Griassdi: Mutually Assisted Slotless Neighbor Discovery

Philipp H. Kindt
Chair of Real-Time Computer Systems
Technical University of Munich (TUM), Germany
kindt@rcs.ei.tum.de

Daniel Yunge
Chair of Real-Time Computer Systems
Technical University of Munich (TUM), Germany
yunge@rcs.ei.tum.de

Gerhard Reinerth
Chair of Real-Time Computer Systems
Technical University of Munich (TUM), Germany
reinerth@rcs.ei.tum.de

Samarjit Chakraborty
Chair of Real-Time Computer Systems
Technical University of Munich (TUM), Germany
chakraborty@rcs.ei.tum.de

ABSTRACT

Recent results show that slotless, purely-interval based neighbor discovery protocols, in which time is assumed to be continuous, achieve significantly lower worst-case discovery latencies than time-slotted protocols. In slotted protocols, the discovery of device A by B and vice-versa occurs within the same slot, and hence the latencies for one-way and two-way discovery are identical. However, in purely interval-based protocols, these latencies are independent from each other, leading to longer mean latencies for two-way discovery. In this paper, we propose a cooperative approach to reduce this two-way discovery latency. In particular, each side broadcasts information on the time-period until its next reception phase takes place. The remote device adjusts its beacon schedule accordingly once a first packet is received. Compared to non-cooperative slotless protocols, this technique can reduce the two-way discovery latency by up to 43 %. We propose a theory to model such protocols and show that with an optimized schedule, our proposed protocol achieves considerably shorter mean latencies than all known protocols, while still guaranteeing worst-case latencies that are similar to the best known solutions. For example, compared to Searchlight-Striped, our proposed protocol achieves by up to 89 % lower mean latencies and by up to 86 % lower worst-case latencies.

CCS CONCEPTS

•**Networks** →**Mobile networks; Mobile ad hoc networks;** *Link-layer protocols;*

KEYWORDS

Neighbor Discovery, MANETs, Wireless Communication

ACM Reference format:
Philipp H. Kindt, Daniel Yunge, Gerhard Reinerth, and Samarjit Chakraborty. 2017. Griassdi: Mutually Assisted Slotless Neighbor Discovery. In *Proceedings of The 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, Pittsburgh, PA USA, April 2017 (IPSN 2017),* 12 pages.
DOI: http://dx.doi.org/10.1145/3055031.3055074

IPSN 2017, Pittsburgh, PA USA

DOI: http://dx.doi.org/10.1145/3055031.3055074

1 INTRODUCTION

Mobile ad-hoc networks have become widespread, driven by the availability of economy-priced low-power radios. However, with shrinking device dimensions and small batteries, energy-consumption is still one of the main concerns in such networks. Especially, initiating a first contact between two devices is an energy-expensive procedure. For example, in applications such as contact tracking, in which a device attempts to record continuously which devices are within its range, the energy spent for neighbor discovery dominates the total energy consumption. To reduce it, duty-cycled discovery protocols, in which each device repeatedly goes to a sleep mode in between the active phases, are widely-used. Only if one device sends a packet at the same point in time at which the other one is listening, both devices rendezvous. To realize fast discoveries with a given duty-cycle (the fraction of time a device is awake), slotted protocols are considered to be an efficient solution. Time is subdivided into multiple, equal-length slots. In each slot, the device may either be active or asleep. Typically, a packet is sent at the beginning and the end of each active slot [2]. In between these two packets, the device listens to the channel for incoming packets. Based on this scheme, multiple schedules of active and sleep-slots have been proposed, which provide deterministic upper latency bounds.

An alternative to slotted protocols is purely interval (PI)-based, continuous time discovery, as depicted in Figure 1. In such protocols, each device broadcasts packets with a certain interval T_a, which is called the *advertising interval.* The remote side switches on its receiver periodically for a duration of d_s time-units, which is called the *scan window.* The repetition period of the reception is called the *scan interval* T_s. Such schemes are, with certain modifications, widely used in low-power protocols such as ANT/ANT+ or Bluetooth Low Energy (BLE). However, until recently, their main properties, in particular their latency performances for given-duty-cycles, have not been well understood. Recently, a theory for computing the discovery latencies of such protocols has been proposed [4]. Protocols based on this theory (e.g.[5]) can achieve lower worst-case latencies than all previously known protocols. The main reason for this is that, unlike in slotted protocols, sending beacons and listening on the channel are temporally decoupled from each other. Therefore, devices can send the optimal number of beacons at the optimal points in time, whereas in slotted protocols, the number of beacons is limited by the number of active slots.

In slotted protocols, the discovery of device B by A takes place within the same slot as the discovery of A by B. Therefore, the

mean latency for two-way discovery is identical to the one for one-way discovery. However, in slotless protocols, the mean discovery latency for two-way discovery is significantly larger than for one-way discovery, even though the worst-case bound is equal in both cases. Because both devices follow the same schedule, the discovery of device A by device B occurs independently from the discovery of device B by device A. The point in time both devices come into range and hence the initial temporal offsets between their packets and scan windows are random. Since the two-way discovery procedure is only complete if both of the two independent probabilistic processes have finished, the two-way latency is defined by the maximum latency of the two procedures.

This paper considers symmetric (i.e., all devices follow the same schedule), asynchronous (i.e., the clocks of all devices are unsynchronized) discovery. We propose a technique to decrease the mean two-way discovery latency of such protocols while still maintaining a similar worst-case latency. Unlike in slotted protocols, there are two independent chances on a successful reception in each period (viz, the reception of a packet from device B by device A and vice-versa). The main insight which we exploit is that the mean two-way discovery latency can be reduced by combining both chances, if both devices carry out the discovery procedure in a cooperative fashion.

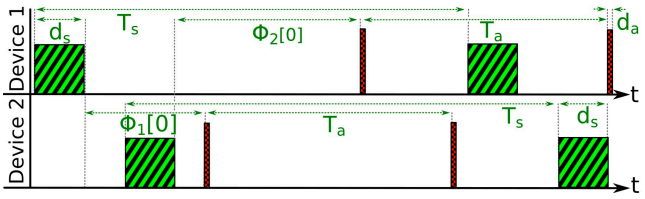

Figure 1: Periodic interval-based discovery

Consider two devices, as depicted in Figure 1. The initial temporal distance $\Phi_n[0]$ between the first scan window of device $n \in \{1, 2\}$ and its temporally left neighboring advertising packet of the remote device is a random variable that represents the point in time at which the two devices come into range. Even though one device most likely discovers its neighbor before being discovered by the opposite one, the total latency for two-way discovery remains the maximum of both latencies. If the earlier of both discoveries could be exploited to accelerate the discovery of the second rendezvous, the total mean latency for mutual discovery could be reduced considerably. In this paper, we propose *Griassdi*[1], a protocol which systematically exploits the temporal correlation of discovering device 2 by device 1 (or vice-versa), given device 2 has already received a packet from device 1. Towards this, every packet sent contains a hint on the time-period after which the next reception phase of device 2 takes place. A device receiving such a hint will adapt its beaconing schedule appropriately to send a packet within the reception period of the opposite device. We call this technique to reduce the mean discovery latency *assisted two-way* (A2W) discovery. In slotted protocols, both devices discover each other within the same slot and therefore, the problem of optimizing the two-way latency does not arise.

The analysis of such protocols is challenging, since assisted two-way discovery imposes a large number of temporal constellations that need to be accounted for by an appropriate combinatorial model. We present a precise theory for modeling such protocols. Based on this, we propose optimized A2W-schedules that reduce the mean discovery latencies by up to 43 % compared to non-assisted solutions, without increasing their worst-case latencies by more than 7 %. The resulting mean latencies for a given duty-cycle are significantly lower than the ones achieved by all known discovery protocols, including all slotted ones. For example, Griassdi achieves by 87 % lower mean latencies than Searchlight [1], while achieving similar worst-case latencies to the PI-kM$^+$Opt-protocol [5], which guarantees the best latency-duty-cycle-relations of all known protocols in the worst-case. Compared to existing work, we make the following contributions:

- We propose the first slotless protocol that systematically exploits mutual assistance to reduce the mean discovery latency in an optimized manner.
- We provide a theory for analyzing PI-based, slotless A2W protocols, which is capable of computing their mean and worst-case discovery latencies.
- We evaluate the performance of our proposed protocol, compare it to existing solutions, and demonstrate its proper functioning using a real-world implementation.

The rest of this paper is organized as follows. In Section 2, we present an overview of related work. In Section 2.2, we propose a high-performance A2W-based protocol. We describe a precise mathematical model for its analysis in Section 4. In Section 5, we evaluate the performance of this protocol and compare it to existing approaches. We present an implementation on a radio in Section 6 and show that the protocol behaves as predicted by the theory using comprehensive experimental data. Finally, we conclude our paper by outlining some directions for future work in Section 7.

2 RELATED WORK

In this section, we give an overview of existing asynchronous, symmetric slotted and slotless neighbor discovery protocols.

2.1 Slotted Discovery Protocols

Slotted protocols subdivide time into multiple, equal-length periods, called slots. In most slots, the device remains asleep (passive slots). In the remaining, active slots, a beacon is sent at the beginning and the end. The device listens for incoming packets in between the two beacons. Whenever the temporal distance between two active slots is smaller than approximately one slot-length d_{sl}, mutual discovery is achieved, since the beacon of one device falls into the reception period of the other device and vice-versa. Since all temporal offsets within one slot lead to the same discovery latency, discrete-time models can be applied for computing the latencies, which allow for relatively simple analysis techniques.

Multiple schedules of active and passive slots, which guarantee deterministic discovery within a certain time-period, have been proposed. One of the first slotted protocols is Disco [2]. Each device chooses two primes p_1 and p_2. Every p_1-th and p_2-th slot is active, whereas the other slots are sleep slots. The Chinese Remainder Theorem states that mutual discovery is assured after $p_1 \cdot p_2$ slots. An extension of this concept is U-Connect [3], in which each device chooses a prime-period of slots p. In each period, the p-th slot is

[1] The term "Griaß Di" means "hello" in some parts of Southern Germany.

active. In addition, in each super-period p^2, the first $\lfloor \frac{p+1}{2} \rfloor$ slots are also active, leading to shorter latencies in terms of slots. In addition, the design of U-Connect allows for shorter slot-lengths, which further reduces the latencies in terms of time. Another performance increase has been achieved by Searchlight [1], which uses one fixed slot in every period in combination with a *probe slot*. The probe slot changes its position in each period to search for the fixed slot. A cooperative approach which broadcasts information on already discovered neighbors is *Acc* [11]. Unlike our approach, Acc is built upon slotted protocols such as Disco, in which mutual discovery between two nodes is always achieved within the same slot. Considering such slotted protocols, Acc can therefore only accelerate the discovery of additional nodes. However, Acc could also be used on top of other protocols, thereby providing speedups also for pairwise discovery.

To achieve even shorter worst-case latencies for a given duty-cycle, an increasing trend towards slotless protocols can be observed. Intermediate steps between slotted and slotless protocols are Lightning [10] and G-Nihao [9], in which special transmission-only slots are used. In such pseudo slots, a beacon is sent and no reception takes place. The most flexible proposed schedule in [9], G-Nihao, can achieve shorter discovery latencies than most slotted protocols described above by sending additional beacons in each period. Further, Lightning applies mutual assistance to achieve two-way discovery, but the protocol is not optimized to systematically exploit this for short mean latencies. Fully slotless protocols completely break away with the assumption of (pseudo)-slots. They are described next.

2.2 Slotless Protocols

Whereas slotted protocols have been studied thoroughly in the literature, the behavior of slotless, purely interval-based protocols, in which each device sends beacons with a certain interval T_a and listens to the channel with another interval T_s, have not been well understood until recently. Since time is continuous, modeling their behavior requires handling infinitesimal amounts of time. A mathematical theory to compute the exact discovery latencies of PI-based protocols has been presented recently [4], which is briefly described below and, in more detail, in Section 4. Unlike the popular belief that the latencies of such protocols are unbounded, the theory reveals that for most parametrizations, deterministic upper-bounds can be achieved. Therefore, they are a deterministic alternative to slotted protocols.

Figure 2 depicts the main idea behind modeling PI-based protocols. The hatched rectangles represent two neighboring scan windows, which are separated by the scan interval T_s. The time between them can be subdivided into multiple partitions. After two devices have come into range, the discovery latency is determined entirely by in which partition the first advertising packet is sent. The larger, colored rounded boxes in Figure 2 depict partitions for which a certain integer-number of advertising intervals (i.e., the

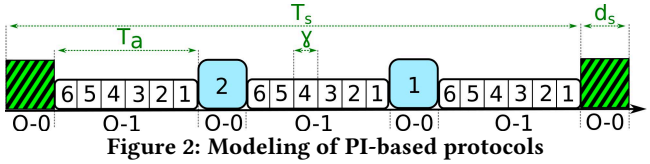

Figure 2: Modeling of PI-based protocols

numbers indicated in the boxes) need to pass until a match of the scan window occurs, because their distances from the scan interval are integer-multiples of the advertising interval. For example, it can be easily observed that if the first advertising packet is sent within the leftmost rounded box, $2 \cdot T_a$ time-units will pass until the right scan window is reached. This case is referred to as an *order-0* process. Order-0 means that, starting from the corresponding partition, the temporal distance from the scan window is reduced in multiples of T_a until reaching it. All areas in which order-0 processes match are marked with *O-0* in the figure. In the remaining parts between two scan windows, one needs to take into account appropriate multiples of T_a and T_s, which is referred to as an order-1 process (marked with *O-1* in the figure). The important parameter is the "overshoot" between such multiples. In our proposed protocol, T_a is always smaller than T_s. Under this assumption, one can compute this overshoot γ by

$$\gamma = \left\lceil \frac{T_s}{T_a} \right\rceil T_a - T_s. \tag{1}$$

In other words, γ is the absolute difference in time between one instance of the scan interval T_s and the multiple of advertising intervals T_a that exceeds T_s for the first time. The parameter γ implies that, whenever $\lceil \frac{T_s}{T_a} \rceil$ advertising intervals pass, the temporal difference between any advertising packet and the next scan window being temporally right of it is reduced by γ time-units. In such areas (indicated with *O-1* in Figure 2), the distance to the next (temporally right) partition in which the order-0 process matches is therefore reduced by multiples of γ and a discovery latency of $\sigma = \lceil \frac{T_s}{T_a} \rceil \cdot T_a$ time-units is incurred in each reduction step. The numbers depicted in the partitions of the O-1 areas in Figure 2 represent the number of γ-intervals until reaching the next O-0 partition. Once an O-0 partition is reached, the scan window is approached directly in steps of T_a, as already explained.

With this scheme, one can compute the discovery latency that applies given the first advertising packet is sent within a particular partition. In each O-1 area, one has to determine the number of γ-intervals until reaching the next O-0 area. From each O-0 area, one has to count the remaining advertising packets until reaching the scan window. For example, in Partition 5 of the leftmost O-1 area, the discovery latency is $d_{nd} = 5 \cdot \lceil \frac{T_s}{T_a} \rceil T_a + 2 \cdot T_a + d_a$. The term d_a denotes the transmission-duration of one packet. However, if $\gamma > d_s$, some parts of each O-1 partition would not match the scan window. In such cases, so-called *higher-order processes* occur, which are not relevant for this paper. Further, in some cases, rather than the "overshoot", the "underflow" needs to be considered. For the protocols designed in this paper, only overshooting processes of order 0 and 1 are relevant. For details on the remaining cases, we refer to [4].

The model presented in [4] is only capable of computing the global mean- and maximum latencies. It cannot compute the latency given a certain offset between an advertising packet and a scan window explicitly. In Section 4, we for the first time present an algorithm that computes all partition borders and the corresponding latencies explicitly, since this is needed for mathematically describing mutual assistance techniques.

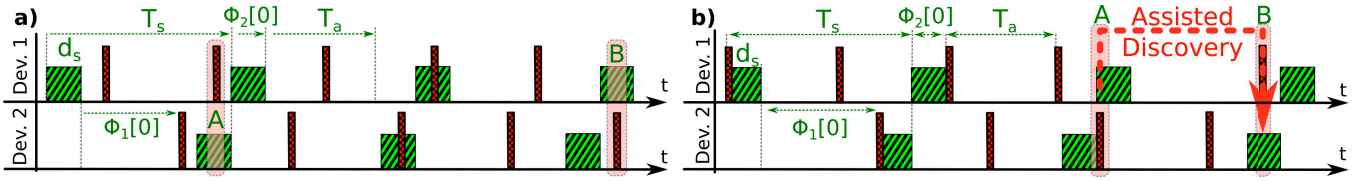

Figure 3: PI-based discovery without (a) and with (b) A2W discovery

One can derive optimized parametrizations of PI-based protocols based on the model described above. Parametrizations that are Pareto-optimal, under the constraint that only order-0 and oder-1 processes are considered, have been presented in [5]. The main idea behind such parametrizations is to choose the parameters such that γ becomes slightly lower than the scan window d_s. As a result, with every advertising interval, the temporal distance to the next order-0 partition shrinks by slightly less than d_s time-units. Larger values of γ would lead to offsets for which the scan window is missed, whereas lower values would increase the duty-cycle without accelerating the discovery. Further, T_s and d_s are chosen such that a certain target duty-cycle is achieved and the product of latency and duty-cycle is minimized, based on the model described above.

When assuming a slot-length of 10 ms and a packet length of 368 μs, the $PI - kM^+Opt$-protocol presented in [5] achieves significantly lower worst-case latencies for every given-duty-cycle than all known slotted protocols. For example, Searchlight [1] takes up to 10× longer to discover a device with the same duty-cycle in the worst-case. The pseudo-slotted protocol G-Nihao [9] is outperformed by a factor of 3.1, respectively.

The protocol proposed in this paper is built upon the PI-kM-framework presented in [5] and on the model presented in [4]. However, it contains multiple significant extensions and improvements. In particular, we propose mutually assisted discovery and extend the model from [4] to account for A2W-based approaches. In addition, while the design of the protocols from [5] is optimized for low worst-case latencies for any given duty-cycle, it does not concern controlling the number of scan intervals after which discovery is guaranteed. In our proposed protocol, this number of scan intervals can be parametrized for low mean latencies, while still achieving almost the same worst-case behavior. Since mutual assistance can only reduce the mean latency if discovery is carried out within a certain multiple of scan intervals, a mathematical framework that provides suitable parametrizations is exploited to maximize the benefits of mutual assistance in this paper.

Compared to all existing solutions, the protocol proposed in this paper achieves shorter mean-discovery latencies for every given duty-cycle, while guaranteeing worst-case latencies almost as low as PI-kM$^+$-Opt. Therefore, it outperforms all known discovery protocols both in terms of mean latencies while at the same time outperforming all protocols (except [5]) in terms of worst-case latencies.

3 MUTUALLY ASSISTED NEIGHBOR DISCOVERY PROTOCOLS

In this section, we describe our proposed discovery protocol *Griassdi* and introduce the technique of A2W discovery.

3.1 Protocol Definition

3.1.1 PI-Based Discovery. Griassdi applies a PI-based scheme, as shown in Figure 3a). The hatched boxes depict the reception periods, whereas the vertical bars show the advertising packets sent. Each device broadcasts packets with an interval T_a. In addition, each device periodically scans the channel with another interval T_s for d_s time-units. If an advertising packet is to be sent while listening, the listening-phase is interrupted for the duration of the transmission, without extending its length.

For the sake of simplicity of exposition, we assume one pair of devices in the rest of the paper. However, our proposed protocol can also be applied for mutual discovery among a larger number of devices, as studied in Section 5.2.3. We further assume that each advertising packet contains some payload (such as device addresses, device type and indicators for the services offered). Two-way discovery occurs once device 2 has received a packet from device 1 (marked with A in Figure 3a)) and device 1 has received a packet from device 2 (marked with B in the figure). Clearly, the receptions of both packets occur independently from each other. Therefore, the time until both devices have mutually discovered themselves is much larger than the time for one-way discovery. In the example depicted in Figure 3a), device 2 discovers device 1 during its first scan window. The latency for one-way discovery is below $1 \times T_s$, whereas the latency for two-way discovery is almost $4 \times T_s$.

3.1.2 Parametrization. In what follows, given a certain duty-cycle η that is to be realized and a packet length d_a, we derive efficient values of T_a, T_s and d_s. To exploit mutual assistance, the discovery procedure must take place within multiple scan intervals. Protocols which guarantee discovery within 1 interval cannot benefit from mutual assistance, since the additional packets would be sent within scan windows in which regular packets would have been sent, anyway. Unlike previous attempts (e.g., [5]), our proposed parametrizations can realize discovery within a certain, adjustable number of scan intervals while still providing optimized latencies. This allows for a systematic exploitation of mutual assistance to reduce the mean discovery latency. We consider two integer values k and M, and require that k divides $M+1$. To achieve short worst-case latencies, we choose the scan interval T_s and the advertising interval T_a as

$$T_s = M \cdot (d_s - d_a) - \epsilon, \tag{2}$$

$$T_a = \left(\frac{M+1}{k} \cdot (d_s - d_a)\right) - \frac{\epsilon}{k}. \tag{3}$$

Equations 2 and 3 form a configuration in which a certain integer-multiple k of advertising intervals is slightly shorter (viz. by ϵ time-units) than $d_s - d_a$. Therefore, after each scan interval, the temporal distance from any packet to the next scan window is reduced by approximately $d_s - d_a$ time-units. This value is optimal, since

larger values would lead to misses of the scan window, whereas shorter values would increase the duty-cycle without decreasing the worst-case latency. Further, this parametrization effects that an integer-number of γ-intervals fits into each scan interval. This is efficient, since for a slightly larger, non-integer number of intervals, an additional full γ-interval would be needed to reach the scan window while only extending the scan interval (and hence, the duty-cycle) by less than γ.

The configuration $k \cdot T_a = (M+1)(d_s - d_a)$ defines a border-case, since the growth of the temporal distance γ is exactly $d_s - d_a$ time-units, and infinitesimally larger values of γ would lead to misses for some offsets. To achieve a well-defined behavior, T_a and T_s need to be somewhat shorter than the ideal values, which is achieved by subtracting ϵ in the Equations above. In the following sections, we assume a small value of e.g. $\epsilon = 0.1$ ps for our theoretic analysis. In practical implementations, ϵ is not relevant, since d_s will be artificially extended to compensate for clock skew, which also avoids this border-case.

In addition to k and M, we introduce a third integer-parameter $R = \frac{M+1}{k}$. As we describe later, R is used to balance the worst-case discovery latency against the mean discovery latency, since R is equivalent to the number of scan intervals that pass in the worst-case until rendezvous. If $R = 1$, the discovery procedure will terminate within 1 interval, and no speedup can be achieved by mutual assistance. Though the duty-cycle-latency-product is influenced by R, there is no linear relation to it, since the protocol adapts all parameters accordingly to any value of R. R can be chosen by the user, and there is an optimum value for low mean latencies. We will study the impact of R in detail, later.

The main metric for the energy-consumption of neighbor discovery protocols is the duty-cycle η, which represents the fraction of time the device is active during operation [3]. By computing the duty-cycle (cf. Equation 5) and the worst-case latency according to the model presented in Section 2, one can show that for realizing a certain duty-cycle η, for a given value of R, the worst-case latency has a local optimum at

$$M_{opt} = \left\lceil \frac{\sqrt{R \cdot \eta + 1}\sqrt{1-\eta} + 1}{\eta} \right\rceil. \quad (4)$$

We further require that R divides $M+1$ and therefore round M to the next higher value of M_{opt} that is, when incremented by 1, divisible by R. Hence, for a given value of R, the values of M and k are fully determined by the equations above. The only degree of freedom left is d_s. It is defined by the target duty-cycle the protocol has to realize. As can be verified easily, the duty-cycle of the proposed protocol is

$$\eta = \frac{d_a T_s + d_s T_a}{T_s T_a}. \quad (5)$$

Using Equations 2 and 3, d_s is chosen such that the desired target-duty-cycle η is realized, if

$$d_s = d_a + \frac{d_a + M(k+1)d_a}{(M\eta - 1)(M+1)}. \quad (6)$$

In practical implementations, one would limit d_s to a minimum value $d_{s,m}$. While in theory, every value greater than d_a is feasible, a certain lower bound, e.g. $d_{s,m} = 10 \cdot d_a$ should be maintained to achieve a high matching probability of additional packets for mutual assistance (see below) and also low collision probabilities (cf. Section 5.2.3).

In summary, using the equations above, given a certain packet length d_a, a target-duty-cycle η and a value of R, the equations above provide optimized values of T_a, T_s and d_s. The packet length d_a and the duty-cycle can be chosen freely (within certain feasible bounds). The value of R allows for a trade-off between worst-case and mean latency, which we describe in detail in Section 5.

3.1.3 A2W Discovery. To accelerate the two-way discovery, we extend this protocol as follows. Every packet sent by each device contains, as a hint for the remote device, the time-period until the center of its temporally next listening-period. Once a device receives an advertising packet from the remote side, it schedules an additional packet according to the hint received. Therefore, the additional packet directly hits the center of the temporally next scan window of the opposite device. With this scheme, device 1 assists device 2 to discover device 1 and vice-versa. Since the probability of a match between an advertising packet and scan window is greatly increased due to coupling both parallel probabilistic processes, the mean latency is reduced. A discovery-procedure using this technique is shown in Figure 3b). Device 1 receives a packet from device 2 at the point in time marked with *A*. Based on the information received, it schedules its next packet such that it will meet the next scan window of device 2 at time *B*. While the mean latency for the first rendezvous between the two devices remains constant, the second rendezvous takes place guaranteed within one scan interval T_s after the first rendezvous. Therefore, whenever there is more time than T_s time-units between the first and the second rendezvous, the discovery latency is reduced.

4 PROTOCOL ANALYSIS

In Section 2, we have described a model for computing the one-way discovery latency of PI-based protocols [4]. However, the theory behind this model is based on describing a single stochastic process, only. For two-way discovery, two such processes need to be considered in parallel. For assisted two-way discovery, these processes are not independent, and therefore the modeling becomes more challenging. In what follows, we extend the one-way theory to account for two-way discovery, including A2W discovery. Towards this, we first present an algorithm to compute the complete set of partitions and one-way latencies, which we call an *explicit model*, because the borders and latencies of all partitions are computed explicitly. In contrast, the model presented in [4] can only compute the global mean and maximum latency among all partitions. We then extend this explicit model towards non-assisted and assisted two-way discovery.

4.1 Explicit Latency Modeling

The discovery latency of our proposed protocol is determined by the initial offset $\Phi[0]$ between the first advertising packet and its temporally left neighboring scan window of the remote side. The range of possible initial offsets is $[0, T_s]$, since all other offsets can be described as periodic repetitions of this interval. As already mentioned, this range can be subdivided into multiple partitions. The discovery latency is determined entirely by into which partition the initial offset falls, whereas the actual point in time within each

partition does not affect the latency (cf. Figure 2). In a set of partitions Ξ, each partition $j \in \Xi$ consists of two borders $l[j] < r[j]$ and a corresponding discovery latency $d[j]$. The purpose of our proposed model is to compute the complete set of partitions Ξ for a given triple of parameters (T_a, T_s, d_s). This set of partitions represents the discovery latencies for all possible initial offsets.

We write $[l, r] \leftarrow d$ to indicate that a new partition with borders l and r having a latency of d is added to the corresponding set Ξ. We assume a set of unprocessed partitions $\Xi_{u,i}$ for a given process-order $i \in \{0, 1\}$[2] and a set of completely processed partitions Ξ_c. The following algorithm can be used for computing the explicit latencies for one-way discovery, which are contained in Ξ_c after its execution. A formal definition is given in Algorithm 4.1.

(1) $\Xi_{u,0}$ is defined by $[0, T_s - d_s] \leftarrow d_a$, Ξ_c by $[T_s - d_s, T_s] \leftarrow d_a$

(2) For each process order $i \in \{0, 1\}$, we compute the appropriate value of γ, as described in Section 2. For order-0 processes, we set $\gamma = T_a$.

(3) Then, for each process order $i \in \{0, 1\}$, we iterate over all unprocessed partitions. Each examined partition is removed from $\Xi_{u,i}$ and is further subdivided into multiple processed or unprocessed partitions, which are added either to $\Xi_{u,i+1}$ or Ξ_c. This is done by the *processPartition()*-function, which is described below. All newly added partitions of $\Xi_{u,i+1}$ are processed in the course of the next order.

(4) If $\Xi_{u,i+1}$ becomes empty, the algorithm terminates.

Algorithm 1 Algorithm for computing Ξ_c

Require: T_a, d_a, T_s, d_s
Ensure: Ξc
1: $\Xi_{u,0} : [0, T_s - d_s] \leftarrow d_a; \Xi_c : [T_s - d_s, T_s] \leftarrow d_a$
2: Compute $\vec{\gamma} = \gamma_0, \gamma_1, \ldots, \gamma_n$
3: **for** i from 0 to 1 **do**
4: **for each** partition $n \in \Xi_{u,i}$ **do**
5: Remove partition n from $\Xi_{u,i}$
6: $(\Xi_{u,i+1}, \Xi_c) = processPartition(n, \Xi_{u,i}, \Xi_c)$
7: **end for**
8: **end for**

The function *processPartition()* is called for every single partition of $\Xi_{u,i}$ once per process order $i \in \{0, 1\}$. After a partition has been processed by this function, this partition is removed from $\Xi_{u,i}$. However, new partitions are added both to $\Xi_{u,i+1}$ and Ξ_c. In the following, we describe the concept behind this function for growing processes (i.e., $\Phi[k]$ becomes larger for growing indices k). Shrinking processes can be handled by a similar scheme.

Based on the borders l and r of the partition to be processed, we compute a lower number N_l and upper number N_u of γ-intervals that fit into this partition. It is

$$N_l = \left\lceil \frac{T_s - d_s - r}{\gamma} \right\rceil, \quad N_u = \left\lfloor \frac{T_s - d_s - l}{\gamma} \right\rfloor. \tag{7}$$

Further, we define d_{nl} as the absolute distance between $N_l \cdot \gamma$ and the right partition border, and d_{nu} as the distance between the left border and $N_u \cdot \gamma$. Due to space constraints, we restrict our descriptions to the most general case of $N_l < N_u$. Special cases can be handled similarly. Figure 4 depicts the basic concept for processing each partition. First, the time period between N_l and N_u is separated into parts in which the current process hits the scan interval (depicted by h in the figure) after a certain number of γ- or T_a-intervals, and the remaining, missing parts (depicted by m in the figure). As already described for shrinking or increasing a temporal distance by γ time-units, a certain multiple of T_a-intervals need to pass. For example, for an order-1-process, if $T_a < T_s$, $\sigma = \lceil \frac{T_s}{T_a} \rceil \cdot T_a$ time-units take place until the temporal distance between an advertising and a scan window is reduced by γ time-units. Following this scheme, the amount of time until reaching the next scan window or the next partition of a lower-order process is computed and added to the latency of this partition. The latency of the partition, before this new addition takes place, is the sum of latencies for all previously processed, lower-order processes. For processes of order 0, the initial latency is d_a time-units (since the latency of one packet-transmission d_a occurs in all cases). For each hitting part (marked with h in Figure 4), a new partition is added to Ξ_c. With d being the existing latency of the partition being currently processed, the latencies for these newly-added partitions are $d + (N_l + 1) \cdot \sigma, (N_l + 2) \cdot \sigma, \ldots, N_u \cdot \sigma$ time-units. Similarly, for each missing part (indicated with m in the figure), a new partition $j \in [N_l + 1, N_u]$ is added to $\Xi_{u,i+1}$. The latencies d_j for such newly added partitions j in $\Xi_{u,i+1}$ are

$$d_j = \begin{cases} d + j \cdot \sigma & , if \quad m_{i+1} = g \\ d + (j+1) \cdot \sigma & , if \quad m_{i+1} = s. \end{cases} \tag{8}$$

$m_{i+1} \in \{s, g\}$ indicates whether the next higher-order process is growing (g) or shrinking (s). For the highest-order process, no non-matching areas exist. The time periods within d_{nl} and d_{nu} are processed similarly.

After the completion of Algorithm 1, $\Xi_{u,n}$ is empty and Ξ_c covers the whole range of possible initial offsets $\Phi[0]$. If all special cases are accounted for, Ξ_c represents the one-way discovery latencies for all possible initial offsets of such a periodic-interval-based protocol.

Figure 5 depicts a complete set of partitions for our proposed protocol, with a target duty-cycle of 10 % and a value of $R = 4$. The upper number in each partition depicts the number of advertising intervals until successful discovery, whereas the lower number depicts the highest process order that occurred. As can be seen, parametrizations as described in Section 3.1.2 lead to a distribution of partitions with equal lengths. As already mentioned, only orders of 0 and 1 are present. We exploit this for computing the latencies for assisted two-way discovery, as described below.

4.2 CDF, Mean and Maximum Discovery Latencies

Based on Ξ_c, we attempt to derive the cumulative distribution functions (CDFs).

4.2.1 One-Way Discovery. To compute the CDF for one-way discoveries, we assign the probability $p_j = \frac{(r[j] - l[j])}{T_s}$ to each partition j, corresponding to its fraction of time in the set of initial offsets.

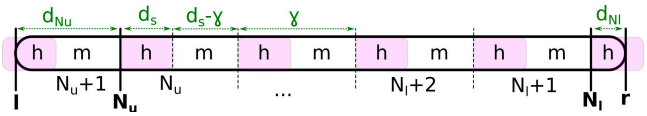

Figure 4: Processing a single partition.

[2] Actually, this algorithm also works for the generic case of higher-order processes, which are not relevant for this paper.

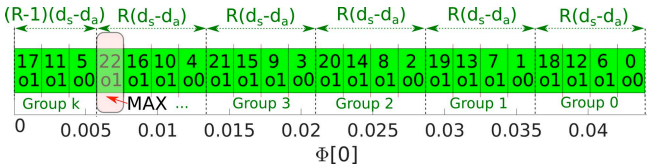

Figure 5: Set of partitions for $\eta = 10\,\%, \mathbf{R} = 4$

The CDF describes the probability $P(d < d_{th})$ of all latencies d being smaller than a certain threshold d_{th}. One can iterate over all partitions j in Ξ_c to find all partitions with latencies below or equal to $d_{th} = d_j$. The probability $P(d < d_j)$ is the sum $\sum p_j$ of these partitions. If this is carried out repeatedly for all $j \in \Xi_c$, the CDF for one-way discovery is obtained. We denote the resulting CDF as P_{1W}.

4.2.2 Two-Way Discovery. The CDF for non-assisted two-way discovery can be derived from P_{1W} as follows. Since both device 1 and device 2 run the same probabilistic process independently from each other, the two-way discovery latency is the maximum of two individually carried out one-way discovery processes. For the two-way CDF P_{2W}, it is therefore

$$P_{2W}(d < d_j) = P_{1W}(d < d_j) \cdot P_{1W}(d < d_j) \forall j \in \Xi_c. \quad (9)$$

4.2.3 Assisted Two-Way Discovery. The CDF for assisted two-way discovery can be computed as follows. Given a number of N_f partitions in Ξ_c, a new set of partitions Ξ_c^* is defined by concatenating N_f periodic repetitions of Ξ_c. A second set of partitions $\Xi_{c,sh}^*$ is defined by concatenating N_f periodic repetitions of Ξ_c, in which the partitions of every repetition $r \in [0, N_f]$ are cyclically shifted by $r \cdot (d_s - d_a)$ time-units to the right. For each pair of partitions in Ξ_c^* and $\Xi_{c,sh}^*$, the minimum latency is derived. The resulting distribution of minimum latencies is denoted by Ξ_{A2W}^*. Clearly, the scan windows of both devices have a random, uniformly distributed temporal distance from each other, which is determined by the random point in time at which both devices come into range. Since the shifted repetition $\Xi_{c,sh}^*$ considers all possible temporal permutations against Ξ_c^* and since always the partition having a shorter latency is chosen, Ξ_{A2W}^* now contains the partitions for the first rendezvous in an A2W protocol. An additional penalty $d_{p,A2W}$ has to be added to each partition of Ξ_{A2W}^* to account for the time until the second discovery takes place. We add $r \cdot (d_s - d_a)$ time-units to the corresponding partition, if the partition from Ξ_c^* has a smaller latency than the one from $\Xi_{c,sh}^*$, or $(N_f - r) \cdot (d_s - d_a)$ time-units, otherwise. This defines the final distribution Ξ_{A2W} for assisted two-way discovery. From Ξ_{A2W}, the CDF can be computed equivalently to P_{1W}.

From these considerations, one can derive a simple equation which approximates P_{A2W} in close proximity. When regarding the assisted two-way discovery procedure as the same probabilistic process being carried out two times in parallel, the instance with the shorter latency determines the overall latency. For the first rendezvous, it is

$$P_{A2W}^{1st}(d < D) = \begin{aligned} & 2 \cdot P_{1W}(d < d_j) \cdot (1 - P_{1W}(d < d_j)) \\ & + P_{1W}(d < d_j)^2 \; \forall j \in \Xi_c, \end{aligned} \quad (10)$$

since for a match, either device 1 needs to have a latency smaller or equal than $d_{th} = d_j$, only, or device 2, or both devices. Due to the A2W mechanism, the second rendezvous happens within T_s time units from the first one. To approximate the delay between the two successful receptions, we add $\frac{1}{2}T_s$ to all partitions of Ξ_c and obtain an approximate CDF P'_{A2W}.

4.3 Worst-Case and Mean Latencies

We could derive the worst-case and mean discovery latencies as the maximum/mean values in Ξ_c, obtained from Algorithm 1. However, for mathematical optimizations of the protocol, a closed-form formulation is needed. The worst-case latencies for device 1 discovering device 2 and vice-versa are identical to the two-way discovery latencies. They can be derived as follows. When applying parametrizations as described in Section 3.1.2, one can observe that the set of partitions Ξ_c for one-way discovery always has the following structure (cf. Figure 5).

- From Equation 2 follows that Ξ_c consists of M partitions of width $d_s - d_a$.
- From Equations 3 and $R = \frac{M+1}{k}$, it follows that there are $k = \frac{M+1}{R}$ partitions in which a match of the order-0 process occurs (cf. Figure 5).
- It follows that the remaining $M - k$ partitions are related to order-1 -processes. Since every two subsequent order-0-partitions have a temporal distance of $T_a = \frac{M+1}{k}(d_s - d_a)$ time-units from each other, it follows that there are $\frac{M+1}{k} - 1 = R - 1$ order-1-partitions temporally left of each order-0-partition (except for the leftmost one, see below).
- Since we have required $M + 1$ to be divisible by R, M is in general not divisible by R. It is $k \cdot R = M + 1$. Therefore, the leftmost order-0-partition has only $R - 2$ neighboring order-1-partitions. In other words, as shown in Figure 5, there are $k - 1$ groups of R partitions with $R - 1$ order-1-partitions and one order-0-partition in each of them. In addition, there is one group at the leftmost part of Ξ_c with $R - 2$ order-1-partitions and 1 order-0-partition.

One needs to recall that for shrinking a temporal distance by γ time-units, $(M + 1) \cdot T_a$ time-units of discovery latency are induced. Therefore, the maximum latency can only be reached from partitions in which the largest number of γ-intervals have to pass. This is fulfilled by the leftmost partition of each group (except for the leftmost partition in the shortened group with only $R - 1$ partitions, which is located at the left of Ξ_c). Among these leftmost partitions, the largest latency is achieved by the group with the largest number of T_a-intervals until reaching the scan window. This is the group $k - 1$, as highlighted with "MAX" in Figure 5. It follows that the worst-case discovery latency is

$$d_m = ((R - 1)k + k - 2)(R(d_s - d_a)) + d_a. \quad (11)$$

Following similar considerations, the mean one-way discovery latency can be derived as

$$\overline{d_{1W}} = \frac{1}{2} \cdot \frac{1}{k}(M - 1)(d_s - d_a)(M + 1) + d_a. \quad (12)$$

One can also observe from Figure 5 that each partition of Ξ_c has its unique latency - no latency occurs more than once. From the elaborations above, it becomes clear that all latencies from $0 \cdot T_a + d_a$ to $(M - 1) \cdot T_a + d_a$ time-units occur, and each partition

has a probability of $1/M$. Therefore, the CDF function for the one-way discovery can also be written as

$$P_{1W}(d < d_j) = \sum_{k=0}^{j} \frac{1}{M}, j \in [0, M-1], \tag{13}$$

with $d_j = j \cdot T_a + d_a$. Similarly, for the non-assisted two-way-discovery, it is:

$$P_{2W}(d < d_j) = \left(\sum_{k=0}^{j} \frac{1}{M} \right)^2, \ j \in [0, M-1]. \tag{14}$$

With Equations 13 and 10, analytical formulations of the CDF for 2W- and A2W-discovery are available. For any random variable X, the expected value can be expressed as $E(x) = \int_0^\infty (1 - P(x < X))dX$. Hence, for the non-assisted two-way discovery, the mean latency is

$$\begin{aligned} \overline{d_{2W}} &= d_a + R(d_s - d_a) \sum_{k=0}^{M-1} \left(1 - \sum_{l=0}^{k} \frac{1}{M} \right)^2 \\ &= \frac{1}{6} R(4M - \frac{1}{M} - 3)(d_s - d_a) + d_a. \end{aligned} \tag{15}$$

Similarly, one can derive the mean latency for A2W-discovery using Equation 10:

$$\begin{aligned} \overline{d_{A2W}} = & d_a + d_p + R(d_s - d_a) \cdot \sum_{k=0}^{M-1} \Bigg(1 - 2 \Bigg(\sum_{l=0}^{k} \left(\frac{1}{M} \right) \cdot \\ & \left(1 - \sum_{l=0}^{k} \left(\frac{1}{M} \right) \right) + \sum_{l=0}^{k} \left(\frac{1}{M} \right)^2 \Bigg) \Bigg). \end{aligned} \tag{16}$$

In this equation, d_p is the additional time penalty for the second discovery after the first rendezvous has taken place. When assuming that all offsets between two neighboring scan intervals occur with the same likelihood, d_p can be approximated with $\frac{M}{2}(d_s - d_a)$. Therefore, the mean A2W-latency becomes approximately

$$\overline{d_{A2W}} \approx d_a + (d_s - d_a) \left(\frac{M}{2} + \frac{R(2M^2 - 3M + 1)}{6M} \right). \tag{17}$$

5 EVALUATION

In this section, we first evaluate the behavior of our proposed protocol for different values of R and different duty-cycles η. Next, we compare its performance to previously known protocols.

The discovery latencies presented in this section have been computed using the theory described above (i.e., by applying the explicit model). We have assumed $d_0 = 0$ and a lower limit of the scan window $d_{s,m} = 10 \cdot d_a$, as already explained. We assume a packet length of 3 bytes, which is the minimum packet length of our proposed protocol: Radios typically need a 1-byte preamble. In addition, we need to send 2 bytes for mutual assistance hints. Additional data can be transferred upon success without any significant delay: If a node receives an A2W-hint having the value 0 s (which indicates that the packet is an A2W packet), the receiving node can schedule an additional packet immediately afterwards, whereas the sending node could listen right after the transmission of the additional A2W packet. This packet can be of arbitrary length.

Even though other duty-cycles can be realized with our proposed protocol, too, we restrict our evaluations to duty-cycles $\eta \in [0.13\,\%, 1.46\,\%]$ in steps of $0.01\,\%$, since this covers a practical range of worst-case latencies: The value $\eta = 0.13\,\%$ corresponds to worst-case latencies of approximately 1 min, whereas $\eta = 1.46\,\%$ corresponds to worst-case latencies of 0.5 s.

5.1 Discovery Latencies

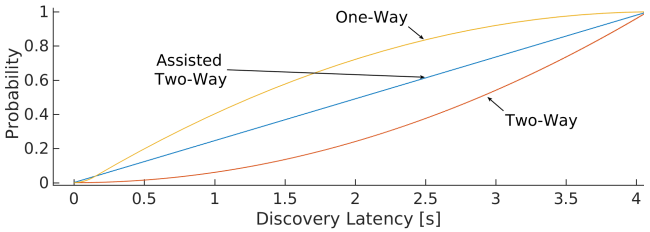

Figure 6: CDF for R = 26, $\eta = 0.5\,\%$

5.1.1 Effect of A2W Discovery. Figure 6 depicts the CDF of our proposed protocol for $\eta = 0.5\,\%$ and $R = 26$. The three depicted curves show the CDFs for one-way, two-way and mutually assisted two-way discovery. All three curves reach a probability of 100 % after the same latency, which means that the worst-case latency is equal for one-way, two-way and assisted two-way discovery. However, as can be seen, the CDF for two-way discovery is convex, indicating a lower mean discovery latency than half the maximum one. In contrast, the CDF for the assisted two-way discovery has a concave shape, which means that the discovery can be performed faster in the average case. However, the speedup depends on the value of R, which is evaluated next.

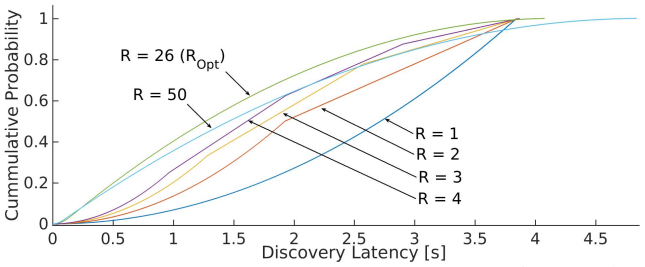

Figure 7: CDF for assisted two-way discovery ($\eta = 0.5\,\%$)

5.1.2 Cumulative Distribution Functions. Figure 7 depicts multiple CDFs for A2W discovery with different values of R. As can be seen, for a value of $R = 1$, the A2W CDF is convex. In fact, it is identical to the non-assisted two-way discovery curve. When increasing R, the CDF becomes increasingly concave, leading to shorter mean latencies. However, there is an optimal value of R, at which the mean latency becomes minimal. It is referred to as R_{Opt}. Increasing R beyond R_{Opt} does not decrease the mean latency anymore, which can also be observed in the Figure for $R = 50 > R_{Opt}$. R_{Opt} depends on the duty-cycle. It can be computed (approximately) by minimizing the product $\eta \cdot d_{mean}$ using Equations 5 and 17. One can also observe that larger values than R_{Opt} increase the worst-case latency. Therefore, $R \in [1, R_{Opt}]$ can be used to trade-off worst-case against mean latencies. We study this trade-off more thoroughly, next.

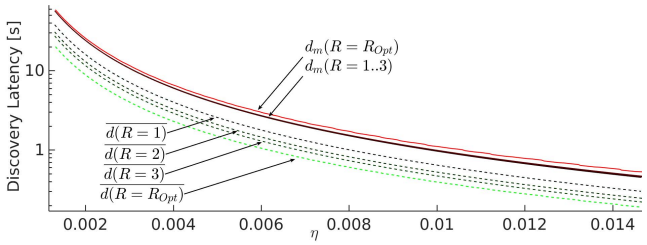

Figure 8: Maximum and mean discovery latencies for different values of R

5.1.3 Worst-case and Mean Latencies. Figure 8 depicts the discovery latencies for sweeping target duty-cycles of $\eta \in [0.13\,\%, 1.46\,\%]$ in steps of 0.01 %. Multiple curves for different values of R are depicted. In addition, there is one curve in which R has been set to $R_{Opt}(\eta)$ for every given duty-cycle. As can be seen, the maximum latencies are only marginally affected by the parameter R, since all maximum latency curves (d_m) nearly lie on each-other. However, the mean latencies are greatly reduced for all duty-cycles. Growing values of R lead to larger reductions of the mean latency $\overline{d}$, as it can also be seen from the CDF.

As already explained, one can choose R freely within 1 and R_{Opt}. $R = 1$ results in the lowest worst-case latency, whereas $R = R_{Opt}$ results in the lowest mean latency. For $\eta = 0.5\,\%$, Table 1 quantifies this trade-off.

R	1	2	3	4	5	6	7	8	9	10	R_{Opt}
G_m [%]	0	0	-1	-1	-2	-2	-2	-3	-3	-4	-7
$\overline{G}$ [%]	0	19	28	32	35	37	39	40	41	41	43

Table 1: Gain of max. (d_m) and mean ($\overline{d}$) latencies for different values of R over the latencies for R = 1

It depicts the gain of mean latencies $\overline{G} = (\overline{d(1)} - \overline{d(R)}/\overline{d(1)}$ for different values of R against the mean latencies for $R = 1$ averaged over all duty-cycles, and the according gains of the maximum latencies G_m, respectively. In other words, it shows how much the worst-case latency is increased for a certain reduction of the mean latency. Evidently, the largest reduction of $\overline{d}$ on average over all examined duty-cycles is 43 %, thereby increasing the worst-case latency d_m by 7 %. Hence, mean latencies can be traded against worst-case latencies by choosing the value of $R \in [1, R_{Opt}]$.

5.2 Comparison Against other Protocols

We have shown that, for given values of R, A2W discovery can greatly reduce the mean latency, without affecting the worst-case latency. In what follows, we compare the performance of Griassdi against multiple existing protocols. We have included both slotted and slotless protocols into this comparison. In particular, we have considered the following protocols.

- **DISCO** [2], one of the first slotted protocols.
- **Searchlight-Striped** [1], since it is often considered as one of the best-performing symmetric slotted protocols. We assume a slot-overflow δ of 0, indicating that its predicted performance is slightly better than in practice.
- **U-Connect** [3], which is frequently included in performance comparisons.
- **G-Nihao** [9], a pseudo-slotted protocol. We assume $\gamma = 2$, as also assumed in [9].
- **PI-kM$^+$-Opt** [5], since it is, to the best of our knowledge, the protocol with the lowest worst-case latency. We assume all parameters to have the values proposed in [5] (i.e., $\epsilon = \frac{1}{32768}, \epsilon_{Ta} = 0$), except for the minimum listening duration $d_{s,l}$. As for our proposed protocol, we assume $d_{s,l} = 10 \cdot d_a$.

Slotted protocols perform best using short slot-lengths. However, since a packet is sent at the beginning and the end of each slot, discovery fails if these packets overlap. Two slots can overlap in time and hence realize discovery if their temporal offset is within $[-d_s, d_s]$. When considering all possible constellations in this interval, there are three sub-intervals of length $2 \cdot d_a$ within $[-d_s, d_s]$, at which the packets of two nodes collide. Hence, the probability that two nodes fail to discover each other is $P_f = \frac{3 \cdot 2 \cdot d_a}{2 \cdot d_s}$. We assume that the probability of two nodes failing to discover each other should be below 10 %, which results in a minimal slot-length of $30 \cdot d_a$. We assume this slot-length for all slotted protocols except U-Connect, which applies a special slot design. This design cannot be used with other schedules, and allows for shorter slot-lengths. Despite this scheme might have additional drawbacks (the hardware has to support clear channel assessments and noise might wakeup the radio), we assume a slot-length of 250 μs for U-Connect, as presented in [3]. Since the lowest possible slot-length for G-Nihao is not clear, we assume a slot-length of $30 \cdot d_a$ for G-Nihao, too, which is shorter than the 10 ms-slots assumed in [9]. Further, all protocols considered perform best for short packet lengths, except U-Connect, for which the packet-length has negligible impact. For determining the minimal required packet length, we assume that the radio needs a one-byte preamble in addition to the payload. For all previously known protocols, we assume that the payload consists of a two-byte node ID, as in [3]. For our proposed protocol, we assume a two-byte payload, as already described. With these considerations, we assume a packet length of 3 bytes for all protocols. We have computed the worst-case latencies of the protocols under consideration according to the equations in Table 2.

Protocol	$d_m(\eta)$
Disco	$\frac{4}{\eta^2} d_{sl}$
U-Connect	$(\sqrt{\frac{1}{2\eta} + \frac{9}{16\eta^2}} + \frac{3}{4\eta})^2 d_{sl}$
Searchlight	$\lceil \frac{1}{2} \cdot \lfloor \frac{1}{\eta} \rfloor \rceil d_{sl}$
G-Nihao	$\left(\frac{d_{sl} + d_a\gamma}{2\gamma\eta d_{sl}} + \sqrt{\frac{d_{sl} + d_a\gamma}{2\gamma\eta d_{sl}} - \frac{d_a}{d_{sl}}} \right)^2 \gamma$
PI-kM$^+$-Opt	According to model algorithm [5]

Table 2: Worst-case discovery latencies of known protocols [2], [3], [1], [5], [9]

5.2.1 Worst-Case Latencies. Figure 9 depicts the computed worst-case latencies for duty-cycles between 0.13 % to 1.46 % in steps of 0.01 % with $R = R_{Opt}$. As can be seen, the worst-case latencies of the PI-kM$^+$-Opt-protocol and our proposed protocol lie almost on the same curve. The worst-case latencies of all other protocols in this comparison are significantly larger. Table 3 depicts the worst-case latency gains of our proposed protocol, defined as $(d_m - d_{m,Griasdi})/d_m)$, with d_m being the worst-case latency of the protocol under consideration and $d_{m,Griassdi}$ the worst-case latency of our proposed protocol. Column 1 and 3 depict the mean

gains over the range of duty-cycles considered, columns 2 and 3 the maximum gains. The first two columns show the gains for $R = 1$, the last two ones show the values for $R = R_{Opt}$. For $R = R_{Opt}$, the worst-case performance of our proposed protocol is slightly reduced compared to $PI-kM^+-Opt$, which has the lowest relation of duty-cycles and worst-case latencies among all known protocols. For smaller values of R, the worst-case performance of both protocols become identical, but no mean latency gains due to mutual assistance can be achieved.

	$R = 1$		$R = R_{Opt}$	
Protocol	Mean	Max	Mean	Max
Disco	0.97	0.97	0.96	0.97
U-Connect	0.83	0.83	0.82	0.82
Searchlight-S	0.87	0.87	0.86	0.86
G-Nihao	0.77	0.77	0.75	0.76
PI-kM$^+$-Opt	0.0	0.0	-0.07	-0.10

Table 3: Gain of worst-case latencies of Griassdi over different protocols

5.2.2 Mean Latencies. Unlike for the worst-case latencies, there are no equations that cover the average behavior of the slotted protocols proposed in the literature [2], [3], [1], [9], [5]. Therefore, we have implemented simulation models for all slotted protocols under consideration. The simulations have been repeated 100, 000× for each slotted protocol, with random offsets in each run. For the PI-kM$^+$-Opt-protocol, the model presented in [4] has been used.

Figure 10 shows the simulated CDF for a duty-cycle of 1.4 %. A duty-cycle of 1.4 % translates to the following protocol-specific parameters: Disco: $p_1 = 137$ and $p_2 = 149$; U-Connect: $p = 107$; G-Nihao: $m = 38$ and $n = 76$. For our proposed protocol, R has been assumed to be R_{Opt}. Clearly, our proposed protocol achieves the lowest mean latency of all protocols under consideration. In particular, the mean latencies for $\eta = 1.4\,\%$ and - in parenthesis - the mean latency gains $(\overline{d} - \overline{d_{Griassdi}})/\overline{d_{Griassdi}}$, are as follows. Disco 4.84 s (96 %), U-Connect 1.42 s (86 %), Searchlight-P 3.59 s

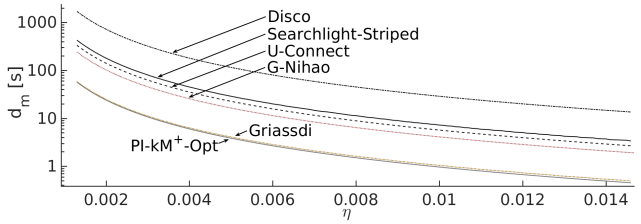

Figure 9: Worst-case latencies d_m of the Griassdi protocol ($R = R_{Opt}$) against other protocols

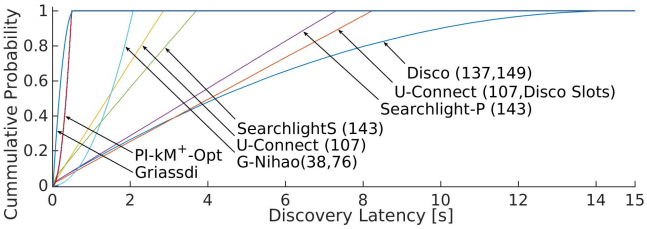

Figure 10: CDF of Griassdi compared to existing protocols for $\eta = 1.4\,\%$, $R = R_{Opt}$

(95 %), Searchlight-S 1.78 s (89 %), G-Nihao 1.39 s (86 %), PI-kM$^+$-Opt 0.33 s (40 %), Griassdi 0.20 s. Compared to the best-performing non-assisted slotless protocol PI-KM$^+$-Opt [5], the reduction of the mean latency by 40 % has a significant impact on practical implementations, since it can be used to reduce the duty-cycle, which allows for increased battery runtimes.

5.2.3 Channel Utilization and Collisions. The channel utilization is defined as the percentage of time packets are being transmitted on the channel. For our proposed protocol, it is $\frac{d_a}{T_a}$. Figure 11 depicts the channel utilization caused by a single device. As can be seen, the channel-utilizations of both our proposed protocol and the PI-kM$^+$-Opt-protocol are larger than the channel utilizations of the slotted ones. However, for the largest duty-cycle considered, it only reaches about 0.7 %.

High channel utilizations increase the probability of packet collisions, which potentially deteriorates the performance of our proposed protocol, especially in busy environments. There are two types of collisions:

Direct collisions: First, there are collisions between the regular packets sent with periodic intervals of multiple nodes. Given the low channel utilizations, these collisions are relatively infrequent. However, if one pair of regular packets collides, all others will also collide, given their periodic nature.

A2W collisions: Second, there are collisions between multiple additional packets sent for mutual assistance, or collisions of A2W packets with regular ones. These collisions have a smaller impact on the discovery latency, since future regular packets can still match a scan window after an A2W packet has collided. However, they appear more frequently. To minimize such collisions, it is not beneficial to send A2W packets within the center of a remote device's scan window, but to randomly distribute them over the whole scan interval $[0, d_s - d_a]$. The collision probability is also influenced by the parameter R, which determines the number of scan intervals after which a guaranteed discovery occurs. Larger values of R reduce the probability of A2W collisions, because there are more scan windows an A2W packet can be sent within.

Collisions have not been accounted for in our theoretical analysis. To study the collision properties, we in what follows present the results of a simulation experiment. We assume $R = R_{opt}$ and further that A2W packets are distributed randomly over a remote device's scan window. Further, we require a minimum scan window of $d_{s,m} = 10 \cdot d_a$. We consider a scenario in which n nodes are constantly within range and in discovery mode. We simulate for a duration of 2× the worst-case latency. For every duty-cycle, the simulation has been repeated for 10, 000 times. Figure 12 depicts the fraction of discovery-procedures which did not terminate successfully within 2× the worst-case latency. This is a metric for direct

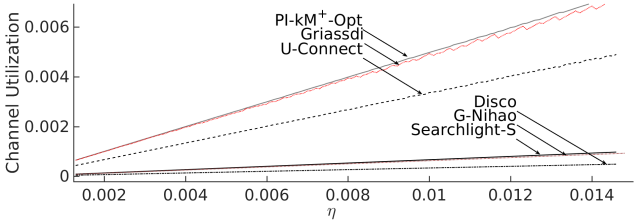

Figure 11: Channel Utilizations

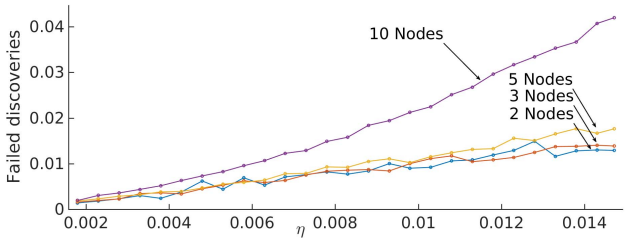

Figure 12: Fraction of discovery procedures which did not terminate after $2 \times d_{nd,m}$

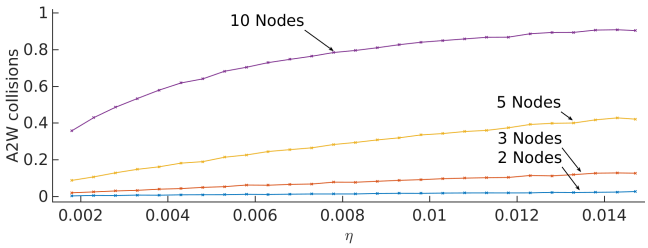

Figure 13: Fraction of discovery procedures with collided A2W packets

collisions. As can be seen, the collision rate is below 5 % for all numbers of nodes studied.

Figure 13 depicts the fraction of discoveries with at least one collided A2W packet. As can be seen, the rate of A2W collisions is significantly higher than for direct collisions, exceeding 80 % for 10 nodes. It needs to be mentioned that the rate of A2W collisions is influenced by the minimum scan window length (here: $10 \cdot d_a$). If the $d_{s,m}$ is doubled, the rate of A2W collisions is approximately halved. However, increasing $d_{s,m}$ reduces the latency-duty-cycle-performance. Finding the optimum value of $d_{s,m}$ to maximize the mean latency for n nodes considering collisions is left for future research. In summary, our proposed protocol has low primary collision rates, but the speedup achieved by mutual assistance is reduced increasingly for growing numbers of nodes.

6 IMPLEMENTATION

6.1 Setup

We have implemented our proposed protocol on two Nordic NRF51822 [7] radios. Based on a modified version of the open-source BLE stack *blessed* [8], we realized the A2W protocol described in this paper on two PCA10005 evaluation boards [6]. Each board has been connected to a laptop using an UART-to-USB adapter. To ensure a proper functioning and to compensate for clock skew, we have extended each scan window by $d_0 = 80\,\mu$s, inducing a duty-cycle error of up to 2 %. The packet length d_a has been 368 μs.

The mechanism for assisted two-way discovery has been realized as follows. Whenever a device receives a packet containing a wakeup hint, it schedules three additional packets around the point in time received. One of them is scheduled to begin at the exact received point in time, whereas the other two are sent 1 ms before and after it. This is done to minimize the number of collisions between the additional A2W-packets and the regular packets of the remote device. No regular packets are sent after a hint has been received from a remote device, until the additional packets have been sent entirely. Afterwards, the device returns to its original schedule. Figure 14 depicts the measured current consumption of a successful discovery procedure. We have used the built-in shunting resistors and measurement amplifiers of the PCA10005 board for this measurement. For a better readability, we have shifted the origin for device 2 by 5 mA. The broader peaks are related to scanning, whereas the smaller ones indicate the transmission of a packet. As can be seen, after device 2 receives a regular packet from device 1, which includes a hint on the next wakeup time of device 1, it schedules three additional packets within the wakeup period of device 1. After that, mutual discovery is complete.

6.2 Experimental Data

The theory presented in this paper indicates a high efficiency of our proposed protocol. To evaluate the validity and accuracy of this theory and to show that this performance is reached in practice, we have conducted a comprehensive set of experiments. In each experiment, the laptop has sent start- and stop-commands to the two radios via the USB/UART connection. The laptop has measured the amounts of time after which the discovery has occurred. The radios have realized each duty-cycle $\eta \in [0.5\,\text{s}, 5.5\,\text{s}]$ in steps of 0.1 %. For each of these duty-cycles, we have carried out 200 two-way discoveries, consisting of 400 one-way discoveries. In total, we thereby measured the latencies of 20, 400 discovery-procedures. In particular, for each discovery-experiment between the two devices, the following steps have been carried out.

(1) Device 1 starts scanning at a point in time t_0.
(2) After a random amount of up to T_s time-units from t_0, device 2 starts scanning.
(3) At $t_0 + 2 \cdot T_s + \rho_{a,1}$, with $\rho_{a,1}$ being a random amount of time between 0 and T_a time-units, device 1 starts advertising.
(4) At $t_0 + 2 \cdot T_s + \rho_{a,2}$ time-units, device 2 starts advertising. $\rho_{a,2}$ is a random amount of time within $[0, T_a]$.

The discovery latencies have been measured relatively to the points in time at which the first advertising packet has been sent, since this corresponds to the point in time at which two devices would come into range for the first time. All random amounts of time were distributed uniformly. The measurements have been aborted if two-way discovery could not been achieved within 70 s, which is longer than the predicted upper bound for all duty-cycles.

6.2.1 Worst-Case Latencies. The solid line in Figure 15 depicts the modeled worst-case latencies for assisted two-way discovery. It has been computed using the theory presented in Section 4. Each circle depicts the measured A2W latency of one experiment. As can be seen, the predicted upper bound is a very accurate estimate of the worst-case latency. The measured points at the top of the figure, with latencies around 70 s, depict collided measurements. In such cases, both devices scheduled all of their advertising packets at overlapping points in time. In addition, especially for larger duty-cycles,

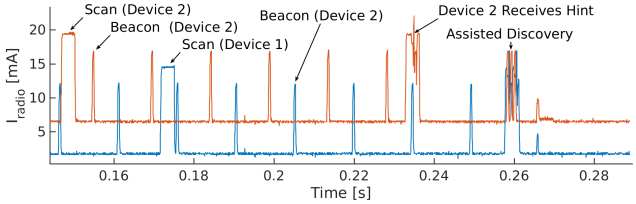

Figure 14: Current measurement during A2W discovery

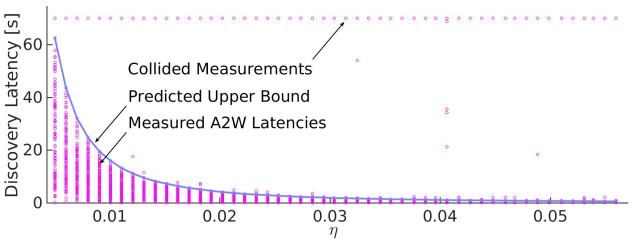

Figure 15: Fit of the model for worst-case latencies

some measurements exceeded the predicted worst-case latency, but did not reach the maximum latency of 70 s. We observed that in some cases, colliding series of packets were subjected to clock skew and therefore stopped colliding before reaching 70 s. Other reasons might be collisions of A2W packets and implementation issues. The total (direct and A2W) rate of all measurements exceeding the predicted worst-case bound has been 7.3 %, which is mainly caused by frequent collisions (due to the large packet length). The total number of one-way measurements which exceeded the predicted bound but did not reach 70 s has been 145 (0.7 %). It needs to be mentioned that collisions are not a problem specific to our proposed protocol. They have been reported for slotted [3], pseudo-slotted [9] and slotless [5] protocols, too. The A2W mechanism has failed 45 times in total. In the rest of this section, we discard all measurements exceeding the predicted upper bound by more than 1 %, in order to evaluate the statistical properties of the non-collided discoveries.

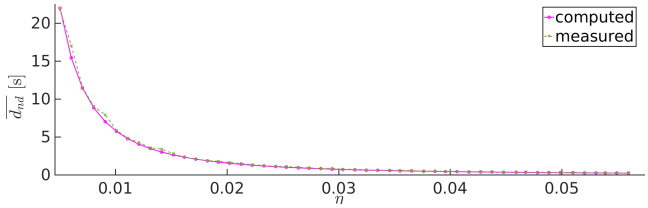

Figure 16: Fit of the model for mean latencies

6.2.2 Mean Latencies. Figure 16 depicts the measured mean latencies for A2W discovery. In addition, the curve predicted by our proposed theory is shown. Collisions have been excluded. As can be seen, the predicted mean latencies are very accurate estimates for the experimental ones. The NRMSE considering all (non-discarded) measurements has been 1.3 %

6.2.3 Statistical Behavior. Our proposed theory cannot only model the overall behavior, such as mean- and worst-case latencies, but can also accurately describe the complete statistical behavior of our proposed protocol. Figure 17 depicts the modeled CDFs (solid lines) together with the measured distributions (dotted lines) for multiple duty-cycles. Measurements exceeding the predicted upper bound d_m by more than 1 % have been discarded, as already described. As can be seen, our predictions lie in close proximity with the experimental data. Overall, our experiments fully confirm our theory. They show that our proposed protocol can achieve the high predicted performances in practice. The low remaining errors are caused by the loopback latency of the USB connection, the granularity of the random amounts of time that form the initial offsets, the compensation measures against clock inaccuracies and, in some

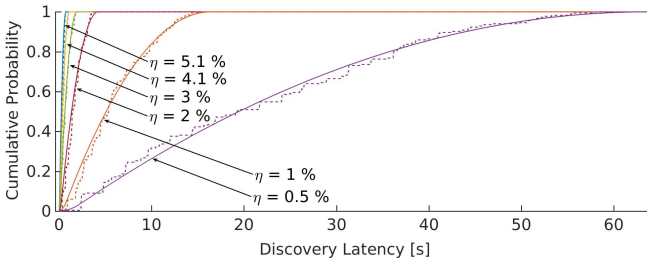

Figure 17: Computed (solid) and measured (dotted) CDFs

rare cases, failed A2W discoveries (e.g., caused by collisions), or implementation issues.

7 CONCLUDING REMARKS

We have presented a technique that exploits mutual assistance to improve the mean latencies for two-way discovery. Based on this technique, we have proposed the protocol Griassdi, which achieves similar worst-case latencies to the best known discovery protocols, but provides up to 43 % shorter mean latencies. Therefore, our proposed protocol is - to the best of our knowledge - the one with the lowest latency-duty-cycle relation of all known ones. In future work, it seems promising to design protocols which exploit higher-order-processes to reach even shorter mean latencies. Since more scan intervals could be skipped by mutual assistance using such processes, the mean latencies could be further reduced.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their helpful comments and Prof. Tian He for shepherding this paper. This work was partially supported by *Sound4All* - a project funded by the federal ministry of education and research of Germany (BMBF) and by the Marie Curie program *oCPS* under the EU grant agreement number 674875.

REFERENCES

[1] M. Bakht, M. Trower, and R.H. Kravets. 2012. Searchlight: won't you be my neighbor?. In *Annual International Conference on Mobile Computing and Networking (MOBICOM)*.
[2] P. Dutta and D. Culler. 2008. Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications. In *ACM Conference on Embedded Network Sensor Systems (SenSys)*.
[3] A. Kandhalu, K. Lakshmanan, and R. Rajkumar. 2010. U-Connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *International Conference on Information Processingin Sensor Networks (IPSN)*.
[4] P.H. Kindt, M. Saur, and S. Chakraborty. 2015. Neighbor discovery latency in BLE-like duty-cycled protocols. *CoRR* abs/1509.04366 (2015).
[5] P.H. Kindt, M. Saur, and S. Chakraborty. 2016. Slotless Protocols for Fast and Energy-Efficient Neighbor Discovery. *CoRR* abs/1605.05614 (2016).
[6] Nordic Semiconductor ASA. 2013. nRF51822 Development Kit User Guide v 1.3. (July 2013). available via nordicsemi.com.
[7] Nordic Semiconductor ASA. 2014. nRF51822 Product Spec. v3.1. (2014). Available via nordicsemi.com.
[8] P. Borges et Al. 2015. Bluetooth Low Energy Software Stack for Embedded Devices (BLESSED). (2015). https://github.com/pauloborges/blessed.
[9] Y. Qiu, S. Li, X. Xu, and Z. Li. 2016. Talk More Listen Less: Energy-Efficient Neighbor Discovery in Wireless Sensor Networks. In *IEEE Conference on Computer Communications (INFOCOM)*.
[10] L. Wei, B. Zhou, X. Ma, D. Chen, J. Zhang, J. Peng, Q. Luo, L. Sun, D. Li, and L. Chen. 2016. Lightning: A High-efficient Neighbor Discovery Protocol for Low Duty Cycle WSNs. *IEEE Communications Letters* PP, 99 (2016), 1–1.
[11] D. Zhang, T. He, Y. Liu, Y. Gu, F. Ye, R. K. Ganti, and H. Lei. 2012. *Acc*: generic on-demand accelerations for neighbor discovery in mobile applications. In *ACM Conference on Embedded Network Sensor Systems, (SenSys)*.