ARTICLE IN PRESS
JID: ADHOC [m5G;May 31, 2016;4:27]
Ad Hoc Networks 000 (2016) 1–15
Contents lists available at ScienceDirect
Ad Hoc Networks
journal homepage: www.elsevier.com/locate/adhoc

Neighbor discovery in mobile sensing applications: A comprehensive survey

Lin Chen[a,*], Kaigui Bian[b]

[a] Lab. Recherche Informatique (LRI-CNRS UMR 8623), Univ. Paris-Sud, 91405 Orsay, France
[b] Inst. Netw. Comp. & Inf. Sys., School of EECS, Peking Univ., Beijing, 100871, China

ARTICLE INFO

Article history:
Received 14 August 2015
Revised 23 March 2016
Accepted 13 May 2016
Available online xxx

Keywords:
Neighbor discovery
Mobile sensing
Wireless networks

ABSTRACT

The ubiquitous deployment of personal mobile devices today has boosted numerous mobile sensing applications where sensing data should be timely collected and exchanged among participating sensors. An important bootstrapping primitive in such applications is neighbor discovery. Designing distributed neighbor discovery protocols in mobile sensing applications is particularly challenging because of the duty cycling operation mode where mobile devices, usually battery-powered, switch between active and dormant modes periodically to conserve energy. In this paper, we give a comprehensive survey on the latest advance and development in this field by covering probabilistic, deterministic and collaborative neighbor discovery approaches developed in the literature. The focus of our survey on the developed neighbor discovery protocols is their design ideas and methodologies that may inspire and guide the development of new solutions in the future research. We also highlight a number of important and relevant research challenges that have not been addressed in the existing literature and that deserve further attention and investigation.

1. Introduction

The ubiquitous deployment of personal mobile devices nowadays, e.g., smart-phones and tablets, has boosted numerous mobile sensing applications ranging from mobile social networking [50,51], intelligent transportation [9,68], proximity-based gaming [1,3], environment and habitat monitoring [18,20,21,81] to participatory and crowd sensing [25,41,74,75]. In these applications, mobile devices usually carry various types of sensors and interact with neighbor devices to exchange sensing data [28,44]. For example, policemen and firefighters need to exchange information and commands in a timely fashion in rescue operations so as to coordinate with each other efficiently [43]; proximity-based gaming applications require players to interact with their nearby peers in real time [55,63].

The bootstrapping primitive that discovers all the neighbors of a mobile device is termed as *neighbor discovery*, which is one of the supporting functionality for many basic networking tasks, such as medium access control, topology control and clustering, routing, etc. An efficient neighbor discovery protocol should enable a node to discover its neighbors within a short delay for other functionalities to launch as quickly as possible.

Devising effective neighbor discovery protocols for distributed mobile sensing applications is a non-trivial task given the stringent energy saving requirement of low-power wireless devices. Particularly, these mobile wireless devices typically switch between active and dormant modes periodically to conserve energy. This energy conservation technique is called *duty cycling*, where *duty cycle* refers to the fraction of time a device is in the active state [4,27]. For example, a device whose duty cycle is 1% activates during one time slot every 100 slots. The duty cycle length is thus 100 slots. Despite its efficiency in saving energy, duty cycling imposes extra difficulty for the design of neighbor discovery protocols to limit neighbor discovery delay. Particularly, the two important design objectives, energy conservation via a duty-cycled operation mode and minimizing neighbor discovery delay, are contradictory to each other. Therefore, designing efficient duty-cycle based neighbor discovery protocols should strike a desired balance between these two conflicting objectives.

Due to the fundamental importance of neighbor discover protocols in mobile sensing applications and the particular design challenge brought by the duty cycling energy conservation technique, we devote this survey to reporting and analysing the recent technical advance and development of energy-efficient neighbor discovery protocols. Aiming at tracing the latest developments in this

* Corresponding author.
E-mail addresses: chen@lri.fr (L. Chen), bkg@pku.edu.cn (K. Bian).

http://dx.doi.org/10.1016/j.adhoc.2016.05.005

field, we attempt to deliver a comprehensive coverage on existing literatures with a proper technical depth to introduce the design idea and philosophy and analyse the pros and cons of each surveyed neighbor discovery solution. We complete the survey by pointing out a number of important and relevant research challenges that have not been addressed in the existing literatures and that deserve further research attention and investigation. There exist a handful of survey articles on neighbor discovery, but they are either generically focused on ad hoc and sensor networks (e.g., [23,64,66]), or address a specific scenario different from our focus (e.g., [59] surveys neighbor discovery in the Internet of Things (IoT) applications).

The remaining sections are organized as follows. Section 2 points out the design challenges of neighbor discovery protocols in mobile sensing applications. Section 3 provides a two-level classification of the existing neighbor discovery protocols in the literature. Sections 4 and 5 provide a comprehensive survey on the direct neighbor discovery protocols by focusing on probabilistic and deterministic approaches, respectively. Section 6 further discusses indirect neighbor discovery protocols. Finally, Section 7 concludes the paper by highlighting important and relevant research challenges that have not been addressed in the existing literatures and that deserve further research attention and investigation.

2. Neighbor discovery protocol design challenges and performance metrics

As pointed out in the Introduction, neighbor discovery is the process of identifying all nodes with which a given node can communicate directly. Specifically, each node in the network broadcasts short messages (or beacons) containing its ID and other information. The node is discovered by its neighbors if the neighbor discovery messages are corrected received and decoded by them. The way how such messages are broadcast (e.g., probability and period) is specified by a neighbor discovery protocol.

If network nodes are rechargeable or have infinite energy resource, the neighbor discovery can be ensured by a simple protocol by letting each node periodically broadcast beacons announcing its presence and always stay active to listen to beacons from its neighbors. The task is also much easier to accomplish if nodes can be tightly synchronised one to another. In [7], Baker et al. developed a distributed two-round round-robin neighbor discovery algorithm under a common clock. However, it is very difficult, even impossible in some cases, to achieve tight synchronisation among local clocks of wireless devices with limited processing power operating in an autonomous ad-hoc manner. Synchronising with external assistance such as GPS or NTP (Network Time Protocol [49]) servers via periodic message exchange [33,44] is usually too energy-consuming and thus too expensive or even impossible for mobile sensors and smart-phones [22,57]. The problem becomes much more tractable if all nodes operate on symmetric wake-up patterns (i.e., operating on the same duty cycle), or at least, the duty cycle length of other nodes are known or can be acquired. However, even these assumptions are sometimes unrealistic in mobile sensing applications because the duty cycle lengths of different nodes are usually asymmetrical, depending on their individual energy constraint. Even if nodes begin with the same duty cycle length, since the network activities are heterogeneous among users of different roles, the available energy of each node will evolve asymmetrically and result in asymmetric duty cycles.

Based on the above argument, we summarise the design challenges of neighbor discovery protocols in mobile sensing applications as follows:

- Stringent energy constraint;
- No network-level time synchronisation;
- Heterogeneous duty cycle length.

The combination of the three challenges renders the design of neighbor discovery protocols in mobile sensing applications far from trivial. Specifically, we use the term *heterogeneous neighbor discovery* to formalize the problem of designing distributed neighbor discovery protocols:

How can two neighbor nodes, that not necessarily operate on the same duty cycle and wake up infrequently and asynchronously, discover each other without any prior coordination or knowledge on their energy conservation parameters and encounter patterns?

Having defined the heterogeneous neighbor discovery problem, we now specify major metrics that quantify the performance of any neighbor discovery protocol:

- *Discovery delay:* The primary performance metric is the neighbor discovery delay. Depending on the application scenarios, we seek to minimise the expected discovery delay or the worst-case discovery delay.
- *Granularity in duty cycle support:* A neighbor discovery protocol need to provide sufficiently fine granularity support to enable sufficient levels of energy conservation.
- *Robustness against clock drift:* The discovery should be ensured even if the clocks of any two nodes are not synchronised and their time difference may be arbitrarily large.
- *Discovery diversity:* In multi-channel networks, it is desirable that a neighbor discovery protocol can achieve discovery on several channels to minimize the probability of neighbor discovery failures due to interference over any wireless channel.

In what follows, we discuss recent advance and development of energy-efficient neighbor discovery protocols for mobile sensing applications addressing the above design challenges.

3. Classification of neighbor discovery protocols

Neighbor discovery protocols can be classified using different criteria. In this paper, we adopt a two-level classification. At the higher level, neighbor discovery protocols can be categorised into *direct* and *indirect* approaches. In direct neighbor discovery approaches [7,8,16,19,30–32,34,35,38–40,46,47,61,69,71–73,77,84], a node is discovered by a neighbor node only if the neighbor node directly hears from this node. In many cases, neighboring devices share common neighbors, which can be exploited to enable indirect neighbor discovery. Indirect neighbor discovery approaches [79,80] use direct neighbor discovery protocols as building blocks and exploit the collaboration of direct discovered neighbors to discover new neighbors indirectly. At the lower level, the direct neighbor discovery protocols, which can be regarded as the baseline scenario of neighbor discovery, are further classified into *probabilistic* and *deterministic* protocols.

Probabilistic protocols [16,30,35,46,47,61,71–73,77] adopt probabilistic strategies at each node. Specifically, each node remains active or asleep with different probabilities. Probabilistic protocols have the advantages of being stationary due to the memoryless nature. Moreover, they usually perform well in the average case by limiting the expected discovery delay. The main drawback of them is the lack of discovery guarantee. This problem is referred to as the long-tail discovery latency problem in which two neighbor nodes may experience extremely long delay before discovering each other.

Deterministic protocols, in contrast, are able to provide guaranteed upper-bound on the worst-case discovery delay [7,8,19,31,32,34,38–40,69,84]. In deterministic neighbor discovery protocols, each mobile node operates according to its wake-up schedule carefully designed to guarantee that any pair of nodes can wake up in at least one common slot. The key element in the deterministic protocol design is how to devise the wake-up schedule to ensure discovery and minimise the worst-case discovery delay, regardless of the duty cycle asymmetry and the relative clock drift. Compared to the probabilistic approaches that work well in the average case while fail to bound the worst-case discovery delay, the deterministic protocols have good worst-case performance while usually have longer expected discovery delay.

To streamline the presentation of our survey, we first discuss direct neighbor discovery approaches by presenting both probabilistic (Section 4) and deterministic approaches (Section 5). We then move to survey the more advanced indirect neighbor discovery protocols (Section 6) which use the direct neighbor discovery protocols as building blocks and exploit the collaboration of the direct neighbors to enable indirect neighbor discovery.

4. Direct neighbor discovery: probabilistic approaches

In this section, we review the probabilistic direct neighbor discovery protocols in the literature. *As a common design objective, the probabilistic approaches aim at minimising the expected neighbor discovery delay.*

Specifically, we start with a detailed analysis on two probabilistic protocols that are developed for generical wireless networks and whose ideas have inspired and guided the design of many other probabilistic neighbor discovery solutions. We then discuss major technological developments designated for specific network and communication models and scenarios such as cognitive radio networks, wireless personal networks and networks where nodes are equipped with directional antennas. Although some of the neighbor discovery protocols are not tailored specifically for mobile sensing applications, yet the methodologies and techniques used in their design represent an important research thrust in neighbor discovery protocol development and have their merit in the context of mobile sensing applications. Therefore, in this section we give a comprehensive survey of these probabilistic approaches.

4.1. Baseline Aloha-like protocols

The first family of probabilistic neighbor discovery protocols are the baseline Aloha-like protocols [12,47].[1] Specifically, the authors of [47] consider a collision-prone slotted network where the total number of nodes n is known. Each node is in one of the following three states: transmit (T), where it broadcasts a discovery message advertising itself, listen (L), where it listens for discovery messages sent by others, or energy-saving (S) where it sleeps and spends zero energy. Under such model, a node a is discovered by another node b in slot t if in slot t, a is in state T, b is in state L and no other nodes is in state T. The authors proposed three operation modes for the birthday protocol based on the probabilities of staying at the three states.

The first mode is Birthday-listen-transmit (BLT). A node in BLT mode is in one of the three states T, L and S. In each slot, a node operates in state T with probability p_t, state L with probability p_l and state S with probability $1 - p_t - p_l$. The expect number of discovered links within the network under the BLT mode in one slot, denoted by $E(h)$, can be derived as

$$E(h) = p_t p_l (1 - p_t)^{n-2}. \quad (1)$$

By tuning the parameters p_t and p_l, we can trade off energy efficiency with discovery performance.

The second mode is Birthday-listen (BL) mode in which each node aims at maximising the number of discovered links. To this end, it is clear that each node sets $p_l = 1 - p_t$, i.e., a node never sleeps. Substituting $p_l = 1 - p_t$ into (1), we have that $E(h)$ is maximised at $p_t = \frac{1}{n}$. Since there are n nodes, each transmitting a fraction $\frac{1}{n}$ of the time and listening the rest of the time, the result can be regarded as the probabilistic analog of a round robin protocol and is thus termed as probabilistic round robin (PRR). There is no energy saving in the BL mode, but the number of discovered neighbors is maximised.

Among the first two modes, the BLT mode can save energy during a long period of network deployment, while the BL mode can quickly discover neighbors at the price of more energy consumption. To combine the advantages of the above two modes, the authors proposed the third mode as an alternative combination of the BLT and BL modes. Specifically, when the network deployment lasts long time, this mode consists of deploying nodes in BLT mode, then when some event occurs, transiting to BL mode for a period of time. The time in BL mode should be relatively short, since it consumes significant energy. The authors concluded their analysis with a numerical study on a specific example of deployment and neighbor discovery process to demonstrate the effectiveness of the baseline Aloha-like protocols.

4.2. Aloha-like protocols with collision detection and unknown number of neighbors

The baseline Aloha-like protocols, with the three operation modes, are a family of simple and flexible neighbor discovery protocols designed for wireless networks where energy conservation is critical. However, the requirement of knowing the number of neighbors n makes them unadaptable in dynamic scenarios with mobile nodes where the number of neighbors may vary in time and is not known to the network nodes. In some cases, even having an estimation of n is unrealistic. Moreover, the protocols suffer from packet collision when the network scales. Motivated by these observations, Vasudevan et al. [72] developed a suite of practical neighbor discovery protocols with collision detection without any estimation on the number of neighbor nodes. The authors started with a synchronised Aloha-like protocol without collision detection where the number of nodes (and thus the number of neighbors of any node) in the network n is known to any node in the network and then removed the assumptions one by one to iron out a protocol that works in practice.

The first Aloha-like protocol works as the BL mode in the baseline Aloha protocol. To derive the expected delay to discover all neighbors, the authors cast the problem to the coupon collector's problem. In the coupon collector's problem, n distinct objects are randomly drawn from an urn with probability $\frac{1}{n}$. The selected objects are then put back into the urn. The minimum number of trials before picking each object at least once can be derived as $n(\ln n + c)$ with c being a constant. To study the neighbor discovery process using the coupon collector's problem, we can create a fictitious coupon collector C who, in each slot, picks a coupon with probability p_s, and picks no coupon with probability $1 - np_s$, where $p_s = \frac{1}{n}\left(1 - \frac{1}{n}\right)^{n-1} \simeq \frac{1}{ne}$ is the probability a node successfully transmits in a given slot. When C collects n distinct coupons, the whole process of neighbor discovery is achieved. Mathematically, the expected delay to discovery all neighbor nodes can be derived as $ne(\ln n + c)$. It can also be proved that the delay is sharply concentrated around its average.

The authors then developed a protocol with collision detection when each node is synchronised and knows n that works as follows. Each slot is further divided into two sub-slots.

- In the first sub-slot, each node transmits a neighbor discovery message with probability $p_i = \frac{1}{ni}$ and receive with probability

[1] The baseline Aloha-like neighbor discovery protocols are also referred to as birthday protocols in [47].

Please cite this article as: L. Chen, K. Bian, Neighbor discovery in mobile sensing applications: A comprehensive survey, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.05.005

$1 - p_i$ where i is a local variable denoting the number of neighbors discovered so far and is initialised to 0.
- In the second sub-slot:
  - Each node operating in the receive mode during the first sub-slot verifies whether the current transmission is successful or not. If yes, it sets $i \leftarrow i + 1$, otherwise it sends a neighbor discovery message in the second sub-slot.
  - For any node operating in the transmit mode during the first sub-slot, if it detects a transmission in the second sub-slot, then it deduces that its transmission in the first sub-slot is unsuccessful. If it does not detect any transmission in the second sub-slot, it deduces that it has been discovered by others and thus switches to receive mode and remains in this mode for the rest of time.

The core idea of the protocol is that each node that has been successfully discovered stops transmitting, thus allowing other nodes to be discovered faster. Conversely, nodes that have not been discovered increase their transmission rate as other nodes get discovered. As the key element in the protocol, collision-detection enables a node to trace the number of nodes that have not been discovered so as to adapt the optimum transmission probability. Collisions are detected as follows: any node in the receive mode checks if the transmission in the first sub-slot is successful or not; if not (meaning a collision), it transmits one bit in the second sub-slot that allows the transmitter to be aware that the transmission is not successful by detecting energy in the second sub-slot. Note that if a collision is detected in the first sub-slot, a node is required to transmit only one bit in the second sub-slot. The second sub-slot is thus much shorter than the first. With a similar analysis based on the coupon collector's problem, the expected discovery delay can be derived as $E[W] = O(ne)$, i.e., a factor of $\ln n$ better than the baseline protocol.

The authors then studied the case with unknown n. The baseline ALOHA-like protocol was modified to be executed in rounds. Round i ($i \geq 1$) is composed of $2^i e(\ln 2^i + c)$ slots with c being a constant. In round i, every node transmits with probability $\frac{1}{2^i}$. Under this round-based neighbor discovery protocol, network nodes exponentially decrease their transmission probabilities. They finally enter a period of time where all of them transmit with probability around $\frac{1}{n}$ for consecutive $ne(\ln n + c)$ slots. The expected discovery delay can be bounded by $2ne(\ln n + c)$. Compared to the case where the number of nodes n is known, the discovery delay doubles in the worst case when n is not known. Similar idea was then applied to adapt the protocol with collision detection with the same performance gap w.r.t. the case where n is known.

The authors then proceeded to the asynchronous case where slots are not aligned at different nodes. In the asynchronous case, each node stays in the receive mode for an exponentially distributed duration whose mean is $\frac{1}{\lambda} = \frac{1}{2\tau n}$ with τ being the transmission duration of a message to maximizes the rate of neighbor discovery. By incorporating the exponentially distributed receiving time, the authors developed both ALOHA-like protocol and collision detection protocol with both known and unknown n. Regarding the expected discovery delay, it was shown that the loss of synchronization and the lack of the knowledge on n resulted each in a factor of two slowdown. The authors also derived a closed-form condition to terminate the neighbor discovery process. For other probabilistic approaches, a reasonable termination criteria is to compute the probability that two neighbors discover each other after t slot, denoted by $z(t)$ and terminate the protocol when $z(t)$ is larger than a threshold. Vasudevan et al. further investigated the neighbor discovery problem in a general multi-hop setting in [70] by showing the Aloha-based protocol is in the worst case a factor $\min(\Delta, \ln n)$ worse than the optimum with Δ denoting the maximum node degree (i.e., the maximum number of neighbors for any node) in the network.

4.3. Aloha-like protocol with multi-packet reception

Motivated by the performance gain brought by multi-packet reception (MPR) technologies (e.g., CDMA and MIMO), Zeng et al. [77] studied the neighbor discovery problem in networks with MPR capacity where multiple packets can be received and decoded by receivers. Inspired by [72], the authors proposed and analyzed a suite of randomized neighbor discovery protocols for MPR networks. Their analysis follows the similar development as that in [72].

Specifically, the authors started with a simple Aloha-based protocol where nodes are synchronised in their transmission schedule and the number of neighbors is known. They showed that the expected discovery time is $O(\ln n)$ in the ideal case where the receivers are able to decode any number simultaneous transmissions. They then studied a more practical scenario, where the number of successful receptions is limited to k. They showed that the expected discovery delay is $O(n \ln n / k)$. They then developed an adaptive Aloha-like protocol for the case with transmission feedback and demonstrated that it yields a $\ln n$ performance gain in the basic Aloha-like protocol. They completed their work by incorporating more practical constraints, particularly the case without knowing n and the asynchronous algorithm operation. They show that these result in at most $\ln n$ factor increase in discovery delay. Generically speaking, the work of [77] can be regarded as a generalisation of [72] in the MPR networks where up to $k \geq 1$ simultaneous transmissions can be decoded, and is degenerated to that of [72] if $k = 1$.

4.4. Probabilistic protocols with directional antennas

There are a body of studies on neighbor discovery protocols for nodes equipped with directional antennas [17,30,54,61,71,83]. In these works, besides the transmission strategies, the antenna scanning strategies should also be carefully designed to achieve efficient neighbor discovery.

Vasudevan et al. [71] proposed a suite of probabilistic protocols. The developed solutions can be classified into two categories, (1) direct-discovery protocols where nodes discover their neighbors only when they actually receive a beacon from their neighbors, (2) gossip-based protocols where nodes spread their neighbors' information in the broadcast neighbor discovery beacon to further increase discovery probability. In synchronised slotted cases, the direct-discovery algorithms work in the similar way as that in [72] with one more dimension on the antenna direction which is also randomly configured. With a similar analysis, the transmission probability that maximises the discovery rate is $p_t = \frac{2\pi}{n\theta}$, where n is the number neighbors, θ is the antenna beam width. In the gossip-based algorithms, the same probability is derived as follows:

$$p_t = \frac{4\pi + (n-1)\theta - \sqrt{[4\pi + (n-1)\theta]^2 - 8n\pi\theta}}{2n\theta}.$$

When nodes are not synchronised, a similar technique to [72] is applied to let the listen intervals of each node to be exponentially distributed with rate $\lambda = \frac{1}{2\tau[(n-1)\frac{\theta}{2\pi}+1]}$ with τ being the transmission duration of a message to maximizes the rate of neighbor discovery.

A number of work treated neighbor discovery as a functional component in the MAC protocol and proposed intergraded neighbor discovery and MAC protocols with directional antenna [30,61]. One such example is the Polling-based MAC (PMAC) protocol proposed in [30] that integrates the neighbor discovery protocol with a scheduling based medium access designed for directional antennas. The key idea that hinges behind the PMAC design is a polling mechanism in which each node polls periodically its neighbors already discovered. Specifically, a node in PMAC operates in one of the following three states: (1) search state where it searches new neighbors, (2) polling state where it polls neighbors already discovered, (3) data transfer state where packets are transmitted, each corresponding to a segment in the MAC frame. In the segment corresponding to the search state, a node randomly points its antenna. If communication is set with a new neighbor, then packets are further exchanged with the neighbor such that the two nodes negotiate in the polling segment and the actual data exchange is performed in the data segment using the schedule negotiated in the polling segment.

4.5. Neighbor discovery in multi-channel wireless networks

Another research thrust in the probabilistic neighbor discovery protocol design focuses on multi-channel wireless networks. A typical example is the cognitive radio (CR) network, where a cognitive node can opportunistically utilize unused spectrum of primary users. In this regard, Mittal et al. proposed a series of multi-channel probabilistic and deterministic neighbor discovery protocols [37,52,53,78] for CR networks where different cognitive nodes in the network may have different perceptions on the available channels. In [6], C. Arachchige et al. developed a leader election protocol to bootstrap a CR network. Specifically, a leader is elected based on the IDs of nodes and the elected leader periodically broadcasts beacons to discover neighbors. Other nodes listen for the beacons and send an acknowledgement upon receiving them. Note that the proposed neighbor discovery protocols for CR networks do not take into account energy efficiency. On the other hand, a number of protocols are proposed for wireless personal area networks such as IEEE 802.15 [2] that support multi-channel neighbor discovery and different beacon intervals (equivalent to duty cycles). In this regard, the works in [35] and [73] performed optimisation on several random strategies to derive optimal or low-complexity neighbor discovery protocols that minimise the expected discovery delay.

4.6. Neighbor discovery in wireless sensor networks

Madan et al. [46] proposed an energy-effieicnt neighbor discovery protocol in wireless sensor networks. They developed a distributed algorithm minimising the power required for neighbor discovery. Specifically, they consider a network with a large number of sensor node randomly deployed in a given area. In such networks, the sensor node distribution is characterised by a Poisson process. The neighbor discovery process is characterised by a Markov decision process, and the neighbor discovery policy can be regarded as a finite automaton, depending on the probability distribution. The design objective is to minimise the expected energy consumption. To that end, the authors presented an example of a finite automaton that implements the derived optimal strategy.

Cohen et al. investigated continuous neighbor discovery in wireless sensor networks [16]. The continuous neighbor discovery refers to the process of maintaining neighbor information after the initial neighbor discovery process. The authors assume the initial neighborhood is already established and proposed an algorithm for continuous neighbor discovery after an initial neighborhood was established using a broadcast SYNC message which is heard by all nodes.

Slot index	1	2	3	4	5	6	7	8	9	10	11	12	...
Node *a* :	0	0	1	0	0	1	0	0	1	0	0	1	...
Node *b* :	0	0	0	1	0	0	0	1	0	0	0	1	...

Fig. 1. Example illustrating the wake-up schedule.

5. Direct neighbor discovery: deterministic approaches

In this section, we proceed to review the deterministic direct neighbor discovery protocols. As analysed in Section 3, the key objectives in the design of deterministic protocols are

- Guaranteeing discovery regardless of the duty cycle asymmetry and the relative clock drift,
- Optimising specific performance metrics such as the worst-case discovery delay and the power-latency (PL) product. We present a detailed performance comparison of major deterministic approaches in Table 1.

Specifically, we first present the technical foundations on the wake-up schedule and formulate the deterministic heterogeneous neighbor discovery problem. We then survey the state-of-the-art deterministic neighbor discovery protocols developed in the literature and provide a synthetic comparison among them using various performance metrics.

5.1. Technical foundations and problem formulation

The recently developed deterministic neighbor discovery solutions usually work on a time-slotted basis because slotted protocols are usually practical to implement and can limit the impact of clock drift among different nodes. Specifically, in the time-slotted paradigm, time is divided into identical slots, measure by nodes' local clock and subject to their clock drift. Under the duty cycling technique, nodes stay awake in a subset of slots, termed as active slots, and sleep for the rest of the slots, called inactive slots, or dormant slots. During an active slot, a node can send and/or receive messages. For a pair of neighbor nodes, they can discover each other if and only if at least one of their active slots overlap more than certain time, e.g., half of the slot duration.

Formally, we define the *wake-up schedule* to characterise the wake-up pattern of a node.

Wake-up schedule definition: *The wake-up schedule of a node* u *is a sequence* $\mathbf{x}_u \triangleq \{x_u^t\}$ $(1 \le t \le T_u)$*, with* T_u *being the period of the sequence,*[2] *and*

$$x_u^t = \begin{cases} 0 & u \text{ sleeps in slot } t \\ 1 & u \text{ wakes up in slot } t \end{cases}.$$

For any two nodes a and b whose wake-up schedules are $\mathbf{x}_a$ and $\mathbf{x}_b$ with periods T_a and T_b, due to the periodicity of their wake-up schedules $\mathbf{x}_a$ and $\mathbf{x}_b$, the tuple $(\mathbf{x}_a, \mathbf{x}_b)$ repeats every consecutive T_aT_b slots. If $\exists t \in [1, T_aT_b]$ and such that $x_a^t = x_b^t = 1$, we say that a and b can discover each other in slot t. Slot t is referred to as a discovery slot between nodes a and b.

The following example gives a further illustration. Consider two nodes a and b whose wake-up schedules are $\mathbf{x}_a = \{0, 0, 1\}$ and $\mathbf{x}_b = \{0, 0, 0, 1\}$, i.e., $T_a = 3$ and $T_b = 4$. The duty cycle lengths of a and b are $d_a = 3$ and $d_b = 4$. The wake-up schedules of a and b are thus repeated each $T_aT_b = 12$ slots, as shown in Fig. 1. We can see that $x_a^{12} = x_b^{12} = 1$, i.e., a and b can discover each other on slot 12.

[2] A random wake-up schedule can be regarded as a special case with $T_u \to \infty$.

Table 1
Comparison of major deterministic neighbor discovery protocols.

Protocol	Parameters	Duty cycle length	Worst-case discovery delay (symmetrical case)	Average discovery delay (symmetrical case)	Power-latency product (asymptotical case)
Quorum (grid)	L	$\sqrt{L}/2$	L	$L/3$	$2\sqrt{L}$
Disco	p_1, p_2	$p_1 p_2/(p_1+p_2)$	$p_1 p_2$	$p_1 p_2/2$	$2\sqrt{L}$
U-Connect	p	$2p^2/(3p+1)$	p^2	$p^2/2$	$3\sqrt{L}/2$
Searchlight	l	$l/2$	$l^2/2$	$l^2/4$	$\sqrt{2L}$

In the above example, the clocks of a and b are synchronised. In practice, the clocks of different nodes may not be synchronised. Consequently, their wake-up schedules may be shifted to each other. To accommodate this situation, we use the concept of *cyclic rotation* to wake-up schedules. Specifically, given a wake-up schedule $\mathbf{x}_a$, we use $\mathbf{x}_a(k)$ to denote a cyclic rotation of $\mathbf{x}_a$ by shifting k $(0 \le k \le T_a - 1)$ slots. In the previous illustrative example, we have $\mathbf{x}_a(2) \triangleq \{0, 1, 0\}$ and $\mathbf{x}_b(1) = \{0, 0, 1, 0\}$.

Using the above formalisation, we can formulate the deterministic heterogeneous neighbor discovery problem can be defined as follows.

Deterministic heterogeneous neighbor discovery problem: *The deterministic heterogeneous neighbor discovery problem consists of devising the neighbor discover schedules for network nodes to limit or minimise the worst-case discovery delay between any two neighbor nodes, regardless of their duty cycle patterns and their relative clock drift.*

Recent technical development on deterministic heterogeneous neighbor discovery mainly consists of using related mathematical tools in group theory and number theory to device neighbor discover schedules that can ensure discovery in the asynchronous and asymmetrical setting. Specifically, the deterministic neighbor discovery protocols developed in the literature can be largely categorised into two major classes, *quorum*-based and *co-primality*-based protocols, which we survey in the following subsections.

5.2. Quorum-based approaches

In quorum-based protocols ([29,31,32,38–40,69] and a number of energy-efficient MAC protocols such as BMAC [58] and S-MAC [76] which implicitly use quorum-based approaches or other algebraic structures [48] to maintain neighbor connections), each node configures its neighbor discover schedule based on a quorum. This pattern ensures that any two nodes have at least an active slot overlapping with each other regardless of their relative time difference.

We first briefly introduce the related theory concerning the quorum systems. Given a cycle length L, we set $U = \{0, 1, \cdots, L\}$ to be a universal set. A quorum system Q consists of subsets of U satisfying the following intersection property: $\forall A, B \in Q$: $A \cap B \neq \emptyset$. Each element in the quorum system Q is called a quorum. For example, consider $U = \{0, 1, 2, 3\}$, $Q = \{\{0, 1, 2\}, \{1, 2, 3\}\}$ is a quorum system as {0, 1} and {1, 2}, have two common elements 1 and 2.

Concerning quorum systems, an important property is the *rotation closure* property. Specifically, a quorum system Q satisfies the rotation closure property if $\forall A, B \in Q, i \in \{0, 1, \cdots, L-1\}$, it holds that $A \cap (B + i) \neq \emptyset$, where $B + i$ denotes a modulo L addition of each element in A, e.g., for a quorum $B = \{0, 1\}$ under $U = \{0, 1, 2\}$, $B + 2 = \{2, 0\}$. Let set A be a subset of U, if $\forall i \in \{0, 1, \cdots, L\}$, $A \cap (A + i) \neq \emptyset$, then $\{A, A+1, \cdots, A+L-1\}$ is a quorum system with rotation closure. Such quorum systems are called *cyclic quorum systems*.

There are two major quorum systems extensively studied in the literature, *grid quorum* system and *cyclic quorum* system. Both of them are rotation closure systems [39,45].

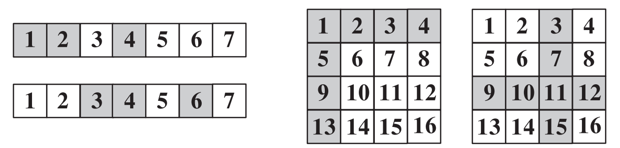

Fig. 2. Example illustrating a cyclic quorum system and a grid quorum system.

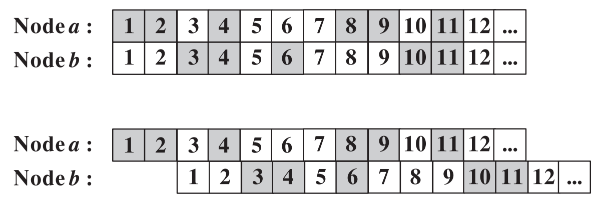

Fig. 3. Example illustrating a neighbor discovery process using the cyclic quorum system in Fig. 2: a and b can discover each other in slots 4, 11 and 6 when their clocks are synchronised or unsynchronised, using the clock of a as reference.

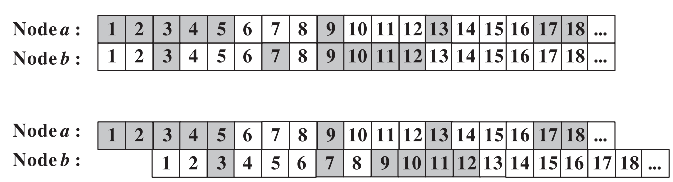

Fig. 4. Example illustrating a neighbor discovery process using the grid quorum system in Fig. 2: a and b can discover each other in slots 3, 9 and 5, 9, 13 when their clocks are synchronised or unsynchronised, using the clock of a as reference.

- *Cyclic quorum system*. Cyclic quorum systems use the theoretical properties of difference sets [65].[3] An example of cyclic quorum systems is shown in Fig. 2 (left) with $U = \{1, 2, 3, 4, 5, 6, 7\}$ and two quorums $A = \{1, 2, 4\}$, $B = \{3, 4, 6\}$.
- *Grid quorum system*. In a grid quorum system, the universal set U is a $\sqrt{L} \times \sqrt{L}$ grid, and a quorum contains elements of a column and a row. An example of a grid quorum systems is shown in Fig. 2 (right). The two quorums in the figure are {1, 2, 3, 4, 5, 9, 13} and {3, 7, 9, 10, 11, 12, 15}.

In quorum-based neighbor discovery protocols, the wake-up schedules are chosen based on quorums. In Figs. 3 and 4, we illustrate a neighbor discovery example using the cyclic and grid quorum systems in Fig. 2 when the clocks of the two neighbor nodes a and b are perfectly synchronized and has a drift of 2 slots. In terms of average discovery delay, the authors showed that the average delay is $\frac{L-1}{2}$ in the case of cyclic quorum system, and $\frac{(L-1)(\sqrt{L}+1)}{3\sqrt{L}}$ which approaches $\frac{L}{3}$ asymptotically in the case of grid cyclic quorum system.

The above quorum-based protocols only support symmetric operation with L being a global system parameter known to all nodes. To address the asymmetrical case where nodes may have het-

[3] A set D: $\{a_1, \cdots, a_k\}(\mod N)$, $a_i \in [0, N-1]$, is called a (N, k, λ)-difference set if for every $d \neq 0$, there are exactly λ ordered pairs (a_i, a_j), $a_i, a_j \in D$ such that $a_i - a_j = d \mod N$.

Slot index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
Node *a* :	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	...
Node *b* :	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	...

Fig. 5. Example illustrating a neighbor discovery process: $m_a = 3$, $m_b = 5$, $t_0^a = 0$, $t_0^b = 2$, discovery occurs in slot 12.

Slot index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
Node *a* :	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1	0	0	1	...
Node *b* :	0	1	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	...

Fig. 6. Example illustrating a neighbor discovery process: $p_a^1 = 3$, $p_a^2 = 5$, $p_b^1 = 5$, $p_b^2 = 7$, $t_0^a = 0$, $t_0^b = 2$, discovery occurs in slots 9 and 12.

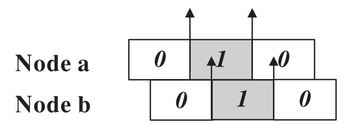

Fig. 7. Beacon transmission strategy in Disco to maximize the discovery probability when slots are not aligned.

erogeneous energy constraints, Lai et al. [39] developed an enhanced protocol to enable two levels of L, say L_1 and L_2, representing two levels of energy conservation. In this context, the wake-up schedules based on grid quorum systems can be directly applied without any modification. The discovery schedules based on cyclic quorum systems should be reconfigured. To that end, the authors developed a fast cyclic quorum construction mechanism in the asymmetrical context. In a recent work [29], Huang et al. developed a quorum-based neighbor discovery protocol for multi-channel wireless sensor networks where nodes have the same duty cycle length.

Another deterministic neighbor discovery protocol for the symmetric duty-cycle case uses the properties of difference sets [84] to ensure discovery. The design of wake-up schedules resembles the approach using the cyclic quorum system. The authors of [84] explored the Multiplier Theorem [5] to obtain the optimal solution. For the asymmetric duty cycles, designing the wake-up schedules using the proposed scheme was proved to be reduced to the vertex-cover problem, an NP-complete problem [36]. In this regard, as pointed in [34], the application of difference sets in solving asynchronous asymmetric duty-cycled neighbor discovery still remains open today.

Quorum-base approaches can provide guaranteed discovery within bounded delay regardless of the time difference between nodes. However, allowing only very limited number of duty cycles (at most two in the current literature) limits their application in more heterogenous applications, which is the downside of these approaches.

5.3. Co-primality-based approaches

In this category, we survey two representative neighbor protocols, Disco and U-Connect.

5.3.1. Disco

To allow heterogeneous duty cycles, Dutta et al. [19] proposed Disco, an asynchronous neighbor discovery protocol enabling wireless devices operating at heterogeneous duty cycles to discover each other without time synchronization. The key challenges in the design of Disco are (1) the heterogeneous and low duty cycles, e.g., 1% and (2) the requirement of guaranteed discovery within limited delay.

To address the above design challenges, Disco bases its design on the adaptation of the Chinese Remainder Theorem [56] to ensure discovery. To illustrate the core idea, we consider a simple example of two nodes a and b whose duty cycle lengths are m_a and m_b, i.e., a and b wake up each m_a and m_b time slots, respectively. Mathematically, the discovery schedule of each node i ($i = a, b$) $\mathbf{x}_i$ can be written as

$$x_i^t = \begin{cases} 1 & [t]_{m_i} = 0 \\ 0 & \text{otherwise} \end{cases},$$

where $[t]_{m_i} = t \bmod m_i$. By the Chinese Remainder Theorem, there exists $t \le m_a m_b$ such that $x_a^t(t_0^a) = x_b^t(t_0^b)$ for any t_0^a and t_0^b, i.e., for any initial time offset t_0^a and t_0^b, a and b are ensured to be able to discover each other within at most $m_a m_b$ slots. Fig. 5 illustrates an example of neighbor discovery process with $m_a = 3$, $m_b = 5$, $t_0^a = 0$, $t_0^b = 2$. It can be noted that discovery occurs in slot 12.

The above simplified solution works if a and b can choose co-prime numbers. However, how to distribute co-prime numbers in a decentralised way to every node without prior coordination is far from trivial, and if two nodes happen to choose the same number, i.e., $m_a = m_b$, they may never discovery each other, e.g., if $m_a = m_b = 5$, b's clock is two slots after a. To solve this problem, Disco lets each node i choose two prime numbers p_i^1 and p_i^2 (assume that $p_i^1 < p_i^2$) and wake up at slots $t_1 p^1$ and $t_2 p^2$ where $t_1, t_2 \ge 1$, i.e., $x_i^t = 1$ if $[t]_{p_i^1} = 0$ or $[t]_{p_i^2} = 0$. By this way, any two nodes must choose a pair of co-prime numbers to ensure discovery as two different prime numbers are by nature co-prime numbers. By the Chinese Remainder Theorem, the discovery is ensured to occur within at most $p_a^2 p_b^2$ slots between a and b. An example illustrating the neighbor discovery process in Disco is shown in Fig. 6.

The choice of prime numbers is a non-trivial design choice in Disco. On one hand, the prime numbers should be chosen to realise the desired duty cycle length. Specifically, for a node who plans to wake up every d slots in average, its two prime numbers p_1 and p_2 should be chosen based on the following formula:

$$\frac{1}{d} \simeq \frac{1}{p_1} + \frac{1}{p_2}.$$

However, on the other hand, the choice of prime numbers have significant impact on discovery delay. The authors illustrated the impact of the choice of prime numbers on the discovery performance by the following example. Given a target duty cycle $d = 50$, there are several prime number combinations to approximate it. One way is to use prime numbers 97 and 103 ($\frac{1}{50} \simeq \frac{1}{97} + \frac{1}{103}$) and another combination is 53 and 883 ($\frac{1}{50} \simeq \frac{1}{53} + \frac{1}{883}$). In the case where two nodes a and b both choose the same combination, the worst-case discovery delay is $97 \times 103 = 9991$ slots and $53 \times 883 = 46799$ slots depending on the combination chosen by them, the latter being 4 times the former. On the other hand, if node a chooses 53 and 883 and node b chooses 57 and 409, the worst-case discovery delay is $53 * 57 = 3201$, much better than that in the first case (46799). From the perspective of limiting the worst-case discovery delay, the nodes are better off choosing more balanced pair of prime numbers.

In practice, slots are rarely aligned due to the difficulty in time synchronisation among nodes. To address the slot non-alignment, in Disco, each node sends a beacon in the beginning and the end of every active slot, thus maximizing the likelihood that overlapping slots result in discovery, as illustrated by Fig. 7. This transmission strategy is also widely applied in other neighbor discovery protocols such as U-Connect [34] and Searchlight [8]. To handle perfect slot alignment, the slot overflowing scheme is developed in [8], where each active slot overflows by δ, a small amount that

Slot index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...
Node *a* :	1	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	...
Node *b* :	0	1	1	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	...

Node *a* :	1	0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	0	1	...
Node *b* :	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	...

Fig. 8. Example illustrating a neighbor discovery process: $t_0^a = 0$, $t_0^b = 2$; symmetrical case (upper): $p_a = p_b = 3$; asymmetrical case (lower) $p_a = 3$, $p_b = 5$; in both cases, discovery occurs in slots 3 and 12.

is sufficient to receive a beacon from another node. Of course, slot overflowing cannot eliminate collisions among multiple nodes. The utility of deterministic neighbor discovery algorithms is to ensure that any pair of neighbors will eventually activate in the same slot, without which discovery can never be achieved. To further limit the impact on beacon collisions, nodes can decrease the number of beacons transmitted by increasing the interval between beacons. Take Disco as an example, this can be achieved by choosing larger prime numbers. The price for lower collision probability is the increase worst-case discovery bound.

Another design issue of Disco is the granularity of duty cycles supported by the protocol. Disco requires that the reciprocal of the desired duty cycle d can be decomposed as the sum of the reciprocal of 2 prime numbers. As a result, many small duty cycles cannot be approximated accurately enough. To address this limitation, Disco adds another parameter, p_3, another prime number that can be chosen by a node to approach the desired duty cycle. In this case, the duty cycle d is

$$\frac{1}{d} \simeq \frac{1}{p_1} + \frac{1}{p_2} + \frac{1}{p_3}.$$

A side effect of adding p_3 is that in this case, users tend to choose larger prime numbers to approximate the duty cycle which may increase the worst-case discovery delay. Consequently, the successful implementation of Disco relies on the tradeoff among several design metrics such as the worst-case discovery delay and the granularity in duty cycle support.

5.3.2. U-Connect

U-Connect is an asynchronous neighbor discovery protocol for nodes with heterogeneous duty cycles, proposed by Kandhalu et al. [34]. Different from the design of Disco, U-Connect uses only one prime number for each node. The wake-up schedule of U-Connect is as follows:

$$x_i^t = \begin{cases} 1 & [t]_{p_i} = 0 \text{ or } 0 \le [t]_{p_i^2} < \frac{p_i+1}{2} \\ 0 & \text{otherwise} \end{cases},$$

where p_i denotes the prime number chosen by node i. An example illustrating the discovery process under U-Connect between two nodes a and b is shown in Fig. 8 with a relative clock drift 2 (specifically, $t_0^a = 0$, $t_0^b = 2$) for both symmetrical case where $p_a = p_b$ and asymmetrical case where $p_a \neq p_b$.

The authors then analysed the worst-case discovery delay of U-Connect. In the asymmetrical case, since p_a and p_b are two different prime numbers and are thus co-prime to each other, from Chinese Remainder Theorem we can prove that a and b are ensured to discovery each other within at most $p_a p_b$ slots, regardless of their time difference. In the symmetrical case where $p_a = p_b = p$, the discovery delay can be shown to not exceed p^2 by distinguishing the case where $|t_0^a - t_0^b| < \frac{p+1}{2}$ and where $|t_0^a - t_0^b| \ge \frac{p+1}{2}$.

The authors also proposed a performance metric to evaluate neighbor discovery protocols. Specifically, the major performance requirements of a neighbor discovery protocol are high energy efficiency and limited neighbor discovery latency. However, they cannot be achieved simultaneously. For example, increasing the energy efficiency by augmenting the duty-cycle length inevitably leads to longer discovery delay. To quantify this design trad-off, the authors defined a compound metric as the power-latency (PL) product Λ, which is the product of the average energy consumption and the worst-case neighbor discovery delay. By using the defined performance metric, they conducted a comparative study on existing deterministic neighbor discovery protocols based on a worst-case discovery delay L.

- Optimal: It was shown in [84] that there exists an optimal wake-up schedule if the discovery delay can be expressed as $L = k^2 + k + 1$, where k can be written as $k = p^m$ with p being a prime number and $m \ge 1$. Within L slots, each node remains active for $k + 1$ slots. Hence the optimal PL product is
$$\Lambda_o = \frac{k+1}{k^2+k+1} \times L = \sqrt{L - \frac{3}{4}} + \frac{1}{2},$$
which tends to $\sqrt{L}$ asymptotically.
- Quorum-based approaches: Take the symmetrical grid quorum system as an example. To limit the worst-case delay not to exceed L, at least a $\sqrt{L} \times \sqrt{L}$ grid quorum system is required, in which each node remains active $2\sqrt{L} - 1$ slots every L slots. The PL product is:
$$\Lambda_q = 2\sqrt{L} - 1,$$
which tends to $2\sqrt{L}$ asymptotically.
- Disco: Consider the case of Disco where each node operates on the same two prime numbers p_1 and p_2, the worst-case discovery delay $L = p_1 p_2$. Each node remains active for $p_1 + p_2 - 1$ slots every L slots. The PL product is:
$$\Lambda_d = L \times \frac{p_1 + p_2}{L} = p_1 + p_2 \ge 2\sqrt{p_1 p_2} = 2\sqrt{L}.$$
- U-Connect: Consider the symmetrical case where each node operates on the prime number p, the worst-case discovery delay $L = p^2$. Each node remains active for $p + \frac{p+1}{2}$ slots every L slots. The PL product is:
$$\Lambda_u = L \times \frac{p + \frac{p+1}{2}}{L} = \frac{3p+1}{2} = \frac{3\sqrt{L}+1}{2},$$
which tends to $\frac{3\sqrt{L}}{2}$ asymptotically.

The above comparison shows that U-Connect outperforms Disco and Quorum-based protocols from the perspective of the PL product. In this research strand, another protocol called WiFlock [60] was proposed by combining neighbor discovery with neighbor maintenance applying a cooperative beaconing method with a temporal clock synchronization.

To summarise, U-Connect achieves the best performance regarding the metric power-latency product by exhibiting 25% decrease compared to quorum-based approach and Disco. However, U-Connect limits the choices to only prime numbers and thus fails to support all duty cycle lengths.

5.4. Searchlight

As the state-of-the-art development in deterministic approaches, Searchlight is another neighbor discovery protocol implicitly using Quorum systems, proposed by Bakht et al. [8]. We discuss Searchlight separately as distinguished to classical Quorum-based approaches, Searchlight does not use classical quorums and has the capability of supporting multiple duty cycles. The design of Searchlight is motivated by the following observation. Probabilistic protocols usually have low discovery delay in the

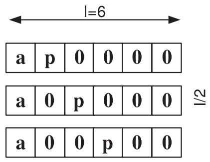

Fig. 9. Example of the wake-up schedule in Searchlight: $l = 6$.

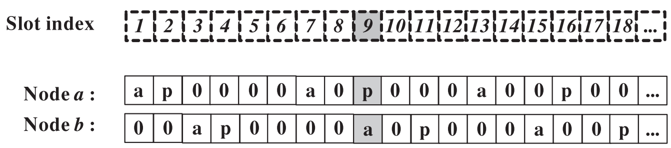

Fig. 10. Example of the neighbor discovery process in Searchlight: $l = 6$.

average case and naturally support asymmetric duty cycles. The main drawback is the failure to limit the worst-case neighbor discovery delay. Deterministic protocols, on the other hand, by either using quorum systems or prime numbers for duty cycles, can theoretically bound the worst-case discovery delay, but have high discovery delay in average. Typically, the average delay in deterministic approaches, such as quorum-based approaches, Disco and U-Connect, is half of the worst-case delay. Searchlight is a deterministic neighbor discovery protocol which can provide a strict upper-bound on the worst-case discovery delay, while improves the average discovery delay by incorporating randomization techniques.

In Searchlight, each node has its own wake-up schedule of period l, which is configured by the target duty cycle length of the node. In each period of l slots, there are two active slots. The first active slot is called the anchor slot and is the first slot of the period. An important property concerning the anchor slots is that for a given node, the relative position of the anchor slot is fixed to the anchor slot of any other node. However, in most cases, the anchor slots of any two nodes cannot overlap with each other due to the non-synchronised clocks of them. Therefore, Searchlight implements a second active slot for each period which is the probe slot. The relative position of a probe changes from one period to another in order to overlap the anchor slot of any other node. Overlaps between probe and anchor slots, probe and probe slots, and anchor and anchor slots all lead to discovery. Since the period of Searchlight is l, the maximum relative offset between any two nodes is bounded by $\frac{l}{2}$ slots. Hence, a probe slot only needs to search the starting $\frac{l}{2}$ slots of each period to hit a anchor slot of any other node. Particularly, the relative position of a probe slot with respect to the anchor slot of the same period begins from 1 and increments by one each period until $\frac{l}{2}$. The same pattern is repeated in Searchlight. Fig. 9 shows an example of wake-up schedule in Searchlight with $l = 6$.

When nodes have symmetrical duty cycle lengths, the worst-case delay of Searchlight is bounded by $\lfloor \frac{l}{2} \rfloor$ periods, or $l \cdot \lfloor \frac{l}{2} \rfloor$ slots. Fig. 10 illustrates a discovery process in Searchlight for $l = 6$.

In the symmetrical case, the discovery is guaranteed between any pair of neighbor nodes because the relative offsets between their anchor slots are constant which depends on the relative offset between their local clocks. However, such discovery cannot be guaranteed in the asymmetrical case where two nodes have different duty-cycle lengths, i.e., different l values. A possible solution, as that in Disco and U-Connect, is to limit l to prime numbers. Searchlight takes another solution to address the duty cycle asymmetry by restricting the duty cycles to the form of p^i where p is the smallest duty cycle length, e.g., 2, 2^2, 2^3, $\cdots$ or 3, 3^2, 3^3, $\cdots$, etc., thus guaranteeing that the duty cycle of one node is multiples of that of the other node. The key idea hinging behind the solution is to keep the constant relative offset between their anchor slots. With this solution, the worst-case discovery delay in Searchlight between any two neighbor nodes a and b operating on l_a and l_b where l_a is a multiple of l_b is $\frac{l_b + \left[\lceil \frac{\lfloor t_a \rfloor}{2} \rceil\right]_{t_b}}{2}$ periods, or $\frac{l_b + \left[\lceil \frac{\lfloor t_a \rfloor}{2} \rceil\right]_{t_b}}{2} \cdot t_a$ slots.

The worst-case discovery in Search light is guaranteed by anchor-probe overlap. However, consider a symmetrical scenario where nodes a and b have a time difference of 1 slot, their probe slot can never overlap due to the constant relative offset of them. Motivated by the phenomena illustrated by this example, Searchlight further incorporates a probabilistic approach in its design, which increase the chance of overlaps between the probe slots of different nodes. Specifically, the relative offset of a probe slot regarding the corresponding anchor slot does not follow the pattern from 1 to $\frac{l}{2}$, but instead is a randomly chosen permutation of $\{1, 2, \cdots, \frac{l}{2}\}$. After integrating the probabilistic component, Searchlight maintains its discovery guarantee due to the overlap between a probe slot of a node a and a anchor slot of any other node b in the symmetrical case where $l_a = l_b$. Meanwhile, the probability of overlaps between two probe slots is significantly increased, leading to smaller average discovery delay. In the asymmetrical case where $l_a \neq l_b$, the introduction of the probabilistic component increases the worst-case discovery delay because the probe slot of the node with the bigger l may pass all the sleeping slots before hitting an anchor slot. To address this problem, the designers of Searchlight proposed a restricted version of the randomized probing, termed as restricted randomized probing, that can achieve the same worst-case discovery delay in the asymmetrical case as the symmetrical case.

To summarise, Searchlight further decreases both the average and the worst-case discovery delay compared to U-Connect and Disco. However, Searchlight still cannot support all duty cycles as it limits the duty cycle choices to power-multiples of the smallest duty cycle.

In Table 1, we give a synthetic comparison among the surveyed deterministic neighbor discovery protocols using various performance metrics. To decide when to terminate the execution of deterministic neighbor discovery protocols, given the upper-bound of the duty cycle length, we can compute the worst-case discovery delay D and simply terminate the discovery after D slots.

6. From direct neighbor discovery to indirect neighbor discovery

In the neighbor discovery protocols discussed previously, discovery occurs only when a node hears directly another node. Such approaches can be referred to as direct neighbor discovery protocols. In many cases, neighboring devices share common neighbors, which can be exploited to enable indirect neighbor discovery. Specifically, if we let devices propagate their neighbor list when they send their neighbor discovery beacons, a node can discover its neighbors indirectly via the received broadcast beacons from its neighbors. Motivated by this observation, Zhang et al. developed two cooperative neighbor discovery approaches to improve the discovery efficiency [79,80]. Their work represents two different design perspectives and approaches. In [79], we are given the energy budget and we seek to optimize discovery efficiency under the energy budget; in [80], we are given the upper-bound of discovery latency and we seek the minimum energy consumption to satisfy the discovery latency requirement.

Compared with the direct neighbor discovery protocols, their work presents a different design architecture with the following characteristics. Firstly, their approaches work on top of existing deterministic neighbor discover protocols (e.g., Disco and U-Connect) as a middleware to increase the neighbor discovery efficiency;

Slot index	0	1	2	3	4	5	6	7	8	9	10	...
Node *s* :	1	0	0	0	0	1	0	0	0	0	1	...
Node *a* :	1	0	0	1	0	0	1	0	0	1	0	...
Node *b* :	-	-	-	1	0	0	0	0	0	0	1	...

Fig. 11. Example of the neighbor discovery process in Acc.

secondly, their approaches exploit the collaboration of the direct neighbors to enable indirect neighbor discovery.

6.1. Accelerating neighbor discovery via indirect neighbor discovery

One of the proposed approach is called Acc [79]. Acc can be implemented as a middleware on existing neighbor discovery protocols (such as Searchlight and U-Connect) to further accelerate the neighbor discovery process. Acc exploit the broadcast nature of wireless media to support both direct neighbor discovery and indirect neighbor discovery. Specifically, the authors of [79] considered a situation where an effective discovery protocol, e.g., Disco, had already been deployed by nodes in a duty-cycled network. In the case where a user needs to accelerate the neighbor discovery process, it runs Acc by waking up in a number of extra slots carefully configured to maximise the performance gain in terms of discovery delay. The major optimisation objective in Acc design is how to assign extra active slots (i.e., extra energy budget) in order to optimize the neighbor discovery efficiency.

Under Acc, each device operates in one of the following modes during the neighbor discovery process:

- *Energy efficient discovery mode*: If a node does not need to accelerate its neighbor discovery process, it simply operates on this mode, which is more energy-efficient, by executing the underlying discovery protocol such as Disco. The only difference is that in the periodical discovery messages sent the node, it includes its neighbor node list containing its duty cycle length and the IDs and duty cycles of its neighbor nodes he has already discovered. In this way, each node can obtain information on the wake-up schedule of both its direct and indirect neighbors.
- *On-demand accelerated discovery mode*: If a node needs to accelerate its neighbor discovery process (e.g., trigger by the applications running on it), it enters the second mode with extra energy budget in terms of extra active slots which allow him to discover new neighbors during these slots. These extra slots are optimised by exploiting two types of neighbor discovery: direct discovery and indirect discovery via its neighbors.

We illustrate the key point in the second mode by the following example provided in [79]. As shown in Fig. 11, after the discovery of node a in slot 0, if s is allocated an extra slot for the coming 10 slot to accelerate the neighbor discovery, it can activate itself in slot 6 to exploit the indirect neighbor discovery via node a because s, who discovers node a in slot 0 and receives the neighbor list of a, is aware that a is planned to be active in slots 3 and 6 and the newly discovered neighbors of a in slot 3 (node b) is supposed to be broadcast by a in slot 6. Consequently, s can advance the discovery of node b by 4 slots from 10 to 6.

To optimise the discovery efficiency, the additional active slots should be carefully chosen given the energy budget. To address this optimisation problem, the authors developed a metric called *spatial-temporal coverage* to quantify the benefit of each potential active slot in the discovery of indirect neighbors. The *temporal diversity* is the number of slots of a neighbor already discovered are active during which s is not active. The *spatial similarity* is the likelihood of a neighbor of a neighbor already known to s being also the neighbor of s. Specifically, the two metrics are calculated as follows, given the actual time t_0 and a future slot t:

- The temporal diversity between two neighbor nodes i and j, defined as $\alpha^{i,j}_{t_0\to t}$, is computed by j as follows:

$$\alpha^{(i,j)}_{t_0\to t} = \frac{|m^{(i,i)}_{t_0\to t}| - |m^{(i,j)}_{t_0\to t}|}{t - t_0},$$

  where $m^{(i,j)}_{t_0\to t}$ denotes the set of common active slot between nodes i and during slots t_0 and t; $|m^{(i,j)}_{t_0\to t}|$ is thus the number of such common active slots; if $i = j$, $m^{(i,i)}_{t_0\to t}$ degenerates to the set of active slots of node i during the period. The numerator is thus the number of slots during which i is active but j is not active. It can be noticed that a larger $\alpha^{(i,j)}_{t_0\to t}$ indicates that the wake-up schedules of i and j are more heterogeneous and consequently, they may bring to each other more information about indirect neighbors which cannot be discovered directly.
- The spatial similarity between two neighbor nodes i and j, defined as $\beta^{i,j}_{t_0\to t}$, is computed by j as follows:

$$\beta^{(i,j)}_{t_0} = \frac{|n^{(i,j)}_{t_0}|}{|n^{(j,j)}_{t_0}|},$$

  where $n^{(i,j)}_{t_0}$ denotes the set of nodes which are already discovered by both nodes i and j as neighbors until slot t_0; when $i = j$, $n^{(j,j)}_{t_0}$ denotes the neighbor list of node j at t_0. It can be noted that a node with larger spatial similarity is more likely to bring potential neighbor information.

Based on the temporal diversity and the spatial similarity, s can rank the potential active slots based on the following metric:

$$\gamma^{s}_{t_0\to t} = \sum_{i\in n^{(s,s)}_{t_0}} \alpha^{(i,s)}_{t_0\to t}\beta^{(i,s)}_{t_0},$$

where $n^{(s,s)}_{t_0}$ is the neighbor list of node s at slot t_0. The above metric signifies that s prefers to activate itself in slots where nodes with higher temporal diversity and spatial similarity are active.

An online activation scheduling algorithm is then proposed that works as follows: s activates the slot with the highest rank given its energy budget in terms of the number of extra activate slots; after every active slot, $n^{(s,s)}_{t_0}$ is updated using latest neighbor information obtained in the active slot; the process is repeated until the budget is used. As an application, the authors further presented an application called Crowd-Alert where Acc was used by taxi drivers to choose directions with fewer competitors but more clients.

6.2. Saving energy via indirect neighbor discovery

The second neighbor discovery protocol developed by Zhang et al. [80] exploits the information from the direct neighbor nodes to reduce the energy consumption (i.e., number of active slots) in the neighbor discovery process, thus increasing its energy efficiency. Specifically, their approach is motivated by the following observation. Existing deterministic neighbor discovery protocols are focused on the pairwise discovery via direct one-hop communication. Besides the direct neighbor discovery, a node can exploit its direct neighbor nodes to enhance its neighbor discovery process. As a consequence, some active slots can be removed to reduce energy consumption without affecting neighbor discovery.

To formulate the phenomena of indirect neighbor discovery, the authors proposed a new system called *extended quorum system*.

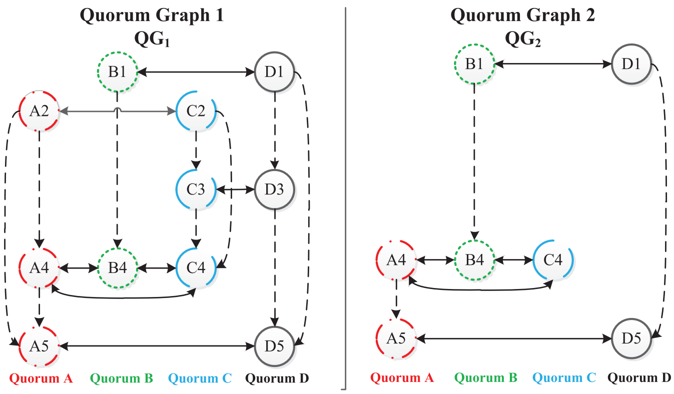

Fig. 12. Example of a quorum graph with the corresponding wake-up schedules [80]: solid lines represent discoveries among different nodes; dashed lines represent neighbor information flow from an early slot to a later slot.

Specifically, they use quorum graph to characterise all the possible paths along which the neighbor information can spread. We illustrate the concept of the extended quorum system by an example provided in [80], as shown in Fig 12. In the quorum graph, vertices in the same row correspond to the same active slot with each row corresponding to a node; vertices in the same column correspond to active slots of the same node; the edges indicate the possible neighbor information propagation flow. Consider the left quorum graph in Fig. 12, A and C can directly discovery each other in slot 2; A and D can discovery each other directly and indirectly via node C who can obtain the information of D in slot 3 and pass this information to A in slot 4.

An important property of a quorum graph is the *reachability*. Specifically, a quorum graph is said to have the property of reachability if any quorum of the graph can get to at least a vertex in any other quorum in the graph. If every quorum in a quorum graph can get to any other quorum via only its own vertices, the quorum graph is said to be *directly reachable*; if the reachability is enabled by the vertices of other quorums, the graph is *indirectly reachable*. It can be easily checked that the left quorum graph in Fig. 12 has direct reachability and the right quorum graph in Fig. 12 has indirect reachability.

For neighbor discovery, the direct and indirect reachability of a quorum graph indicate that the neighborhood information of a node can propagate to others directly and indirectly via other nodes, respectively. A directly reachable quorum graph is called a *legacy quorum system*. A indirectly reachable quorum graph is called an *extended quorum system*. The quorum graphs in existing deterministic neighbor discovery protocols are legacy quorum systems.

The objective of the work in [80], termed as *quorum reachability minimisation* problem, is to pick minimum number of vertices in a given legacy quorum system to maintain the reachability between any pair of quorums. The selected subset of vertices lead to a new quorum graph representing an extended quorum system. Concretely, solving the quorum reachability minimization problem consists of removing the redundant active slots in a decentralised way to achieve better energy performance without affecting neighbor discovery performance.

Since the quorum reachability minimization problem is NP-hard, the authors developed a heuristic algorithm to tackle it. The key idea is to select the rows that contribute to the global reachabilities with the fewest active slots in each iteration until the reachability is achieved for the whole graph. Quantitatively, the authors defined the metric $C_x = \frac{T_x}{N_x}$ for a row x, where T_x is the number extra reachabilities by picking vertices in row x, N_x denotes the number of vertices in row x. At each iteration, the row with the largest C_x value is activated. A snapshot of the execution of the developed heuristic algorithm is illustrated in Fig. 13: in the first iteration (Step 2), row 4 is selected because it give the maximal contribution to reachabilities $C_4 = \frac{6}{3} = 2$ (adding row 4 enables mutual discovery between A, B and C, thus $T_4 = 6$); similarly in the second iteration (Step 3), row 1 is selected; in the third iteration (Step 4), row 5 is selected and the reachability of the graph is achieved; the algorithm is thus terminated.

The authors then numerically evaluated the performance of the heuristic algorithm as an augmenting middleware for neighbor discovery. Their experiments showed that the algorithm could decrease energy consumption by more than 50% while increasing the discovery delay by maximum 5% compared to other neighbor discovery protocols.

7. Conclusion and perspective

In this paper, we have surveyed the latest developments in neighbor discovery protocols in mobile sensing applications by covering probabilistic, deterministic and collaborative approaches. Whereas numerous neighbor discovery solutions have been developed in the past few years, there are still several major research challenges that deserve further research attention and investigation.

7.1. Combining probabilistic and deterministic approaches: towards a hybrid solution

We have surveyed major probabilistic and deterministic neighbor discovery protocols in the literature. Probabilistic protocols have the advantages of being stationary due to the memoryless nature. They are thus particularly robust in decentralised networks. Moreover, they usually perform well in the average case by limiting the expected discovery delay. The main drawback of them is the lack of discovery guarantee. Compared to the probabilistic approaches that work well in the average case while fail to bound the worst-case discovery delay, deterministic protocols have good worst-case performance while usually have longer expected discovery delay.

A natural question that arises is whether and how we can combine the probabilistic and deterministic approaches to achieve a desired balance between the worst-case and average discovery delay. Searchlight has made some efforts by introducing a probabilistic component in the deterministic wake-up schedule. However, their solution is specific to the design of searchlight. In a broad sense, it remains an open while important research question to quantify the trade-off between the worst-case and the average discovery delay if we add a probabilistic component in a deterministic approach or a deterministic component in a random approach.

7.2. Fine granularity in supporting duty cycles

One of the key challenges in devising deterministic neighbor discovery protocols is to support heterogeneous duty cycle lengths with sufficiently fine granularity, ideally all the duty cycle lengths. Existing deterministic protocols cannot fully support all duty cycle lengths due to their constrained choices on either prime numbers (Disco, U-Connect) or power-multiples of the smallest duty cycles (Searchlight). As a consequence, they can only support a limited power consumption configurations. For example, only less than 20% of the duty cycles less than 1000 are prime numbers. Although in Disco, a third prime number can be added to further increase the granularity of supported duty cycles, nodes usually need to choose large prime numbers to approximate the duty cycle which may increase the worst-case discovery delay.

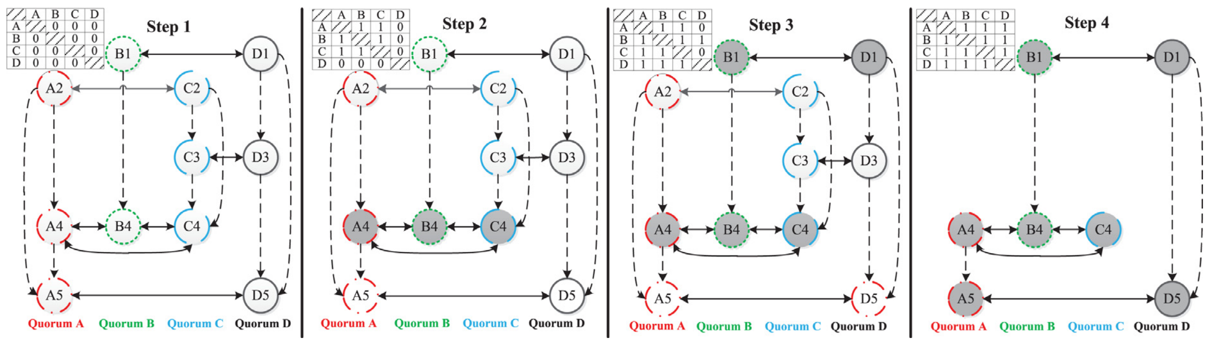

Fig. 13. Example illustrating the heuristic algorithm [80].

The work of Zhang et al. [79] partially addresses this problem by reserving some energy budget to be activated when the applications need a quick discovery. However, their approach does not provide deterministic worst-case bound on the discovery delay and still relies on the underlying neighbor discovery protocols such as Disco and U-Connect. The work of [14,15] uses another co-primality-based approach based on consecutive odd numbers to support more duty cycles, but still the approach cannot support all duty cycles. We thus argue that a theoretical design framework is called for to enable the support for arbitrary duty cycle lengths without any prior coordination and to quantify the performance tradeoff among different metrics such as energy-consumption, worst-case discovery delay when nodes have arbitrary duty cycle lengths.

Moreover, from the perspective of energy conservation, it is beneficial for nodes to dynamically adjust their duty cycles based on energy regiments, contact patterns and mobility, etc. How to make the existing neighbor discovery protocols adaptable or how to design novel dynamic and adaptable solutions consists of a pertinent avenue for future research. A possible starting point is to allow devices to switch between a set of duty cycles with guaranteed discovery.

7.3. Exploiting device mobility

Most neighbor discovery protocols presented in the survey do not make specific assumptions about the mobility patterns to achieve neighbour discovery. In other words, they can achieve discovery without exploiting knowledge concerning mobility patterns. This is sometimes an advantage as they can be applied to either static or mobile devices and are robust to mobility as their performance does not depend on mobility. However, there are many practical scenarios where by exploiting mobility patterns of devices, neighbor discovery can be facilitated. Hence, how to design mobility-aware neighbor discovery protocols that exploit device mobility to facilitate neighbor discovery, either limiting discovery delay or increasing energy efficiency, is a pertinent research direction. For example, a possible idea is to adapt a more flexible duty cycle mode and to allocate more active slots when mobility is important to ensure more agile neighbor discovery.

7.4. Multiple channel neighbor discovery with heterogeneous duty cycles

Another research dimension is to study the neighbor discovery problem in the multi-channel environment with heterogeneous duty cycles. Nowadays, more and more modern wireless devices (even tiny sensor nodes) are able to operate on a wide swath of spectrum subdivided into multiple orthogonal channels so as to get extra performance gain by exploiting parallel transmissions and reducing both intra and inter-network interferences. This reality inspires a compelling research question: how to design neighbor discovery protocols when the mobile devices can hop across multiple channels?

The multi-channel paradigm, combined with the duty cycle based operation mode of wireless nodes, poses three major challenges for devising neighbor discovery protocols. The first two challenges are the lack of clock synchronisation and the asymmetrical duty cycle lengths, which has been extensively addressed in the existing literature. The third one, brought by the multi-channel paradigm, is the asymmetry of channel perceptions among nodes, i.e., different nodes may have different set of accessible channels and they can discovery each other only if they are on the same channel at the same time.

Given the above challenges, we can formulate the problem of neighbor discovery in multi-channel networks as follows: *How can two neighbor nodes operating on different duty cycles, without clock synchronisation and common channel perceptions, discover each other within a limited time?*

Chen et al. have addressed the problem of multi-channel neighbor discovery problem by requiring a pair of neighbor nodes to be able to discovery each other on every commonly accessible channels [14]. However, the worst-case discovery delay of the proposed solution may become significant when the number of channels is large. A number of solutions have been proposed to enable pairwise rendezvous in cognitive radio networks (cf. [10,11,13,24,26,42,62,67,82]), which exhibit some similarity with the neighbor discovery problem we investigate. However, the problem of heterogeneous duty cycles is absent there as cognitive nodes just hop across channels to rendezvous with their communication peers without falling into asleep. Therefore, new research efforts are needed to address the three design challenges in a holistically way towards devising efficient and robust neighbor discovery protocols.

7.5. Practical implementation and integration of neighbor discovery with other network functionalities

Last but not least, it is important to conduct experimental evaluation and implementation of the proposed neighbor discovery protocols using commodity hardware. Furthermore, it is beneficial to integrate the neighbor discovery functionality into the media access protocol by jointly designing the two supporting primitives because the two are closed coupled one with the other. For example, neighbor discovery is important for the media access arbitration. Conversely, neighbor discovery is also built on the messages exchanged among neighboring nodes and hence, on the underlying MAC protocol. There are some existing solutions that integrate the neighbor discovery functionality into other functional components such as MAC protocols [30,61] and group management mechanisms [60]. However, they are focused on specific applications and functionalities. There is still a long way to go towards

Please cite this article as: L. Chen, K. Bian, Neighbor discovery in mobile sensing applications: A comprehensive survey, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.05.005

designing and prototyping a complete set of MAC primitives integrating neighbor discovery and related primitives that can satisfy application layer requirements in terms of delay and energy consumption.

Acknowledgments

The work of L. Chen is supported by the ANR grant Green-Dyspan (ANR-12-IS03).

References

[1] Nintendo 3ds - streetpass. http://www.nintendo.com/3ds/hardware, 2010.
[2] IEEE 802.22 working group. http://www.ieee802.org/22/, 2012.
[3] Sony ps vita - near. http://us.playstation.com/psvita, 2012.
[4] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Netw. 7 (3) (2009).
[5] I. Anderson, Combinatorial Designs and Tournaments, Oxford University Press, 1998.
[6] C. Arachchige, S. Venkatesan, N. Mittal, An asynchronous neighbor discovery algorithm for cognitive radio networks, in: Proceedings DySPAN, 2008.
[7] D.J. Baker, A. Ephremides, The architectural organization of a mobile radio network via a distributed algorithm, IEEE Trans. Commun. 29 (11) (1981) 1694–1701.
[8] M. Bakht, M. Trower, R.H. Kravets, Searchlight: won't you be my neighbor, in: Proceedings Mobicom, 2012.
[9] J. Biagioni, T. Gerlich, T. Merrifield, J. Eriksson, Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones, in: Proceedings SenSys, 2011.
[10] K. Bian, J.-M. Park, R. Chen, A quorum-based framework for establishing control channels in dynamic spectrum access networks, in: Proceedings Mobicom, 2009.
[11] K. Bian, J.-M. Park, R. Chen, Asynchronous channel hopping for establising rendezvous in cognitive radio networks, in: Proceedings Infocom, 2011.
[12] S.A. Borbash, A. Ephremides, M.J. McGlynn, An asynchronous neighbor discovery algorithm for wireless sensor networks, Ad Hoc Networks 5 (7) (2007) 998–1016.
[13] L. Chen, K. Bian, L. Chen, C. Liu, J. Park, X. Li, A group-theoretic framework for rendezvous in heterogeneous cognitive radio networks, in: Proceedings MobiHoc, 2014.
[14] L. Chen, K. Bian, M. Zheng, Heterogeneous multi-channel neighbor discovery formobile sensing applications: theoretical foundationand protocol design, in: Proceedings MobiHoc, 2014.
[15] L. Chen, R. Fan, K. Bian, L. Chen, M. Gerla, T. Wang, X. Li, On heterogeneous neighbor discovery in wireless sensor networks, in: Proceedings Infocom, 2015.
[16] R. Cohen, B. Kapchits, Continuous neighbor discovery in asynchronous sensor networks, IEEE/ACM Trans. Netw. 19 (1) (2011) 69–79.
[17] J. Du, E. Kranakis, O.M. Ponce, S. Rajsbaum, Neighbor discovery in a sensor network with directional antennae, in: Algorithms for Sensor Systems, in: Lecture Notes in Computer Science, 7111, Springer Berlin Heidelberg, 2012, pp. 57–71.
[18] P. Dutta, P.M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, A. Woodruff, Common sense: participatory urban sensing using a network of handheld air quality monitors, in: Proceedings SenSys, 2009.
[19] P. Dutta, D. Culler, Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications, in: Proceedings Sensys, 2008.
[20] P. Dutta, L. Subramanian, Human-enabled microscopic environmental mobile sensing and feedback, AAAI Spring Symposium: Artificial Intelligence for Development, 2010.
[21] S.B. Eisenman, E. Miluzzo, N.D. Lane, R.A. Peterson, G.-S. Ahn, A.T. Campbell, The bikenet mobile sensing system for cyclist experience mapping, in: Proceedings SenSys, 2007.
[22] J. Elson, K. Römer, Wireless sensor networks: a new regime for time synchronization, Sigcomm Comput. Commun. Rev. 33 (1) (2003).
[23] V. Galluzzi, T. Herman, Survey: discovery in wireless sensor networks, Int. J. Distrib. Sensor Netw. (2014).
[24] R. Gandhi, C.-C. Wang, Y.C. Hu, Fast rendezvous for multiple clients for cognitive radios using coordinated channel hopping, in: Proceedings Secon, 2012.
[25] R. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges, IEEE Commun. Mag. 49 (11) (2011) 32–39.
[26] Z. Gu, H. Pu, Q.-S. Hua, F.-C.-M. Lau, Improved rendezvous algorithms for heterogeneous cognitive radio networks, in: Proceedings Infocom, 2015.
[27] K. Han, J. Luo, Y. Liu, A. Vasilakos, Algorithm design for data communications in duty-cycled wireless sensor networks: a survey, IEEE Commun. Mag. 51 (7) (2013).
[28] J.-H. Huang, S. Amjad, S. Mishra, Cenwits: a sensor-based loosely coupled search and rescue system using witnesses, in: Proceedings SenSys, 2005.
[29] T. Huang, H. Chen, L. Cui, Y. Zhang, Easind: neighbor discovery in duty-cycled asynchronous multichannel mobile WSNs, Int. J. Distrib. Sensor Netw. (2013).
[30] G. Jakllari, W. Luo, S.V. Krishnamurthy, An integrated neighbor discovery and mac protocol for ad hoc networks using directional antennas, IEEE Trans. Wireless Commun. 6 (3) (2007).
[31] J.-R. Jiang, Expected quorum overlap sizes of quorum systems for asynchronous power-saving in mobile ad hoc networks, Comput. Netw. 52 (17) (2008) 3296–3306.
[32] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, T.-H. Lai, Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks, ACM/Springer MONET 10 (1,2) (2005) 169–181.
[33] H. Jun, M.H. Ammar, M.D. Corner, E.W. Zegura, Hierarchical power management in disruption tolerant networks using traffic-aware optimization, Comput. Commun. 32 (16) (2009) 1710–1723.
[34] A. Kandhalu, K. Lakshmanan, R. Rajkumar, U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol, in: Proceedings IPSN, 2010.
[35] N. Karowski, A.C. Viana, A. Wolisz, Optimized asynchronous multi-channel discovery of ieee 802.15.4-based wireless personal area networks, IEEE Trans. Mobile Comput. 12 (10) (2013) 1972–1985.
[36] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, 1972, pp. 85–103.
[37] S. Krishnamurthy, N. Mittal, R. Chandrasekaran, S. Venkatesan, Neighbor discovery in multi-receiver cognitive radio networks, Intl J. Computers and Applications (IJCA) 31 (1) (2009) 50–57.
[38] S. Lai, Heterogenous quorum-based wakeup scheduling for duty-cycled wireless sensor networks, Virginia Polytechnic Institute and State University, 2009 Ph.D. thesis.
[39] S. Lai, B. Ravindran, H. Cho, Heterogenous quorum-based wakeup scheduling in wireless sensor networks, IEEE Trans. Comput. 59 (11) (2010).
[40] S. Lai, B. Zhang, B. Ravindran, H. Cho, CQS-Pair: cyclic quorum system pair for wakeup scheduling in wireless sensor networks, in: Proceedings OPODIS, 2008, pp. 295–310.
[41] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A. Campbell, A survey of mobile phone sensing, IEEE Commun. Mag. 48 (9) (2010) 140–150.
[42] Z.Y. Lin, H. Liu, X.W. Chu, Y.-W. Leung, Jump-stay based channel hopping algorithm with guaranteed rendezvous for cognitive radio networks, in: Proceedings Infocom, 2011.
[43] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J.A. Stankovic, D. Siu, Automatic and robust breadcrumb system deployment for indoor firefighter applications, in: Proceedings MobiSys, 2010.
[44] T. Liu, C.M. Sadler, P. Zhang, M. Martonosi, Implementing software on resource-constrained mobile sensors: experiences with impala and zebranet, in: Proceedings MobiSys, 2004.
[45] W. Luk, T. Huang, Two new quorum based algorithms for distributed mutual exclusion, in: Proceedings ICDCS, 1997.
[46] R. Madan, S. Lall, An energy-optimal algorithm for neighbor discovery in wireless sensor networks, ACM/Springer MONET 11 (3) (2006) 317–326.
[47] M.J. McGlynn, S.A. Borbash, Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks, in: Proceedings MobiHoc, 2001.
[48] T. Meng, F. Wu, G. Chen, On designing neighbor discovery protocols: a code-based approach, in: Proceedings INFOCOM, 2014.
[49] D.L. Mills, Computer Network Time Synchronization: The Network Time Protocol, CRC Press, 2011.
[50] E. Miluzzo, N.D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S.B. Eisenman, X. Zheng, A.T. Campbell, Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application, in: Proceedings SenSys, 2008.
[51] E. Miluzzo, M. Papandrea, N.D. Lane, A.M. Sarroff, S. Giordano, A.T. Campbell, Tapping into the vibe of the city using vibn, a continuous sensing application for smartphones, in: Proceedings 1st International Symposium on from Digital Footprints to Social and Community Intelligence, 2011.
[52] N. Mittal, S. Krishnamurthy, R. Chandrasekaran, S. Venkatesan, Y. Zeng, On neighbor discovery in multi-channel cognitive radio networks, J. Par. Dist. Comput. (JPDC) 69 (7) (2009) 623–637.
[53] N. Mittal, Y. Zeng, S. Venkatesan, R. Chandrasekaran, Randomized distributed algorithms for neighbor discovery in multi-hop multi-channel heterogeneous wireless networks, in: Proceedings ICDCS, 2013.
[54] R. Murawski, E. Felemban, E. Ekici, S. Park, S. Yoo, K. Lee, J. Park, Z.H. Mir, Neighbor discovery in wireless networks with sectored antennas, Ad Hoc Netw. 10 (1) (2012) 1–18.
[55] A. N., N. Yiannoutsou, A review of mobile location-based games for learning across physical and virtual spaces, J. Univ. Comput. Sci. 18 (15) (2012) 2120–2142.
[56] I. Niven, H.S. Zuckerman, H.L. Mongomery, An Introduction to the Theory of Numbers, John Wiley & Sons, 1991.
[57] J. Paek, J. Kim, R. Govindan, Energy-efficient rate-adaptive gps-based positioning for smartphones, in: Proceedings MobiSys, 2010.
[58] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings SenSys, 2004.
[59] R. Pozza, M. Nati, S. Georgoulas, K. Moessner, A. Gluhak, Neighbour discovery for opportunistic networking in internet of things scenarios: a survey, IEEE Access 3 (2015) 1101–1131.
[60] A. Purohit, B. Priyantha, J. Liu, Wiflock: collaborative group discovery and maintenance in mobile sensor networks, in: Proceedings IPSN, 2011, pp. 37–48.
[61] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, S. Polit, Ad hoc networking with directional antennas: a complete system solution, IEEE J. Selected Areas Commun. 23 (1) (2005) 496–506.
[62] J. Shin, D. Yang, C. Kim, A channel rendezvous scheme for cognitive radio networks, IEEE Comm. Let. 14 (10) (2010) 954–956.
[63] D. Souza, E. Silva, G.C. Delacruz, Hybrid reality games reframed potential uses in educational contexts, Games Culture 1 (3) (2006) 231–251.

Please cite this article as: L. Chen, K. Bian, Neighbor discovery in mobile sensing applications: A comprehensive survey, Ad Hoc Networks (2016), http://dx.doi.org/10.1016/j.adhoc.2016.05.005

[64] K. Sreekanth, S. Sarfaraj, A survey on neighbor discovery in asynchronous wireless sensor network, Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2 (9) (2012).
[65] D.R. Stinson, Combinatorial Designs: Constructions and Analysis, Springer Verlag, 2003.
[66] W. Sun, Z. Yang, K. Wang, Y. Liu, Hello: A generic flexible protocol for neighbor discovery, in: Proc. INFOCOM, 2014.
[67] N. Theis, R. Thomas, L. DaSilva, Rendezvous for cognitive radios, IEEE Trans. Mobile Comput. 10 (2) (2011) 216–227.
[68] A. Thiagarajan, J. Biagioni, T. Gerlich, J. Eriksson, Cooperative transit tracking using smart-phones, in: Proceedings SenSys, 2010.
[69] Y.-C. Tseng, C.-S. Hsu, T.-Y. Hsieh, Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks, in: Proceedings Infocom, 2002.
[70] S. Vasudevan, M. Adler, D. Goeckel, D. Towsley, Efficient algorithms for neighbor discovery in wireless networks, IEEE/ACM Trans. Netw. 21 (1) (2013) 69–83.
[71] S. Vasudevan, J. Kurose, D. Towsley, On neighbor discovery in wireless networks with directional antennas, in: Proceedings Infocom, 2005.
[72] S. Vasudevan, D. Towsley, D. Goeckel, R. Khalili, Neighbor discovery in wireless networks and the coupon collector's problem, in: Proceedings MobiCom, 2009.
[73] A. Willig, N. Karowski, J.-H. Hauer, Passive discovery of ieee 802.15.4-based body sensor networks, Ad Hoc Networks 8 (7) (2010) 742–754.
[74] T. Yan, V. Kumar, D. Ganesan, Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones, in: Proceedings MobiSys, 2010.
[75] T. Yan, M. Marzilli, R. Holmes, D. Ganesan, M. Corner, mcrowd: a platform for mobile crowdsourcing, in: Proceedings SenSys, 2009.
[76] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: Proceedings Infocom, 2002.
[77] W. Zeng, S. Vasudevan, X. Chen, B. Wang, A. Russell, W. Wei, Neighbor discovery in wireless networks with multipacket reception, in: Proceedings MobiHoc, 2011.
[78] Y. Zeng, N. Mittal, S. Venkatesan, R. Chandrasekaran, Fast neighbor discovery with lightweight termination detection in heterogeneous cognitive radio networks, in: Proceedings 9th International Symposium on Parallel and Distributed Computing (ISPDC), 2010.
[79] D. Zhang, T. He, Y. Liu, Y. Gu, F. Ye, R.K. Ganti, H. Lei, Acc: generic on-demand accelerations for neighbor discovery in mobile applications, in: Proceedings SenSys, 2012.
[80] D. Zhang, T. He, Y. Liu, F. Ye, R.K. Ganti, H. Lei, EQS: neighbor discovery and rendezvous maintenance with extended quorum system for mobile sensing applications, in: Proceedings ICDCS, 2012.
[81] P. Zhang, C.M. Sadler, S.A. Lyon, M. Martonosi, Hardware design experiences in zebranet, in: Proceedings SenSys, 2004.
[82] Y. Zhang, Q. Li, G. Yu, B. Wang, Etch: efficient channel hopping for communication rendezvous in dynamic spectrum access networks, in: Proceedings Infocom, 2011.
[83] Z. Zhang, B. Li, Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: algorithms and comparisons, IEEE Trans. Wireless Commun. 7 (5) (2008) 1540–1549.
[84] R. Zheng, J.C. Hou, L. Sha, Asynchronous wakeup for ad hoc networks, in: Proceedings MobiHoc, 2003.

Lin Chen received his B.E. degree in Radio Engineering from Southeast University, China in 2002 and the Engineer Diploma from Telecom ParisTech, Paris in 2005. He also holds a M.S. degree of Networking from the University of Paris 6. He currently works as associate professor in the department of computer science of the University of Paris-Sud. His main research interests include modeling and control for wireless networks, distributed algorithm design and game theory.

Kaigui Bian received his Ph.D. degree in computer engineering from Virginia Tech, Blacksburg, USA in 2011. He is currently an associate professor in the Institute of Network Computing and Information Systems, School of EECS at Peking University. His research interests include mobile computing, cognitive radio networks, network security and privacy.