需选择性地主动苏醒发现那些可能成为自身邻居的节点。

2 相关工作

根据唤醒调度模式的确定性，邻居发现算法可以分为概率性邻居发现和确定性邻居发现。在确定性邻居发现算法中，根据节点是否主动苏醒发现邻居，又可以进一步划分为主动式邻居发现和被动式邻居发现。

2.1 概率性邻居发现

概率性邻居发现算法采用随机唤醒的调度方式，通过设置节点在每个时间片所处状态的概率，可以获得较低的平均发现时延，但是最坏情况下发现时延可能会非常大。Mcglynn 等[6]提出了一种典型的概率性邻居发现算法 Birthday，它利用生日悖论的概率性原理，即在随机抽取 n 个人组成的集合中，当 n 很大时，有人生日在同一天的概率会很大。因此，节点以某种概率来选择当前时隙节点的状态（包括睡眠、发送、侦听），通过细致地设置各个状态的概率值，节点在一段连续时间内会有很高的概率可以相互发现。该算法能够实现邻居发现过程中能量消耗和发现时延之间的良好平衡，但它无法限定最坏情况下的发现时延。针对这个问题，You 等[7]提出了 ALOHA-Like 算法，它分析任意节点发现 $n-1$ 个邻居节点的期望时间，有很高的概率能限定最坏情况下的发现时延。

2.2 确定性邻居发现

确定性邻居发现中，节点采用固定的唤醒调度模式，即节点处于睡眠或苏醒状态按照预定的睡眠苏醒调度表周期性地重复。

2.2.1 被动式邻居发现

被动式邻居发现算法是指节点完全按照预定的苏醒时刻苏醒进行邻居发现。一个典型的被动式邻居发现算法是 Jiang 等[8]提出的 TQS（torus quorum system）邻居发现算法。该算法将节点工作周期编排成 $t\times w$ 矩阵，工作周期长度为 $n=t\times w$。节点任选其中一列 c 的所有元素，再从所选 $c+i(i=1,\cdots,\frac{w}{2})$ 列任意位置选择 $\frac{w}{2}$ 个元素，作为节点的苏醒时间片。TQS 能实现较优的能耗时延指标，但是不适用于每个节点采用不同的唤醒调度模式的情况。为了解决这个问题，Zheng 等[9]实现了一种不需要时隙对准的异步苏醒协议（AWP, asynchronous wakeup protocol）算法，该算法中节点可以根据其在网络中的不同角色灵活设置占空比。

AWP 将邻居发现问题简化为一个图的最小顶点覆盖问题，而解决图的最小顶点覆盖问题都是集中式的方法，但集中式的方法在节点分散的无线传感网中并不适用。

针对 TQS 算法及 AWP 算法的不足，Dutta 等[10]提出了 Disco 算法。Disco 算法中每个节点选择 2 个素数作为其工作周期，节点大部分时间处于睡眠状态，每个节点拥有一个独立的计数器，一旦节点计数器能够整除其任何一个素数工作周期，使该节点处于苏醒状态。根据中国剩余定理[11]，2 个节点的苏醒时隙必然能够周期性地重叠或部分重叠，Disco 算法能够确保一直在邻居范围内的 2 个节点能够在一定的时限内相互发现。

为了统一解决节点唤醒调度模式异步和同步的问题，Kandhalu 等[12]提出了 U-Connect 邻居发现算法。U-Connect 算法选择素数 q 作为其基本工作周期，然后，在连续 T 个时隙内构建 $q\times q$ 的网格矩阵（$T=q\times q$），并在矩阵内任选某列和某行的一半时隙作为节点的苏醒时隙，当该行后面的时隙数不足一半时，返回到该行的首列继续选择。U-Connect 算法融合了中国剩余定理和 TQS 算法的思想。U-Connect 算法的能耗—时延指标是最优理论系统的 1.5 倍左右，在能耗时延均衡方面优于 TQS 算法和 Disco 算法。

2.2.2　主动式邻居发现

已有的邻居发现算法有很多，但是它们都基于节点被动式苏醒进行邻居发现。Chen 等[13]提出了一种节点主动苏醒的邻居发现算法 Q-connect。该算法分析了节点从睡眠状态切换到苏醒状态所需的能耗，考虑到该部分能耗不可忽视，Q-connect 算法将节点时间片细分为几个部分，在第一部分主动苏醒广播 beacon 消息，在第四部分主动苏醒接收 beacon 消息。Q-connect 算法能提高网络的能效利用率，但是节点平均发现时延较大。

为了充分利用 beacon 消息减少发现时延，Qiu 等[4]提出了一种主动式的邻居发现算法 Nihao。Nihao 算法基于“多说少听”（TMLL, talk more listen less）的原则，节点在一个周期（$m\times n$）的前 m 个时间片处于“听”（苏醒）状态，并在整个周期中每隔 m 个时间片进入“说”（主动苏醒发送一个 beacon 消息）状态，即一共发送 n 个 beacon 消息。在 Nihao 算法中，beacon 消息可以在节点睡眠状态主动苏醒发送，因此，使节点在较少时间片内相互发现，减少发现时延。

Kindt 等[14]提出了一种主动式邻居发现算法 Griassdi。在 Griassdi 算法中，节点周期性地苏醒与发送 beacon 消息，且苏醒与 beacon 消息的发送相互独立。节点在某个时间片内苏醒收到其他节点的 beacon 消息后，会根据其他节点的下一次苏醒时间，调整自身下一次发送 beacon 消息的时刻，以实现快速与其他节点相互发现。

这部分算法都要求节点在一个时间片的部分时刻发送 beacon 消息，Chen 等[15]提出了一种不需要发送 beacon 消息的节点主动苏醒发现邻居 GBD（group-based discovery）算法。在 GBD 算法中，相互发现的节点位于一个组中。组中的节点根据移动低占空比网络的时空特性，选择性地将它们部分现有邻居的睡眠和苏醒时刻分享给新发现的加入组中的邻居节点。新发现的节点就能快速地获得周围节点的苏醒时刻表，在周围节点苏醒的时刻主动苏醒，从而判断 2 个节点是否是邻居。针对节点密度较大的网络，Chen 等[15]又对 GBD 算法进行改进，提出了 AGBD（advanced group-based discovery）算法。在 AGBD 算法中，节点遵循一个基于移动网络时空特性的选择机制，这种机制能减少节点部分不必要的主动苏醒，但是可能会漏掉邻居节点。GBD 算法的优点是平均发现时延较低，节点能在潜在的邻居节点苏醒时主动苏醒进行邻居发现，减小了网络低占空比特性增大的发现时延。但是，GBD 算法存在 2 个方面的不足。

1) 节点选择性地主动苏醒进行邻居发现，无法限定最坏情况的最大发现时延。

2) 节点基于概率的选择性转发会降低节点发现邻居的可靠性。

以上是针对低占空比网络的一些典型邻居发现算法，可以发现，概率性邻居发现算法能取得较低的网络能耗且以较小的时间发现大部分邻居，但是无法限定最坏情况下的全网最大发现时延。确定性邻居发现中，被动式邻居发现算法时延往往较大，主动式邻居发现算法能取得较小的发现时延，但是需要节点主动苏醒或主动发送 beacon 消息，网络能耗较大。本文将针对这些算法的优缺点进行建模，并提出优化的解决方案。

3　系统模型和问题描述

本节将给出 MLDC-WSN 的网络模型及本文研究

问题的描述，同时对相关术语进行定义。

3.1 相关定义

定义 1 通信半径[16]。节点以一定功率通信能保障消息被其他节点接收到的最远距离。

定义 2 节点的发现概率[17]。距离在通信范围内的 2 个节点能相互发现的概率。

定义 3 节点的发现比率[17]。网络的任意节点发现的邻居节点数目占所有通信范围内节点数目的比率。

定义 4 发现时延[17]。从节点移动到通信范围内的时刻开始到它们相互发现时的时间间隔。

3.2 网络模型

假设一个拥有 n 个移动节点的无线传感器网络 $G_{\text{net}}=\{V, E\}$，其中，V 表示网络中节点的集合，E 表示网络中节点间是否连通的信息。G_{net} 中节点采用低占空比工作模式，将移动节点的工作状态定义为睡眠（sleep）—苏醒（active）状态，如图 1 所示，即节点大部分时间处于 sleep 状态，关闭除去定时器以外所有其他功能模块，只在少部分时间，如第 3、12 个时间片处于 active 状态，感知数据和通信。节点能在任何时候主动苏醒发送数据分组，但是只能在计划的苏醒时刻接收数据分组。

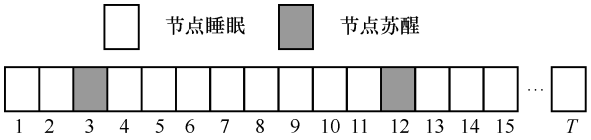


图 1 占空比示意

为了考虑网络分布较为均匀的情况，网络的节点采取 Random Waypoint 移动模型。在 Random Waypoint 移动模型[18]中，每个移动节点随机选取一个方向作为目标方向，然后选定一个范围在$[0, v_{\max}]$的恒定速度 v 移动，其中，$v_{\max}$ 为节点能移动的最大速度。节点间方向及速度的选取相对独立。网络中节点在特定的时间段 Δt 内移动，节点在一个 Δt 时间段内移动的距离范围为$[0, v_{\max}\times\Delta t]$。

3.3 问题描述

网络中的节点 i 与节点 j 能在某个时刻 t 实现相互发现需要满足 2 个条件。

1) 在某个时间段 Δt 内，节点 i 与节点 j 在通信范围内，假设网络中节点的通信半径一致为 R。

2) 节点 i 与节点 j 在 Δt 内同时处于苏醒状态，且 Δt 不小于节点实现邻居发现所需的最小时间。

节点间同时满足以上 2 个条件，才能被称为节点 i 与节点 j 能在$[t, t+\Delta t]$实现相互发现。

在 MLDC-WSN 中，节点的邻居发现时延是指从 2 个节点可以相互通信的时刻 t_0 开始，到它们完成相互发现的时刻 $t+\Delta t$ 为止的时间间隔 δ，即

$$\delta=t+\Delta t-t_0 \tag{1}$$

节点的发现概率是指在通信范围内的 2 个节点 i 和节点 j 能相互发现的概率 P_{ij}，P_{ij} 的取值范围为[0,1]。$P_{ij}(t)=0$ 表示节点在 t 时间内无法相互发现，$P_{ij}(t)=1$ 表示节点在 t 时间内肯定能相互发现。用比率 $P^i(t)$表示节点 i 在 t 时间内发现的邻居节点占所有通信范围内邻居的比率。因此，MLDC-WSN 中邻居发现问题可以归纳为以最小的能耗实现发现时延 δ 的最小化与以最小的时间实现发现比率 P^i 的最大化问题，即

$$\{\min \delta, \max P^i\} \tag{2}$$

其中，$\delta=t+\Delta t-t_0$，由于 Δt 通常是常量无法调节，只有 $t-t_0$ 可以通过算法进行优化。在已有的能实现较低发现时延的算法中，通常无法使 P 接近 1。本文提出一种新的节点主动式邻居发现算法 SPND，能实现低能耗的快速邻居发现，以下是其详细的描述。

4 算法设计

本节将给出 SPND 算法的基本思想及详细设计，并从理论上将 SPND 算法和已有的邻居发现算法进行对比。考虑到已有的典型算法采用了节点的主动苏醒来减小邻居发现时延，SPND 算法专注于结合网络移动性模型，减少节点主动苏醒次数，从而实现低能耗的主动式邻居发现。

4.1 算法基本思想

在 SPND 算法中，对于刚部署的传感器网络，每个节点分布式地开展邻居发现工作。针对移动传感器网络的特点，将节点的邻居发现分为 2 步。步骤 1 根据基于组的邻居发现算法构建节点初始邻居集合，依据这样一个原则：新加入组中的邻居节点将自身的邻居集合信息分享给组中的其他节点，组中的其他节点就能快速地获取潜在的邻居节点的苏醒时间，从而在该时间主动苏醒进行邻居发现。步骤 2 结合节点的移动模型，选择性地指定节点的主动苏醒时刻，节点主动苏醒发现该时刻的邻居集合，依据这样一个原则：网络中节点经过一段时间的移动后，节点只选择那些移动后可能在通信范围内的潜在邻居节点进行主动

式苏醒的邻居发现；将肯定在邻居范围内的节点按既定的睡眠苏醒状态进行邻居发现；将肯定移动出邻居范围内的节点剔除出邻居集合。

4.2　算法详细设计

SPND 算法分为 2 个阶段。网络刚部署，构建节点初始邻居集合为第一阶段。选择性地指定节点的主动苏醒时刻，节点主动苏醒发现该时刻的邻居集合为第二阶段。网络中的节点分布式地完成第一阶段的邻居发现过程，然后节点间完全异步地从第一阶段切换到第二阶段。

4.2.1　构建初始邻居集合

MLDC-WSN 部署后，首先在 t 时刻，网络中任意一个节点 i 分布式地生成自己的邻居集合 $G_t^i=\{i\}$。当 G_t^i 中节点个数为 1 时，节点 i 将依次在自己的苏醒时刻发送广播消息来发现邻居，直到在某个时刻发现了第一个邻居节点 k，并将节点 k 加入集合 G_t^i 中，此时，节点 i 的集合 $G_t^i=\{i,k\}$ 中节点个数为 2。当 $G^i\geqslant 2$ 时，如图 2 所示，节点 i 按以下步骤进行下一步的邻居发现。

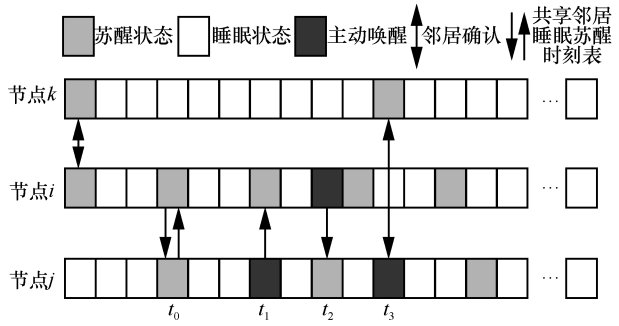


图 2　构建初始邻居集合

1) 网络中的每个节点在苏醒时刻会广播一个消息，告知网络自己的存在。如图 2 所示，在某个时刻 t_0，节点 i 与节点 j 在通信范围内，收到了彼此的广播消息，即节点 i 和节点 j 相互发现成为邻居节点并能获取彼此的睡眠苏醒时刻表。

2) 节点 j 将在节点 i 的下一次苏醒时刻 t_1 主动苏醒，将已经位于 G^j 中的邻居节点的睡眠苏醒时刻表信息发送给节点 i。节点 i 则会在节点 j 的下一次苏醒时刻 t_2 将已经位于 G^i 中的邻居节点（如节点 k）的睡眠苏醒时刻表信息发送给节点 j。

3) 节点 j 在收到节点 i 的邻居信息后，将依次在节点 i 的邻居节点（如节点 k）的下一次苏醒时刻 t_3 主动苏醒，然后发送消息，来确认该节点是否是节点 j 的邻居。节点 i 以同样的方式确认节点 j 的邻居是否是节点 i 的邻居。

具体算法如算法 1 所示。

算法 1　构建初始邻居集合

```
1) for (网络中每一个节点 i) {
2) while (t<T) {
3)    if (节点 i 苏醒发现节点 h) {
4)          节点 i 和节点 h 共享 G;
5)    } else {节点 i 睡眠}
6)    if (节点 h 的邻居苏醒) {
7)          节点 i 主动苏醒;
8)          节点 i 与节点 k 确认;
9) } }     }
```

每当节点 i 的邻居集合 G^i 加入新的邻居节点，循环执行与新加入节点的邻居进行确认的步骤，即算法 1 中的步骤 6)~步骤 8)，直到收到的所有邻居信息都被确认完毕，完成第一阶段的初始邻居集合构建。

4.2.2　选择性指定苏醒时刻

节点构建完初始邻居集合即进入第二阶段的邻居发现工作，即当节点 i 周围有节点移动后，为节点 i 选择性地指定其后主动苏醒进行邻居发现的时刻。网络中节点采用 Random Waypoint 移动模型，即节点每次移动的最大距离为 $\Delta S=v_{\max}\times\Delta t$，方向随机。在网络中，假设节点的通信半径一致为 R。节点 i 在某个时刻 t 的邻居集合为 G_t^i，经过一段时间 Δt 的移动后，节点 i 的新邻居集合为 $G_{t+\Delta t}^i$。如图 3 所示，根据网络模型可知，G_t^i 中与节点 i 的距离在 $[0, R-\Delta S)$范围内的节点肯定在 $G_{t+\Delta t}^i$ 中，与节点 i 的距离在$[R+\Delta S, +\infty)$范围内的节点肯定不在 $G_{t+\Delta t}^i$ 中，距离在$[R-\Delta S, R+\Delta S)$范围内的节点可能在 $G_{t+\Delta t}^i$ 中。因此，对于节点 i 而言，在经过 Δt 时间移动后，下一次进行邻居发现时，可以只需考虑距离在$[R-\Delta S, R+\Delta S)$范围内的节点。

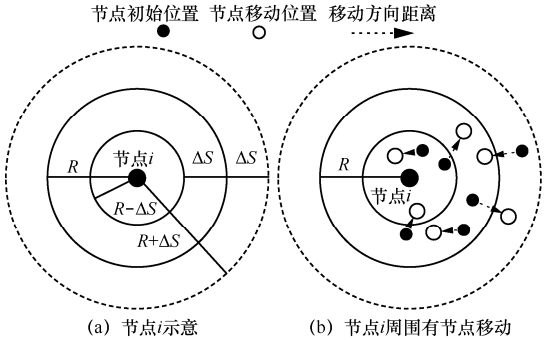
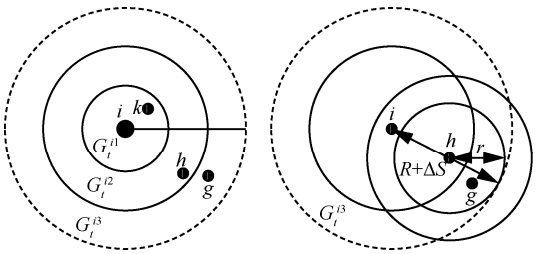


图 3　节点移动示意

传感器网络中，节点可以通过 RSSI 信号强度来大致估计互为邻居的节点间的距离。对于已经获取了邻居集合 G_t^i 的节点 i 而言，如图 4(a)所示，可以根据节点间的距离将邻居集合 G_t^i 划分为 2 个子集合 G_t^{i1} 和 G_t^{i2}。其中，集合 G_t^{i1} 中节点 k 具有的性质是节点 k 与节点 i 之间的距离 l_{ik} 在[0, $R-\Delta S$)范围内。集合 G_t^{i2} 中节点 h 具有的性质是节点 h 与节点 i 之间的距离 l_{ih} 在[$R-\Delta S$, $R+\Delta S$)范围内。



(a) 子集合划分　　(b) 获取集合G_t^{i3}中节点

图 4　集合划分示意

对于在集合[R, $R+\Delta S$)范围内的节点 g，可以通过节点 h 来获取。如图 4(b)所示，令 $r=R+\Delta S-l_{ih}$，节点 h 位于集合 G_t^{i2} 内，并且拥有自己的邻居集合 G_t^h，则需要将集合 G_t^h 中与节点 h 距离在[0, r]范围内的邻居节点的信息发送给节点 i，构建节点 i 的邻居集合 G_t^{i3}。具体算法如算法 2 所示。

算法 2　集合划分与确认邻居节点

1) for (网络中每一个节点 i) {
2)　if (节点 i 的邻居集合 G_i 非空) {
3)　　for (集合 G_i 中任意节点 j){
4)　　if (节点 i 和节点 j 的距离<$R-\Delta S$){
5)　　节点 j 加入集合 G_t^{i1};
6)　　}else if ($R-\Delta S$<节点 i 和节点 j 的距离<$R+\Delta S$) {
7)　　节点 j 加入集合 G_t^{i2};
8)　　}　}
9)　　if (G_t^{i2} 中节点 h 的邻居节点 k 与节点 h 的距离<r){
10)　　节点 k 加入集合 G_t^{i3};
11)　　}
12)　if (G_{part}^i 中节点 l 苏醒){
13)　　节点 i 主动苏醒与节点 l 确认;
14) }　}　}

节点 i 获取了 3 个集合 G_t^{i1}、G_t^{i2} 和 G_t^{i3} 后，则只取 $G_{\text{part}}^i = G_t^{i2} + G_t^{i3} - G_t^{i1}$ 中节点的苏醒时刻主动苏醒进行发现邻居的确认，将 G_t^{i1} 中所有节点及集合 G_{part}^i 中确认为邻居的节点加入集合 $G_{t+\Delta t}^i$ 中，完成在第二阶段的邻居发现过程。

4.3　理论分析

针对发现时延，将 SPND 与已有的典型算法进行对比，并分析 SPND 算法的能耗和发现概率。首先，从理论上定性地证明 SPND 算法发现时延优于已有的被动式苏醒邻居发现算法，然后定量地比较节点发现所有邻居节点时延的期望值。

4.3.1　发现时延分析

假定一对已经相互发现的邻居节点 i 和节点 j，节点 i 的邻居集合中包含节点 k，下文分析节点 j 发现节点 k 的时延。为了简化问题模型，假定节点 j 与节点 k 在一个周期开始之前恰好移动到通信范围内。设集合 C 为节点 j 和节点 k 可能的发现时间集合，当节点 i 与节点 j 以被动式的苏醒进行邻居发现时，则有 $C=\tau_j \bigcap \tau_k$，即节点 j 与节点 k 的最小发现时延为 min C。当采用 SPND 算法构建节点初始邻居集合时，节点 j 能通过节点 i 提前获取节点 k 的苏醒时刻，主动苏醒进行邻居发现，即会增加节点 j 的工作模式 τ_j 中的苏醒时间片。节点 j 新的工作模式为 τ_j'，则有 $C'=\tau_j' \bigcap \tau_k$。因为 $\tau_j \subseteq \tau_j'$，则有 $C \subseteq C'$。即选择性主动苏醒邻居发现算法的发现时延小于或等于传统的被动式苏醒邻居发现算法的发现时延。

不失一般性地，本文选取 Disco 算法作为定量比较的对象。与其他被动式苏醒算法 U-connect 和 Searchlight 一样，Disco 算法中网络中的每个节点在网络部署前规划好自身的睡眠苏醒时刻表，网络部署后，节点按时刻表苏醒进行邻居发现。其特性是能限定节点 j 和节点 k 发现的最大时延 $\delta_{\max}^{jk}$，即当节点 j 与节点 k 在通信范围内的时间 $t>\delta_{\max}^{jk}$ 时，肯定能相互发现。当 $t<\delta_{\max}^{jk}$ 时，节点 j 和节点 k 能相互发现的概率为 $p=f(j, k, t)$。即节点 j 与节点 k 在 t 时间内相互发现的概率可以表示为

$$P_{jk}(t)=\begin{cases} f(j,k,t), & t\in\left[0,\ \delta_{\max}^{jk}\right) \\ 1, & t\geqslant\delta_{\max}^{jk} \end{cases} \tag{3}$$

则节点 j 采用 Disco 算法在 t 时间内发现所有 $n-1$ 邻居节点的概率可以表示为 $P_D^j(t)=\prod_{k=0}^{n-1} P_{jk}(t)$。它的

密度函数可以表示为

$$p_D^j(t)=\sum_{k=0}^{n-1}\left[P_{jk}(t)'\prod_{h=0,h\neq k}^{n-1}P_{jk}(t)\right] \tag{4}$$

通过概率密度函数计算节点 j 发现所有邻居节点的期望时间为

$$\begin{aligned}t_D^j=E_D^j(t)&=\int_0^{\delta_{\max}^j}tp_D^j(t)\mathrm{d}t\\&=\int_0^{\delta_{\max}^j}t\sum_{k=0}^{n-1}\left[P_{jk}(t)'\prod_{h=0,h\neq k}^{n-1}P_{jk}(t)\right]\mathrm{d}t\end{aligned} \tag{5}$$

其中，$\delta_{\max}^j$ 为节点 j 发现所有邻居节点中时延最大值。

当采用 SPND 算法构建节点初始邻居集合时，节点 i 被动苏醒，在时间 t 内发现通信范围内第一个邻居节点 j 的概率 $P_S^i(t)$ 可以表示为

$$P_S^i(t)=1-\prod_{j=0}^{n-1}\left(1-P_{ij}(t)\right) \tag{6}$$

它的概率密度函数为

$$p_S^i(t)=\sum_{k=0}^{n-1}\left[P_{ij}(t)'\prod_{k=0,k\neq j}^{n-1}\left(1-P_{ik}(t)\right)\right] \tag{7}$$

通过概率密度函数计算节点 i 发现所有邻居节点的期望时间为

$$\begin{aligned}\mathrm{E}_S^i(t)&=\int_0^{\delta_{\min}^i}tp_S^i(t)\mathrm{d}t\\&=\int_0^{\delta_{\min}^i}t\sum_{j=0}^{n-1}\left[P_{ij}(t)'\prod_{k=0,k\neq j}^{n-1}\left(1-P_{ik}(t)\right)\right]\mathrm{d}t\end{aligned} \tag{8}$$

其中，$\delta_{\min}^i$ 表示节点 i 发现通信范围内第一个邻居节点的时延。节点 i 发现了第一个邻居节点后，将其加入邻居集合中，并与集合中的邻居共享邻居节点的睡眠苏醒时刻表。此后，节点 i 将主动苏醒与潜在的邻居节点进行确认，直到发现所有通信范围内的节点。因此节点 i 发现所有 n 个邻居节点的期望时间 t_S^i 满足以下条件。

$$t_S^i\leqslant\mathrm{E}_S^i(t)+2\max(T_h),h\in[0,n-1] \tag{9}$$

其中，$\max(T_h)$是节点 h 的连续 2 个苏醒时间最大间隔。节点 i 以 $\mathrm{E}_S^i(t)$ 的期望时间发现第一个邻居节点后，最多还需要 $2\max(T_h)$的时间来进行邻居确认和分享自身的邻居集合信息。第 5 节将通过仿真实验，比较 Disco 算法的期望发现时延 t_D^j 和 SPND 算法的期望发现时延 t_S^i。

4.3.2　苏醒及主动苏醒次数分析

节点用于邻居发现的苏醒时间片主要包括 2 个部分：按既定的睡眠苏醒时刻表上苏醒的时间片和主动苏醒的时间片。在 SPND 算法中，节点 j 通过邻居节点 i 获取了节点 i 的邻居后，通过增加节点 j 的工作模式 τ_j 中的苏醒时间片进行邻居发现，能减小发现时延，增大发现概率。为了减少增加的苏醒时间片带来的能耗，本文设计了选择性地指定节点主动苏醒时刻。

假设网络中节点一致分布，密度为 ρ，即网络在单位面积的范围内拥有 ρ 个节点，则在节点 i 周围距离 R 范围内平均有 $N^i=\pi R^2\rho$ 个节点。现有的节点主动式苏醒的算法中，节点 i 需要主动苏醒发现在集合 G_t^{i1}、G_t^{i2} 和 G_t^{i3} 中的所有潜在邻居节点，即平均主动苏醒 $N_t^i=\pi\left(R+\Delta S\right)^2\rho$ 次。在本文设计中，节点 i 只需主动苏醒发现在集合 G_t^{i2} 和 G_t^{i3} 中的所有潜在邻居节点，即平均主动苏醒次数为

$$N_t^{i\prime}=\left(\pi\left(R+\Delta S\right)^2-\pi\left(R-\Delta S\right)^2\right)\rho \tag{10}$$

节点 i 可以减少约 φ 次的主动苏醒次数，φ 表示为

$$\varphi=\frac{N_t^i-N_t^{i\prime}}{N_t^i}\times100\%=\frac{\left(R-\Delta S\right)^2}{\left(R+\Delta S\right)^2} \tag{11}$$

其中，ΔS 是网络中节点一次移动的最大距离，可见在 ΔS 不为 0 的情况下，节点 i 能减少的苏醒次数是很可观的一部分。

4.3.3　能耗分析

无线传感器网络中节点的能耗 E_n 主要包括 3 个部分：处理器模块的能耗 E_p、通信模块的能耗 E_c 和感知模块的能耗 E_s。处理器模块有 3 个工作状态，分别是运行态（run）、空闲态（idle）和睡眠态（sleep）。通信模块一般有 6 种状态，分别是发送（send）、接收（recv）、关闭（off）、空闲（idle）、睡眠（sleep）和信道检测评估（CCA/ED）。感知模块则只有开（on）、关（off）2 种状态。在评估网络邻居发现能耗时，可以认为感知模块独立于其他 2 个模块，因此，本文评估 SPND 算法能耗 $E_{n\text{-SPND}}$ 时主要考虑处理器模块与通信模块的能耗，即

$$E_{n\text{-SPND}}=E_p+E_c \tag{12}$$

在节点苏醒及主动苏醒的时间片内，处理器可能处于运行态或空闲态，运行态处理器一个时间片内正常执行指令需要消耗能量 $E_{p\text{-run}}$，空闲态部分功能暂停执行，能耗 $E_{p\text{-idle}}$ 相对较小。通信模块则可能处于发送、接收或信道检测评估状态。在节点处于睡眠状态时，处理器模块与通信模块也都处于睡

眠态，此时节点的大部分模块关闭，系统能耗 $E_{p\text{-sleep}}$、$E_{c\text{-sleep}}$ 最小。

根据文献[12]可知，节点在一个苏醒状态时间片内进行信道检测评估的能耗 $E_{c\text{-CCA/ED}}$ 和收发数据需要的能耗 $E_{c\text{-send}}$、$E_{c\text{-recv}}$ 基本相等。即节点在苏醒及主动苏醒的时间片内，通信模块消耗 E_c 的能量是一定的，处理器模块根据处于运行态与空闲态的不同，能耗有差异，但都大于节点处于睡眠态的处理器模块能耗 $E_{p\text{-sleep}}$。

因此，本文将 SPND 算法的能耗 $E_{n\text{-SPND}}$ 模型定义为节点处于苏醒及主动苏醒的次数 N 与节点每个时间片内苏醒的能耗 E_n 的乘积，然后加上节点处于睡眠状态的能耗，即

$$\begin{aligned}&E_{n\text{-SPND}}\\&=N\left(E_p+E_c\right)+N_S\left(E_{p\text{-sleep}}+E_{c\text{-sleep}}\right)\\&=N\left(E_{p\text{-run}}+E_{p\text{-idle}}\right)+NE_c+N_S\left(E_{p\text{-sleep}}+E_{c\text{-sleep}}\right)\\&=N_{\text{run}}E_{p\text{-run}}+N_{\text{idle}}E_{p\text{-idle}})+NE_c+N_S\left(E_{p\text{-sleep}}+E_{c\text{-sleep}}\right)\end{aligned}\tag{13}$$

其中，N 表示节点苏醒及主动苏醒次数，N_S 表示节点处于睡眠的时间片个数。N_{run} 表示 N 中处理器模块处于运行态的次数，N_{idle} 表示 N 中节点处理器模块处于空闲态的次数。E_c 表示通信模块进行一次收发数据或信道检测评估能耗的均值。$E_{c\text{-sleep}}$ 表示通信模块一个时间片内处于睡眠的能耗。

4.3.4 发现概率分析

现有的主动式苏醒算法中，GBD 算法能实现时延、能耗较优的邻居发现，本文将与该算法比较节点 i 发现所有邻居的概率。在 GBD 算法中，节点 i 根据节点间的距离 l_{ik} 和 l_{ij} 判断是否在节点 k 的苏醒时刻主动苏醒，其中，k 是节点 j 的邻居。计算节点 k 是节点 i 的邻居的概率式为

$$P_{j,ik}\left(l_{jk},l_{ij}\right)=\begin{cases}\dfrac{1}{\pi}\arccos\left(\dfrac{l_{jk}^2+l_{ij}^2-R^2}{2l_{jk}l_{ij}}\right), l_{jk}+l_{ij}>R\\1\text{ , }l_{jk}+l_{ij}\leqslant R\end{cases}\tag{14}$$

通过为 $P_{j,ik}(l_{jk},\ l_{ij})$ 设置不同的临界值来判断节点 i 是否主动苏醒发现节点 k。该算法仅通过概率来判断是否主动苏醒进行邻居发现，这会导致节点 i 可能主动苏醒发现不在邻居范围内的节点，也可能错过主动苏醒发现邻居范围内的节点，造成节点 i 在 t 时间内发现所有邻居节点的概率 $P^i(t)$ 较小。

在 SPND 算法中，节点 i 的 3 个子集合 G_t^{i1}、G_t^{i2} 和 G_t^{i3} 中，G_t^{i1} 的节点在 $t+\Delta t$ 时刻仍然是节点 i 的邻居，通过被动式苏醒可以发现。G_t^{i2} 和 G_t^{i3} 中的节点在 $t+\Delta t$ 时刻可能是节点 i 的邻居，通过主动式苏醒可以发现，即节点 i 能发现 3 个子集合 G_t^{i1}、G_t^{i2} 和 G_t^{i3} 中所有邻居节点。需要注意的是，在实际网络情况下，由于存在分组丢失率、节点移动可能频繁或速度较快、子集合 G_t^{i3} 可能无法获取$[R,\ R+\Delta S)$范围内的所有节点等问题，造成节点 i 无法在 $t+\Delta t$ 时刻发现所有邻居节点。但是这些假设的存在只会影响 SPND 算法的性能，不会改变 SPND 算法发现概率高的可靠性。

5 实验及分析

本文通过仿真实验测试 SPND 算法性能，并将 SPND 算法的实验数据与已有的典型算法的数据进行对比。为了减少实验误差造成的影响，每组实验进行 1 000 次，数据取平均值。

5.1 实验环境

本文使用 C++语言搭建了一个仿真平台，然后在该平台上分别实现了 SPND 算法与 Disco、GBD、Birthday 这 3 个对比算法的仿真。在仿真中，假设网络部署在 1 000 m×1 000 m 的范围内，由完全随机分布的 500 个移动低占空比节点组成。所有节点均在网络部署区域范围内移动，且节点的移动采用 Random Waypoint 移动模型。节点默认的平均移动速度为 1 m/s，在 1±0.3 m/s 范围内随机取值，即节点最大移动速度为 1.3 m/s，最小移动速度为 0.7 m/s。节点的占空比设置为 5%，表示节点每 20 个时间片将会有一个时间片处于苏醒状态，其他时间片处于睡眠状态。网络中节点的默认通信半径一致为 100 m。网络中节点邻居信息的 *TTL*（即邻居信息失效的时间）设置为 5 000 个时间片，每个时间片为 10 ms。这个设定是合乎实际场景的，例如，一个 CC2420 无线传感器节点收发 1 B 的数据大约需要 0.032 ms，因此，在网络完全异步的情况下，10 ms 的时间片也能在极大限度内保证 2 个节点能在一个同时苏醒的时间片内完成邻居发现工作。

5.2 性能表现

本文分别测定了网络平均发现时延、节点苏醒次数、节点能耗及限定时延内的节点发现比率这 4

个方面的数据，通过这4组仿真实验数据进行SPND算法与其他算法的对比分析。

5.2.1　网络平均发现时延

本节测定了不同网络占空比和节点移动速度情况下网络的平均发现时延。首先测定了网络中每个节点发现所有邻居节点的发现时延，然后取所有节点发现时延的平均值作为整个网络的平均发现时延。设置网络节点移动速度为 1 m/s，占空比从 1% 到 10%，得到占空比不同的发现时延如图 5(a)所示。设置网络中节点占空比为 3%，移动速度从 1 m/s 到 10 m/s，得到移动速度不同的发现时延如图 5(b)所示。

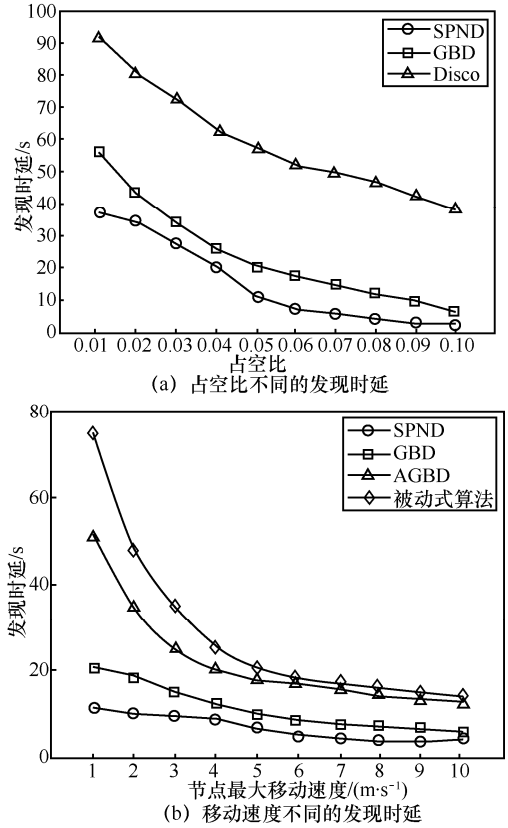


图 5　网络平均发现时延

由图 5(a)可以发现，随着节点占空比的增加，节点拥有更多的苏醒时间片进行邻居发现，能减少节点的邻居发现时延。当节点占空比达到 0.05 及以上时，SPND 算法能实现 10 s 以内的邻居发现时延。由图 5(b) 可以发现，随着节点移动速度的增大，所有算法的发现时延均有一定程度的减小。这是因为节点移动速度增大，造成节点在通信范围内的时间减少，节点能发现的邻居节点数目也就较少，因此节点发现时延减小，整个网络的平均发现时延随之减小。

5.2.2　节点苏醒次数

本节主要测定了不同占空比和移动速度情况下节点发现所有邻居节点需要的苏醒次数。这里的苏醒次数包括有节点按睡眠苏醒时刻表被动苏醒的次数以及节点根据邻居集合信息主动苏醒的次数，得到仿真实验数据如图 6 所示。由图 6 可知，SPND 算法中节点主动苏醒平均次数占节点总苏醒平均次数不到 30%，带来的额外能耗优于已有的主动苏醒邻居发现算法。

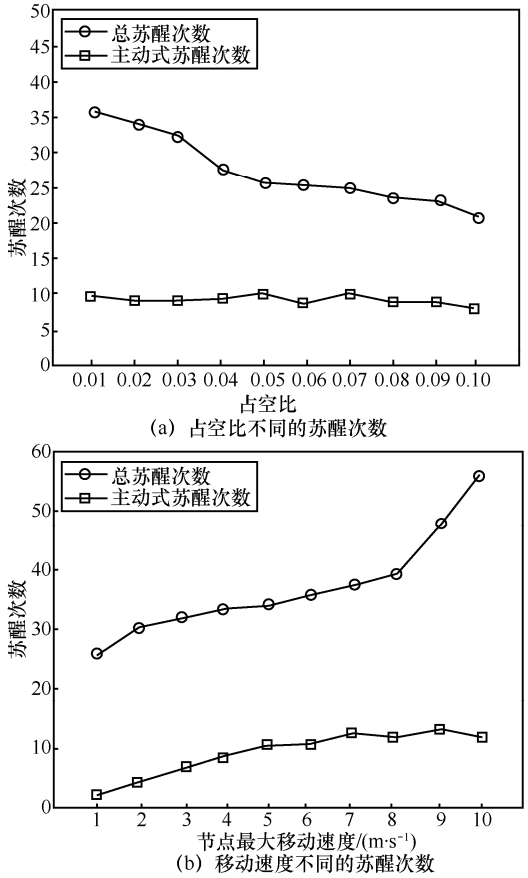


图 6　节点苏醒次数

由图 6(a)可以发现，在节点占空比增大时，节点的总苏醒次数会减小，而节点的主动苏醒次数基本保持稳定。这是因为节点占空比的变化不影响节点在某个时刻需要发现的邻居节点的数目，而较大的占空比使节点拥有更多的被动苏醒时间，因此，总苏醒次数会减小。由图 6(b)可以发现，在节点移动速度增大时，节点主动苏醒次数会先增加后持平，总苏醒次数也随之增加。这是因为增大的移动速度会使节点周围邻居集合变化快，节点需要更多的主动苏醒时间发现新的邻居节点。而当节点速度增大到一定程度，使节点在通信范围内的时间不足以进

行邻居发现工作，节点主动苏醒次数就保持稳定。

5.2.3 节点能耗

本节将展示 SPND 算法的网络能耗，并与现有的主动式邻居发现算法 GBD 进行对比。在此处的网络能耗中，仿真实验考虑了与节点邻居发现相关的处理器模块的能耗和通信模块的能耗。各项参数值设置如表 1 所示。

表 1 能耗分析参数值

参数	设定值/J
$E_{p\text{-run}}$	1×10^{-6}
$E_{p\text{-idle}}$	0.3×10^{-6}
$E_{p\text{-sleep}}$	1×10^{-9}
E_c	1.5×10^{-5}
$E_{c\text{-sleep}}$	1×10^{-6}

仿真实验分别测定了占空比不同与节点移动速度不同情况下，网络中任意一个节点完成一次邻居发现的平均能耗，得到的数据如图 7 所示。

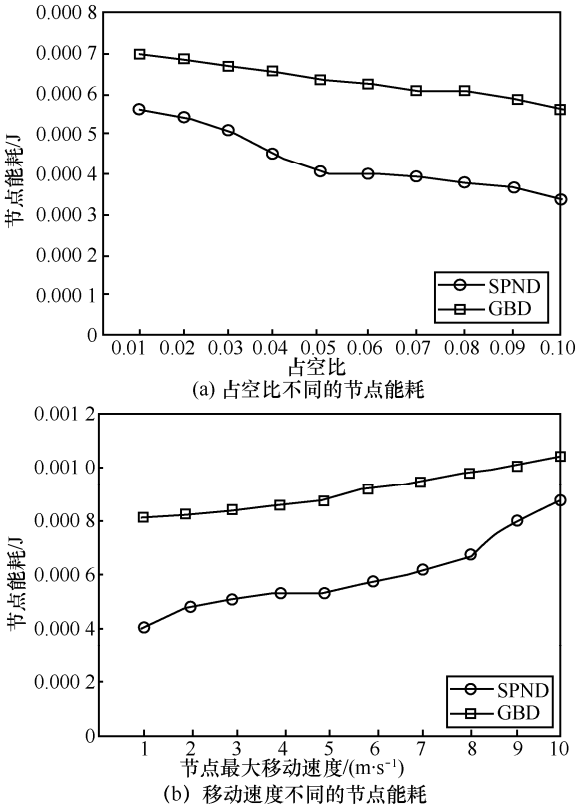


图 7 节点平均能耗

由图 7(a)可以发现，SPND 算法中节点能实现较小的能耗，且随着占空比的增加，节点用于邻居发现的能耗能减少很可观的一部分，而 GBD 算法中节点能耗则趋于稳定。由图 7(b)可以发现，网络中节点能耗会因为节点的移动速度增大而增大，但是 SPND 算法总能取得比 GBD 算法更小的网络能耗。

5.2.4 限定时延内的节点发现比率

本节将展示在限定时延情况下的节点发现比率，并将 SPND 算法与经典的 Disco 算法与 Birthday 算法进行对比。当网络中节点占空比为 0.05，节点移动速度为 3 m/s 时，得到节点限定时延的发现比率如图 8 所示。

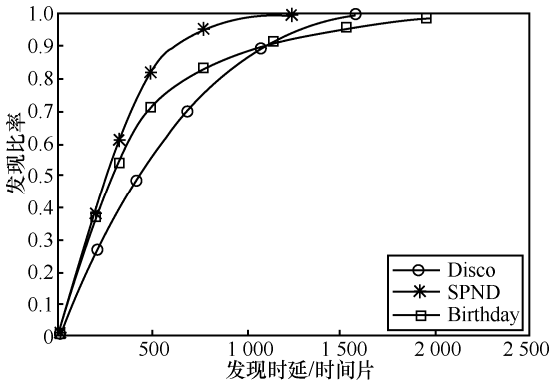


图 8 限定时延的发现比率

由图 8 可以发现，当限定时延较小时，SPND 算法与 Birthday 算法一样，发现比率上升较快，Disco 算法上升较慢。而到一定比率后，Birthday 算法出现了上升较慢的问题，而 SPND 算法能与 Disco 算法一样保持较快的上升速度。最终，SPND 算法能实现在 1 300 个时间片左右实现发现比率收敛到 1，比 Disco 算法快大约 200 个时间片。

5.3 实验总结

本文通过将 SPND 算法与其他算法的性能进行比较，分别从网络平均发现时延、节点苏醒次数、节点能耗及限定时延内的节点发现比率这 4 个方面进行对比。仿真实验数据表明，SPND 算法的邻居发现时延比 GBD 算法减小了约 19%，同时网络能耗减少了约 30%。此外，在限定时延的发现比率方面，SPND 发现比率收敛到 1 的速度比 Disco 算法快 13%。算法性能对比如表 2 所示。

表 2 算法性能对比

算法	发现时延	网络能耗	主动苏醒次数	发现比率收敛
SPND	极小	极小	较小	较快
GBD	较小	较大	较大	较快
Disco	较大	较小	—	较快
Birthday	较大	较小	—	较慢

其中，Disco、Birthday 算法均属于被动式苏醒算法，节点不会主动苏醒。

6　结束语

本文针对移动低占空比无线传感网中节点主动苏醒发现邻居能耗较高的问题，提出了一种新的低能耗的选择性主动苏醒快速邻居发现（SPND）算法。该算法在使用共享邻居信息方法构建节点初始邻居集合后，在下一步发现邻居时，通过划分节点邻居子集合，有选择地进行指定节点主动苏醒时刻，实现节点低能耗的邻居发现。理论分析及实验表明，SPND 算法在网络能耗大幅度优于 Disco、GBD 等算法的同时，还能实现发现时延比 Disco、GBD 等算法更小。

在主动式邻居发现算法中，节点根据邻居信息主动苏醒，要求苏醒时能与潜在邻居节点通信，因此，对网络中节点时钟同步的要求较高。但是，网络经过一段时间的工作后，节点的时钟会发生漂移。因此，下一步将针对节点时钟会发生漂移的异步网络进行研究。

参考文献：

[1] RAZAQUE A, ELLEITHY K M. Low duty cycle, energy-efficient and mobility-based boarder node-MAC hybrid protocol for wireless sensor networks[J]. Journal of Signal Processing Systems, 2015, 81(2): 265-284.

[2] GUO S, YANG Y, WANG C. DaGCM: a concurrent data uploading framework for mobile data gathering in wireless sensor networks[J]. IEEE Transactions on Mobile Computing, 2016, 15(3): 610-626.

[3] 陈权, 高宏. 低占空比无线传感器网络中基于动态切换的实时路由协议[J]. 通信学报, 2015, 10(36): 224-234.
CHEN Q, GAO H. Dynamic switching based real-time routing in low-duty-cycle wireless sensor networks[J]. Journal on Communications, 2015, 10(36): 224-234.

[4] QIU Y, LI S, XU X, et al. Talk more listen less: energy-efficient neighbor discovery in wireless sensor networks[C]// IEEE International Conference on Computer Communications. 2016: 1-9.

[5] MENG T, WU F, CHEN G. Code-based neighbor discovery protocols in mobile wireless networks[J]. IEEE Transactions on Networking, 2016, 24(2): 806-819.

[6] MCGLYNN M J, BORBASH S A. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks[C]// The ACM MobiHoc. 2001: 137-145.

[7] YOU L, YUAN Z, YANG P, et al. ALOHA-like neighbor discovery in low-duty-cycle wireless sensor networks[C]//2011 IEEE Wireless Communications and Networking Conference (WCNC). 2011: 749-754.

[8] JIANG J R, TSENG Y C, HSU C S, et al. Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks[C]// Proceedings of Mobile Networks and Applications. 2005,10(12):169-181.

[9] ZHENG R, HOU J C, SHA L. Asynchronous wakeup for ad hoc networks[C]// The 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing. 2003: 35-45.

[10] DUTTA P, CULLER D. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications[C]//The 6th ACM Conference on Embedded Network Sensor Systems. 2008: 71-84.

[11] NIVEN I, ZUCKERMAN H S. An introduction to the theory of numbers. 4th ed[J]. American Mathematical Monthly, 1961, 68(6): 401-405.

[12] KANDHALU A, LAKSHMANAN K, RAJKUMAR R R. U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol[C]//The 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. 2010: 350-361.

[13] CHEN H, LOU W, WANG Z, et al. On achieving asynchronous energy-efficient neighbor discovery for mobile sensor networks[J]. IEEE Transactions on Emerging Topics in Computing, 2016, PP(99): 1-12.

[14] KINDT P H, YUNGE D, REINERTH G, et al. Griassdi: mutually assisted slotless neighbor discovery[C]//ACM/IEEE International Conference on Information Processing in Sensor Networks. 2017: 93-104.

[15] CHEN L, SHU Y, GU Y, et al. Group-based neighbor discovery in low-duty-cycle mobile sensor networks[J]. IEEE Transactions on Mobile Computing, 2016, 15(8): 1996-2009.

[16] 陈良银, 颜秉姝, 张靖宇, 等. 移动低占空比传感网邻居发现算法[J]. 软件学报, 2014, 25(6): 1352-1368.
CHEN L Y, YAN B S, ZHANG J Y, et al. Neighbor discovery algorithm in mobile low-duty-cycle wireless sensor networks[J]. Journal of Software, 2014, 25(6): 1352-1368.

[17] HUANG T, CHEN H, ZHANG Y, et al. EasiND: effective neighbor discovery algorithms for asynchronous and asymmetric-duty-cycle multi-channel mobile WSNs[J]. Wireless Personal Communications, 2015, 84(4): 3031-3055.

[18] CAMP T, BOLENG J, DAVIES V. A survey of mobility models for ad hoc network research[J]. Wireless Communications and Mobile Computing, 2002, 2(5): 483-502.

[作者简介]

梁俊斌（1979-），男，广西南宁人，博士，广西大学教授，主要研究方向为无线传感器网络。

周翔（1995-），男，湖北鄂州人，广西大学硕士生，主要研究方向为无线传感器网络。

李陶深（1957-），男，广西邕宁人，博士，广西大学教授，主要研究方向为无线 Mesh 网络、分布式工程数据库、遗传优化设计、网络计算与信息安全等。