

# Cooperation in Neighbor Discovery

Niels Karowski  
TKN TU-Berlin  
niels.karowski@tu-berlin.de

Andreas Willig  
CSSE University of Canterbury  
andreas.willig@canterbury.ac.nz

Adam Wolisz  
TKN TU-Berlin  
wolisz@tkn.tu-berlin.de

**Abstract**—Neighbor discovery is a fundamental task in wireless networks in order to find potential communication partners, discover services, and to efficiently access the wireless resource. In this paper we focus on the passive neighbor discovery in multi-channel environments. We present different discovery approaches with low computational complexity that utilize cooperation between devices by exchanging discovery results in order to improve the discovery process. Furthermore, we present an optimized discovery approach that uses multiple transceivers and evaluate the bounds on the performance under idealized conditions. We analyze the impact of different network parameters such as number of channels and number of neighbors on the discovery times of the developed approaches by using a simulative evaluation. By putting the discovery times achievable with the cooperative or “gossiping” approaches into perspective with the multi-transceiver results, we can quantify the value of cooperation in terms of numbers of transceivers.

**Index Terms**—Discovery, multi-channel, cooperation, gossip.

## I. INTRODUCTION

Nowadays humans are surrounded by an abundance of different wireless networks, leading to situations where *many* networks of the same or different wireless technologies are available in the same spot. Switching on a WiFi transceiver in a densely populated city often reveals dozens of access points in close vicinity, people wearing wireless body sensor networks can observe many other body sensor networks when many people flock together, cars equipped with Wireless Access in Vehicular Environment (WAVE) transceivers exchange information with nearby cars or roadside infrastructure in intelligent transportation systems (ITS) and so on. In many scenarios it is useful or even required for a device to *discover* other devices of the same wireless technology, be it to exchange information (e.g. positions and headings in vehicular networks), to use services offered by other devices (e.g. Internet access), or to negotiate on how to share physical resources like spectrum or time to achieve greater efficiency and reduce resource competition.

Neighbor discovery can be done in two fundamentally different ways: in *active* discovery the searching node actively sends packets, to which some other device belonging to an external network has to respond to complete discovery. In contrast, in *passive* discovery a searching node does not transmit extra packets, but instead just listens for “external” transmissions. Hence, the passive approach consumes fewer channel resources, but discovery will generally take longer than in the active case and the discovery time will depend on the activity levels of an external network.

Passive discovery is greatly helped by a characteristic shared by many modern wireless technologies: the periodic transmission of beacon packets. For example, the IEEE 802.15.4 wireless personal area network standard offers periodic beacon transmissions in its beamed mode (as does IEEE 802.15.6), many WiFi access points are configured to transmit beacons periodically and so on. At the same time, however, discovery is made more complicated by the fact that very often these technologies can actually pick one of several distinct frequency channels to operate on, and to discover a neighbor requires to receive a beacon (or any other packet) on the right frequency channel. In the absence of any additional information a searching node has to scan *all* available channels according to some search strategy.

In previous work [1], [2], [3], [4] we have considered sub-optimal and optimal search strategies for the problem of discovering *all* neighbors and found that the number of frequency channels and the set of allowable beacon periods (assumed to be always integer multiples of some base period) have a substantial impact on the discovery time. We have formulated optimization problems and identified optimal search strategies for some selected classes of allowable beacon sets. A key assumption in previous work has always been that the searching node only has a single transceiver and works in isolation, i.e. it does not collaborate with external devices or networks in the search process.

In this paper we extend this previous work in two directions. First, we consider the effect which cooperation has on discovery times when all involved nodes only have a single transceiver. By cooperation we mean that each node includes the devices it has discovered so far into its beacons (indicating their frequency channel, beacon period and “phase shift” or timing offset with respect to the node sending the beacon) so that a searching node receiving such a beacon can learn about the included devices at once. Secondly, we consider the effects of using multiple transceivers in the discovery process and evaluate the achievable speedups under idealized conditions, giving bounds on the performance that can be expected under more realistic assumptions. By putting the discovery times achievable with this cooperative or “gossiping” approach into perspective with the multi-transceiver results, we can quantify the value of cooperation in terms of numbers of transceivers. Our results (obtained in an idealized setting and under a number of simplifying assumptions) suggest that increasing the number of transceivers for listening indeed can reduce the discovery times substantially, with marginal additional

returns for larger numbers of transceivers. Furthermore, for varying numbers of channels the gossiping strategies achieve a performance that is comparable to having between two and four transceivers for listening. When increasing the number of neighboring devices while keeping the number of channels fixed to 8, gossiping becomes even more beneficial and approaches the optimal performance of having eight transceivers.

The remaining paper is structured as follows: In the following Section II we discuss related work. In Section III our system model is described and Section IV presents our developed discovery strategies that are evaluated in Section V. Finally, Section VI concludes this paper.

## II. RELATED WORK

The approaches presented in the following split time into time slots and describe schedules in which certain time slots are marked as active slots. A slot is called active when a neighbored device transmits a beacon (or any other packet) in it or when the searching node decides to listen in it. Discovery occurs when an active slot of the searching node overlaps with an active slot of at least one neighbor on the same channel.

First we present neighbor discovery approaches in which devices perform the discovery without any cooperation. These non-cooperative neighbor discovery strategies can be classified into deterministic and probabilistic approaches with respect to the selection of active slots. Deterministic approaches work either with quorums, prime numbers or static/dynamic slot schemes. In quorum approaches discovery is based on the intersection of schedules generated either by a grid or a cyclic pattern [5], [6]. In the former case time slots are arranged in a square matrix from which each device picks one column and one row to serve as its active slots. In the latter case discovery is achieved by constructing schedules based on cyclic difference sets which have guaranteed overlaps. With approaches based on prime numbers, such as DISCO [7] and U-Connect [8], devices are active during time slots that are multiples of each device's selected prime numbers. Discovery is then ensured by the Chinese Remainder Theorem. Most recent work has been published in the area of static/dynamic slot schemes. Discovery schedules generated by, e.g., Searchlight [9], Hello [10] and BlindDate [11] consist of multiple cycles, where in each cycle there is one active slot at a fixed position and at least one dynamic slot whose position is shifted each cycle. Probabilistic approaches have in common that devices select their operational state for any time slot out of at least two states, *transmit* and *listen*, with a predefined probability [12], [13], [14]. In [12], devices may also select with some probability an additional state *sleep*.

The approaches presented in the following utilize cooperation or coordination between devices to improve the discovery results. First, we present approaches in which devices cooperate by exchanging discovery results. An approach named the *rumor-based scheme* for IEEE 802.15.4 is presented in [15], in which a mobile device is searching for a particular fixed network. In order to speed up the discovery, devices exchange knowledge about the frequency channels on which fixed devices can be found and their beacon period, but not

about their time offset. If the target device is among the rumored devices, the scan is focused on the corresponding channel. In [16] devices form groups after performing a pairwise discovery. If a new node is discovered by a group member, it will schedule an additional active slot when the new node will be active the next time in order to transmit its known neighbors. The set of reported neighbors is filtered by a spatial selection using an estimation of the closeness of nodes based on communication range and node density as well as a temporal selection based on the communication range and maximum node velocity. The approaches presented in [17], [18] assume that devices possess an omnidirectional as well as a directional antenna. Furthermore, devices know their location. Neighbors discovered via the omnidirectional antenna are propagated using the directional antenna [18] and vice versa [17]. In these works no quantitative comparison with multi-transceiver strategies is carried out, and we investigate a wider range of cooperative strategies.

The following approaches use additional signaling between the devices performing the discovery or with an external source to support the discovery process. In [19] nodes form groups and coordinate their listening and beacon transmission schedules in order to achieve high propagation speed of group membership information as well as to decrease discovery times for new nodes finding and joining the group. A similar approach is described in [20] that focuses on continuous neighbor discovery rather than on the initial discovery which is used for building segments of devices. If a segment member discovers a new device, it issues a wake-up message that is distributed to all other members via the already known links of the segment, allowing to discover all hidden links inside the segment. Devices in [21] exploit existent infrastructure Access Points (AP's) as an external source of synchronization to coordinate energy-efficient scanning to discover each other and form or join existing clusters. An approach named CQuest [22] assumes that devices have a Low Power (LP) and High Power (HP) radio. Devices within the communication range of the LP radio agree on a subset of devices by using a contention-based approach that defines which devices will perform the energy-expensive scan using the HP radio. Neighbors discovered via the HP radio are included in the hello messages transmitted with the LP radio. The cooperative approach presented in [23] utilizes an infrastructure-supported neighbor discovery in the 2.4GHz range with the help of an AP to schedule data transmissions using a mmWave radio. In contrast to these works, the approaches considered in this paper do not require extra signaling messages, they only require the ability to include additional information into the beacon messages.

## III. SYSTEM MODEL

We consider devices interested in exploring their network environment in order to discover *neighbors*, that is, other devices or networks within the communication range operating on a shared set of wireless channels. The goals include looking for potential communication partners, devices providing specific services, or avoiding conflicting resource allocations.

All devices use the same wireless technology, are static and are placed in the same location, so that we can disregard the effects of mobility, channel errors, hidden-terminal scenarios and so on, and to make the task for one device to discover *all* other devices meaningful.

We assume that each device announces its presence by periodically broadcasting beacon signaling messages at certain individual **Beacon Intervals (BI's)**. If a neighbor is a network, beacons might be transmitted by a selected device acting as network coordinator. The discovery is performed by passively listening to the transmitted beacon messages. The only a-priori knowledge shared by all devices is the set of possible operating channels and the set of BI's that might be used by the neighbors.

We denote the shared set of wireless channels by  $C$ , and the set of allowable BI's by  $B \subset \mathbb{N}^+$ . We denote the set of devices by  $D$  and the set of transceivers involved in the discovery by each device by  $R$ . The usage of multiple transceivers by a device can either be achieved by using multiple radios (which we assume for the purposes of this paper) or by having a coordinator splitting up the discovery process and tasking multiple slave devices. We furthermore assume that all considered devices have the same number of transceivers. The operating channel of device  $\nu \in D$  is denoted by  $c_\nu \in C$  and the interval at which it sends its beacons is given by  $b_\nu \in B$ . The set of BI's  $B$  might be determined by the choice of communication technology or a set of common policies.

We assume that time is divided into time slots and that the maximum beacon transmission time (time required to send one beacon) is much smaller than one time slot. All devices magically agree on time slot boundaries. We denote the offset, that is, the first time slot, in which device  $\nu$  sends a beacon, by  $\delta_\nu \leq b_\nu$ . We denote the set of beaconing time slots of a device  $\nu$  by  $\mathcal{T}_\nu = \{\delta_\nu + i \cdot b_\nu : i \in \mathbb{N}_0\}$ . We call the resulting set of (channel, time slot) pairs,  $\mathcal{B}_\nu = \{c_\nu\} \times \mathcal{T}_\nu$ , the beacon schedule of device  $\nu$ .

With the given notation, each device  $\nu$  is assigned a triple  $(c_\nu, b_\nu, \delta_\nu)$ , which we call a **device configuration**. Each device includes a device identification and its own device configuration into its beacons, so that for a neighbored device it suffices to receive just one beacon to know the configuration. We denote the set of possible device configurations for a given BI set  $B$  and a given set of channels  $C$  by  $K_{BC} = \{(c, b, \delta) : c \in C, b \in B, \delta \in \{1, \dots, b\}\}$ . For a time slot  $t$  and a BI  $b$ ,  $\delta_b(t) = (t \bmod b) + 1$  shall denote the unique offset such that a neighbor with configuration  $(c, b, \delta_b(t))$ ,  $c \in C$ , transmits its beacon in time slot  $t$ . Observe that  $\delta_b(t)$  has periodicity  $b$ , that is,  $\delta_b(t) = \delta_b(t + b)$  for each  $t, b$ .

Additionally, we assume that each configuration is assigned to at most one device. This assumption, together with the assumption that beacon packets transmitted in the same slot do not collide with each other, is designed to keep the problem deterministic, as otherwise repeated collisions between beacons in the same slot / frequency would lead to randomized discovery times. We denote by  $K(c, t) = \{(c, b, \delta_b(t)) \in K_{BC} : b \in B\}$  the set of all configurations

from  $K_{BC}$  transmitting a beacon in slot  $t$  on channel  $c$ .

The devices perform the discovery by selecting time slots during which each of their transceivers  $r \in R$  listen on particular channels in order to overhear beacons transmitted by neighbors. If we fix a particular transceiver  $r \in R$ , then this transceiver follows a particular **listening schedule**  $\mathcal{L}$ , which specifies a contiguous set of time slots in which to listen, and for each such time slot indicates the channel on which to listen, i.e.  $\mathcal{L} = \{(c, t) : c \in C, t \in \{1, \dots, |\mathcal{L}|\}\}$ . Since we assume that a transceiver cannot simultaneously listen on multiple channels, we demand  $t = t' \implies c = c'$  for two pairs  $(c, t) \in \mathcal{L}$  and  $(c', t') \in \mathcal{L}$ . Each transceiver of a device will have its own listening schedule. We furthermore assume that the time a transceiver requires to switch channels is negligible.

If no beacons are lost and the set of neighbors does not change during the discovery, and if a schedule  $\mathcal{L}$  contains at least one element from the beacon schedule  $\mathcal{B}_\kappa$  for each configuration  $\kappa \in K_{BC}$ , one can discover all neighbors operating with any BI  $b \in B$  on a channel  $c \in C$ . Additionally, we denote the set of previously scanned (channel, time slot) pairs of a listening schedule  $\mathcal{L}$  by  $\mathcal{L}_p$ .

Devices may include information obtained during discovery in their beacon messages in order to support the discovery process of nearby devices. The information included about discovered neighbors (which we call gossip information) consists of their identifier, operating channel, BI, and time of contact, relative to the transmission of the beacon transporting the information. Using this information devices are able to determine the corresponding device configuration  $(c_\nu, b_\nu, \delta_\nu)$  which may be used for modifying the listening schedules. We denote the set of (channel, time slot) pairs resulting from the algorithm processing the gossip information as  $\mathcal{L}_g$ .

In the following results are obtained under the idealizing assumption that there are no beacon losses due to collisions or interference, no deaf periods when switching between channels (switching time is 0), and that beacons which would cross time slot borders are received correctly (beacon transmission/reception time is 0). There is no jitter in beacon transmissions. Furthermore, we assume that devices are synchronized to time slot boundaries.

#### IV. COOPERATION IN NEIGHBOR DISCOVERY

A main goal of this paper is to investigate cooperative discovery strategies (i.e. strategies in which devices include their current knowledge about existing neighbors and their configuration into their beacons to help each other in the discovery process) as a means of “poor-man’s-parallelization” of the discovery process. Some benefits of cooperation are resource sharing among devices (e.g. energy required to listen for beacon transmissions) or to speed up the discovery process.

To investigate the effectiveness of cooperative discovery, we compare the cooperative strategy not only against optimal single-transceiver strategies, but also against (optimal) strategies in which a discovering device possesses multiple transceivers and uses them in a coordinated fashion. In the next Section IV-A we present a range of cooperative or “gossip-based” approaches utilizing the exchange of discovery results

and in Section IV-B we describe a discovery approach using multiple transceivers.

#### A. Gossip-based Discovery Approaches

In the following we present a number of cooperative or gossip-based approaches. All devices only use single radio transceivers and can only listen on one channel at a time. However, each device includes all the information it has collected so far (device/network identifiers and their configuration) into its own beacons, where we assume that beacons can be arbitrarily large (note that in Section V-D we also consider the case of bounded beacon sizes).

The starting point for the design of our gossip-based algorithms is the GREEDY DTR-SWT strategy presented in [4], which we briefly summarize and which henceforth we simply refer to as the GREEDY strategy. We represent by  $\mathcal{L}_P(t)$  the previous listening schedule (pairs of (channel, time slot)) followed in the slots before time slot  $t$ . By our worst-case assumption that all possible configurations from  $K_{BC}$  are actually used by neighboring devices,  $\mathcal{L}_P(t)$  then also determines the set of configurations  $\mathcal{D}(t)$  we have already discovered before time  $t$ . In time slot  $t$  we pick the channel  $c$  on which to listen as:

$$c = \arg \max_{c' \in C} |K(c', t) \setminus \mathcal{D}(t)| \quad (1)$$

i.e. it picks the channel on which the largest number of yet-undiscovered configurations can be discovered in a greedy fashion. This selection rule is then repeatedly applied from slot to slot until all configurations have been discovered.

To properly explain our gossiping strategies, we need a few more notions. We refer to configurations which the searching device has discovered by receiving a beacon sent by the owning device as a **direct discovery**, whereas a configuration only learned from a neighbor's beacon and not directly observed before is called an **indirect discovery**. For an indirectly discovered configuration there are two choices for setting its discovery time: in the case of **immediate accounting (IMM)** the discovery time is set to be the time where the foreign beacon listing the configuration in its payload has been received (i.e. where the searching device has received gossip information). In the case of **verified accounting (VER)** the discovery time is set to the (later) time where a direct beacon is received from the indirectly discovered configuration. The discovery time of a directly discovered configuration is clearly the time slot in which the discovery has been made.

In our first gossiping strategy, termed GSP-UNMOD-IMM, indirectly discovered configurations are immediately accounted for, but they are not added to the set  $\mathcal{D}(t)$  of already known configurations that is used in the underlying GREEDY algorithm to calculate the next channel to listen on. In other words: indirectly discovered configurations do not lead to a modification of the listening schedule.

In all the remaining strategies the indirectly discovered configurations are added to  $\mathcal{D}(t)$  and hence can modify the listening schedule (and hopefully shorten it). To achieve this, the new information is included into the current knowledge  $\mathcal{D}(t)$  while using the GREEDY channel selection rule (Equation 1) as before. In the GSP-MOD-IMM strategy, indirectly

discovered configurations are immediately accounted for. The GSP-MOD-IMM-REV strategy builds on the GSP-MOD-IMM strategy (in particular, indirect discoveries are immediately accounted for), but adds a “re-visiting” feature: by virtue of including indirectly discovered configurations into the set  $\mathcal{D}(t)$  it can well happen that the schedule calculated after the discovery will not lead to any direct beacon reception from an indirectly discovered device, and we may miss out on any information that this device might have for us. Therefore, in addition to the (modified) GREEDY schedule calculated in response to indirect discoveries, an additional schedule is created in which extra slots are scheduled to receive a beacon once from indirectly discovered devices. Indirectly discovered devices with larger BI are prioritized in the creation of the schedule. This additional schedule “overrides” the modified GREEDY schedule, i.e. when both schedules prescribe some channel for a given time slot  $t$ , the channel coming from the additional schedule is chosen. Finally, the GSP-MOD-VER is similar to the GSP-MOD-IMM-REV strategy, but applies verified accounting instead of immediate accounting. This may be appropriate when indirectly discovered configurations are not immediately trusted – for example, in a multi-hop network a neighbor might report a configuration hidden from the discovering device.

In scenarios in which the number of gossip entries in beacon messages is limited and not sufficient to include all discovery results, the strategies have to make a choice. Selecting the entries just based on how recently they have been discovered (i.e. the point in time when the last beacon was received) favors devices with a small BI which can generally be detected quickly even without the usage of gossiping. Therefore, we give preference to configurations with a smaller *last contact time in multiples of their BI*. With this we aim to avoid penalizing neighbors operating with larger BI, e.g. a neighbor operating with BI 10 and a relative last contact time of -16 time slots will be prioritized over an entry with BI 5 and a relative last contact time of -10 time slots.

Table I gives an overview of the discovery strategies.

#### B. Multi-Transceiver Optimized (MT-OPT) Strategy

In the MT-OPT approach we assume that a single device with  $r$  separately controllable transceivers sets out to discover its neighborhood (instead of a single device with multiple radios one could equivalently consider distributing the discovery task within one network to multiple closely coordinated member nodes). The searching node does not make use of any information found in the beacons of other devices beyond the configuration and identification of that device. Broadly, the MT-OPT strategy is given by an optimal solution to an ILP that is an extension of the GENOPT strategy presented in [4] supporting multiple transceivers. The result of the optimization are listening schedules  $\mathcal{L}_r$  for each transceiver  $r \in R$ , so that the Mean Discovery Time (MDT) (see Section V-B) for all configurations  $(c, b, \delta)$  (with  $c \in C$ ,  $b \in B$  and  $\delta \in \{1, \dots, b\}$ ) is being minimized. We refer to the MT-OPT strategy with  $R$  transceivers as MT-OPT- $R$ .

TABLE I: Overview of the discovery strategies and properties

Strategy	Description
GREEDY [4]	Low complex discovery strategies that greedily selects the channel with the maximum number of discoverable configurations in each time slot.
MT-OPT-R	Optimized discovery strategy obtained by solving an Integer Linear Program (ILP) that utilizes up to $R$ transceivers in parallel.
GSP	In gossip ( <b>GSP</b> ) based discovery strategies devices include information about their discoveries in broadcasted beacon messages.
<b>GSP Property</b>	
MOD / UNMOD	When a device receives new gossip information, the discovery strategy can either <b>MOD</b> ify the listening schedule or stick to the original schedule keeping it <b>UNMOD</b> ified.
IMM / VER	Specifies the discovery time in case of indirect discovery via gossiping. It can either be <b>IMM</b> ediately accounted or accounted later to a point in time when a beacon is received directly from the device previously being gossiped and therefore <b>VER</b> ified.
REV	Discovery strategies may alter the listening schedule to <b>RE</b> visit already discovered neighbors in order to gather additional gossip information.

First, we define the decision variables  $x_{c,t,b,r}$  and the auxiliary variables  $h_{c,t,r}$  as follows:

$$x_{c,t,b,r} = \begin{cases} 1, & \text{if configuration } (c, b, \delta_b(t)) \text{ is detected by} \\ & \text{transceiver } r \text{ during scan of channel } c \text{ in} \\ & \text{time slot } t \\ 0, & \text{otherwise} \end{cases}$$

$$h_{c,t,r} = \begin{cases} 1, & \text{if scan is performed by transceiver } r \text{ on} \\ & \text{channel } c \text{ at time slot } t \\ 0, & \text{otherwise} \end{cases}$$

In the following  $LCM(X)$  denotes the least common multiple of a set  $X \subset \mathbb{N}^+$ . Then MT-OPT can be formulated as follows:

$$\begin{aligned} \min \quad & \frac{1}{|C||B|} \sum_{c \in C} \sum_{b \in B} \sum_{t=1}^{LCM(B)|C|} \sum_{r \in R} x_{c,t,b,r} \frac{t}{b} \\ \text{s.t.} \quad & \sum_{m=0}^{\frac{LCM(B)|C|}{b}-1} \sum_{r \in R} x_{c,mb+\delta,b,r} = 1 \\ & \text{for all } c \in C, b \in B, \delta \in \{1, \dots, b\} \end{aligned} \quad (C1)$$

$$\begin{aligned} x_{c,t,b,r} &\leq h_{c,t,r} \\ &\text{for all } c \in C, b \in B, r \in R, \\ &\quad t \in \{1, \dots, LCM(B)|C|\} \end{aligned} \quad (C2)$$

$$\begin{aligned} \sum_{c \in C} h_{c,t,r} &\leq 1 \\ &\text{for all } t \in \{1, \dots, LCM(B)|C|\}, r \in R. \end{aligned} \quad (C3)$$

In this formulation, constraint (C1) ensures that each configuration is detected, (C2) ensures that a configuration  $(c, b, \delta_b(t))$  can only be detected by transceiver  $r$  if channel  $c$  is scanned during time slot  $t$  by  $r$ , (C3) makes sure that each transceiver  $r$  scans at most one channel during a time slot.

## V. EVALUATION

In this section we first detail the parameter settings of the analyzed scenarios and the evaluation methodology. We then

describe the performance metrics used in this paper. Finally, we present the evaluation results when varying important network parameters: the number of channels, the number of transceivers and the number of neighbors.

### A. Setting

In order to evaluate the proposed strategies for different numbers of channels and devices we use a custom-made discrete-event simulator written in Java. We draw random samples for the BI sets  $B$  and apply all algorithms to the same set of samples. Due to the high computational complexity of the MT-OPT strategy, we had to restrict the size of the studied scenarios, that is, the number of channels, as well as the number and magnitude of elements in the BI sets.

First we draw  $|B|$  from a uniform distribution over  $[3, 6]$  and then draw individual BI's from a uniform distribution over  $[1, 10]$  for a total of 50 BI sets  $B$ . For each drawn BI set  $B$ , number of channels and number of devices we run 30 replications, in each of which the device configurations (for the given parameters) are generated randomly (but they are unique). More specifically, each device configuration is assigned a channel, a BI and start offset  $\delta$  randomly according to a uniform distribution.

We vary the number of devices between 5 and 80 and the number of channels between 2 and 30. Since listening schedules for the MT-OPT strategy are obtained by solving an ILP requiring lots of computational power, the maximum number of channels for this strategy is limited to 10. In those experiments where we vary the number of devices we consistently use eight channels. In those experiments where we vary the number of channels we use 15 devices.

For comparison we included the GREEDY strategy in the evaluation. This strategy does not utilize any cooperation in the discovery process and is based on fixed listening schedules for a specific set of channels and set of BI's.

All results presented in the following are accompanied by confidence intervals for a confidence level of 95%.

### B. Performance Metrics

We next describe the performance metrics that we are using to evaluate the different algorithms. The evaluation is performed from the view of the device with the lowest identifier, i.e. a simulation run is finished when this device has discovered all its neighbors.

For each of the following metrics the results are normalized to the results achieved by the best possible non-cooperative strategy, i.e. the strategy of listening on all channels at the same time (where all neighbors are discovered when their first beacon is transmitted). For a specific scenario and specific scheme the considered metrics are computed and then normalized. Then the mean of the normalized values over the 30 scenarios for a BI set  $B$  is computed. Lastly, the mean of the averaged results of all 50 drawn BI sets  $B$  is computed.

The discovery time of a neighbor is the difference between the time at which the listening device starts operating (at time 0) and the time at which the first beacon of this neighbor is received. In case of strategies which apply immediate

accounting for indirectly discovered neighbors (see Section IV-A), a discovery is also recorded when gossip information of a previously unknown neighbor is received.

We mainly use two metrics. The Mean Discovery Time (MDT) is the mean of the discovery times of all neighbors discovered. The MDT is an important metric in scenarios in which devices are interested in finding some subset of all existing neighbors, e.g. devices looking for one particular access point. The Last Discovery Time (LDT) is the discovery time of the last neighbor discovered. The LDT is important in scenarios in which a device is interested in discovering the complete neighborhood, e.g. devices that want to start a new network and use the discovery results to find a suitable channel and BI combination that does not interfere with already existing networks.

### C. Results for Unbounded Beacon Size

In this section we present results for the case of an unbounded beacon size, i.e. there is no constraint on the number of gossip entries in a beacon message.

1) *MDT*: We first consider results for the MDT.

a) *Varying Number of Channels*: In Figure 1a we show the normalized MDT of all considered strategies (including three different versions of the MT-OPT scheme with different numbers of transceivers) for varying number of channels. The following points are noteworthy:

- If we compare the results of the GSP-MOD-IMM, GSP-UNMOD-IMM and GSP-MOD-IMM-REV strategies with the results for the MT-OPT approach with different numbers of transceivers, it can be seen that the best gossiping strategy show a performance equivalent to between two and four transceivers (and optimal scheduling of their listening activities).
- Surprisingly, the GSP-UNMOD-IMM strategy (in which the listening schedule is **not** recomputed after indirect discoveries) achieves a slightly smaller MDT than the GSP-MOD-IMM strategies (which attempt to shorten the schedule after indirect discoveries). This suggests that attempting to shorten the schedule based on indirect discoveries can actually be harmful. One possible explanation for this is that by dropping these later slots from our schedule we deprive the listener of opportunities to hear directly from indirectly discovered neighbors and pick up any additional gossip information they might have obtained in the meantime. Note that, heuristically, the later we hear a beacon from a neighbor, the more information is included in them.
- The results for the GSP-MOD-VER scheme correspond to the usage of one to two transceivers in the MT-OPT strategy. However, it appears that with increasing number of channels the GSP-MOD-VER scheme starts to outperform MT-OPT-2.

b) *Varying Number of Neighbors*: In Figure 1b we show the impact of the number of neighbors on the normalized MDT of the discovery strategies (considering five different versions

of MT-OPT) while keeping the number of channels fixed to the default value of 8. The following points are interesting:

- With increasing number of neighbors the MDT of the GSP-MOD-IMM and GSP-UNMOD-IMM strategies decreases monotonically and reaches the performance of the MT-OPT-6 strategy at 80 neighbors.
- Interestingly, the MDT of the GSP-MOD-VER is not monotonically decreasing but rather shows a minimum around 20 neighbors and increases again with larger numbers of neighbors. This is caused by the recomputation of listening schedules forcing to validate gossiped information. With lower numbers of neighbors the validation decreases the MDT even though it is still higher than the results of the other three strategies assuming all gossip information to be correct. Since the number of neighbors has no impact on the length of the original listening schedule, the forced validation slots added to the schedule result in a delayed discovery of other neighbors that have not yet been gossiped. The impact of the validation of neighbors can also be seen in the results of the GSP-MOD-IMM-REV strategy. It results in a larger MDT than for the GSP-MOD-IMM and GSP-UNMOD-IMM strategies, with an equivalent of around 4 transceivers using the MT-OPT approach with 80 neighbors.

2) *LDT*: We next consider the LDT.

a) *Varying Number of Channels*: Figure 1c depicts the normalized LDT with varying number of channels. Broadly, we can observe the same trends as for the MDT and varying number of channels, with some slight differences. Compared to the MDT results the GSP-MOD-IMM-REV strategy has a higher LDT than the GSP-UNMOD-IMM strategy and very similar results to GSP-MOD-IMM. The LDT of the GSP-MOD-VER strategies is lower than MT-OPT-2 for higher numbers of channels. Furthermore, the LDT of the GSP-UNMOD-IMM strategy is closer to MT-OPT-4 compared to the results of the MDT.

b) *Varying Number of Neighbors*: Figure 1d shows the normalized LDT when varying the number of neighbors. The LDT of the GSP-UNMOD-IMM scheme decreases towards the optimum of MT-OPT-8 with increasing number of neighbors. When looking at the GSP-MOD-IMM-REV strategy, the combination of removing discovered configurations during schedule recomputation and having to revisit gossiped neighbors results in large and even increasing LDT values for larger numbers of neighbors. Furthermore, the results are similar to GSP-MOD-VER but with an offset caused by the immediate accounting property of GSP-MOD-IMM-REV, which simply counts discoveries earlier than GSP-MOD-VER does.

### D. Results for Bounded Beacon Size

In the results presented so far we have assumed that beacons are sufficiently large and can contain all discoveries that a device has ever made. In the following we introduce restrictions on the number of configurations that can be reported in a beacon and compare the resulting MDT performance of selected strategies against the performance achievable with an

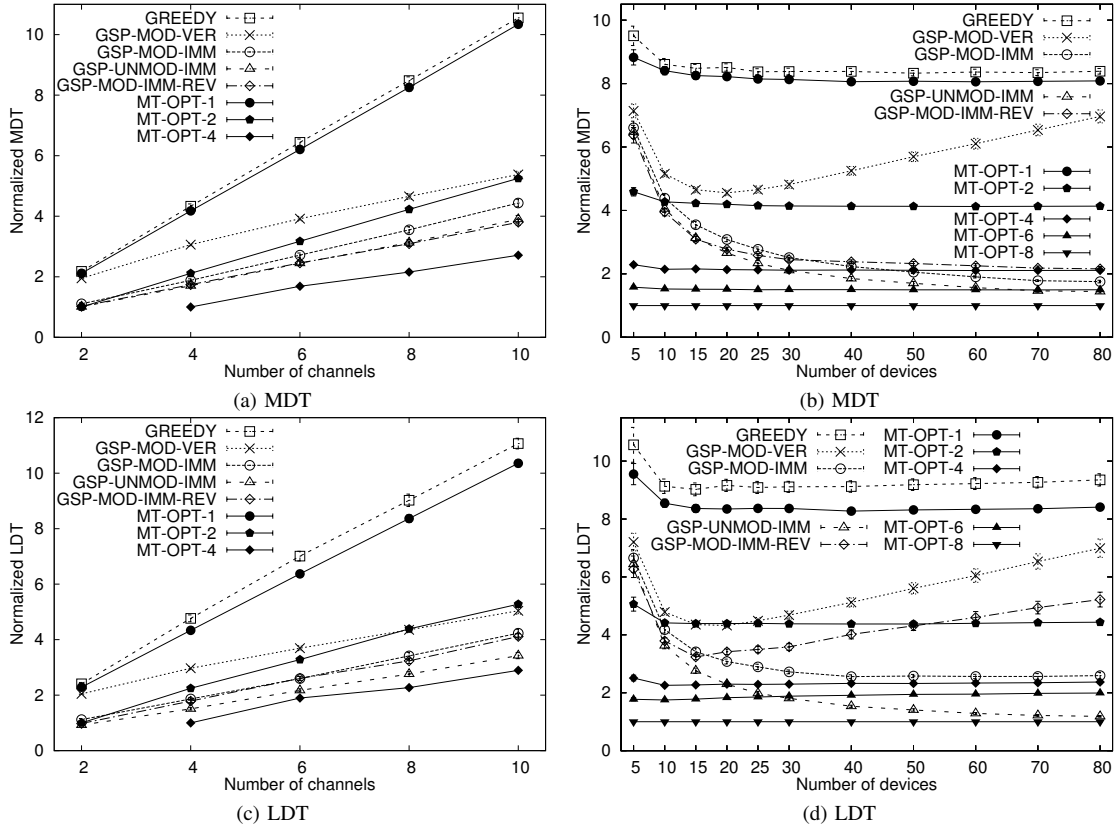


Fig. 1: Normalized MDT and LDT under varying number of channels and varying number of neighbors.

unlimited beacon size. We again vary the number of channels and the number of neighbors. The number of gossip entries has been set to 1, 5 and unlimited entries. Please note that a limitation on the number of configurations that can be included also introduces the problem of *selecting* the configurations to include into beacons when many more are known than can be included. The selection rule is explained in Section IV-A.

1) *Varying Number of Channels:* In Figure 2a the normalized MDT of the gossip-based strategies is shown for different limits for the number of gossip entries and for varying numbers of channels (with 15 neighbors). As expected, allowing devices to exchange more gossip information per beacon message decreases the MDT for all analyzed strategies when considering the same number of channels and neighbors.

In contrast to the results of Section V-C1 in which only a small number of channels has been analyzed (so we could include results for the ILP-based MT-OPT approach), the GSP-MOD-IMM-REV strategy has the lowest MDT among the gossip-based strategies. Since the number of neighbors is set to the fixed value of 15, higher numbers of channels result in a longer time required to detect all neighbors. Therefore, receiving additional gossip information in the process of validating/revisiting indirectly discovered neighbors helps with finishing the detection of all neighbors in scenarios with a large number of channels and/or a BI set  $B$  with large entries.

Even though the GSP-MOD-VER strategy considers a discovery of a neighbor only if one of its beacons is directly

received, it has a lower MDT than GSP-MOD-IMM and GSP-UNMOD-IMM for large numbers of channels and limited number of gossip entries per beacon.

2) *Varying Number of Neighbors:* Figure 2b depicts the normalized MDT for limited beacon entries and different numbers of neighbors. The MDT of GSP-MOD-VER and GSP-MOD-IMM-REV suffers from the process of validating or revisiting neighbors as described in V-C1b and is increasing for more than 25 neighbors for limited beacon entries. On the other hand, for an unlimited number of entries the MDT of GSP-MOD-IMM-REV is continuously decreasing when increasing the number of neighbors. However, for more than 25 neighbors the GSP-MOD-IMM-REV strategy has the highest MDT among the strategies assuming gossip entries to be true.

A large number of gossip entries has only minor impact on the MDT of GSP-MOD-VER; the results for 5 entries and an unlimited number of entries are very close to each other. Furthermore, for large numbers of neighbors the gap between the results of the different allowed number of beacon entries vanishes. The results of the GSP-MOD-IMM and GSP-UNMOD-IMM strategy are almost similar when only a single beacon entry is allowed. However, for larger number of entries the GSP-UNMOD-IMM achieves a faster MDT.

## VI. CONCLUSION

This paper presented different neighbor discovery approaches utilizing cooperation between devices in order to

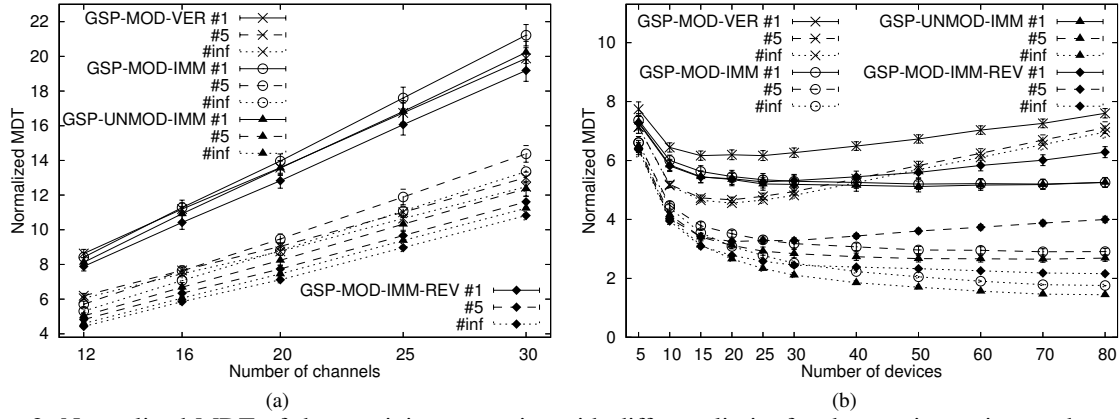


Fig. 2: Normalized MDT of the gossiping strategies with different limits for the gossip entries per beacon

improve the discovery process. We have developed gossip-based discovery strategies that exchange discovery results by including them in periodically transmitted beacon messages. The gossip-based approaches have been compared to an optimal multi-transceiver approach whose listening schedules are optimized regarding the Mean Discovery Time (MDT).

We have shown that by only exchanging discovery results via beacon messages, devices can achieve a significant improvement in the MDT and Last Discovery Time (LDT) depending on the number of channels and neighbors comparable to the utilization of up to the optimal number of transceivers being equal to the number of channels. Our future work will take mobility and partially connected networks into account.

#### ACKNOWLEDGEMENT

This work has been partially funded by the DAAD programme FITweltweit and by the European Horizon 2020 programme under grant agreement number 688116 (eWINE).

#### REFERENCES

- [1] A. Willig, N. Karowski, and J.-H. Hauer, "Passive Discovery of IEEE 802.15.4-based Body Sensor Networks," *Elsevier Ad Hoc Networks Journal*, vol. 8, no. 7, pp. 742–754, 2010.
- [2] N. Karowski, A. Viana, and A. Wolisz, "Optimized Asynchronous Multichannel Neighbor Discovery," in *Proc. of IEEE INFOCOM*, April 2011.
- [3] N. Karowski, A. Viana, and A. Wolisz, "Optimized Asynchronous Multichannel Discovery of IEEE 802.15.4-Based Wireless Personal Area Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1972–1985, October 2013.
- [4] N. Karowski and K. Miller, "Optimized Asynchronous Passive Multichannel Discovery of Beacon-Enabled Networks," Telecommunication Networks Group, Technische Universität Berlin, TKN Technical Report Series TKN-15-002, March 2015.
- [5] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wake-up scheduling in wireless sensor networks," *Computers, IEEE Transactions on*, vol. 59, no. 11, pp. 1562–1575, Nov 2010.
- [6] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," *Elsevier Computer Networks Journal*, vol. 43, no. 3, pp. 317–337, 2003.
- [7] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *ACM SenSys*, Nov. 2008.
- [8] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: ACM, 2010, pp. 350–361.

- [9] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor?" in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012, pp. 185–196.
- [10] W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: A generic flexible protocol for neighbor discovery," in *INFOCOM, 2014 Proceedings IEEE*, April 2014, pp. 540–548.
- [11] K. Wang, X. Mao, and Y. Liu, "Blinddate: A neighbor discovery protocol," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 26, no. 4, pp. 949–959, April 2015.
- [12] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *ACM MobiHoc*, Oct. 2001.
- [13] S. Vasudevan, D. Towsley, D. Dennis, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *ACM MobiCom*, Sep. 2009.
- [14] R. Khalili, D. L. Goeckel, D. Towsley, and A. Swami, "Neighbor discovery with reception status feedback to transmitters," in *IEEE Infocom*, Mar. 2010.
- [15] S. Ali, E. T. Yazdi, and A. Willig, "Investigations on passive discovery schemes for IEEE 802.15.4 based body sensor networks," in *Telecommunication Networks and Applications Conference (ATNAC), 2013 Australasian*, Nov 2013, pp. 89–94.
- [16] L. Chen, Y. Shu, Y. Gu, S. Guo, T. He, F. Zhang, and J. Chen, "Group-based neighbor discovery in low-duty-cycle mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2015.
- [17] J. Du, E. Kranakis, and A. Nayak, "Cooperative neighbor discovery protocol for a wireless network using two antenna patterns," in *2012 32nd International Conference on Distributed Computing Systems Workshops*, June 2012, pp. 178–186.
- [18] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," vol. 4, mar. 2005, pp. 2502 – 2512 vol. 4.
- [19] A. Purohit, B. Priyantha, and J. Liu, "Wiflock: Collaborative group discovery and maintenance in mobile sensor networks," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, April 2011, pp. 37–48.
- [20] R. Cohen and B. Kapchits, "Continuous neighbor discovery in asynchronous sensor networks," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 1, pp. 69–79, Feb 2011.
- [21] D. Camps-Mur and P. Loureiro, "e<sup>2</sup>d wi-fi: A mechanism to achieve energy efficient discovery in wi-fi," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1186–1199, June 2014.
- [22] M. Bakht, J. Carlson, A. Loeb, and R. Kravets, "United we find: Enabling mobile devices to cooperate for efficient neighbor discovery," in *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, ser. HotMobile '12. New York, NY, USA: ACM, 2012, pp. 11:1–11:6.
- [23] H. Park, Y. Kim, I. Jang, and S. Pack, "Cooperative neighbor discovery for consumer devices in mmwave ad-hoc networks," in *2012 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2012, pp. 100–101.