Randomized Distributed Algorithms for Neighbor Discovery in Multi-Hop Multi-Channel Heterogeneous Wireless Networks

Neeraj Mittal Yanyan Zeng S.Venkatesan R. Chandrasekaran
Department of Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA
Emails: neerajm@utdallas.edu zengyanyan@student.utdallas.edu venky@utdallas.edu chandra@utdallas.edu

***Abstract*—An important first step when deploying a wireless ad hoc network is *neighbor discovery* in which every node attempts to determine the set of nodes it can communicate within one wireless hop. In the recent years, *cognitive radio (CR)* technology has gained attention as an attractive approach to alleviate spectrum congestion. A CR transceiver can operate over a wide range of frequencies possibly spanning multiple frequency bands. A CR node can opportunistically utilize unused wireless spectrum without interference from other wireless devices in its vicinity. Due to spatial variations in frequency usage and hardware variations in radio transceivers, different nodes in the network may perceive different subsets of frequencies available to them for communication. This *heterogeneity* in the available channel sets across the network increases the complexity of solving the neighbor discovery problem in a CR network. In this paper, we design and analyze several randomized algorithms for neighbor discovery in such a (heterogeneous) network under a variety of assumptions.**

***Keywords*-multi-hop multi-channel wireless network; cognitive radio technology; heterogeneous channel availability; neighbor discovery; randomized algorithm; asynchronous system; clock drift**

I. Introduction

Neighbor discovery is an important step in forming a *self-organizing* wireless ad hoc network without any support from an existing communication infrastructure [1], [2]. When deployed, nodes initially may have no prior knowledge of other nodes that they can (directly) communicate with. The results of neighbor discovery can then be used to solve other important communication problems such as medium access control [3], [4], clustering [5], [6], collision-free scheduling [7], [8], and topology control [9], [10]. Many algorithms for solving these problems implicitly assume that all nodes know their one-hop and sometimes even two-hop neighbors.

Cognitive radio (CR) technology has recently emerged as a promising approach for improving spectrum utilization efficiency and meeting the increased demand for wireless communications [11]. A CR node can scan a part of the wireless spectrum, and identify unused or underutilized channels in the spectrum [11]. CR nodes in a network can then use these channels opportunistically for communication among themselves even if the channels belong to licensed users. The licensed users are referred to as the *primary users*, and CR nodes are referred to as the *secondary users*. (Of course, when a primary user arrives and starts using its channel, the secondary users have to vacate the channel.) Due to spatial variations in frequency usage/interference, hardware variations in radio transceivers and uneven propagation of wireless signals, different nodes in the network may perceive different subsets of frequencies available to them for communication. This gives rise to a *multi-hop, multi-channel, heterogeneous wireless network*, abbreviated as *M^2HeW network*. The focus of this paper is on solving the neighbor discovery problem in an M^2HeW network.

Related Work: Most of the prior work on neighbor discovery assumes a single channel wireless network (*e.g.*, [1], [2], [12]–[17]. Some of the neighbor discovery algorithms for a single channel network, such as those proposed in [2], can be easily extended to work for a multi-channel network (including a heterogeneous network). Let the collective set of all channels over which radio nodes in the network are capable of operating be referred to as the *universal channel set*. The main idea is to execute a separate instance of single-channel neighbor discovery algorithm on all channels in the universal channel set *concurrently*. A node only participates in instances that are associated with channels in its available channel set. However, this simple approach has several disadvantages. First, it requires that all nodes have to agree on the composition of the universal channel set. Second, the time complexity of the algorithm for multi-channel network (obtained as above) will always be *linear* in the size of the universal channel set. This is true even if the available channel set of all nodes contain a single common channel. This may happen if all channels in the universal channel set but one are busy and cannot be used by any node in the network. In many cases, the available channel sets of nodes may be much smaller than the universal channel set. Third, all nodes should start executing the algorithm at the same time. Otherwise, different nodes may tune to different channels in the same time slot, thereby causing the multi-channel neighbor discovery algorithm to fail.

Neighbor discovery algorithms for a multi-channel wireless network have been proposed in [18]–[22]. Work in

1063-6927/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCS.2011.92

IEEE computer society

[18] implicitly assumes that all channels are available to all nodes, and only considers a single hop network, whereas the work in [19] assumes that a node has multiple interfaces. Work in [20]–[22] assume a synchronous system in which all nodes initiate neighbor discovery at the same time. Moreover, the proposed algorithms are deterministic in nature, and have high time complexity that depends on the *product* of network size (actual [21], [22] or maximum [20]) and universal channel set size. Further, work in [21], [22] assumes that nodes can detect collisions.

Our Contributions: Our main contribution in the paper is a *randomized* neighbor discovery algorithm for an *M²HeW network* when the system is *asynchronous* that works under the following two assumptions: (a) nodes know an upper bound on the maximum degree of any node in the network, and (b) the maximum drift rate of the clock of any node is bounded by a small value (specifically, $\frac{1}{7}$). It does not require clocks of different nodes to be synchronized. Further, clocks of any two nodes may have arbitrary offset or skew with respect to each other.

Our algorithm for an asynchronous system is based on that for a synchronous system. Therefore, as additional contributions, we also present a suite of *randomized* neighbor discovery algorithms for an *M²HeW network* when the system is *synchronous* under a variety of assumptions such as whether: (i) nodes start executing the neighbor discovery algorithm at the same time and (ii) nodes know a good upper bound on the maximum degree of any node in the network.

Roadmap: The rest of the paper is organized as follows. We describe our system model for an M²HeW network in Section II. We present several randomized neighbor discovery algorithms for a synchronous system under a variety of assumptions and analyze their complexity in Section III. We present a randomized neighbor discovery algorithm for an asynchronous system and analyze its complexity in Section IV. Finally, Section V concludes the paper.

II. System Model

We assume a multi-hop multi-channel heterogeneous wireless (M²HeW) network consisting of one or more radio nodes. Let N denote the total number of radio nodes. Nodes do not know N. Each node is equipped with a transceiver, which is capable of operating over multiple frequencies or channels. However, at any given time, a transceiver can operate (either transmit or receive) over a single channel only. Further, a transceiver cannot transmit and receive at the same time. Transceivers of different nodes need not be identical; the set of channels over which a transceiver can operate may be different for different nodes.

Different nodes in a network may have different sets of channels available for communication. For example, in a cognitive radio network (a type of M²HeW network), each node can scan the frequency spectrum and identify the subset of unused or under-used portions of the spectrum, even those that have been licensed to other users or organizations [11]. A node can potentially use such frequencies to communicate with its neighbors until they are reclaimed by their licensed (primary) users [11]. Due to spatial variations in frequency usage/interference and hardware variations in radio transceivers, different nodes in the network may perceive different subsets of frequencies available to them for communication. We refer to the subset of frequencies or channels that a node can use to communicate with its neighbors as the *available channel set* of the node. For a node u, we use $\mathcal{A}(u)$ to denote its available channel set. We use S to denote the size of the largest available channel set, that is, $S = \max_u |\mathcal{A}(u)|$. Note that nodes do not know S.

We say that a node v is a *neighbor* of node u on a channel c if u can reliably receive any message transmitted by v on c provided no other node in the network is transmitting on c at the same time, and vice versa. For ease of exposition, in this paper, we assume that the communication graph is *symmetric*; our algorithms, however, can be easily extended to handle asymmetric communication graphs as well [23]. For a node u and a channel $c \in \mathcal{A}(u)$, we use $\Delta(u, c)$ to denote the number of neighbors, also known as *degree*, of u on c. We use Δ denote the maximum degree of any node on any channel, that is, $\Delta = \max_u \max_{c \in \mathcal{A}(u)} \Delta(u, c)$.

Note that, if nodes u and v are neighbors of each other on some channel, then u has to discover v and v has to discover u separately. It is convenient to assume two separate links—one from u to v and another from v to u. We use (u, v) to denote the link from u to v. We refer to the set of channels on which the link (u, v) can operate as the *span* of (u, v) and denote it by $span(u, v)$. Note that $span(u, v) \subseteq \mathcal{A}(u) \cap \mathcal{A}(v)$. We refer to the ratio of $|span(u, v)|$ to $|\mathcal{A}(v)|$ as the *span-ratio* of the link (u, v). Note that the span-ratio of any link lies between $\frac{1}{S}$ and 1. Further, the span-ratio of (u, v) may be different from the span-ratio of (v, u) (because $|\mathcal{A}(u)|$ may be different from $|\mathcal{A}(v)|$). We use ρ to denote the minimum span-ratio among all links. Note that nodes do not know ρ. Intuitively, ρ can be viewed as a measure of the *degree of homogeneity* in the network—the larger the value, the more homogeneous the network is in terms of channel availability for nodes and links. The running time of our algorithms is *inversely* proportional to ρ. When all links can operate on all available channels (an assumption made frequently in the literature), $\rho = 1$, which minimizes the running time of our algorithms. In general, the more heterogeneous the network is, the larger is the running time of our algorithms.

If nodes v and w are neighbors of node u on channel c and both v and w transmit on c at the same time, then their transmissions *collide* at u. If u listens on c at that time, then u hears only noise. We do not assume that nodes can detect collisions, that is, distinguish between background noise and collision noise.

For ease of exposition, in this paper, we assume that all channels or frequencies over which an M^{2}HeW network can operate have similar propagation characteristics. As a result, if a communication link from node u to node v can operate over some channel $c \in \mathcal{A}(u) \cap \mathcal{A}(v)$ then it can also operate over all channels in $\mathcal{A}(u) \cap \mathcal{A}(v)$. Our algorithms, however, can be easily adapted to handle diverse propagation characteristics of different channels [23].

We use "log" to refer to the logarithm to the base 2 and "ln" to refer to the natural logarithm. In this paper, we investigate the neighbor discovery problem both when the system is synchronous and when the system is asynchronous. We now describe how we model each type of system.

Synchronous System:: We assume that the execution of the system is divided into *synchronized time-slots*. In each time slot, each node can be in one of the following three modes: (i) *transmit mode* on some channel in its available channel set, (ii) *receive mode* on some channel in its available channel set, or (iii) *quiet mode* when the transceiver is shut-off.

Asynchronous System:: We assume that every node is equipped with a clock. Clocks of different nodes are not required to be synchronized. For a clock C and time t, we use $C(t)$ to denote the value of C at t. For a node u, we use C_u to denote the clock of u. Clock of a node may have non-zero drift and the drift rate may change over time. Drift rate of a clock C at time t is given by $\frac{dC}{dt} - 1$. If C is an ideal clock, then, $\forall t, \Delta t \geq 0$, $C(t + \Delta t) - C(t) = \Delta t$; thus $\frac{dC}{dt} = 1$. At any given time, different clocks may have different drift rates. Further, drift rate of the same clock may change over time both in magnitude and sign. We do, however, assume that the magnitude of the maximum drift rate is *bounded* and denote the bound by δ. This implies that $\forall t, \Delta t \geq 0$,

$$(1-\delta)\Delta t \;\leq\; C(t+\Delta t) - C(t) \;\leq\; (1+\delta)\Delta t \qquad (1)$$

For an ideal clock, $\delta = 0$. In practice, δ is quite small (*e.g.*, 10^{-6} seconds/second).

III. Neighbor Discovery in Synchronous System

We first describe our algorithms assuming that all nodes start neighbor discovery at the same time. We then relax this assumption and describe an algorithm for the case when nodes may start neighbor discovery at different times. Our algorithm for the asynchronous case builds upon the last algorithm.

A. Identical Start Times

We first assume that the nodes know some upper bound on maximum node degree. The bound need not be tight and, in fact, may be quite loose (since the dependence is logarithmic on the value of the upper bound.) But all nodes should agree on a *common* upper bound. We then relax this restriction and describe an algorithm for the case when such knowledge is not available.

1) Knowledge of Loose Upper Bound on Maximum Node Degree: Let Δ_{est} denote an upper bound on the maximum node degree as known to all nodes. The execution of the algorithm is divided into *stages*. Each stage consists of $\lceil \log(\Delta_{est}) \rceil$ time-slots. In each time-slot of a stage, a node randomly chooses a channel from its available channel set and transmits on that channel with a certain probability (and listens on that channel with the remaining probability). Specifically, in time-slot i of a stage, where $1 \leq i \leq \lceil \log(\Delta_{est}) \rceil$, a node u transmits on the selected channel, say c, with probability $\min\left(\frac{1}{2}, \frac{|\mathcal{A}(u)|}{2^i}\right)$ and listens on c with the remaining probability.

Algorithm 1: Neighbor discovery algorithm for a synchronous system with identical start times and knowledge of upper bound on maximum node degree.

// Algorithm for node u

Input: An upper bound on maximum node degree, say Δ_{est}.
Output: The set of neighbors along with the subset of channels that are common with the neighbor.

while *true* **do**
  // execute a stage
  for $i \leftarrow 1$ **to** $\lceil \log(\Delta_{est}) \rceil$ **do**
    // time-slot i of the current stage
    $c \leftarrow$ channel selected uniformly at random from $\mathcal{A}(u)$;
    $p \leftarrow \min\left(\frac{1}{2}, \frac{|\mathcal{A}(u)|}{2^i}\right)$;
    tune the transceiver to c;
    switch to transmit mode with probability p and receive mode with probability $1-p$;
    if *(in transmit mode)* **then**
      transmit a message containing $\mathcal{A}(u)$;
    else if *(heard a clear transmission)* **then**
      let the received message be sent by node v containing set A;
      add $\langle v, A \cap \mathcal{A}(u) \rangle$ to the set of neighbors;

A formal description of the algorithm is shown in Algorithm 1. We next analyze its running time. Consider a node u and let node v be its neighbor on a channel c. Note that $1 \leq \Delta(u,c) \leq \Delta \leq \Delta_{est}$. Let $k = \max(1, \lceil \log \Delta(u,c) \rceil)$. Clearly, we have:

$$2^{k-1} \;\leq\; \Delta(u,c) \;\leq\; 2^k \qquad (2)$$

We say that a time-slot t *covers* the link (v,u) on channel c if during t: (i) v transmits on c, (ii) u listens on c, and (iii) no other neighbor of u transmits on c. The three conditions collectively ensure that u receives a clear message from v (on c during t). We say that a stage s covers the link (v,u) on channel c if some time-slot of s covers (v,u) on c; also, s covers (v,u) if s covers (v,u) on some $c \in span(v,u)$. Finally, we say that a sequence of stages covers (v,u) if some stage of the sequence covers (v,u).

Consider a stage s and let τ denote the time-slot of s that satisfies (2). We first compute the probability that

(v, u) is covered on c during τ. Let $A(\tau, c)$, $B(\tau, c)$, $C(\tau, c)$ denote the three events corresponding to the three conditions (i), (ii) and (iii), respectively, required for coverage. Note that the three events are *mutually independent*. Therefore we first compute the probability of occurrence of events $A(\tau, c)$, $B(\tau, c)$ and $C(\tau, c)$ separately. We then compute the probability that s covers (v, u). The steps involved in computing the probabilities are fairly standard and can be found elsewhere [23]. We only state the results here. We have:

$$\Pr\{A(\tau, c)\} \geq \frac{1}{2\max(S, \Delta)} \tag{3}$$

$$\Pr\{B(\tau, c)\} \geq \frac{1}{2|\mathcal{A}(u)|} \tag{4}$$

$$\Pr\{C(\tau, c)\} \geq \frac{1}{4} \tag{5}$$

$$\Pr(s \text{ covers } (v, u)) \geq \frac{\rho}{16\max(S, \Delta)} \tag{6}$$

Now, consider a sequence of $M = \frac{16\max(S,\Delta)}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$ stages, where ϵ denotes an upper bound on the probability that neighbor discovery fails to complete successfully. We show that the probability that the link (v, u) is not covered within M stages is at most $\frac{\epsilon}{N^2}$. The steps for computing the probability are fairly standard and can be found elsewhere [23]. Formally,

$$\Pr((v, u) \text{ is not covered within } M \text{ stages}) \leq \frac{\epsilon}{N^2} \tag{7}$$

Finally, we have:

$$\begin{aligned} &\Pr(\text{neighbor discovery does not finish within } M \text{ stages}) \\ &= \Pr(\text{some link is not covered within } M \text{ stages}) \\ &\leq (\text{number of links in the network}) \times \frac{\epsilon}{N^2} \leq \epsilon \end{aligned} \tag{8}$$

Therefore, we have the following theorem:

Theorem 1. *Algorithm 1 guarantees that each node discovers all its neighbors on all channels within* $O\left(\frac{\max(S,\Delta)}{\rho} \log(\Delta_{est}) \log\left(\frac{N}{\epsilon}\right)\right)$ *time-slots with probability at least* $1 - \epsilon$.

2) No Knowledge of Maximum Node Degree: One way to derive a neighbor discovery algorithm when knowledge about maximum node degree is not available is as follows. Starting with an estimate of one for the maximum node degree, repeatedly run an instance of the algorithm that assumes knowledge about the maximum degree for a certain number of time-slots with geometrically increasing values for the estimate [2]. This approach cannot be used here because it requires computing the exact number of time-slots for which an instance of the knowledge-aware algorithm ought to be run such that, in case the estimate is correct, neighbor discovery completes with a desired success probability. Computing the number of time-slots for our (knowledge-aware) algorithm requires nodes to *a priori* know the values for other system parameters, namely N, S and ρ, whose values may not be known in advance. We instead employ the following approach used in [24]. Starting with an estimate of two, we repeatedly execute an instance of a stage with sequentially increasing values for the estimate.

Algorithm 2: Neighbor discovery algorithm for a synchronous system with identical start times and no knowledge of maximum node degree.

// Algorithm for node u

Output: The set of neighbors along with their available channel set

$d \leftarrow 2$;
while *true* **do**
    execute an instance of stage described in Algorithm 1 with Δ_{est} set to d;
    $d \leftarrow d + 1$;

A formal description of the algorithm is shown in Algorithm 2. It can be easily verified that, once d becomes at least Δ, each stage thereafter contains a time-slot that satisfies (2). To reach the success probability of at least $1-\epsilon$, from the analysis of Algorithm 1, it suffices for an execution to contain $M = \frac{16\max(S,\Delta)}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$ stages each of which consists of a time-slot that satisfies (2). In other words, neighbor discovery completes successfully with probability at least $1 - \epsilon$ within $\Delta + M$ stages. Note that $M = \Omega(\Delta)$. Therefore we have:

Theorem 2. *Algorithm 2 guarantees that each node discovers all its neighbors on all channels within* $O(M \log M)$ *time-slots with probability at least* $1 - \epsilon$*, where* $M = \frac{16\max(S,\Delta)}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$.

B. Variable Start Times

We assume that nodes know a "good" upper bound on the maximum node degree. Although the algorithm works even if the upper bound is loose, the running time of the algorithm may be too large since it depends linearly on the value of the upper bound.

The main idea behind our algorithm is to ensure that the transmission probability of a node is *same* for every time-slot (but may be different for different nodes). This allows us to prove that a given link is covered in a time slot with "sufficiently high" probability. Let Δ_{est} denote an upper bound on the maximum node degree as known to all nodes. In each time-slot, a node u randomly selects a channel from its available channel set, say c. It then transmits on c with probability $\min\left(\frac{1}{2}, \frac{|\mathcal{A}(u)|}{\Delta_{est}}\right)$ and listens on c with the remaining probability. A formal description of the algorithm is shown in Algorithm 3.

For the analysis of the running time, as in the case of Algorithm 1, we can compute the probability of occurrence

Algorithm 3: Neighbor discovery algorithm for a synchronous system with variable start times and knowledge of upper bound on maximum node degree.

// Algorithm for node u

Input: An upper bound on maximum node degree, say Δ_{est}.
Output: The set of neighbors along with the subset of channels that are common with the neighbor.

$p \leftarrow \min\left(\frac{1}{2}, \frac{|\mathcal{A}(u)|}{\Delta_{est}}\right)$;
while *true* **do**
  $c \leftarrow$ channel selected uniformly at random from $\mathcal{A}(u)$;
  tune the transceiver to c;
  switch to transmit mode with probability p and receive mode with probability $1-p$;
  if *(in transmit mode)* **then**
    transmit a message containing $\mathcal{A}(u)$;
  else if *(heard a clear transmission)* **then**
    let the received message be sent by node v containing set A;
    add $\langle v, A \cap \mathcal{A}(u)\rangle$ to the set of neighbors;

of events $A(\tau, c)$, $B(\tau, c)$ and $C(\tau, c)$. We have:

$$\begin{aligned}\Pr\{A(\tau, c)\} &= \frac{1}{|\mathcal{A}(v)|} \times \min\left(\frac{1}{2}, \frac{|\mathcal{A}(v)|}{\Delta_{est}}\right) \\ &= \frac{1}{\max\{\, 2|\mathcal{A}(v)|, \Delta_{est}\,\}} \\ &\geq \frac{1}{\max(2S, \Delta_{est})} \qquad (9)\end{aligned}$$

It can be verified that the inequalities for $\Pr\{B(\tau, c)\}$ in (4) and $\Pr\{C(\tau, c)\}$ in (5) are still valid (although the proofs have to be slightly modified). Let T_s be the time by which all nodes have initiated neighbor discovery. Using an analysis similar to that for Algorithm 1, we can prove the following result:

Theorem 3. *Algorithm 3 guarantees that each node discovers all its neighbors on all channels within $O\left(\frac{\max(2S,\Delta_{est})}{\rho}\log\left(\frac{N}{\epsilon}\right)\right)$ time-slots after T_s with probability at least $1-\epsilon$.*

Note that we no longer have a factor of $O(\log(\Delta_{est}))$ in the time complexity because we do not have stages any more.

IV. Neighbor Discovery in Asynchronous System

Our neighbor discovery algorithm for an asynchronous system is based on our neighbor discovery algorithm for a synchronous system with variable start times (Algorithm 3). Let Δ_{est} denote an upper bound on the maximum node degree as known to all nodes. In addition, our algorithm makes the following assumption about the maximum drift rate of the clock of any node δ:

Assumption 1. *The maximum drift rate of the clock of any node is bounded by $\frac{1}{7}$ seconds/second.*

The offset or skew between clocks of any two nodes may be arbitrarily large and, in fact, may grow with time. We now describe how to extend Algorithm 3 to solve the neighbor discovery algorithm in an asynchronous system.

Each node divides its time into equal-sized *frames*. Frames of different nodes are *not* required to be synchronized and may, in fact, be *misaligned*. A node measures the duration of a frame using its *local* clock. The length of a frame *as measured by a node using its local clock* is same for all nodes, say L. Note that, because of the clock drift, duration of a frame, when projected on real-time, may be different from L (shorter than L in case of positive drift and longer than L in case of negative drift). Specifically, it can be verified that the length of a frame in real-time lies in the range:

$$\frac{L}{1+\delta} \;\leq\; \text{frame length in real-time} \;\leq\; \frac{L}{1-\delta} \qquad (10)$$

A node further divides each frame into three equal-sized slots—equal with respect to its local clock. Therefore the duration of a slot as measured by a node using its local clock is $\frac{L}{3}$. At the beginning of each frame, a node u randomly selects a channel from its available channel set, say c. It then transmits on c during each slot of the frame with probability $\min\left(\frac{1}{2}, \frac{|\mathcal{A}(u)|}{3\Delta_{est}}\right)$ and listens on c during the entire frame with the remaining probability. In the former case, u transmits the same message during each slot of the frame. In the latter case, u listens for any clear messages it may receive during *any part* of the frame. The partition of a frame into slots is not important for a receiving node; they are only used by a transmitting node. Also, a node may receive multiple clear messages while listening during a single frame. The execution of a node with respect to its local clock is shown in Fig. 1. The execution of the network with respect to common real-time is shown in Fig. 2.

A formal description of the algorithm is shown in Algorithm 4. We now analyze the running time of the algorithm. In our analysis, unless otherwise stated, time refers to *real-time*. Of course, nodes do not have access to the real-time. We define two notions that we use in our analysis as follows:

Definition 1 (aligned pair). *We say that a pair of frames $\langle f, g\rangle$ is* aligned *if at least one slot of f lies completely within g.*

Definition 2 (overlapping frames). *For a frame f and a node u, let $overlap(f, u)$ denote the set of frames of u that overlap in real-time with f. Also, let $overlapAll(f)$ denote the subset of all frames that overlap in real-time with f.*

For example, as per the execution shown in Fig. 2, pairs $\langle f_1, g_1\rangle$ and $\langle f_2, h_1\rangle$ are aligned whereas the pair $\langle f_1, h_1\rangle$ is not. Also, $overlap(g_2, v) = \{f_1, f_2\}$ and $overlapAll(g_2) = \{f_1, f_2, g_2, h_1, h_2\}$.

One of the arguments we commonly use in our analysis is: "x adjacent slots of a node cannot strictly contain y adjacent

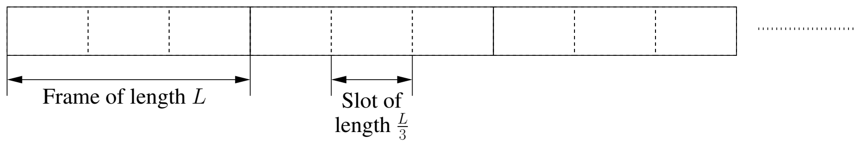

Figure 1. Execution of a node with respect to its local clock. All frames are of length L. All slots are of length $\frac{L}{3}$.

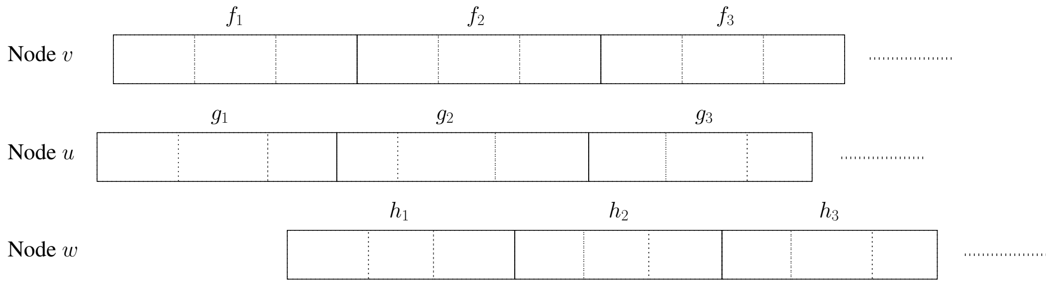

Figure 2. Execution of the network with respect to real-time. Frames may be of different lengths (even within the same node). Slots may be of different lengths (even within the same frame).

Algorithm 4: Neighbor discovery algorithm for an asynchronous system with knowledge of upper bound on maximum node degree.

```
// Algorithm for node u
Input: An upper bound on maximum node degree, say Δ_est.
Output: The set of neighbors along with the subset of
        channels that are common with the neighbor.

p ← min(1/2, |A(u)|/(3Δ_est));
while true do
    c ← channel selected uniformly at random from A(u);
    tune the transceiver to c;
    switch to transmit mode with probability p and receive
    mode with probability 1 − p;
    if (in transmit mode) then
        transmit a message containing A(u) during each slot
        of the frame;
    else
        // in receive mode
        foreach (clear message received during the frame)
        do
            let the received message be sent by node v
            containing set A;
            add ⟨v, A ∩ A(u)⟩ to the set of neighbors;
```

slots of another node" for certain specific values of x and y with $x < y$. To prove this, we argue that, for the statement to be false, the following inequality must hold:

$$\frac{xL}{3(1-\delta)} > \frac{yL}{3(1+\delta)}$$

and then show that it contradicts Assumption 1. The above inequality must hold for the statement to be false because the left hand side denotes the largest possible length of the time interval containing x adjacent slots, and the right hand side denotes the smallest possible length of the time interval containing y adjacent slots.

Lemma 4. *A frame of a node overlaps with at most three frames of any other node. Formally,*

$$\forall f, u :: |overlap(f, u)| \leq 3$$

Proof: Assume, on the contrary, that a frame of some node, say f, overlaps with at least four frames of another node. This implies that there are at least two consecutive frames that are strictly contained within f. From (10), a frame of a node can strictly contain two frames of another node only if the following condition holds:

$$\frac{L}{1-\delta} > \frac{2L}{1+\delta} \quad \Longrightarrow \quad \delta > \frac{1}{3}$$

This contradicts Assumption 1. □

Consider the link from node v to node u on channel c. Also, consider frames f of v and g of u such that the pair $\langle f, g\rangle$ is aligned. We extend the notion of a link covered by a time-slot (defined in Section III-A1) to a link covered by an aligned pair of frames. Specifically, the pair of aligned frames $\langle f, g\rangle$ *covers* the link (v, u) on channel c if: (i) v transmits on c during f, (ii) u listens on c during g, and (iii) no other neighbor of u, say w, transmits on c during any frame in $overlap(g, w)$. Also, $\langle f, g\rangle$ covers (v, u) if $\langle f, g\rangle$ covers (v, u) on some channel $c \in span(v, u)$. The three conditions collectively ensure that u receives a clear message from v (on c during g) provided $\langle f, g\rangle$ is aligned. We have:

Lemma 5. *If $\langle f, g\rangle$ is aligned, then $\langle f, g\rangle$ covers (v, u) with probability at least $\frac{\rho}{8\max(2S, 3\Delta_{est})}$.*

Proof: Analogous to $A(\tau, c)$, $B(\tau, c)$ and $C(\tau, c)$ defined in Section III-A1, let $\hat{A}(f, g, c)$, $\hat{B}(f, g, c)$, $\hat{C}(f, g, c)$ denote the three events corresponding to the three conditions (i), (ii) and (iii), respectively, required for coverage. Note that the three events are *mutually independent*. As before, we compute the probability of occurrence of the three events separately.

Computing the probability of occurrence of $\hat{A}(f,g,c)$: We have:

$$\begin{aligned}\Pr\{\hat{A}(f,g,c)\} &= (v \text{ selects } c \text{ at the beginning of } f) \wedge \\ &\quad (v \text{ chooses to transmit during } f) \\ &= \frac{1}{|\mathcal{A}(v)|} \times \min\left(\frac{1}{2}, \frac{|\mathcal{A}(v)|}{3\Delta_{est}}\right) \\ &= \min\left(\frac{1}{2|\mathcal{A}(v)|}, \frac{1}{3\Delta_{est}}\right) \\ &\geq \frac{1}{\max(2S, 3\Delta_{est})} \qquad (11)\end{aligned}$$

Computing the probability of occurrence of $\hat{B}(f,g,c)$: We have:

$$\begin{aligned}\Pr\{\hat{B}(f,g,c)\} &= (u \text{ selects } c \text{ at the beginning of } g) \wedge \\ &\quad (u \text{ chooses to listen during } g) \\ &= \frac{1}{|\mathcal{A}(u)|} \times \left\{1 - \min\left(\frac{1}{2}, \frac{|\mathcal{A}(u)|}{3\Delta_{est}}\right)\right\} \\ &\quad \left\{\min(x,y) \leq x\right\} \\ &\geq \frac{1}{|\mathcal{A}(u)|} \times \left(1 - \frac{1}{2}\right) = \frac{1}{2|\mathcal{A}(u)|} \qquad (12)\end{aligned}$$

Computing the probability of occurrence of $\hat{C}(f,g,c)$: Let $\mathcal{N}(u,c)$ denote the set of neighbors of u on c. Note that, if $\mathcal{N}(u,c)$ only contains v, then $\Pr\{\hat{C}(f,g,c)\} = 1$. Otherwise, we have:

$$\begin{aligned}&\Pr\{\hat{C}(f,g,c)\} \\ &= \prod_{\substack{w \in \mathcal{N}(u,c) \\ w \neq v}} \Pr\begin{pmatrix} w \text{ does not transmit on } c \text{ during any frame} \\ \text{in } overlap(g,w) \end{pmatrix} \\ &= \prod_{\substack{w \in \mathcal{N}(u,c) \\ w \neq v}} \prod_{h \in overlap(g,w)} \Pr\begin{pmatrix} w \text{ does not transmit on } c \\ \text{during } h \end{pmatrix} \\ &= \prod_{\substack{w \in \mathcal{N}(u,c) \\ w \neq v}} \prod_{h \in overlap(g,w)} \{1 - \Pr(w \text{ transmits on } c \text{ during } h)\} \\ &\geq \prod_{\substack{w \in \mathcal{N}(u,c) \\ w \neq v}} \prod_{h \in overlap(g,w)} \left\{1 - \frac{1}{|\mathcal{A}(w)|} \times \min\left(\frac{1}{2}, \frac{|\mathcal{A}(w)|}{3\Delta_{est}}\right)\right\} \\ &= \prod_{\substack{w \in \mathcal{N}(u,c) \\ w \neq v}} \prod_{h \in overlap(g,w)} \left\{1 - \min\left(\frac{1}{2|\mathcal{A}(w)|}, \frac{1}{3\Delta_{est}}\right)\right\} \\ &\quad \left\{\min(x,y) \leq y\right\} \\ &\geq \prod_{\substack{w \in \mathcal{N}(u,c) \\ w \neq v}} \prod_{h \in overlap(g,w)} \left(1 - \frac{1}{3\Delta_{est}}\right) \\ &\quad \left\{\text{using Lemma 4, } |overlap(g,w)| \leq 3\right\} \\ &\geq \prod_{\substack{w \in \mathcal{N}(u,c) \\ w \neq v}} \left(1 - \frac{1}{3\Delta_{est}}\right)^3 \\ &= \left(1 - \frac{1}{3\Delta_{est}}\right)^{3(|\mathcal{N}(u,c)|-1)} \\ &\quad \left\{|\mathcal{N}(u,c)| - 1 = \Delta(u,c) - 1 \leq \Delta_{est}\right\} \\ &\geq \left(1 - \frac{1}{3\Delta_{est}}\right)^{3\Delta_{est}} \\ &\quad \left\{\forall x \geq 2, \left(1 - \frac{1}{x}\right)^x \text{ is a monotonically increasing} \right. \\ &\qquad \left. \text{function of } x \text{ and thus } \geq \frac{1}{4}\right\} \\ &\geq \frac{1}{4} \qquad (13)\end{aligned}$$

Finally, using a derivation similar to that for (6), it can be shown that $\langle f,g \rangle$ covers (v,u) with probability at least $\frac{\rho}{8\max(2S,3\Delta_{est})}$. □

For a frame f, we use $node(f)$ to denote the node to which f belongs. For example, as per the execution show in Fig. 2, $node(f_1) = v$ and $node(h_1) = w$. We now define a precedence relation between pairs of frames referred to as *frame-pairs* for short.

Definition 3 (precedence relation). *For frame-pairs $\langle f,g \rangle$ and $\langle p,q \rangle$, we say that $\langle f,g \rangle$* precedes *$\langle p,q \rangle$, denoted $\langle f,g \rangle \sqsubset \langle p,q \rangle$, if (i) $node(f) = node(p)$, (ii) $node(g) = node(q)$, (iii) the start time of f is before that of p, and (iv) the start time of g is before that of q.*

For example, as per the execution shown in Fig. 2, $\langle f_1,g_1 \rangle \sqsubset \langle f_2,g_2 \rangle$ and $\langle f_1,g_1 \rangle \sqsubset \langle f_3,g_2 \rangle$ but $\langle f_1,g_1 \rangle \not\sqsubset \langle f_1,g_2 \rangle$. We next define a notion on a sequence of frame-pairs that intuitively enables us to treat coverage provided by different frame-pairs as essentially independent events.

Definition 4 (admissible sequence). *A sequence σ of M frame-pairs $\left\{\langle f_{i_k}, g_{j_k} \rangle\right\}_{1 \leq k \leq M}$ is* admissible *with respect to the link (v,u) if it satisfies the following conditions:*

1) *$\forall i : 1 \leq k \leq M : node(f_{i_k}) = v$ and $node(g_{j_k}) = u$,*
2) *$\forall i : 1 \leq k < M : \langle f_{i_k}, g_{j_k} \rangle \sqsubset \langle f_{i_{k+1}}, g_{j_{k+1}} \rangle$,*
3) *$\forall i : 1 \leq k \leq M : \langle f_{i_k}, g_{j_k} \rangle$ is aligned, and*
4) *$\forall i : 1 \leq k < M : overlapAll(g_{i_k}) \cap overlapAll(g_{i_{k+1}}) = \emptyset$.*

For a sequence σ of frame-pairs that is admissible with respect to (v,u), we say that σ covers (v,u) if some frame-pair in the sequence covers (v,u).

Lemma 6. *Let σ be a sequence of $\frac{8\max(2S,3\Delta_{est})}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$ frame-pairs such that σ is admissible with respect to (v,u). Then the probability that σ does not cover (v,u) is at most $\frac{\epsilon}{N^2}$.*

Proof: For convenience, let $M = \frac{8\max(2S, 3\Delta_{est})}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$. Let E_k with $1 \le k \le M$ denote the event that $\langle f_{i_k}, g_{j_k} \rangle$ covers (v, u) on some channel. Note that the occurrence of the event E_k depends *only* on frames in $overlapAll(g_{j_k})$. Since σ is an admissible sequence, for all x and y, with $1 \le x < y \le M$, $overlapAll(g_{j_x}) \cap overlapAll(g_{j_y}) = \emptyset$. In other words, the set of frames that overlap with the frame g_{j_x} are distinct from the set of frames that intersect the frame g_{j_y}. Since, for each frame, a node chooses its action randomly, events E_ks are *mutually independent* of each other. Using a derivation similar to that of (7), it can be shown that σ does not cover (v, u) with probability at most $\frac{\epsilon}{N^2}$. □

We next show that any execution of the network must contain a "sufficiently long" sequence of admissible frame-pairs. In the rest of this section, let T_s denote the time by which all nodes have initiated the neighbor discovery algorithm, Algorithm 4. To show the existence of a "'sufficiently long" admissible sequence, we first show that, for any instant of time T after T_s and, for any pair of neighboring nodes, there is an aligned pair of frames after but "close" to T.

Lemma 7. *Consider a link from node v to node u and some instant of time T with $T \ge T_s$. Let f_i (respectively, g_i) with $i \ge 1$ denote the i^{th} full frame of node v (respectively, node u)* after T. *Then some frame in $\{f_1, f_2\}$ is aligned with some frame in $\{g_1, g_2\}$.*

Proof: Note that the frames f_1 and f_2 together contain six slots. Let a_i for $i = 1, 2, \ldots, 6$ denote the start times of the six slots numbered in the increasing order of their start times (see Fig. 3). Likewise, let b_i for $i = 1, 2, \ldots, 6$ denote the start times of the six slots of the frames g_1 and g_2.

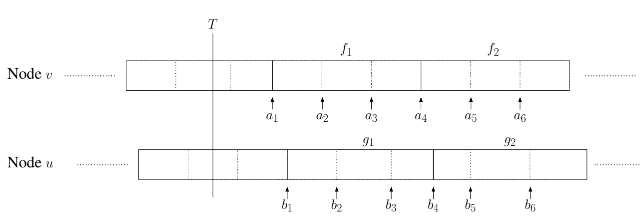

Figure 3. Slots used in the proof of Lemma 7.

Claim 1: We first show that $b_1 \le a_5$. Assume, on the contrary, that $a_5 < b_1$. Note that, by definition of g_1, there is only a partial frame of u between T and b_1. This, in turn, implies that a partial frame of u contains at least four slots of v. This can happen only if the following condition holds:

$$\frac{L}{1-\delta} > \frac{4L}{3(1+\delta)} \quad \Longrightarrow \quad \delta > \frac{1}{7}$$

This contradicts Assumption 1.

Claim 2: Likewise, we can show that $a_1 \le b_5$.

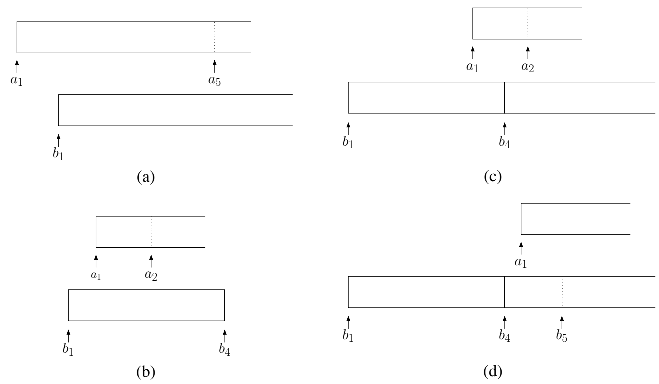

Figure 4. Various cases in the proof of Lemma 7.

Claim 3: If $a_{i-1} \le b_1 \le a_i$ for some i with $1 < i \le 5$, then $b_1 \le a_i < a_{i+1} \le b_4$. It suffices to show that $a_{i+1} \le b_4$. Assume, on the contrary, that $b_4 < a_{i+1}$. This implies that $a_{i-1} \le b_1 < b_4 < a_{i+1}$. In other words, two adjacent slots of v strictly contain an entire frame of u. This can happen only if the following condition holds:

$$\frac{2L}{3(1-\delta)} > \frac{L}{1+\delta} \quad \Longrightarrow \quad \delta > \frac{1}{5}$$

This contradicts Assumption 1.

We now prove the lemma statement. We consider two cases depending on whether f_1 or g_1 has earlier start time.

- *Case 1 ($a_1 < b_1$):* In this case, we have $a_1 < b_1 \le a_5$ (see Fig. 4(a)). The rightmost inequality follows from Claim 1. The time interval from a_1 to a_5 consists of four contiguous slots of v: $[a_1, a_2]$, $[a_2, a_3]$, $[a_3, a_4]$ and $[a_4, a_5]$. Clearly, b_1 lies in at least one of them. (If b_1 lies on the boundary of two slots, we select the earlier one.) In any case, using Claim 3, we can show that at least one of the slots $[a_2, a_3]$, $[a_3, a_4]$, $[a_4, a_5]$ or $[a_5, a_6]$ is contained in the frame g_2. This implies that either $\langle f_1, g_1 \rangle$ or $\langle f_2, g_1 \rangle$ is aligned.
- *Case 2 ($b_1 \le a_1$):* If $a_2 \le b_4$, then $b_1 \le a_1 < a_2 \le b_4$ (see Fig. 4(b)). In other words, the first slot of the frame f_1 is contained with the frame g_1, which implies that $\langle f_1, g_1 \rangle$ is aligned. Therefore assume that $b_4 < a_2$. We have two subcases depending on where a_1 lies relative to b_4. Let b_7 denote the end time of the frame g_2.
  - *Case 2.1 ($a_1 \le b_4$):* In this case, we show that the slot $[a_2, a_3]$ is contained within the frame $[b_4, b_7]$ (see Fig. 4(c)). To that end, we first prove that $a_3 < b_7$. If not, then $a_1 \le b_4 < b_7 \le a_3$. In other words, two adjacent slots of v strictly contain an entire frame of u. This can happen only if the following condition holds:

$$\frac{2L}{3(1-\delta)} > \frac{L}{1+\delta} \quad \Longrightarrow \quad \delta > \frac{1}{5}$$

This contradicts Assumption 1. Therefore, we have $b_4 < a_2 < a_3 < b_7$, which implies that the pair $\langle f_1, g_2 \rangle$ is aligned.

– *Case 2.2 ($b_4 < a_1$):* In this case, we show that the slot $[a_1, a_2]$ is contained within the frame $[b_4, b_7]$ (see Fig. 4(d)). To that end, we show that $a_2 \leq b_7$. If not, then $a_1 \leq b_5 < b_7 < a_2$. (The inequality $a_1 \leq b_5$ follows from Claim 2.) In other words, one slot of v strictly contains at least two adjacent slots of u. This can happen only if the following condition holds:

$$\frac{L}{3(1-\delta)} > \frac{2L}{3(1+\delta)} \quad \Longrightarrow \quad \delta > \frac{1}{3}$$

This contradicts Assumption 1. Therefore, we have $b_4 < a_1 < a_2 \leq b_7$, which implies that the pair $\langle f_1, g_2 \rangle$ is aligned.

In all cases, we show that one of the four pairs $\langle f_1, g_1 \rangle$, $\langle f_1, g_2 \rangle$, $\langle f_2, g_1 \rangle$ or $\langle f_2, g_2 \rangle$ is aligned. □

We now show that existence of a "sufficiently long" admissible sequence in any execution of Algorithm 4.

Lemma 8. *Consider a link from node v to node u. Further, consider an execution of the network after T_s that contains at least M full frames of u as well as v. Then the execution contains a sequence of at least $\frac{M}{6}$ frame-pairs such that the sequence is admissible with respect to (v, u).*

Proof: The proof is by construction. The construction is in two steps. In the first step, we construct a sequence of frame-pairs γ that is "almost" admissible in the sense that it satisfies the first three properties of an admissible sequence but may not satisfy the fourth property. We show that γ contains at least $\frac{M}{2}$ frame-pairs. In the second step, using γ, we construct a sequence of frame-pairs σ that satisfies all four properties of an admissible sequence. We also show that σ contains at least $\frac{M}{6}$ frame-pairs.

Constructing γ: To obtain the first frame-pair $\langle f_{i_1}, g_{i_1} \rangle$ that is aligned, we apply Lemma 7 to T_s. Now, assume that we have already selected k frame-pairs satisfying the first three properties of an admissible sequence. Let the k^{th} frame-pair be denoted by $\langle f_{i_k}, g_{i_k} \rangle$. To select the next frame-pair, let T_k be defined as the *earlier* of the end times of frames f_{i_k} and g_{i_k}. To obtain the next frame-pair that is aligned, we apply Lemma 7 to T_k. Let the frame-pair be denoted by $\langle f_{i_{k+1}}, g_{i_{k+1}} \rangle$. Clearly, the extended sequence (consisting of $k + 1$ aligned frame-pairs) satisfies the first and third properties of an admissible sequence. The second property holds because the start times of both f_{i_k} and g_{i_k} are *before* T_k, whereas the start times of both $f_{i_{k+1}}$ and $g_{i_{k+1}}$ are *after* T_k. We repeatedly select aligned frame-pairs using Lemma 7 until we run out of frames of either u or v. We now establish a lower bound on the length of γ. Note that, when selecting the $(k+1)^{st}$ pair, the first full frame of node v after T_k (namely frame f_1 in the statement of Lemma 7) is *adjacent* to the frame f_{i_k}. This follows from the definition of T_k. This, in turn, implies that the frame $f_{i_{k+1}}$ obtained using Lemma 7 is within a distance of two of the frame f_{i_k}. Likewise, the frame $g_{i_{k+1}}$ is within a distance of two of the frame g_{i_k}. As a result, γ contains at least $\frac{M}{2}$ frame-pairs.

Constructing σ: To construct a sequence σ that also satisfies the fourth property of an admissible sequence, we choose every *third* frame-pair of γ starting with the first frame-pair $\langle f_{i_1}, g_{i_1} \rangle$. Clearly, σ also satisfies the first three properties of an admissible sequence (since it is a subsequence of γ). Let the k^{th} frame-pair of σ be denoted by $\langle f_{j_k}, g_{j_k} \rangle$. To prove that σ satisfies the fourth property as well, assume, on the contrary, that some frame, say h, overlaps with two consecutive frames g_{j_k} and $g_{j_{k+1}}$ for some k. Note that, since we selected only every third frame-pair of γ to construct σ, there are at least two other frames of u between g_{j_k} and $g_{j_{k+1}}$. This implies that h overlaps with at least four frames of u, which contradicts Lemma 4. This establishes that the length of σ is at least $\frac{M}{6}$. □

Finally, we have the main result.

Theorem 9. *Let T_s be the time by which all nodes have initiated neighbor discovery. Also, let T_f be the* ear-liest *time by which each node has executed at least $\frac{48 \max(2S, 3\Delta_{est})}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$ full frames since T_s. Then Algorithm 4 ensures that each node discovers all its neighbors on all channels with probability at least $1 - \epsilon$ by time T_f.*

Proof: Consider a link (v, u). From Lemma 8, the execution from T_s to T_f contains a sequence of at least $\frac{8 \max(2S, 3\Delta_{est})}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$ frame-pairs such that the sequence is admissible with respect to (v, u). From Lemma 6, the probability that (v, u) is not covered by T_f is at most $\frac{\epsilon}{N^2}$. This implies that the probability that some link in the network is not covered by T_f is at most ϵ. □

We now bound the length of the interval $T_f - T_s$.

Theorem 10. *Let T_s and T_f be as defined in Theorem 9. Then the length of the interval $T_f - T_s$ is upper bounded by $\left\{\frac{48 \max(2S, 3\Delta_{est})}{\rho} \ln\left(\frac{N^2}{\epsilon}\right) + 1\right\} \left(\frac{L}{1-\delta}\right)$.*

Proof: By our choice of T_f, there exists some node such that $T_f - T_s$ contains exactly $\frac{48 \max(2S, 3\Delta_{est})}{\rho} \ln\left(\frac{N^2}{\epsilon}\right)$ full frames of that node. The first frame of that node in the execution may be partial frame. From (10), the length of each frame is upper bounded by $\frac{L}{1-\delta}$. Combining the three facts, we obtain the result. □

V. Conclusions

In this paper, we have presented several randomized algorithms for neighbor discovery in an M^{2}HeW network for both a synchronous system and an asynchronous system. Our algorithms and/or its analysis can be easily extended for cases when (a) the communication graph is asymmetric, (b) channels are not reliable, and/or (c) channels have diverse propagation characteristics. Due to space limitations, details have been omitted and can be found elsewhere [23].

References

[1] M. J. McGlynn and S. A. Borbash, "Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks," in *Proc. 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2001, pp. 137–145.

[2] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor Discovery in Wireless Networks and the Coupon Collector's Problem," in *Proc. 15th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2009, pp. 181–192.

[3] L. Bao and J. J. Garcia-Luna-Aceves, "Transmission Scheduling in Ad Hoc Networks with Directional Antennas," in *Proc. 8th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2002, pp. 48–58.

[4] R. Choudhury, X. Yang, R. Ramanathan, and N. Vaidya, "Using Directional Antennas for Medium Access Control in Ad Hoc Networks," in *Proc. 8th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, Sep. 2002, pp. 59–70.

[5] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, Sep. 1997.

[6] W. Heinzelman, A. Chandrasekaran, and H. Balakrishnan, "Energy Efficient Communication Protocol for Wireless Microsensor Networks," in *Proc. Hawaii International Conference on Systems Sciences*, Jan. 2000, pp. 3005–3014.

[7] S. Gandham, M. Dawande, and R. Prakash, "Link Scheduling in Wireless Sensor Networks: Distributed Edge-Coloring Revisited," *Journal of Parallel and Distributed Computing (JPDC)*, vol. 68, no. 8, pp. 1122–1134, Aug. 2008.

[8] S. Gandham, Y. Zhang, and Q. Huang, "Distributed Time-Optimal Scheduling for Convergecast in Wireless Sensor Networks," *Computer Networks (COMNET)*, vol. 52, pp. 610–629, 2008.

[9] R. Ramanathan and R. Hain, "Topology Control of Multihop Radio Networks using Transmit Power Adjustment," in *Proc. 19th IEEE Conference on Computer Communications (INFOCOM)*, Mar. 2000, pp. 404–413.

[10] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "A Cone-Based Distributed Topology-Control Algorithm for Wireless Multi-Hop Networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 147–159, 2005.

[11] D. Cabric, S. M. Mishra, D. Willkommen, R. W. Brodersen, and A. Wolisz, "A Cognitive Radio Approach for Usage of Virtual Unlicensed Spectrum," in *Proc. 14th IST Mobile Wireless Communications Summit*, Jun. 2005.

[12] R. Zheng, J. Hou, and L. Sha, "Asynchronous Wakeup for Ad Hoc Networks," in *Proc. 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Jun. 2003, pp. 35–45.

[13] S. Gallo, L. Galluccio, G. Morabito, and S. Palazzo, "Rapid and Energy Efficient Neighbor Discovery for Spontaneous Networks," in *Proc. ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, Oct. 2004, pp. 8–11.

[14] V. Dyo and C. Mascolo, "Efficient Node Discovery in Mobile Wireless Sensor Networks," in *Proc. IEEE international Conference on Distributed Computing in Sensor Systems (DCOSS)*, Jun. 2008, pp. 478–485.

[15] P. Dutta and D. Culler, "Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications," in *Proc. ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Nov. 2008, pp. 71–84.

[16] E. B. Hamida, G. Chelius, A. Busson, and E. Fleury, "Neighbor Discovery in Multi-Hop Wireless Networks: Evaluation and Dimensioning with Interferences Considerations," *Discrete Mathematics and Theoretical Computer Science*, vol. 10, no. 2, pp. 87–114, 2008.

[17] D. Yang, J. Shin, J. Kim, and C. Kim, "Asynchronous Probing Scheme for the Optimal Energy-Efficient Neighbor Discovery in Opportunistic Networking," in *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Mar. 2009, pp. 1–4.

[18] G. Alonso, E. Kranakis, C. Sawchuk, R. Wattenhofer, and P. Widmayer, "Probabilistic Protocols for Node Discovery in Ad-Hoc Multi-Channel Broadcast Networks," in *Proc. 2nd Annual Conference on Adhoc Networks and Wireless (ADHOCNOW)*, 2003, pp. 101–115.

[19] A. Raniwala and T. c. Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network," in *Proc. 24th IEEE Conference on Computer Communications (INFOCOM)*, Mar. 2005, pp. 2223–2234.

[20] S. Krishnamurthy, M. Thoppian, S. Kuppa, R. Chandrasekaran, N. Mittal, S. Venkatesan, and R. Prakash, "Time-efficient Distributed Layer-2 Auto-configuration for Cognitive Radio Networks," *Computer Networks (COMNET)*, vol. 52, no. 4, pp. 831–849, Mar. 2008.

[21] N. Mittal, S. Krishnamurthy, R. Chandrasekaran, S. Venkatesan, and Y. Zeng, "On Neighbor Discovery in Multi-Channel Cognitive Radio Networks," *Journal of Parallel and Distributed Computing (JPDC)*, vol. 69, no. 7, pp. 623–637, Jul. 2009.

[22] Y. Zeng, N. Mittal, S. Venkatesan, and R. Chandrasekaran, "Fast Neighbor Discovery with Lightweight Termination Detection in Heterogeneous Cognitive Radio Networks," in *Proc. 9th International Symposium on Parallel and Distributed Computing (ISPDC)*, Jul. 2010, pp. 149–156.

[23] Y. Zeng, "Distributed Algorithms for Communication Problems in Heterogeneous Cognitive Radio Networks," Ph.D. dissertation, Department of Computer Science, The University of Texas at Dallas, Richardson, Texas, U.S.A., 2011.

[24] K. Nakano and S. Olariu, "Randomized Leader Election Protocols in Radio Networks with no Collision Detection," in *Proc. Annual International Symposium on Algorithms and Computation (ISAAC)*, Dec. 2000, pp. 362–373.