Ad Hoc Networks 16 (2014) 142–164

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A comprehensive analysis on the use of schedule-based asynchronous duty cycling in wireless sensor networks

Ricardo C. Carrano [*,1], Diego Passos [2], Luiz C.S. Magalhães [1], Célio V.N. Albuquerque [2]

Laboratório MídiaCom, Universidade Federal Fluminense, R. Passo da Pátria 156, bl.E sala 408, Niterói, RJ 24210-240, Brazil

ARTICLE INFO

Article history:
Received 11 October 2012
Received in revised form 15 November 2013
Accepted 23 December 2013
Available online 1 January 2014

Keywords:
Asynchronous duty cycling
Discovery latency
Energy efficiency

ABSTRACT

Duty cycling is a fundamental mechanism for battery-operated ad hoc networks, such as Wireless Sensor Networks, Delay Tolerant Networks, and solar-powered Wireless Mesh Networks. Because of its utter importance, it has been proposed in a wide variety of flavors, one of the most prominent being that of the asynchronous mechanisms. In particular, schedule-based duty cycling has earned attention due to its low requirements and simplicity of implementation.

Despite its potential, a comprehensive and realistic study on the neighbor discovery latency that results from schedule-based asynchronous duty cycling is still missing. This paper fills in this gap, by providing accurate models for major schedule-based mechanisms: Block Designs, Quorum systems and Disco. The provided models consider message loss probability and yield more precise estimations than traditional models. Based on this improved accuracy, the relative latency, a new metric for studying the trade off between latency and power, is proposed as a substitute to the power-latency product. Finally, a practical mapping of which schedule is more adequate for given requirements of latency, energy savings and link reliability is presented.

1. Introduction

Energy efficiency is fundamental to most Ad Hoc Networks. In Wireless Sensor Networks (WSNs), Delay Tolerant Networks (DTNs) and also in some Wireless Mesh Networks (WMNs), for example, nodes are battery-powered and often operate under severe energy constraints. In such applications, the duty cycling of the radio interface is a necessity, since the radio is usually responsible for a significant amount of a node's power drain [1,2].

However, duty cycling demands coordination. There must be ways to guarantee that a node will find an active neighbor for relaying its data, and do it timely. Such coordination may be achieved with synchronous mechanisms, in which nodes keep a common time reference, or with asynchronous mechanisms, where a common clock is unnecessary. There are also hybrid mechanisms in which the network is divided into clusters that are synchronized internally, while the communication between the clusters remains asynchronous.

In comparison to synchronous mechanisms, asynchronous contenders have the advantage of not relying on additional hardware, such as GPS or extra-precision clocks, and of generating less control traffic. Because of that, a plethora of asynchronous schemes has been proposed during the last decade. One of the most prolific categories of asynchronous duty cycling schemes is based on the use of special *wakeup schedules*. These schedules are alternations of active and inactive time slots selected in a way that guarantees that nodes will have a minimum overlapping active time, independent of their synchronization. During these periods, nodes may exchange messages and go back to

[*] Corresponding author. Tel.: +55 21 2629 5486.
E-mail addresses: carrano@midiacom.uff.br (R.C. Carrano), dpassos@midiacom.uff.br (D. Passos), schara@midiacom.uff.br (L.C.S. Magalhães), celio@ic.uff.br (C.V.N. Albuquerque).
[1] Department of Telecommunications Engineering.
[2] Institute of Computing.

1570-8705/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.adhoc.2013.12.009

sleep afterwards. We will refer to these overlapping active slots as *opportunities of discovery*.

As it has been pointed out [3,4], asynchronous mechanisms may result in long delays that accumulate over multihop paths. The *sleep waiting problem*, or *data forwarding interruption problem* [4], for instance, refers to the time a node has to wait until the next-hop neighbor wakes up. In fact, as duty cycling is reduced, neighbor discovery time (NDT) tends to increase, which is the fundamental trade off of the asynchronous approach and a core issue to our analysis.

Since discovery latency may be a hindrance to the use of asynchronous duty cycling, it is crucial to understand it thoroughly. However, to the best of our knowledge, analyses published so far do not apprehend important aspects of latency. Firstly, they tend to adopt an oversimplified model for the latency that assumes, for instance, that the opportunities of discovery occur at the end of the cycle. On top of that, they fail to consider the possibility of message loss. As channels are always imperfect, a *discovery opportunity* does not necessarily lead to *neighbor discovery*, meaning that many opportunities may be necessary until neighbors can communicate. Even when this fact is acknowledged [5] it is not incorporated to an analytical model. The models provided in this paper address these two issues and, as we demonstrate, provide more accurate estimations for the NDT.

The proposed models estimate the NDT for the most cited schedule-based asynchronous mechanisms: Block Designs [6], Quorum systems (Grid and Torus) [5,7] and Disco [8], and were validated with real implementations on sensor motes and with statistical simulations. Moreover, their improved accuracy allowed us to introduce a new metric – the *relative latency* – which provides a level playing field for comparisons between the mechanisms.

The work is concluded with a synoptic table that compares all asynchronous mechanisms presented. As it will be demonstrated, though Block Designs have a substantial advantage over the other proposals in most scenarios, it is not a suitable choice for all deployments. Also, the link quality, and the resulting probability of frame reception, may favor one or another mechanism in non-obvious ways.

The contributions of this paper may be divided into four groups (a more detailed list of contributions is provided in Section 7):

1. General contributions to the understanding of asynchronous duty cycling proposals, with formal definition and survey of the category of schedule-based asynchronous duty cycling mechanisms.
2. Proposal of accurate models for determining the neighbor discovery time (NDT) for the main asynchronous schedule-based duty cycle mechanisms, that incorporate message loss and relative time offsets between the neighbors. Current models are grossly inaccurate, with errors easily amounting to 400% for low quality links (e.g. $p < 0.2$). With the proposed models this error is typically inferior to 10%.
3. Practical comparison between the mechanisms. With improved models and the introduction of a new metric (relative latency) that allows direct and fair comparisons between the mechanisms, we demonstrate that the previous comparison, based on the power-latency metric is misleading. We show that the schedules perform comparatively better or worse depending on the link quality, and we indicate which schedule performs better given certain practical requisites.
4. In-depth analysis and discussion of aspects of each mechanism, that were not provided in current literature.

The rest of this paper is organized as follows. Section 2 defines the category of schedule-based asynchronous duty cycling and quickly reviews the other important asynchronous mechanisms. Sections 3–5 are dedicated to the most important categories of schedule-based asynchronous mechanisms, namely Block Designs (Section 3), Quorum systems (Section 4), and mechanisms based on prime numbers, such as Disco (Section 5). In each of these sections, the main mechanisms are studied and an analytical model for the estimation of the neighbor discovery time (NDT) is presented. Moreover, all models are validated through tests in real sensor motes and statistical simulations. Section 6 provides useful comparisons between the mechanisms, supported by the presented models. Section 7 presents our final remarks and a synoptic table of the most important findings of our analysis.

2. Asynchronous duty cycling

It is generally accepted that synchronizing nodes in a multihop wireless network is hard and costly [1,5], requiring extra hardware or processing capacities that may be too demanding for certain nodes, or adding frequent control traffic, which takes airtime and drains precious energy for transmission. In response to that, the asynchronous branch of proposals is prolific and diverse [6–13]. We start this section with a quick review of the main asynchronous mechanisms, and then proceed to a formal definition of the category of our interest: schedule-based mechanisms.

2.1. Overview of the asynchronous duty cycling mechanisms

Asynchronous duty cycling mechanisms tend to present similar issues, the most relevant being an increase in latency due to sleep waiting. Another, that may also affect synchronous mechanisms, is *idle listening*, which happens when a node wakes up in vain, i.e. when no traffic is directed to it. Asynchronous mechanisms differ in how they try to reduce latency or idle listening, while achieving low duty cycles.

In *preamble sampling* [9] mechanisms, every node goes to sleep asynchronously and wakes up periodically to check for channel activity. If a preamble is heard during this check-up period, the node remains active, waiting for the incoming frame. If not, the node goes back to sleep. What guarantees that the frame will be detected is the duration of the preamble – longer than the duration of active and sleep times together.

Another category of mechanisms is based on *receiver-initiated transmissions* [10]. In this category, instead of signaling that it has data to transfer, the willing sender waits for a periodic beacon from the receiver, and transmits the frame only after that beacon is heard. This method substitutes the periodic beacons for the long preamble.

On-demand wakeup [11] is based on the idea that a node may be removed from the sleep state whenever necessary. The mechanism usually relies on another communication interface, generally called the *wakeup radio* – a low power radio that would listen to a wakeup signal and send an interrupt to the CPU which, in turn, would activate the primary radio.

In *random duty cycling* [12], the idea is that, in sufficiently dense deployments, nodes can go to sleep and wake up randomly, since there is a high probability that there will be enough active nodes anytime.

Finally, the *schedule-based mechanisms*, which will be more strictly characterized in the next subsection, are based on the design of wakeup/sleep schedules that always guarantee an overlap of the active time between neighbors.

Most asynchronous mechanisms come in the form of redesigned MAC layers (as preamble-sampling and receiver initiated), while others rely on extra hardware to work properly (such as on-demand wakeup) or need a minimum network density (as the random mechanisms). Schedule-based mechanisms are the less demanding of the asynchronous techniques, not requiring extra hardware, or modifications to the MAC layer. They are also topology-independent.

2.2. Schedule-based asynchronous duty cycling

In schedule-based asynchronous mechanisms, nodes divide time into cycles, further subdivided into slots, either active or inactive, according to the selected *wakeup schedule*. Each cycle is a repetition of the previous. Fig. 1 illustrates the idea. To facilitate further formulations we will, hereinafter, use the terms *schedule* and *scheme*, defined as follows:

Definition 1 (*Schedule and scheme*). Given a cycle of n slots numbered from 0 to $n-1$, we define a *schedule in n*, $S[n] = \{a_0, \ldots, a_{m-1} \in [0, \ldots, n-1]\}$, as a set of active slots in that cycle ($m = \mid S[n] \mid$ is the number of active slots) and a *scheme in n*, $\chi[n] = \{S[n]_0, S[n]_1, \ldots, S[n]_z\}$ as a set of *schedules* in n.

Example 1 (*Scheme*). $\chi[5] = \{\{0,1,2\},\{2,3,4\},\{0,1,4\}\}$ is an example of a scheme in 5 slots, consisting of 3 schedules, each with 3 active slots.

In Example 1, all schedules of the scheme present the same duty cycle (60%). This is not necessarily true for all schemes, but it is an important characteristic that the literature usually refers to as *symmetry* of a scheme.

Definition 2 (*Symmetric scheme*). We define a *symmetric scheme* as one that results in symmetric duty cycle, i.e. a scheme where all schedules present the same cardinality: $\mid S[n] \mid = m, \forall S[n] \in \chi[n]$.

One advantage of the schedule approach is its simplicity of implementation, since it is sufficient that nodes operate under the selected wakeup schedule. That schedule, however, must be designed in a way that any two nodes will have overlapping active time, irrespective of their time offsets. What ensures that the schedule provides these opportunities of discovery is the property of *rotation closure* [5]. To define such property, we first define the operation of *rotation of a schedule*:

Definition 3 (*Rotation of a schedule*). A rotation of a schedule $S[n] = \{a_0, \ldots, a_{m-1}\}$ by $i \in \mathbb{Z}$, noted as $\rho(S[n], i)$ is a schedule $S[n]' = \{(a_k + i) \text{ modulo } n, \forall a_k \in S[n]\}$.

Example 2 (*Rotation of a schedule*). Given a schedule $S[5] = \{0,1,2\}$, from the scheme in Example 1, the rotation $S[5]' = \rho(S[5], 3) = \{0,3,4\}$. It is not hard to see that a rotation of a schedule in n is also a schedule in n.

We now move to the definition of the rotation closure property. This is the property that allows an asynchronous scheme to deal with the different time offsets experienced by nodes in a non-synchronized network. Without this property, there is the risk of the *deafness problem*, when nodes become disconnected from the rest of the network for never waking up in tandem with any of its neighbors.

Definition 4 (*Rotation closure property*). A scheme $\chi[n]$ presents the rotation closure property if, and only if:

- $\forall S[n]_i, S[n]_j \in \chi[n] : S[n]_i \cap \rho(S[n]_j, k) \neq \emptyset, k = 0, \ldots, n-1$

Example 3 (*Schemes and the rotation closure property*). The scheme $\chi[4]_a = \{\{0,1,2\},\{1,2,3\},\{0,2,3\}\}$ presents the rotation closure property, while $\chi[4]_b = \{\{0,1\},\{0,2\},\{1,2\}\}$ does not, since $\{0,1\} \cap \rho(\{1,2\},1) = \emptyset$. Notice that the intersection between any two elements is a necessary condition for the rotation closure. However, as seen in the example of $\chi[4]_b$, this condition is not sufficient.

Finally, we define the Neighbor Discovery Time, the quantity extensively used throughout the text to analyze and compare the main asynchronous schemes.

Definition 5 (*Neighbor Discovery Time*). Given two nodes A and B, operating under a duty cycle scheme, and sending beacon messages every active slot, the *NDT* is the number of time slots, counted from an initial slot when the nodes enter communication range, until the slot when B successfully receives a beacon from A.

We are particularly interested in obtaining the expected value for the NDT (*E[NDT]*), considering the following hypothesis: any two nodes are turned on at random independent moments, hence their time references may have an offset that we assume as uniformly distributed; and the initial slot is also a random and uniform variable.

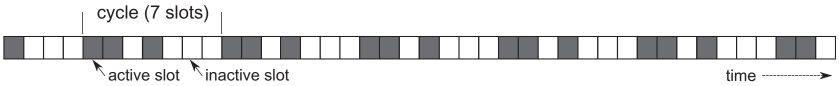


Fig. 1. An example of wakeup schedule. Time is divided into cycles, in this example consisting of 7 slots, either active (dark) or inactive (white).

The design of asynchronous wakeup schedules is, therefore, the quest for low duty cycle schedules, which satisfy the rotation closure property, while providing acceptable *E[NDT]*.

As noted in [5,7], some *Quorum systems*, as the Grid and the Torus [5] present the rotation closure property. Quorum systems consist of a set system (set of sets) where any two elements present non-null intersections (which, as noted in Example 3, is not a synonym for having the rotation closure property). Quorum systems have been studied in the context of distributed systems at least since 1985 [14], though initially as a solution to the mutual exclusion problem, i.e. avoiding the concurrent use of a resource. Quorum systems will be presented in Section 4.

Another category of schedule-based mechanisms builds schedules from prime numbers, where the rotation closure property is guaranteed by the Chinese Remainder Theorem [15]. An example is Disco [8], presented in Section 5.

But optimum schedules, in terms of duty cycling, as demonstrated in [14], come from *Block Designs*. An asynchronous mechanism based on Block Designs is presented in [6], and recently, the idea was revisited in [16], in which the authors advocate for its use in an energy-efficient mechanism for neighbor discovery. Block Designs are presented in Section 3.

3. Block Designs

Block Designs come from the mathematical area of Combinatorics, and were first introduced as technique to design experiments in agriculture [17]. A Block Design[3] is a set system (a set of sets called *blocks*) that can be defined as follows:

Definition 6 (*Block Design*). Given a finite set V of v elements and integers $k, \lambda \geqslant 1$ a *Block Design*, represented as $\{v, k, \lambda\}$, will have exactly v *blocks* $(B_0 \ldots B_{v-1} \subset V)$ of k elements and the following properties:

1. Each and every element of V occurs in exactly k blocks.
2. Any two blocks will have exactly λ elements in common.

Example 4 (*Block Design*). Suppose, for instance, that $V = \{0,1,2,3,4,5,6\}$. A Block Design $\{7,3,1\}$ (Fig. 2) would be the set of 7 blocks consisting of 3 elements each, such that any two of these blocks would have exactly one element in common: $\{[0,1,3],[1,2,4],[2,3,5],[3,4,6],[4,5,0],[5,6,1],[6,0,2]\}$.

In relation to our previous definitions, a *block* is equivalent to a *schedule*, while a *design* is equivalent to a *scheme*. The rotation closure property is ensured by the Property 2 in Definition 6.

The duty cycle for a Block Design is k/v, which corresponds to 43% in Example 4, or 1.03% for a $\{9507,98,1\}$ design. Table 1 presents some examples of Block Designs, providing the resulting duty cycles and the list of active slots.

To date, networking literature has considered mainly, if not exclusively, a particular category of Block Design where $\lambda = 1$, commonly referred to as *projective planes*.[4] A projective plane is a Block Design that takes the form $\{s^2 + s + 1, s + 1, 1\}$, where the parameter s is called *order*. It is conjectured that there only exists projective planes for values of s which are powers of primes [18]. This implies that there are no projective planes for $k = 7$, 11, 13, 15, and so on. In reality, the number of known Block Designs is relatively small – 145, of which only 35 are projective planes [19]. All known Block Designs were included in our analysis and are presented in Tables 2 and 3.

One reason for the attention received by projective planes ($\lambda = 1$) comes from the fact that they provide the optimal schedules in terms of duty cycle [14]. That means that, given a fixed number of time slots, projective planes will generate the wakeup schedule with the minimum rate of active to inactive slots that present the rotation closure property.

The literature sometimes refers to *perfect difference sets* [16] which form directly from projective planes and result in the same schedule of operation. Because of the equivalence we will refer only to projective planes throughout this text. A formal definition of difference sets may be found in [18].

3.1. NDT for Block Design schedules

Differently from the simple expressions in the literature, that consider the NDT as the same as the cycle length and do not account for message loss probability, in this work, we introduce a model for the NDT that includes link quality and is extensible for all values of λ, not only projective planes. As we will see, current models incur in prediction errors that may greatly understimate or overestimate NDT.

3.1.1. Methodology

The NDT model for Block Designs was obtained by combinatorial analysis of all the possible intersections between the schedules of two unsynchronized nodes operating with random offsets.

This approach works for all Block Designs, not only for projective planes. The key point is to determine how the λ encounter opportunities are distributed within a cycle, as a function of all the possible v offsets between the neighbors, and also considering that many cycles may be necessary for the discovery, depending on p. The method is further described in Appendix A.

3.1.2. NDT model

We argue that our model, presented in Eq. (1), provides an estimate of the latency (neighbor discovery time) in asynchronous duty cycling schemes based on Block Designs.

[3] In networking literature, Block Designs and Symmetric Block Designs are commonly treated as synonyms, and usually defined with lack of mathematical rigor. For conciseness, we will proceed likewise. The reader is referred to [18] for a formal definition.

[4] Actually, the term projective plane comes from Geometry and not from Combinatorics. However, the two concepts bear the same mathematical formulation and are usually considered as equivalent.

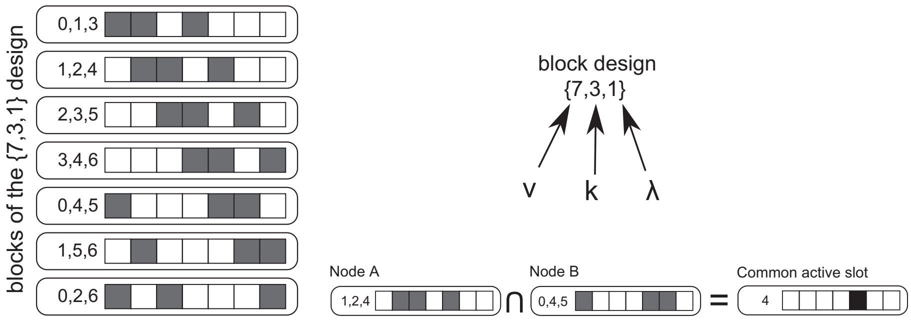


Fig. 2. A $\{7,3,1\}$ Block Design, its elements (blocks) and an example of two nodes operating under two different blocks ([1,2,4] and [0,4,5]), that intersect at slot 4.

Table 1
Some examples of Block Designs, with the resulting duty cycles and the list of active slots. Slots are numbered from 0 to $v-1$, where v is the first parameter of the design, and also the cycle length. Note that, in $\{7,3,1\}$, the set of active slots 0, 1, 3 is completely equivalent to 1, 2, 4 or any other rotation of the schedule.

Design	Duty cycle (%)	Active slots
$\{7,3,1\}$	42.86	0, 1, 3
$\{183,14,1\}$	7.65	0, 12, 19, 20, 22, 43, 60, 71, 76, 85, 89, 115, 121, 168
$\{9507,98,1\}$	1.03	0, 1, 3, 37, 52, 191, 308, 332, 433, 914, 919, 984, 1093, 1155, 1231, 1238, 1600, 1678, 1723, 1732, 1755, 1773, 1826, 1930, 1938, 2099, 2116, 2141, 2457, 2712, 2859, 3058, 3187, 3466, 3524, 3655, 3675, 3748, 4139, 4145, 4183, 4297, 4301, 4518, 4528, 4600, 4720, 4777, 4964, 5043, 5054, 5176, 5268, 5329, 5356, 5496, 5526, 5601, 5617, 5851, 6151, 6173, 6491, 6539, 6759, 6778, 6792, 6878, 7021, 7163, 7226, 7290, 7490, 7650, 7747, 7860, 7941, 8028, 8056, 8154, 8304, 8339, 8370, 8438, 8450, 8505, 8534, 8574, 8797, 9005, 9048, 9094, 9107, 9133, 9154, 9270, 9326, 9400
$\{11,5,2\}$	45.45	0, 2, 3, 4, 8

Table 2
Known symmetric Block Designs with $\lambda = 1$ (projective planes), sorted by duty cycle.

$\{v,k,\lambda\}$	Duty cycle (%)	$\{v,k,\lambda\}$	Duty cycle (%)	$\{v,k,\lambda\}$	Duty cycle (%)
$\{7,3,1\}$	42.86	$\{553,24,1\}$	4.34	$\{3541,60,1\}$	1.69
$\{13,4,1\}$	30.77	$\{651,26,1\}$	3.99	$\{3783,62,1\}$	1.64
$\{21,5,1\}$	23.81	$\{757,28,1\}$	3.70	$\{4161,65,1\}$	1.56
$\{31,6,1\}$	19.35	$\{871,30,1\}$	3.44	$\{4557,68,1\}$	1.49
$\{57,8,1\}$	14.04	$\{993,32,1\}$	3.22	$\{5113,72,1\}$	1.41
$\{73,9,1\}$	12.33	$\{1057,33,1\}$	3.12	$\{5403,74,1\}$	1.37
$\{91,10,1\}$	10.99	$\{1407,38,1\}$	2.70	$\{6321,80,1\}$	1.27
$\{133,12,1\}$	9.02	$\{1723,42,1\}$	2.44	$\{6643,82,1\}$	1.23
$\{183,14,1\}$	7.65	$\{1893,44,1\}$	2.32	$\{6973,84,1\}$	1.20
$\{273,17,1\}$	6.23	$\{2257,48,1\}$	2.13	$\{8011,90,1\}$	1.12
$\{307,18,1\}$	5.86	$\{2451,50,1\}$	2.04	$\{9507,98,1\}$	1.03
$\{381,20,1\}$	5.25	$\{2863,54,1\}$	1.89		

Eq. (1), for which a proof is provided in (Appendix A), takes the parameters v (number of slots per cycle) and λ (number of discovery opportunities per cycle) from the Block Design and the probability of discovery p to estimate the expectancy for the neighbor discovery time, *E[NDT]*.

$$E[NDT] = \frac{v+1}{p(\lambda+1)} - \frac{(v+1)(1-p)^{\lambda} - (\lambda+1)}{(\lambda+1)[(1-p)^{\lambda} - 1]} \tag{1}$$

Some special cases can be derived from Eq. (1) and provide more practical and insightful expressions for the NDT. Three of this cases are presented on Table 4. Case 1 presents the behavior of *E[NDT]* when $\lambda = 1$, i.e. for projective planes. In this case, as p increases, *E[NDT]* tends to a little less than half a cycle ($\frac{v-1}{2}$). Case 2 is a particular case of Case 1 and presents an intuitive result for near-perfect links ($p \sim 1$): the waiting time will range from 0 (immediate) to v slots (a complete cycle) with mean equal to $v/2$. This comes from the fact that, for $\lambda = 1$ and $p = 1$, the NDT follows a discrete uniform distribution. Finally, Case 3 also considers perfect links (and can be extrapolated to near-perfect links without prejudice), but now there are many opportunities per cycle. The model shows that for a given cycle duration, designs with higher λ will reduce *E[NDT]* at the expense of a higher duty cycle.

3.1.3. Model validation

In order to validate the proposed model, duty cycling schedules based on Block Designs were implemented in real sensor motes running TinyOS [20]. Neighbor discovery was emulated and measured and the resulting NDT was compared with the predictions of the model. There are, however, two limitations to this method. The first is the

Table 3
Known Block Designs with $\lambda > 1$, sorted by duty cycle. The quantity in parenthesis indicates the number of variations known for each design (there are, for instance, 10 distinct variations for the {1023,511,255} design).

$\{v, k, \lambda\}$	Duty cycle (%)	$\{v, k, \lambda\}$	Duty cycle (%)	$\{v, k, \lambda\}$	Duty cycle (%)
{1023,511,255} (10)	49.95	{271,135,67}	49.82	{43,21,10} (2)	48.84
{599,299,149}	49.92	{263,131,65}	49.81	{35,17,8}	48.57
{587,293,146}	49.91	{255,127,63} (4)	49.80	{31,15,7} (*)	48.39
{571,285,142}	49.91	{251,125,62}	49.80	{23,11,5}	47.83
{563,281,140}	49.91	{239,119,59}	49.79	{19,9,4}	47.37
{547,273,136}	49.91	{227,113,56}	49.78	{15,7,3}	46.67
{523,261,130}	49.90	{223,111,55}	49.78	{11,5,2}	45.45
{511,255,127} (6)	49.90	{211,105,52}	49.76	{364,121,40}	33.24
{499,249,124}	49.90	{199,99,49}	49.75	{121,40,13} (4)	33.06
{491,245,122}	49.90	{191,95,47}	49.74	{40,13,4}	32.50
{487,243,121}	49.90	{179,89,44}	49.72	{109,28,7}	25.69
{479,239,119}	49.90	{167,83,41}	49.70	{901,225,56}	24.97
{467,233,116}	49.89	{163,81,40}	49.69	{677,169,42}	24.96
{463,231,115}	49.89	{151,75,37}	49.67	{341,85,21}	24.93
{443,221,110}	49.89	{143,71,35}	49.65	{197,49,12}	24.87
{439,219,109}	49.89	{139,69,34}	49.64	{133,33,8}	24.81
{431,215,107}	49.88	{131,65,32}	49.62	{101,25,6}	24.75
{419,209,104}	49.88	{127,63,31} (6)	49.61	{85,21,5}	24.71
{383,191,95}	49.87	{107,53,26}	49.53	{37,9,2}	24.32
{379,189,94}	49.87	{103,51,25}	49.51	{781,156,31}	19.97
{367,183,91}	49.86	{83,41,20}	49.40	{156,31,6}	19.87
{359,179,89}	49.86	{79,39,19}	49.37	{400,57,8}	14.25
{347,173,86}	49.86	{71,35,17}	49.30	{585,73,9}	12.48
{331,165,82}	49.85	{67,33,16}	49.25	{820,91,10}	11.10
{323,161,80}	49.85	{63,31,15}	49.21	{1464,133,12}	9.08
{311,155,77}	49.84	{63,31,15}	49.21	{2380,183,14}	7.69
{307,153,76}	49.84	{59,29,14}	49.15	{4369,273,17}	6.25
{283,141,70}	49.82	{47,23,11}	48.94		

Table 4
Three special cases derived from the model.

Case 1: $\lambda = 1$	Case 2: $\lambda = 1$ and $p = 1$	Case 3: $p = 1$
$E[NDT] = \frac{v}{p} - \frac{v+1}{2}$	$E[NDT] = \frac{v-1}{2}$	$E[NDT] = \frac{v-\lambda}{\lambda+1}$

practical number of repetitions, as each measure takes an average time in the order of many seconds or a few minutes, depending on the length of the design (v). Since the number of possible offsets is in the order of v, thousands of repetitions may be necessary to validate a single point (a combination of v and p) of the model. The second reason is the practical difficulty in fine tuning the message delivery rate (p) under real test conditions and for long periods.

To overcome these limitations, two additional categories of tests were performed in support to the tests with real implementations. Statistical simulations on the R environment [21] allowed for a great number of repetitions and the exact fixation of p. However, as the statistical simulation do not emulate the radio channel behavior, simulations in the TinyOS simulator (TOSSIM [22]) were also performed.

These three categories of validation cover the preponderant aspects of the modeled phenomenon and were performed for each of the schedule-based asynchronous duty cycling schemes studied. The results for the experiments with Block Designs are presented in Table 5.

The results presented in the first line of Table 5 were performed with TelosB motes, manufactured by Memsic [23]. Because neighbor discovery is a phenomenon involving only two peers, the tests used two sensor motes (*A* and *B*), with no loss of generality. In fact, the number of neighbors may influence NDT in two ways. First, an increase in network density may result in an increase in collisions and consequent waste of discovery opportunities. However, this phenomenon is captured by our models in the parameter "p" – an increase in the rate of collisions result in a decrease in the reception probability. The second way in which the number of neighbors may affect NDT is if, instead of considering the discovery of a given neighbor, we are concerned with the discovery of *any* neighbor. We argue that this only makes sense from the perspective of specific higher layer protocols, while in the canonical case *all* neighbors must be discovered. Moreover, this can again be incorporated to our model, by changing p to represent the probability of discovering any of the available neighbors. As the number of neighbors is unknown beforehand, it seems more appropriate to consider the more representative case, where a given node A discovers a given node B.

The experiment worked as follows. Mote *A* operates continually on an activity schedule taken from the Block Design under test and sends a beacon at every active slot. Mote *B* is activated after a random wait, in order to guarantee the random offset between *A* and *B*. When activated, B starts a counter that stops when a beacon from *A* is received. This counter measures the NDT. After reporting this measurement, *B* waits a random time and then restarts the process.

The two motes were placed in an environment with low interference in the selected channel (channel 26 of the IEEE 802.15.4 standard) and only a few centimeters apart. The delivery probability was periodically monitored and was of 100% during all measurements, indicating that p was, at least, very close to 1 during the entire experiment.

Table 5
Summary of the validation results for the Block Design NDT estimation model presented in Eq. (1).

Validation method	Scenarios	Prediction error
Implementation on TelosB motes	Block Designs $\{9507, 98, 1\}$ and $\{183, 14, 1\}$ with $p \sim 1$	<1%
Implementation on TOSSIM (TinyOS Simulator)	Block Designs $\{9507, 98, 1\}$ and $\{183, 14, 1\}$ with $p = 0.78$	3–4%
Statistical simulations in the *R* environment	All known Block Designs; all probabilities from 5% to 100% with increments of 5%	See Fig. 4

Two different designs were used: $\{183, 14, 1\}$ and $\{9507, 98, 1\}$. In both cases, slots were 97.7 ms long (100 ticks of a 1024 Hz clock). According to the model, for $p = 1$, the NDT would be equivalent to 91 slots for $\{183, 14, 1\}$ and to 4753 slots for the $\{9507, 98, 1\}$ design. For each schedule, 500 measurements were taken and the average measured NDT was respectively equivalent to 90.1 and to 4712.5 slots – a difference of less than 1% from the model prediction.

For the simulations in TOSSIM, the parameters were taken from examples of typical WSNs, with channel parameters resembling those of outdoor links in open-field and low-vegetation environment, with no strong noise or interfering devices and with the nodes placed 30 m apart. For such scenario, simple preliminary tests determined that p was 0.78. The tests were run 2000 times each and, as shown in Table 5, the accuracy of the model prediction was better than 96%.

Finally, statistical simulations performed in the *R* environment were used to test the model over a significant range of values of p – from 0.05 to 1.0, with increments of 0.05. This method allows a great number of repetitions – 40,000 measures were taken for each value of p and all known Block Designs were tested.

The results also confirmed the predictions of the proposed model. Fig. 3 shows the results for the worst case found (the design $\{7, 3, 1\}$) and two other typical results (for designs $\{9507, 98, 1\}$ and $\{4369, 273, 17\}$). Even for the worst case, the accuracy of the model is better than 99% for all cases where $p > 0.4$. Moreover, for most designs this is also true for lower values of p.

Fig. 4 shows the mean and maximum errors for all designs with respect to the results obtained through simulation. Again, mean errors of less than 1% are typical. It is also clear that maximum errors are not significantly higher than mean errors. Note that, if instead of using the proposed model, the same analysis is performed using the cycle length (as in the current literature), the maximum error will scalate to 95% (for $p = 0.05$), with an average error of 52%, for any projective plane. And these results would be even worse for designs with $\lambda > 1$.

3.2. Discussion

The three graphs in Fig. 5 provide a qualitative understanding on the way *E[NDT]* grows with the variation of the three parameters (p, λ and v) of the proposed model. Fig. 5a shows the variation of *E[NDT]* as we fix the number of slots (v) and vary the probability of reception (p) and the number of discovery opportunities per cycle (λ). As expected, *E[NDT]* decreases as p or λ increase. As p and λ increase, the surface flattens and *E[NDT]* changes less dramatically. Outside this area, an exponential growth of *E[NDT]* is observed.

In Fig. 5b, the fixed parameter is p and the resulting curve clearly demonstrates the linear relation between *E[NDT]* and v, which differs from the non-linear gains resulting from the increase of λ. A similar analysis can be made to Fig. 5c – cutting the cycle length in half will halve the discovery time, while increasing the delivery probability (for example, by increasing the transmission power, or by reducing the distance) may have a dramatic effect, particularly for marginal links (low values of p).

The proposed model also permits a comparison between projective planes and Block Designs with $\lambda > 1$. While projective planes provide minimal duty cycle, the augmented frequency of discovery opportunities caused by an increase in λ may reduce the NDT as the link quality deteriorates (low values of p). Fig. 6a plots all known Block Designs in the $NDT \times DC$ (DC = duty cycle) space for $p = 0.9$. Projective planes are represented by circles, while the other Block Designs are represented by crosses. As expected, projective planes provide the lowest duty cycles – all Block Designs with duty cycles of less than 5% are projective planes, and only three Block Designs with $\lambda > 1$ present duty cycles of less than 10%, as can be seen in the expanded area (and in Table 3).

Fig. 6b provides the same information for $p = 0.1$. The points for $\{4369, 273, 17\}$ and $\{273, 17, 1\}$ are marked in the expanded areas of both Figs. 6a and b. It is possible to notice the exchange in the relative positions of these points – while $\{273, 17, 1\}$ is better (lower NDT) for $p = 0.9$, the same does not happen when $p = 0.1$.

Fig. 7a and b continue this analysis and provide a direct comparison between two pairs of Block Designs, always opposing a projective plane with another Block Design with bigger λ and same approximate duty cycle. Fig. 7a compares $\{273, 17, 1\}$ (duty cycle 6.23%) and $\{4369, 273, 17\}$ (duty cycle 6.25%). For $p < 0.26$ the NDT for the latter is the shorter one. Likewise, as we can see in Fig. 7b, the NDT for $\{57, 8, 1\}$ (duty cycle 14.04%) is only shorter than the NDT of $\{400, 57, 8\}$ (duty cycle 14.25%) while $p > 0.44$.

Before moving onto other schedules, is it worth noting the fact that no known Block Design provides duty cycles of less than 1% which, as the rest of this work will prove, is the major drawback of the Block Design scheme.

4. Quorum systems

A Quorum is a set system where the intersection of any two elements is never null. At least three systems may be found throughout the networking literature: the Grid [5,7],

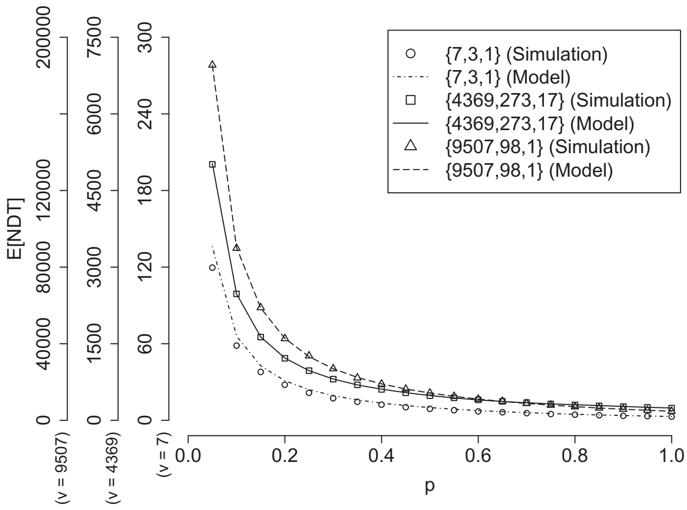


Fig. 3. Results of statistical simulations, obtained in the R environment (points) in comparison to the proposed model (lines). Errors are noticeable only for small values of both p and v. Confidence intervals were too small due to the high number of runs (40,000) and omitted for better legibility.

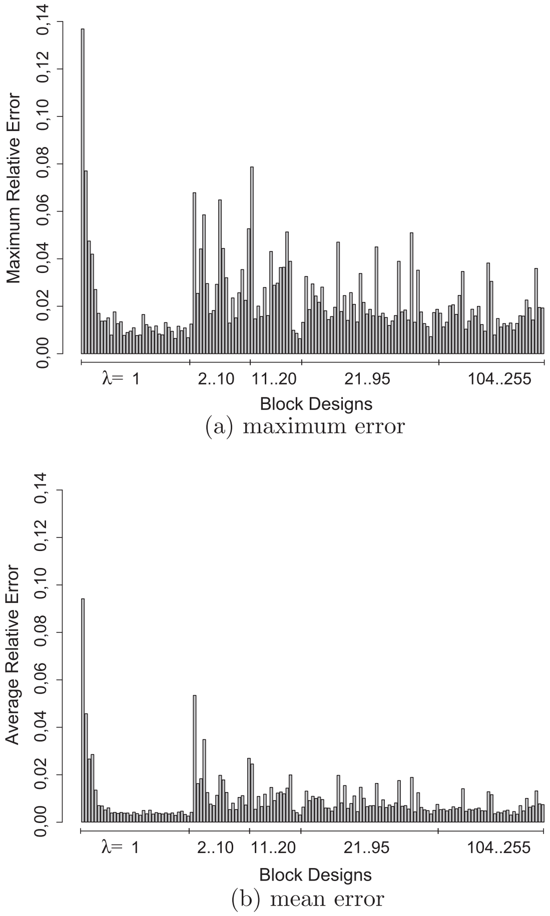


Fig. 4. Maximum and mean errors in statistical simulations. Projective planes are closer to the vertical axe.

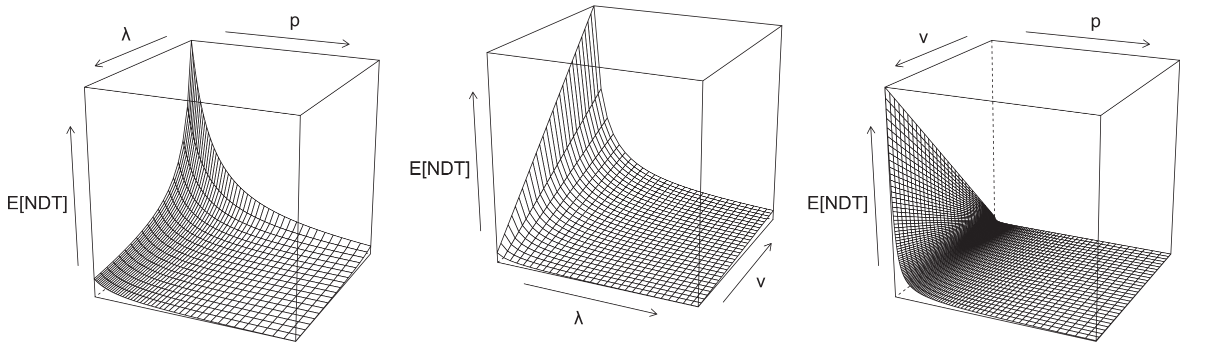
the Torus [5,7] and the e-Torus [5]. The last one was proposed as a basis to an adaptive mechanism where nodes are capable of determining their level of mobility – an added complexity that is not assumed by other mechanisms. Because of that, and because of the fact that the e-Torus is actually an extension of the Torus system, we will limit our analysis to the first and more universally applicable two mechanisms – the Grid and the Torus system.

4.1. Grid Quorum

In a Grid Quorum, each participant selects one line and one column in an $n \times n$ matrix. When used for wakeup schedules in asynchronous duty cycling, each node will divide time into cycles of n^2 slots and decide whether they are active or not, depending on their *line* and *column*, as if the slots were cells in that matrix. It follows that for any two nodes there will be at least two common active slots as illustrated in Fig. 8. The resulting duty cycle for an $n \times n$ grid is $\frac{2n-1}{n^2}$.

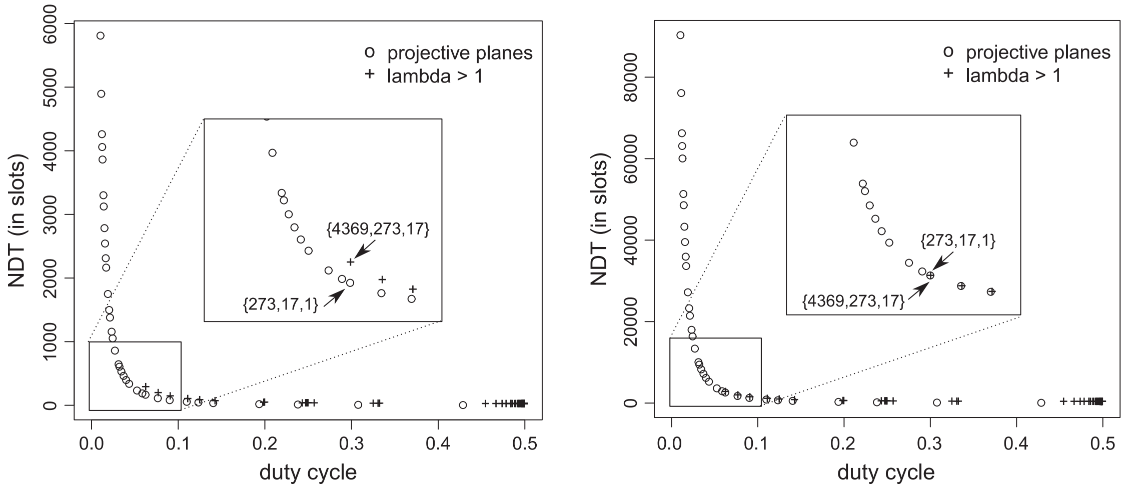
4.1.1. Preliminary analysis on line and column selection

Proposers of the Grid Quorum did not indicate which column or line should be selected as active to reduce the NDT. In fact, because of the cyclic repetition of the schedule, the selection of the active line does not have an impact. However, we found that the selection of the active column impacts the expected NDT in complex ways. To understand that, we performed a series of simulations and comparisons. In our analysis, a maximum difference of 6% was found between the best and the worst column choice, for grids with $n = 5$, 10, 20, 30, 40, 50, 100 and 200, computed with probabilities from 0.05 to 1, with increments of 0.05. The typical difference between best and worst column was near 3%. Table 6 shows part of these results ($n = 10$, 50, 100 and 200; and $p = 0.1, 0.2, \ldots, 1.0$). For each cell, the first value is the index of the best column for that combination of n and p; the second value is the percentage difference in the NDT between the central and the best columns; and the third is the percentage difference between the best and the worst columns.



(a) v fixed (b) p fixed (c) λ fixed

Fig. 5. The behavior of Eq. (1) – linear growth of *E[NDT]* with v and non-linear decrease, with both p and λ. As the NDT is infinite for $p = 0$, all graphs are plotted for $p \in [0.1, 1]$.



(a) $p = 0.9$ (b) $p = 0.1$

Fig. 6. NDT and duty cycle for all known Block Designs, for $p = 0.9$ and for $p = 0.1$. The most efficient Block Designs are within the expanded areas.

Apart from giving an idea on the impact of the column selection (something on the order of 3%) in the NDT, Table 6 shows that, in general, choosing central columns results in shorter NDT in comparison to low or high columns. However, the most interesting result, as clearly demonstrated by the variations on the first number in each cell, is that the best column depends, to a great extent, on the value of p. If one takes $n = 50$ as an example, the best column changes erratically as p changes (16, 45, 4, 23, ...). Therefore, we conclude that, though the column selection impacts the NDT, an optimal selection is impractical.

For the purposes of our analysis and the building of the model presented in the next section, a central column was always selected.

4.1.2. Methodology for obtaining an NDT model

Deriving the expected NDT for a Grid system is not as straightforward as it is for other mechanisms. This difficulty is due mainly to the many different ways in which two grids with different offsets intersect. By empirically analysing the behavior of this intersection patterns, using the R language and applying methods of interpolation and curve fitting, we were able to find an expression for the NDT, presented in Eq. (2).

We obtained a 6th degree polynomial expression, and tested it with many interesting values of n to reassure its accuracy and, finally, we derived a compact quadratic version that approximates the complete expression with accuracy better than 99% for all schedules with duty cycles lower than 3% ($n > 100$). For the details of the complete methodology, refer to Appendix B.

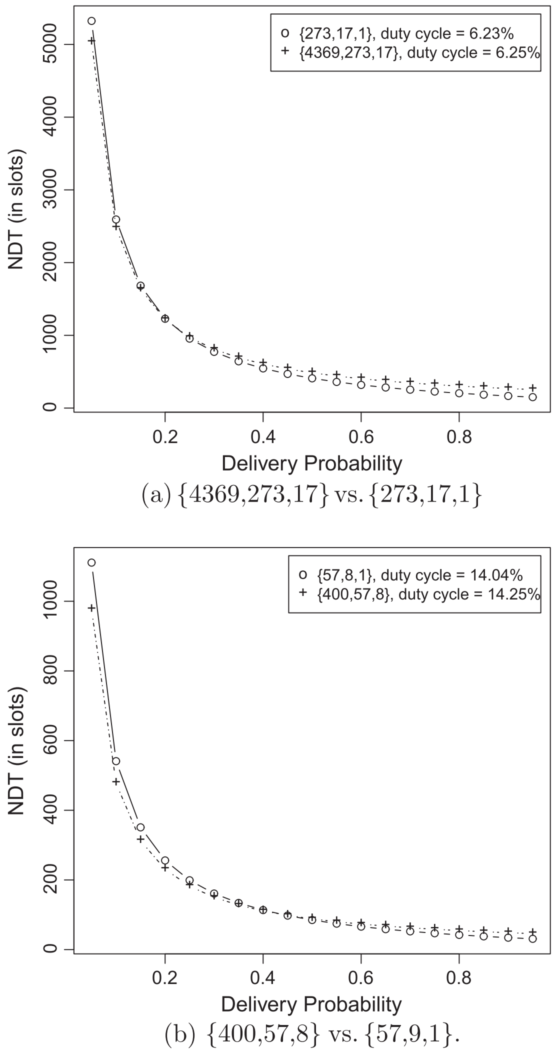
4.1.3. NDT model

Eq. (2) is our expression for the expected NDT, when two nodes operate under an $n \times n$ grid. It takes the parameters p, the probability of discovery, and n, the grid dimension.

$$E[NDT] = \frac{(3 - p)n^2}{6p} \tag{2}$$

4.1.4. Model validation

For the purpose of validating the model presented in Eq. (2), Grid schedules were implemented in TelosB motes and



(a) {4369,273,17} vs. {273,17,1}

(b) {400,57,8} vs. {57,9,1}.

Fig. 7. Comparisons between a projective plane and another Block Design ($\lambda > 1$) with similar duty cycle show that the better NDT depends on the link quality p.

simulated in TOSSIM. Also, statistical simulations were performed for many values of n and p. The methodology employed was the same used for Block Designs, as described in Section 3.1.3. Results are presented in Table 7.

Differently from the case of Block Designs, where all known schedules could be tested, the number of different grids is infinite and the experiments must be limited to a number of interesting values of n. For the tests in real motes and in TOSSIM, 20×20 and 100×100 grids were used. For the statistical simulations, all values of n from 20 to 50, plus $n = 100$, 500, 1000, 5000 and 10,000 were tested. For $n = 10,000$, the duty cycle is already very low (0.02%) while the NDT will be in the order of millions of slots, even for high values of p. Therefore, we infer that grids with $n > 10,000$ would be of limited use. Likewise, any value of $n < 20$ would result in duty cycles of more than 10%, and were also considered uninteresting and omitted.

In comparison to the model presented for Block Designs, the model represented by Eq. (2) is less accurate. The reason, as already noted, comes from the many forms in which two grids intersect, that result in a variety of schedules in terms of the number and distributions of intersecting slots. Nevertheless, the error is under 10%, still a better prediction than that of the models currently found in the literature [8,16]. As will be discussed on Section 6, the available models use the upper-bound probability-unaware estimation, in which $NDT = n^2$. That method results in errors of more than 100% even for $p = 1$. Moreover, the accuracy achieved by our model is sufficient for a direct comparison between the schedules, as discussed in Section 6.

4.2. Torus Quorum

The Torus Quorum system improves the Grid Quorum in terms of duty cycling. It relies on the fact that, to guarantee the overlapping of active slots, it is not necessary to select a complete line. Instead, if n is the number of columns and c is the selected column, it is enough that nodes select half plus one of the slots in each column $c + i, i = 1, \ldots, \lfloor n/2 \rfloor$, as illustrated in Fig. 9.

In the example of Fig. 9, the duty cycle of the Torus is 15% against 24% of the Grid with the same number of lines and columns. The duty cycle of a Torus system is $\frac{3}{2n}$, for even values of n, or $\frac{3n-1}{2n^2}$, for odd values of n.

4.2.1. Methodology for obtaining an NDT model

In order to obtain a model for the estimation of the NDT for Torus systems, a similar procedure to the one for the Grid system was used. However, for Torus, the process must start by treating odd and even values of n differently. In the end of the process, we were able to unify both cases in a single quadratic expression (Eq. (3)). Because intersection patterns in Torus are more well behaved than in Grid, the resulting expression for Torus is more accurate. The specificities of the methodology are detailed in Appendix C.

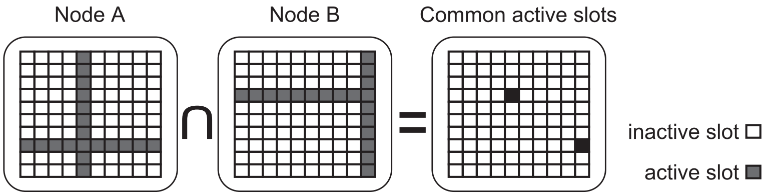


Fig. 8. A 10×10 Grid Quorum. Any nodes selecting a column and a line will have at least two common active slots (more if the line or the column coincide).

Table 6
Results of the statistical simulations to investigate the impact of column selection in NDT. For each grid dimension (n), the table shows respectively: the index of the best column; the percentage difference between the central column and the best column; and the percentage difference between the best and worst columns.

p	$n = 10$	$n = 50$	$n = 100$	$n = 200$
0.1	5/0.00/1.77	16/0.55/2.06	52/0.14/2.81	84/1.35/2.90
0.2	7/1.26/2.19	45/0.34/2.25	68/1.69/2.92	94/2.54/2.89
0.3	6/0.15/2.39	4/1.87/2.75	94/0.66/3.00	37/1.30/2.65
0.4	3/0.43/1.51	23/0.43/1.89	85/0.47/3.13	81/1.39/2.54
0.5	6/0.82/2.38	9/0.34/2.41	58/0.54/2.44	130/0.42/2.85
0.6	7/1.22/1.82	29/1.17/1.52	67/1.07/2.13	190/0.59/2.46
0.7	4/1.01/1.94	7/0.88/2.39	23/1.41/2.16	166/1.06/2.35
0.8	6/0.84/1.98	13/1.71/2.51	45/0.72/1.91	113/0.70/2.48
0.9	7/0.51/1.94	21/0.73/1.36	41/1.26/2.01	122/1.19/2.14
1.0	8/0.41/2.22	40/1.04/1.60	50/0.00/1.92	23/0.94/1.93
Mean	5.9/0.66/2.02	20.7/0.91/2.07	58.3/0.80/2.44	104/1.15/2.52

4.2.2. NDT model

Eq. (3) is our NDT model for nodes operating under Torus schedules. It takes the parameters p, the probability of discovery, and n, the grid dimension.

$$E[NDT] = \frac{(2-p)n^2}{2p} \tag{3}$$

4.2.3. Model validation

Again, for validating the model presented in Eq. (3), Torus schedules with $n = 20$ and $n = 100$ were implemented in TelosB motes and in TOSSIM. Also, statistical simulations were performed for many values of n and p. The methodology employed was the same used for the other schedules (described in Section 3.1.3). Results are presented in Table 8.

The model for the Torus Quorum presents better accuracy than the model for the Grid Quorum. In fact, the number of ways that Torus schedules intersect is more limited than the number of ways that grids intersect, which results in a simpler analysis.

4.3. Discussion

An interesting question, regarding Quorum Systems, is determining to which extension Torus schemes are a substitute for Grid schemes. In one hand, the resulting duty cycle, for the same number of columns and lines, will be smaller in the case of the Torus. On the other hand, if the NDT is considered, the comparison is not as simple.

Table 9 summarizes the NDT and the duty cycle for Grid and Torus. Working these equations for a fixed duty cycle, we conclude that Grid systems will result in lower NDT if the probability of discovery p is lower than approximately 0.5 (the exact value is $p < 0.545$, but given the approximations involved, it is only possible to state that for low quality links, the Grid systems is advantageous if both NDT and duty cycle are considered). As a result, Grid and Torus systems may be considered to have different application scenarios. As we will see, this is an incentive to the proposal of our relative latency metric, in Section 6.

5. Prime number-based mechanisms

The last category of asynchronous schedule-based duty cycling mechanisms studied is that of the schedules that rely on properties of the prime numbers. If two nodes select different prime numbers, m and n, and activate every m_{th} and the n_{th} slots in a cycle respectively, there will be a moment when both will be active, irrespective of their time offsets.

Conversely, if both nodes select the same prime number, there will be no overlapping slot, unless these nodes are synchronized (offset = 0). But forcing nodes to always select different prime numbers would require a coordination mechanism (and not needing such coordination is the main strength of schedule-based mechanisms). In U-connect [13], this problem is solved by making the node activate not only the multiples of the selected prime number, m, but also some of the slots, such that $0 < slotnumber < \frac{m+1}{2}$, at the beginning of every cycle of m^2 slots.

Making a node select one prime number severely limit the choices in terms of duty cycling (which would be $1/m$, where m is the selected prime number). The authors of Disco [8] propose that each node selects two prime numbers, instead of one, and activate the slots multiple to either of the primes, allowing more flexibility in the selection of the operating duty cycle. As an added bonus, the demand that each node selects different numbers is waived – the Chinese Remainder Theorem guarantees the rotation closure property [8].

Fig. 10 illustrates the use of Disco. For conciseness and clarity nodes A and B select small primes ($\{5,7\}$ for A; $\{3,13\}$ for B). Clearly, to achieve lower duty cycles, larger primes must be selected. The proposers of Disco classify the pair of primes into *balanced* – for primes that are consecutive or close (e.g. $\{37,43\}$), and *unbalanced* – when primes are not close (e.g. $\{23,157\}$). They also refer to the *symmetry* of a selection referring to the fact that nodes may select the same pair (*symmetric* selection), or a different pair of primes (*asymmetric* selection).[5]

[5] This is a slightly different definition of symmetry than the one given in Definition 2 where symmetry refers to schedules that result in the same duty cycle, which are not necessarily the same schedule.

Table 7
Summary of the validation results for the Grid Quorum estimation model presented in Eq. (2).

Validation method	Scenarios	Prediction error (%)
Implementation on TelosB motes	20×20 and 100×100 Grids with $p \sim 1$	<8
Implementation on TOSSIM (TinyOS Simulator)	20×20 and 100×100 Grids with $p = 0.78$	<9
Statistical simulations in the *R* environment	n = 20–50, 100, 500, 1000, 5000, 10,000; All probabilities from 5% to 100% with increments of 5%	<10

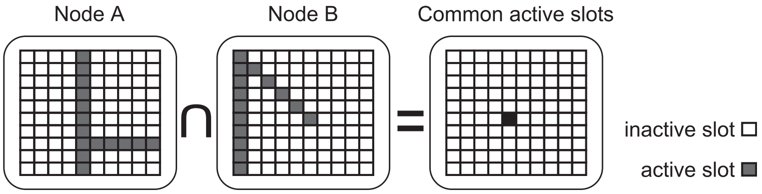


Fig. 9. An example of a 10×10 Torus system built upon a 10×10 matrix. It is not necessary to activate a complete line in order to ensure the overlapping of active slots. In fact, it is not even necessary to select slots in the same line.

Table 8
Summary of the validation results for the Torus Quorum estimation model presented in Eq. (3).

Validation method	Scenarios	Prediction error
Implementation on TelosB motes	20×20 and 100×100 Torus with $p \sim 1$	$< 3\%$ (100×100) $< 6\%$ (20×20)
Implementation on TOSSIM (TinyOS Simulator)	20×20 and 100×100 Torus with $p = 0.78$	$< 4\%$ (100×100) $< 6\%$ (20×20)
Statistical simulations in the *R* environment	n = 20–50, 100, 500, 1000, 5000, 10,000; All probabilities from 5% to 100% with increments of 5%	$< 1\%$

Table 9
NDT and duty cycle for the Grid and Torus systems.

	NDT	Duty cycle
Grid	$\frac{(3-p)n^2}{6p}$	$\frac{2}{n}$
Torus	$\frac{(2-p)n^2}{2p}$	$\frac{3}{2n}$

Though the authors of Disco do not provide proof, they present empirical results suggesting that the best configuration, in terms of NDT, happens when nodes select unbalanced-asymmetric pairs. We, once again, argue that forcing nodes to select different schedules is undesirable under asymmetric duty cycle, because of the incurring communication costs. Moreover, the authors of Disco use only symmetric pairs during the analysis of their mechanism. They also state that, after unbalanced-asymmetric pairs, the second best selection are balanced-symmetric pairs.

For the reasons above, during our modeling, we assume that nodes select the same balanced pair of primes. In fact, in Section 5.2 we prove that, according to our model of NDT, balanced pairs are the best match for symmetric selection and, since our model provides good accuracy, we believe that this is a strong argument for balanced pairs.

5.1. NDT model for Disco schedules

This section presents our model for the NDT of nodes operating under Disco. It includes a summary of the methodology employed (fully presented in Appendix D) and the results for the validation tests.

5.1.1. Methodology

In order to obtain a model of the NDT for Disco, we again analyzed every possible distribution of discovery opportunities that may occur for two unsynchronized nodes, accounting for the probability of discovery for each opportunity.

Although, as in the case of Block Designs, a combinatorial analysis is feasible, such analysis is more complex, since a change in the time offset between the neighbors affects the number of discovery opportunities per cycle. Therefore, the proof for this model, presented in Appendix D is the longer between the four schedules, and the resulting expression, even in its compact quadratic version, presented in Eq. (4), is also slightly more cumbersome.

5.1.2. NDT model

We argue that Eq. (4), gives an approximation of the expected NDT with an error proportional to $\frac{1}{q_1 q_2}$. The Eq. (4), for which a proof is provided in Appendix D, takes the parameters q_1 and q_2 (the prime numbers) and the probability of discovery p.

$$E[NDT] = \frac{q_1 q_2 (p^2 - 3p + 3)}{3p(2-p)} \tag{4}$$

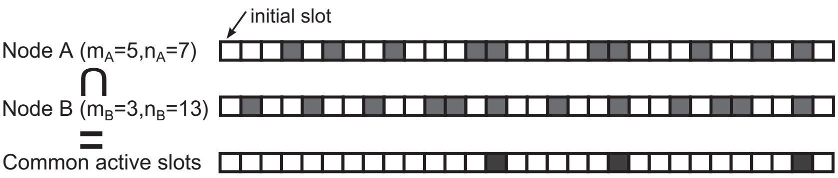


Fig. 10. Two nodes operating under Disco. Node A selects prime numbers 5 and 7, and activates slots that are multiple of these numbers. Node B selects 3 and 13 and acts likewise.

5.1.3. Model validation

The model presented in Eq. (4) was validated using the same methodology used for all other schemes. Results are presented in Table 10. For the implementations in real motes and in TOSSIM, the pairs {37,43} and {193,197} were selected. They are both *balanced pairs*, as already explained.

The statistical simulations were performed with two groups of pairs: (1) all pairs of consecutive primes that result in duty cycles between 0.75% and 2% (30 pairs of primes between {101,103} and {269,271}), (2) 24 pairs of primes that result in duty cycles between 0.9999% and 1.0001%. Group 1 is a large set of balanced pairs, that shows the model is accurate for a useful range of duty cycles. Group 2 shows that, for a relatively low duty cycle, the model is accurate for many different balancing levels. The results, summarized in Table 10, show the good estimation provided by the model presented in Eq. (4). Errors were less than 2% in the tests with motes, and tests in TOSSIM and statistical simulations presented errors of less than 3% and 2%, respectively. Notice that, in Group 1, even with relatively large duty cycles, the error was small. Since the error drops with the decrease of the duty cycles, errors even smaller are expected for pairs with larger primes. In fact, this trend was confirmed in our simulations (for example, the error was 2.07% for {101,103} and 0.96% for {269,271}).

5.2. Discussion

Through the selection of two prime numbers, it is possible for a node using Disco to operate under almost any desired duty cycle, the question being which two prime numbers to select among all possibilities. As already mentioned, the authors of Disco state that if two nodes select the same pair of primes (symmetric selection) the best results, in terms of NDT, would come from balanced pairs. Such affirmative is based on empirical data. However, it is possible to prove that, within the framework of our model, such is true for all intended duty cycles. The proof is straightforward:

Proof. The duty cycle (DC) for Disco is given by the expression bellow (q_1 and q_2 are the two selected prime numbers):

$$DC = \frac{q_1 + q_2 - 1}{q_1 \cdot q_2} \tag{5}$$

Thus,

$$q_1 = \frac{q_2 - 1}{DC \cdot q_2 - 1} \tag{6}$$

By replacing the value of q_1 given by Eq. (6) into Eq. (4), we obtain the NDT for Disco as a function of q_2 and DC. We then proceed by computing the derivative of the resultant expression with respect to q_2. By solving the equation $NDT' = 0$ for q_2, we arrive at the two following solutions:

$$q_2 = \frac{\sqrt{1 - DC} + 1}{DC} \tag{7a}$$

$$q_2 = -\frac{\sqrt{1 - DC} - 1}{DC} \tag{7b}$$

Checking the second derivative NDT'', we find that the solution (7a) minimizes the NDT, while (7b) maximizes it. Notice that (7b) is not a feasible value for q_2 anyway, because q_2 must be, at least, $1/DC$ (q_1 would be negative otherwise). By substituting (7a) back into Eq. (6), we obtain:

$$q_1 = \frac{\sqrt{1 - DC} + 1}{DC} \tag{8}$$

Hence, the minimum NDT happens when $q_1 = q_2$. Clearly, there may be the case that q_1 in Eq. (8) is not a prime number, and also, $q_1 = q_2$ is not a possible selection for Disco. However, for given values of DC and p, NDT is a convex function in the interval $(1/DC, \infty)$. Therefore, the best feasible pair of primes that result in the lowest possible NDT are the ones with q_1 and q_2 as close as possible of the values given by Eqs. (7a) and (8) and, hence, as close as possible to each other. □

6. Comparisons and the relative latency metric

Schedule-based asynchronous mechanisms are usually compared through the *power-latency metric* [13,16], which is calculated as the product of the duty cycle and the latency. In this computation, the latency is, in fact, the cycle length, and the power is represented by the duty cycle. Henceforth, the way it has been used, the power-latency metric intrinsically assumes $p = 1$. As a result, the power-latency represents the maximum NDT over perfect links, and implies that a mechanism will be always better or worse, for all link qualities, which we proved to be erroneous.

The power latency metric values for the studied schemes are presented in Table 11. Though practical and useful as a first approximation, an analysis based on such formulations results, for instance, that the Torus Quorum would always present a lower latency than Grid Quorum or Disco, for the same duty cycle.[6] However, we proved that

[6] For all values of $N > 3$.

Table 10
Summary of the validation results for Disco estimation model presented in Eq. (4).

Validation method	Scenarios	Prediction error (%)
Implementation on TelosB motes	Prime pairs {37,43} and {193,197} with $p \sim 1$	<2
Implementation on TOSSIM (TinyOS Simulator)	Prime pairs {37,43} and {193,197} with $p = 0.78$	<3
Statistical simulations in the R environment	Consecutive pairs of primes from {101,103} to {269,271} (30 pairs); plus 24 pairs of primes that result in duty cycle of about 1% (0.9999–1.0001%); all probabilities from 5% to 100% with increments of 5%	<2

Table 11
Power-latency product as it appears in the literature.

Mechanism	Latency	Duty cycle	Power-latency product
Block Designs $\{v,k,\lambda\}$	$N = v$	$DC = \frac{k}{v} = \frac{s+1}{s^2+s+1} = \frac{\sqrt{N-\frac{3}{4}}+\frac{1}{2}}{N} \sim \frac{1}{\sqrt{N}}$	$\sqrt{N}$
Torus Quorum $(n \times n)$	$N = n^2$	$DC = \frac{3\sqrt{N}}{2N}$	$3\sqrt{N}/2$
Grid Quorum $(n \times n)$	$N = n^2$	$DC = \frac{2\sqrt{N}-1}{N}$	$2\sqrt{N} - 1$
Disco, balanced $\{q_1, q_2\}$	$N = q_1 \cdot q_2$	$DC = \frac{q_1+q_2}{q_1 q_2} \sim \frac{2}{\sqrt{N}}, \left(q_1 \sim q_2 \sim \sqrt{N}\right)$	$2\sqrt{N}$

Table 12
The relative latency metric.

Mechanism	Cycle length	Latency	Duty cycle	Relative latency
Block Design (projective plane) $\{s^2+s+1, s+1, 1\}$	$L = s^2+s+1$	$\frac{L}{p} - \frac{L+1}{2}$	$DC \sim \frac{1}{\sqrt{L}}$	$\frac{2-p}{2p} \cdot \frac{1}{DC^2}$
Torus Quorum $(n \times n)$	$L = n^2$	$\frac{(2-p)L}{2p}$	$DC = \frac{3\sqrt{L}}{2L}$	$\frac{9(2-p)}{8p} \cdot \frac{1}{DC^2}$
Grid Quorum $(n \times n)$	$L = n^2$	$\frac{(3-p)L}{6p}$	$DC = \frac{2\sqrt{L}-1}{L}$	$\frac{2(3-p)}{3p} \cdot \frac{1}{DC^2}$
Disco, balanced $\{q_1, q_2\}$	$L = q_1 \cdot q_2$	$\frac{L(p^2-3p+3)}{3p(2-p)}$	$DC \sim \frac{2}{\sqrt{L}}$	$\frac{4(p^2-3p+3)}{3p(2-p)} \cdot \frac{1}{DC^2}$

this is not true, when the probability of discovery (p) is considered. The power-latency product is also misleading since the N in the formulae, which actually is the cycle length, is not the same for all schemes.

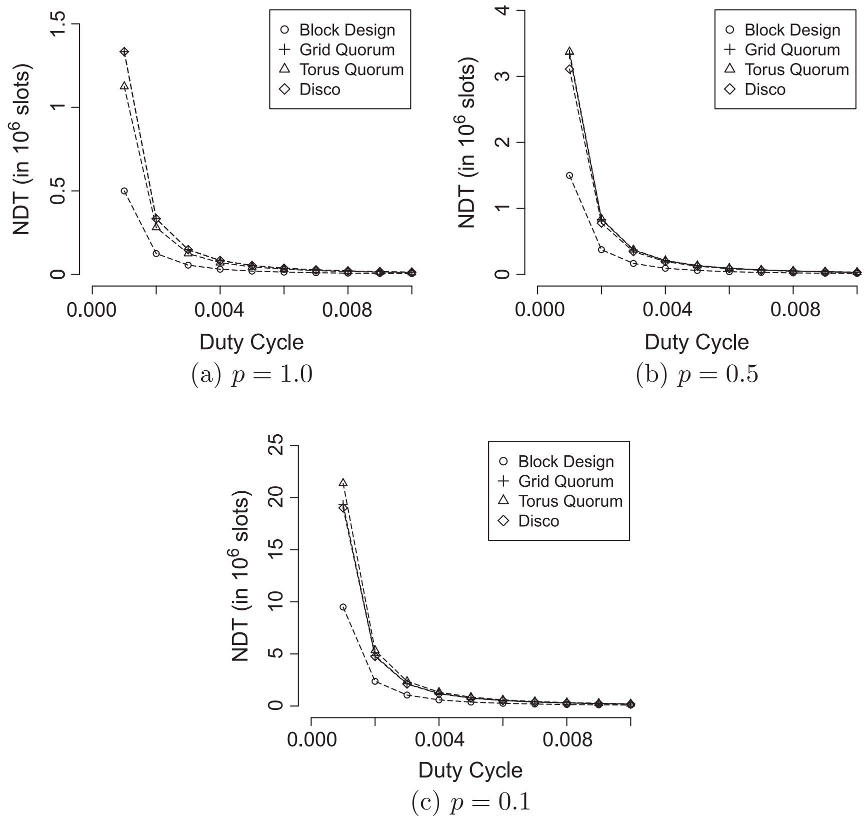
For improved accuracy and direct comparison, we propose an alternative metric, the *relative latency*, which provides the NDT as a function of the duty cycle and of p. The relative latency values for the four studied schedules are presented in Table 12.

The third column in Table 12 shows the latency as a function of the cycle length L, as given by the NDT models provided for the four asynchronous schedules in the previous sections. The relative latency is obtained by combining the expressions provided for duty cycle and latency. First we derive L as a function of DC (fourth column), and then we substitute this value in the latency expression in the third column. For example, for the Torus Quorum, from the forth column, $L = \frac{9}{4DC^2}$ and the latency is $\frac{(2-p)L}{2p}$. By substituting L in the second expression we obtain the relative latency provided in the fifth column.

The relative latency metric shows that, for a fixed duty cycle, Block Designs will (when available) result in shorter NDT, while for the other three schemes, the second best result would depend on the value of p. For example, if we take a target duty cycle of 2%, and consider $p = 1$, the resulting latency would be equivalent to 1250 slots for Block Design, 2812 slots for Torus and 3333 slots for Grid and Disco. If, however, $p = 0.5$, the latencies would be 3750 (Block Design), 8437 (Torus), 8333 (Grid) and 7777 (Disco) slots, changing the results quantitatively and qualitatively.

The direct relation of duty cycle and latency – two of the most important parameters of duty cycling schedules – are only made possible by the relative latency metric. Fig. 11 provides plots for the NDT as a function of the duty cycle for the four studied asynchronous schedules, and for three different link qualities ($p = 1.0, 0.5$ and 0.1). Once more, these graphs show the importance of considering p in the analysis of the NDT. Clearly, while Block Designs always perform best, the relative performance of the other three schedules depend on the link quality.

To help confirm the results brought by the relative latency metric and the reassure the accuracy of proposed NDT models, a direct comparison between the four schedule-based asynchronous duty cycling mechanisms studied (Block Designs, Grid Quorum, Torus Quorum and Disco) was performed by implementing the four mechanisms in TelosB motes. The selected schedules result in the same duty cycle of 1.03% (selected, because it is the lowest achievable by using Block Designs). The results after 400 trials for each method are presented in Fig. 12. The motes were tested in conditions where practically no frames were lost ($p \sim 1$). The results indicate that Block Designs perform significantly better than the other schedules in terms of average NDT. The Torus Quorum performed second best, and the Grid Quorum and Disco



(a) $p = 1.0$ (b) $p = 0.5$ (c) $p = 0.1$

Fig. 11. Relative latency used for direct comparison between Block Designs, Grid, Torus and Disco, for three different link qualities (p).

presented similar results. The results also indicate the good prediction of the contributed models. As expected, the worst prediction occurred for the Grid Quorum but, in all cases, an error of less than 4% was observed.

In order to compare the mechanisms in a wide range of reception probabilities (from 0.05 to 1.0 with increments of 0.05) we once more resorted to statistical simulations. The results for this experiment are presented in Fig. 13 and confirm that the probability of discovery is an important variable in determining which schedule will result in shorter NDT. Again, as seen in Fig. 13a, Block Designs perform better, while all other three mechanisms present similar results. Fig. 13b allows a better comparison between the two Quorum Systems and Disco, by normalizing the NDT in relation to the results for the Block Design. Clearly, though Torus performed better than Grid or Disco for high quality links, as p drops, the opposite happens and Torus results in higher NDT, in comparison to Disco and Grid.

Finally, we compare the literature assumption that the cycle length may be used as an indicator of the NDT with the predictions of our models. Fig. 14 shows how greatly both approaches differ. The four schedules were selected in order to provide diversity. In each graph, a different schedule (Block Design, Grid, Torus and Disco) and duty cycle (1.03%, 0.5%, 0.19% and 0.05%) was selected and, in each case, the result from our proposed models were compared to the cycle length, for values of p (delivery probability) from 0.05 to 1.0, with increments of 0.05. The cycle length does not change with link quality and it is represented by a horizontal line. In opposition, our models capture the effects of random offsets and of link quality, yielding more accurate estimations. Since the accuracy of our models

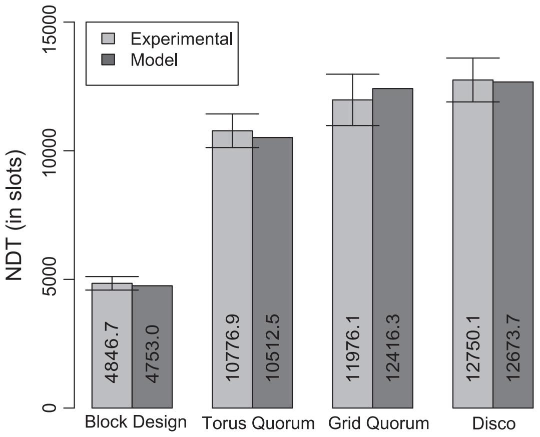
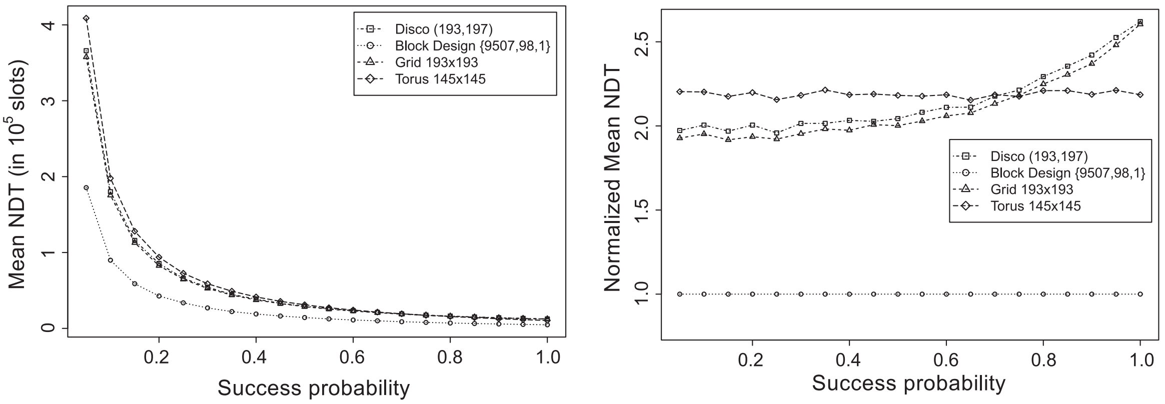


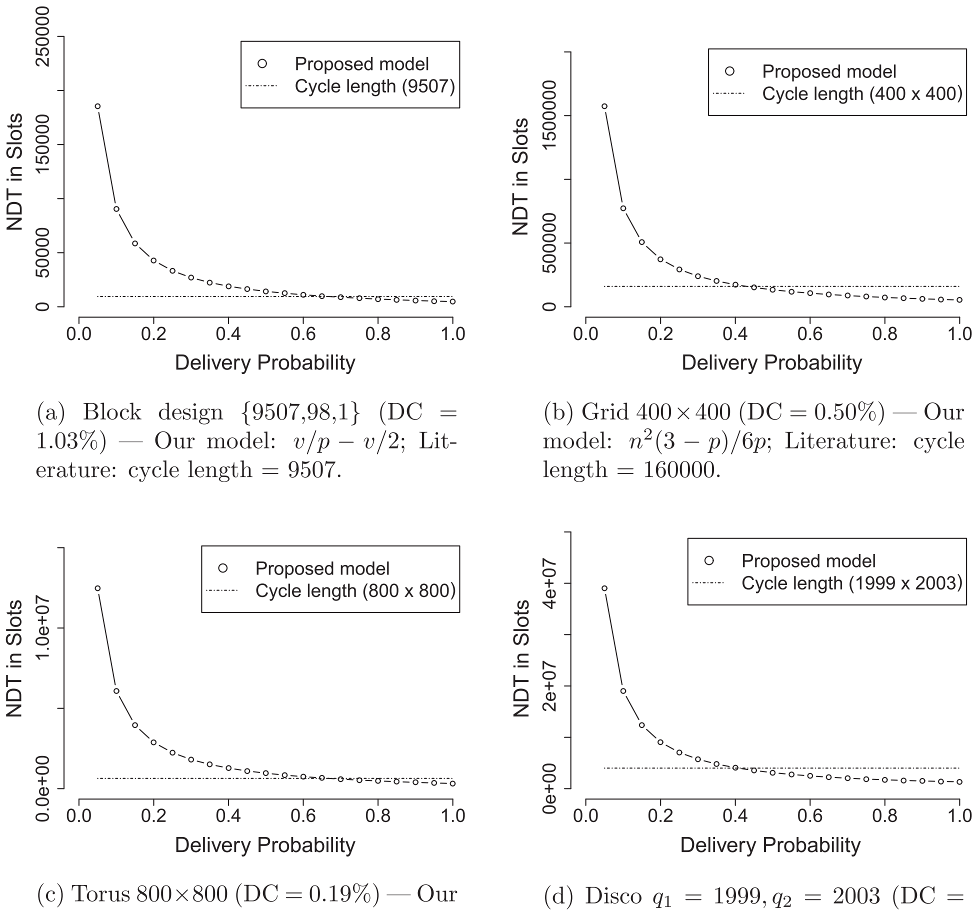
Fig. 12. Experimental NDT for four schedules implemented in TelosB motes ({9507,98,1} Block Design, 193 × 193 Grid Quorum, 145 × 145 Torus Quorum and Disco with the primes {193,197}). The confidence intervals of 95% are indicated with error bars.



(a) Mean NDT

(b) Normalized NDT

Fig. 13. A comparison between the four studied mechanisms. All schedules result in a duty cycle of 1.03%. Confidence intervals were too small due to the high number of runs (40,000) and omitted for better legibility.



(a) Block design {9507,98,1} (DC = 1.03%) — Our model: $v/p - v/2$; Literature: cycle length = 9507.

(b) Grid 400×400 (DC = 0.50%) — Our model: $n^2(3-p)/6p$; Literature: cycle length = 160000.

(c) Torus 800×800 (DC = 0.19%) — Our model: $n^2(2-p)/2p$; Literature: cycle length = 640000.

(d) Disco $q_1 = 1999, q_2 = 2003$ (DC = 0.05%) — Our model: $q_1q_2(p^2 - 3p + 3)/(3p(2-p))$; Literature: cycle length = 4003997.

Fig. 14. A comparison between our proposed models and current literature.

Table 13
Synoptic table – final comparison between Block Designs, Grid Quorum, Torus Quorum and Disco. For each mechanism: the relative latency metric, the minimum duty cycle supported, the possibility of asymmetric duty cycle operation (asym?), and the scenario where the mechanism is indicated.

Mechanism	Relative latency	Min. duty cycle	Asym?	Best scenario
Block Designs (proj. planes)	$\frac{2-p}{2p} \cdot \frac{1}{DC^2}$	1.03%	No	All cases where a symmetric duty cycle higher than 1% is acceptable
Torus Quorum	$\frac{9(2-p)}{8p} \cdot \frac{1}{DC^2}$	Any	No	Symmetric duty cycle of less than 1% and good quality links
Grid Quorum	$\frac{2(3-p)}{3p} \cdot \frac{1}{DC^2}$	Any	Yes	Asymmetric duty cycle of less than 1% or poor quality links
Disco, balanced	$\frac{4(p^2-3p+3)}{3p(2-p)} \cdot \frac{1}{DC^2}$	Any	Yes	Asymmetric duty cycle of less than 1% or poor quality links

was established, the conclusion is that using the cycle length as a measure of latency is a gross simplification. For low quality links, the NDT is in fact many times longer than the cycle length (from 10 to 20 times as long, depending on the schedule). Likewise, for good quality links, an overestimation of 100% is typical if the cycle length is assumed, while our models provide errors of less than 10% for the depicted cases (in some cases, as in the Block Designs or Torus, the error of our model is less than 1%).

6.1. Summary of the comparisons

Table 13 summarizes our comparison between the four schedule-based asynchronous duty cycling mechanisms. As demonstrated, Block Designs are indisputable if the same duty cycle can be applied to all nodes in the network, i.e. if symmetric duty cycle is acceptable. However there is currently no known Block Design that provides less than 1% of duty cycle – which is a significant disadvantage.

For duty cycles inferior to 1% the decision is more complex. For good quality links Torus performs better than Grid or Disco, but it does not support asymmetric operation. For poor quality links, or when assymetric operation is required, Grid and Disco surpass Torus. There is a slight advantage towards the Grid, but depending on the target duty cycle Disco may present better granularity.

7. Conclusions

We proposed four different models for the estimation of NDT for the four main mechanisms of schedule-based asynchronous duty cycle mechanisms. Each of these models are significantly more accurate than previous models in the literature. Thanks to these models we were able to provide useful comparisons between schedule-based asynchronous duty cycle and to support practical decisions on which mechanism should be used in each scenario.

These were the main contributions presented in this paper:

- A quick survey of the asynchronous duty cycling mechanisms.
- A formal definition and comprehensive review of the asynchronous schedule-based duty cycle schemes.
- Models for the estimation of the neighbor discovery time (NDT) for four important asynchronous schedule-based duty cycle schemes, that incorporate message loss.
- Proof that the incorporation of message loss changes the comparison between mechanisms both quantitatively and qualitatively.
- Proposal of a new metric (relative latency) that allows comparisons between the schemes.
- A synoptic comparison between the four mechanisms, indicating the scenario where they are most advantageous.

There are also contributions specific to schemes:

- Block Designs: List of all known Block Designs from literature, including designs with $\lambda > 1$ and analysis on the usefulness of designs with $\lambda > 1$.
- Quorum: Experimental data that indicate that the best selection for active column, in terms of NDT, depends on the value of p, and indication that central columns perform better than average.
- Disco: Proof that, if the proposed model is employed, balanced pairs are the best selection in terms of NDT in symmetric selection.

Appendix A. Proof: NDT model for Block Design

Proof. Let A and B be two nodes operating under a scheme of asynchronous duty cycling based on a Block Design $\{v, k, \lambda\}$. As a simplification, suppose that A and B operate under different offsets (i.e. under different blocks). Define $e_i, 1 \leqslant i \leqslant \lambda$, as the ith common active slot between both nodes within a given cycle (i.e. the ith opportunity of discovery in a cycle). Fig. A.15 exemplifies such definitions with an instance where two nodes operate under a {15,7,3} design. Clearly, for $0 < i < \lambda$:

$$E[e_{i+1} - e_i] = \frac{v+1}{\lambda+1} \quad \text{and} \quad E[e_1] = \frac{v+1}{\lambda+1} - 1$$

In this case, we can calculate *E[NDT]* from the definition of expectancy:

$$E[NDT] = \sum_{c=1}^{\infty}\sum_{i=1}^{\lambda} t_i^c \cdot p_i^c \tag{A.1}$$

where t_i^c is the time when the i_{th} discovery opportunity happens within cycle c and p_i^c is the probability that the discovery happens on that moment. But,

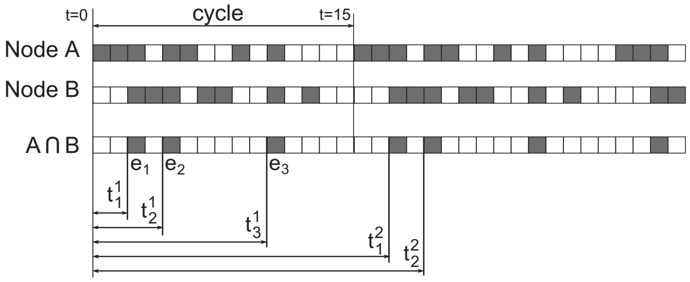


Fig. A.15. Two nodes operating under a {15,7,3} design, with an offset of two slots, give an example for the parameters used in the proof.

$$t_i^c = t_1^c + \sum_{j=1}^{i-1}\{e_{i+1} - e_i\}$$

$$E[t_i^c] = \left[v(c-1) + \left(\frac{v+1}{\lambda+1} - 1\right)\right] + (i-1)\frac{v+1}{\lambda+1} \tag{A.2}$$

and

$$p_i^c = p(1-p)^{\lambda(c-1)+i-1} \tag{A.3}$$

By substituting (A.2) and (A.3) in (A.1) and solving the summation, we find that:

$$E[NDT] = \frac{v+1}{p(\lambda+1)} - \frac{(v+1)(1-p)^{\lambda} - (\lambda+1)}{(\lambda+1)[(1-p)^{\lambda} - 1]} \qquad \square$$

Appendix B. Derivation of the NDT model for the Grid Quorum

In order to derive a model of the NDT for the Grid Quorum, we developed a script using the R language to compute the sum of the times needed until the first opportunity of encounter for all n^4 combinations of offset and starting slot, for $n = 1, \ldots, 60$. Fig. B.16a shows the values found using this computation. By applying the finite difference method to the data, it is possible to identify this curve as a 6th degree polynomial. Using a simple interpolation method, we find that the mean time until the first opportunity of encounter is given by:

$$E[e_1] = \frac{2n^6 - 2n^5 - 3n^3 + 4n^2 - n}{6n^4} \tag{B.1}$$

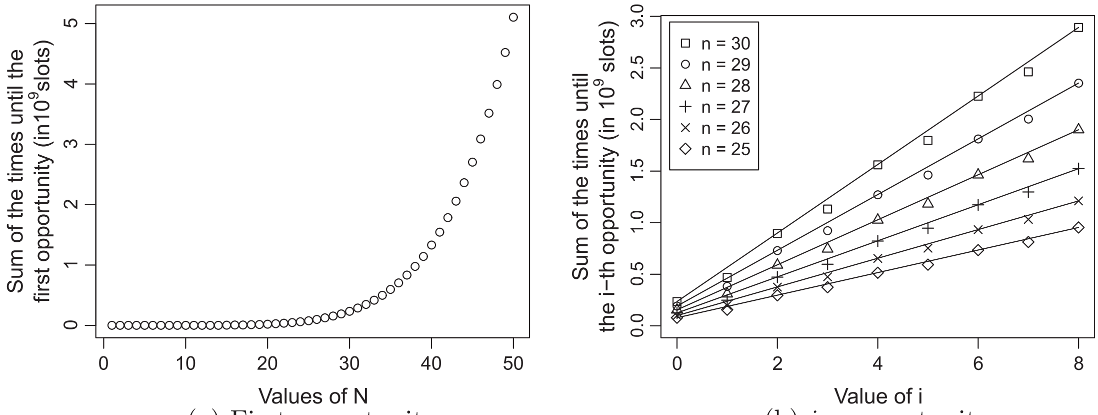
We reinforced the validity of Eq. (B.1) by additionally testing the values $n = 1000$ and $n = 10,000$, to find that the curve also fits perfectly to these values. We then proceeded our analysis by executing the same computation, but for the ith opportunity (notice that for the first opportunity, we consider $i = 0$). Fig. B.16b shows the obtained results for some values of n and i. One can notice that, for a given n, the curve is approximately linear in i. The function intersects the Y-axis at the values given by Eq. (B.1) and a reasonable angular coefficient can be computed by taking any two points. For all values of n, we chose to compute the angular coefficient based on the values for $i = 0$ and $i = 10,000$. Interestingly, the obtained coefficients behave approximately as a 6th degree polynomial function of n. By applying the least square method, we found the following approximation for the value of the angular coefficients:

$$AngularCoefficient(n) = an^6 - bn^5 + cn^4 - dn^3 + en^2 - fn + g \tag{B.2}$$

where the values of $a \ldots g$ are given on Table B.14.

Therefore, by combining Eqs. (B.1) and (B.2) we can find an approximation for the mean time until the i_{th} opportunity of encounter:

$$E[e_i] = E[e_1] + i \times AngularCoefficient(n) \tag{B.3}$$



(a) First opportunity

(b) i_{th} opportunity

Fig. B.16. Sum of the times until the first and the i_{th} opportunity for all combinations of offsets and starting slots as a function of n in a Grid system.

Table B.14
Values of constants $a \ldots g$ on Eq. (B.2).

a	b	c	d	e	f	g
0.4999	1.4636	8.3847	51.0981	315.5027	1066.6953	1119.9465

Finally, given the success probability p of a discovery, one can find the expected NDT for a Grid system of dimension n as:

$$\begin{aligned} E[NDT] = -\{[(6a-2)n^6 + (2-6b)n^5 + 6cn^4 + (3 \\ -6d)n^3 + (6e-4)n^2 + (1-6f)n + 6g]p \\ -6an^6 + 6bn^5 - 6cn^4 + 6dn^3 - 6en^2 \\ +6fn - 6g\} \cdot \frac{1}{6n^4p} \end{aligned} \quad (B.4)$$

However, for values of $n > 100$, Eq. (B.4) can be approximated, with less than 1% of error, to Eq. (B.5):

$$E[NDT] = \frac{(3-p)n^2}{6p} \quad (B.5)$$

Since $n > 199$ is the necessary condition to achieve a duty cycle of 1% (a typical requirement), such simplification is a reasonable approximation for many useful scenarios.

Appendix C. Derivation of the NDT model for Torus Quorum

Although the process for obtaining the expression for Torus is similar to the one used for Grid, in Torus we must treat odd and even values of n differently.

For even values of n, the first opportunity e_1 is given by:

$$e_1 = \frac{12n^6 - 16n^5 - 6n^4 + 11n^3 - 9n^2 + 2n}{24n^4} \quad (C.1)$$

and the i_{th} opportunity:

$$\begin{aligned} e_i = \frac{12n^6 - 16n^5 - 6n^4 + 11n^3 - 9n^2 + 2n}{24} + i \\ \cdot \frac{an^6 - bn^5 + cn^4 - dn^3 + en^2 - fn + g}{n^4}, \end{aligned} \quad (C.2)$$

where the values for constants $a \ldots g$ are given in Table C.15.

Finally, the NDT for even values of n:

$$\begin{aligned} E[NDT] = -\{[(24a-12)n^6 + (16-24b)n^5 + (24c \\ +6)n^4 + (-24d-11)n^3 + (24e+9)n^2 \\ +(-24f-2)n + 24g]p - 24(an^6 + bn^5 \\ -cn^4 + dn^3 - en^2 + fn - g)\} \cdot \frac{1}{24n^4p} \end{aligned} \quad (C.3)$$

Similarly, for odd values of n:

$$e_1 = \frac{12n^6 - 12n^5 - 6n^4 + 25n^3 - 30n^2 + 11n}{24n^4} \quad (C.4)$$

$$\begin{aligned} e_i = \frac{12n^6 - 12n^5 - 6n^4 + 25n^3 - 30n^2 + 11n}{24} + \\ i \cdot \frac{an^6 - 1.bn^5 + cn^4 - dn^3 + en^2 - fn + g}{n^4}, \end{aligned} \quad (C.5)$$

where the values for constants $a \ldots g$ are given in Table C.16.

$$\begin{aligned} E[NDT] = -\{[(24a-12)n^6 + (12-24b)n^5 + (24c+6)n^4 \\ +(-24d-25)n^3 + (24e+30)n^2 + (-24f-11)n \\ +24g]p - 24(an^6 + bn^5 - cn^4 + dn^3 - en^2 \\ +-g)\} \cdot \frac{1}{24n^4p} \end{aligned} \quad (C.6)$$

A simplification similar to that used in the case of the Grid model can also be used for the Torus – for $n > 100$. Eqs. (C.3) and (C.6) can be unified and simplified to the more manageable form presented in Eq. (C.7), with less than 1% of error.

$$E[NDT] = \frac{(2-p)n^2}{2p} \quad (C.7)$$

Appendix D. Proof: NDT model for Disco

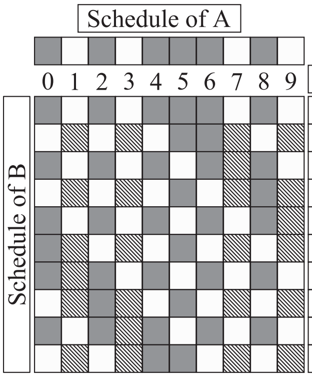
Suppose nodes A and B follow duty cycle schedules generated by Disco using parameters q_1 and q_2 (by definition, both prime numbers). Assume, without loss of generality, that we use the current slot of A as a time reference. Let $\theta, 0 \leqslant \theta < q_1 \cdot q_2$, denote the offset between the current slots of A and B. The intersection between the schedules used by A and B can be computed based on the solutions for the following modular equations (where $x_a \ldots x_d$ are the variables):

Table C.16
Values of constants $a \ldots f$ on Eq. (C.5).

a	b	c	d	e	f	g
0.9992	1.9079	7.1622	21.2087	54.0977	80.9621	43.0001

Table C.15
Values of constants $a \ldots f$ on Eq. (C.2).

a	b	c	d	e	f	g
0.9994	2.4187	6.4341	22.6085	74.7884	148.5750	118.8220



		Solutions to the modular equations							
		$\Theta + 2x_a \equiv 0(\text{mod}2)$		$\Theta + 2x_b \equiv 0(\text{mod}5)$		$\Theta + 5x_c \equiv 0(\text{mod}2)$		$\Theta + 5x_a \equiv 0(\text{mod}5)$	
Case	Θ	x_a	$\Theta + 2x_a$	x_b	$\Theta + 2x_b$	x_c	$\Theta + 5x_c$	x_d	$\Theta + 5x_d$
4	0	$\mathbb{Z}$	0,2,4,6,8	...,-5,0,5,...	0	...,-2,0,2,...	0	$\mathbb{Z}$	0,5
3	1	none	none	...,-3,2,7,...	5	...,-1,1,3,...	6	none	none
1	2	$\mathbb{Z}$	0,2,4,6,8	...,-1,4,9,...	0	...,-2,0,2,...	2	none	none
3	3	none	none	...,-4,1,6,...	5	...,-1,1,3,...	8	none	none
1	4	$\mathbb{Z}$	0,2,4,6,8	...,-2,3,8,...	0	...,-2,0,2,...	4	none	none
2	5	none	none	...,-5,0,5,...	5	...,-1,1,3,...	0	$\mathbb{Z}$	0,5
1	6	$\mathbb{Z}$	0,2,4,6,8	...,-3,2,7,...	0	...,-2,0,2,...	6	none	none
3	7	none	none	...,-1,4,9,...	5	...,-1,1,3,...	2	none	none
1	8	$\mathbb{Z}$	0,2,4,6,8	...,-4,1,6,...	0	...,-2,0,2,...	8	none	none
3	9	none	none	...,-2,3,8,...	5	...,-1,1,3,...	4	none	none

Fig. D.17. Possible offsets for the schedule generated by disco using parameters 2 and 5. White slots are inactive, slots with diagonal lines are active, and gray slots are active at the same time of an active slot of the schedule with offset 0.

$$\theta + q_1 \cdot x_a \equiv 0 \ (\text{mod} q_1) \tag{D.1a}$$
$$\theta + q_1 \cdot x_b \equiv 0 \ (\text{mod} q_2) \tag{D.1b}$$
$$\theta + q_2 \cdot x_c \equiv 0 \ (\text{mod} q_1) \tag{D.1c}$$
$$\theta + q_2 \cdot x_d \equiv 0 \ (\text{mod} q_2) \tag{D.1d}$$

In each equation, the left-hand side represents a set of active slots for *B*, while the right-hand hand represents a set of active slots for *A*. For example, the left-hand side of Eq. (D.1a) represents all active slots for *B* due to the parameter q_1 (by adding θ, we use the time reference of *A*). The right-hand side represents all active slots for *A* due to the parameter q_1 (all slots that are congruent to 0 modulo q_1). By solving each equation and replacing the values of $x_a \ldots x_d$ on the left-hand sides, one can obtain all slots in the intersection.

Consider, for example, the case $q_1 = 2$ and $q_2 = 5$. Fig. D.17 shows all possible offsets between the schedules of *A* and *B*. Since we use the current slot of *A* as a time reference, its duty cycling follows the schedule for offset 0. Node *B*, on the other hand, can use any of the schedules, depending on its offset with respect to *A*. For each schedule, each white box represents an inactive slot. The gray boxes are active slots which are also active for node *A* (i.e., slots that belong to the intersection between this schedule and the schedule used by *A*). The boxes with diagonal lines are active slots which are not in the intersection. The figure also relates each offset with the solutions for Eqs. (D.1a)–(D.1d). Consider, for instance, the schedule for offset 2. Every $x_a \in \mathbb{Z}$ is a solution for Eq. (D.1a), resulting in slots $0, 2, 4, 6, 8$ being active in the intersection.[7] Likewise, Eq. (D.1b) has a solution for slot 0 and Eq. (D.1c) has a solution for slot 2. However, Eq. (D.1d) has no solution.

In order to compute the NDT, we need to analyze the intervals (number of inactive slots) between two consecutive active slots in the intersection of the schedules. Based on the offset, there are four distinct cases, also indicated in Fig. D.17. Case 1 happens when the offset is multiple of q_1, but not of q_2. Case 2 happens when the offset is multiple of q_2, but not of q_1. Case 3 happens when the offset is neither multiple of q_1, nor of q_2. Case 4 happens when the offset is multiple of both q_1 and q_2. The expected NDT considering all cases can be computed by a weighted average of the expected NDT for each particular case. On the following sections, each case will be discussed individually.

D.1. Case 1: θ is a Multiple of q_1 but not of q_2

In this case, every $k_a \in \mathbb{Z}$ is a solution for Eq. (D.1a). Therefore, all slots multiple of q_1 belong to the intersection. Eq. (D.1b) has solutions of the form $x_b = q_2 \cdot k_b - \frac{\theta}{q_1}$, $\forall k_b \in \mathbb{Z}$. Hence, all slots of the form $q_1 q_2 (k_b + 1)$ are solutions. Notice, however, that these slots are also multiples of q_1. For Eq. (D.1c), solutions are of the form $x_c = q_1 \cdot k_c, \forall k_c \in \mathbb{Z}$, which results in slots of the form $\theta + q_1 q_2 \cdot k_c$. Again, all slots are multiple of q_1. Finally, Eq. (D.1d) has no feasible solutions.

The analysis of these equations shows that the only slots active in the intersection between the schedules of *A* and *B* are the multiples of q_1. Hence, every two consecutive active slots in the intersection are separated by $q_1 - 1$ inactive slots. As a consequence, the expected time until the first opportunity of encounter is $\frac{(q_1 - 1)}{2}$ slots. If the first opportunity fails, new opportunities happen every q_1 slots. Therefore, the expected NDT for this case, given the success probability of a discovery *p*, is:

$$E[NDT]_1 = \frac{q_1 - 1}{2} + \left(\frac{1}{p} - 1\right) \cdot q_1 = \frac{2q_1 - p(q_1 + 1)}{2p} \tag{D.2a}$$

D.2. Case 2: θ is a multiple of q_2 but not of q_1

This case is analogous to the previous one. The only slots active in the intersection are the multiples of q_2. Therefore, every two consecutive active slots in the intersection are separated by $q_2 - 1$ inactive slots. Hence, the expected NDT for this case, given the success probability of a discovery *p*, is:

$$E[NDT]_2 = \frac{q_2 - 1}{2} + \left(\frac{1}{p} - 1\right) \cdot q_2 = \frac{2q_2 - p(q_2 + 1)}{2p} \tag{D.3a}$$

D.3. Case 3: θ is not multiple of neither q_1 nor q_2

In this case, Eqs. (D.1a) and (D.1d) have no solutions. According to the Chinese Remainder Theorem[15], Eqs.

[7] We consider only the slots in the represented cycle (slots 0 to 9). The complete list of slots would be 0, 2, 4, 6, 8, 10,. . ..

(D.1b) and (D.1c) have, respectively, solutions of the form $x_b = [x_b]_0 + k_b \cdot q_2$ and $x_c = [x_c]_0 + k_c \cdot q_1, \forall k_b, k_c \in \mathbb{Z}$ (where $[x_b]_0$ and $[x_c]_0$ denote any particular solution for each equation). This means that the active slots in the intersection are of the form $(\theta + q_1 \cdot [x_b]_0) + k_b \cdot q_1 q_2$ or $(\theta + q_2 \cdot [x_c]_0) + k_c \cdot q_1 q_2$. This implies that for each cycle of $q_1 q_2$ slots, there are two active slots in the intersection (or one, if $q_1 \cdot [x_b]_0 \equiv q_2 \cdot [x_c]_0 \pmod{q_1 q_2}$).

Let $s_b = (\theta + q_1 \cdot [x_b]_0) + k_b \cdot q_1 q_2$ and $s_c = (\theta + q_2 \cdot [x_c]_0) + k_c \cdot q_1 q_2$ be two active slots in the intersection between the schedules used by node A and B, for some k_b and k_c so that s_b and s_c belong to the same cycle. Let $\Delta_\theta = s_c - s_b$ be the difference between both slots. Finally, let ND_{q_1,q_2} denote the set of all natural numbers less than $q_1 q_2$ that are not divisible by neither q_1 or q_2. Notice that all offsets that fall into this case belong to ND_{q_1,q_2}. Then, the following lemmas hold:

Lemma 1. *For any offset θ in Case 3, $\Delta_\theta \in ND_{q_1,q_2}$.*

Proof. From the definition of Δ_θ it follows that:

$$\Delta_\theta = (\theta + q_2 \cdot [x_c]_0) + k_c \cdot q_1 q_2 - [(\theta + q_1 \cdot [x_b]_0) + k_b \cdot q_1 q_2]$$

$$\Delta_\theta = q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 - (q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2) \tag{D.4a}$$

We can rewrite this equation in two equivalent forms:

$$q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 = \Delta_\theta + q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2 \tag{D.5a}$$

$$q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2 = q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 - \Delta_\theta \tag{D.5b}$$

Since $[x_b]_0$ and $[x_c]_0$ are solutions for Eqs. (D.1b) and (D.1c):

$$\theta + \Delta_\theta + q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2 \equiv 0 \pmod{q_1} \tag{D.6a}$$

$$\theta + q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 - \Delta_\theta \equiv 0 \pmod{q_2} \tag{D.6b}$$

Simplifying the expressions and rearranging the terms, it follows that:

$$\Delta_\theta \equiv -\theta \pmod{q_1} \tag{D.7a}$$

$$\Delta_\theta \equiv \theta \pmod{q_2} \tag{D.7b}$$

By definition, θ is not divisible by neither q_1 nor q_2. Hence, Δ_θ must have the same property. Moreover, since Δ_θ is the distance between two slots in the same cycle, $0 < \Delta_\theta < q_1 q_2$. □

Lemma 2. *Let $f : ND_{q_1,q_2} \mapsto ND_{q_1,q_2}$ be the function that maps any offset $\theta \in ND_{q_1,q_2}$ to its respective Δ_θ. The function f is a bijection.*

Proof. Suppose *f* is not a bijection, i.e., there exists at least one Δ_i such that $f(\theta_i) = f(\theta_j) = \Delta_i$, for some distinct $\theta_i, \theta_j \in ND_{q_1,q_2}$. Since Δ_i is a valid value of distance between active slots in an intersection, we can substitute Δ_θ for Δ_i in Eqs. (D.7a) and (D.7b):

$$\Delta_i \equiv -\theta \pmod{q_1} \tag{D.8a}$$

$$\Delta_i \equiv \theta \pmod{q_2} \tag{D.8b}$$

Solving the system for θ, we must obtain at least θ_i and θ_j as solutions. However, the Chinese Remainder Theorem guarantees that all solutions for this system are congruent modulo $q_1 q_2$. This implies either $\theta_i \notin ND_{q_1,q_2}$ or $\theta_j \notin ND_{q_1,q_2}$, which contradicts the initial hypothesis. □

Given the value of Δ_θ for a given offset, it is possible to compute the expected time until the first encounter opportunity. If the initial slot is located between s_b and s_c, the expected time until the first encounter opportunity is $\frac{\Delta_\theta - 1}{2}$. Otherwise, the expected time until the first encounter opportunity is $\frac{q_1 q_2 - \Delta_\theta - 1}{2}$. Hence, on average, the time until the first opportunity is given by:

$$\left[\Delta_\theta \frac{\Delta_\theta - 1}{2} + (q_1 q_2 - \Delta_\theta) \frac{q_1 q_2 - \Delta_\theta - 1}{2}\right] \cdot \frac{1}{q_1 q_2}$$
$$= \frac{(q_1 q_2)^2 + (-2\Delta_\theta - 1) q_1 q_2 + 2\Delta_\theta^2}{2 q_1 q_2} \tag{D.9}$$

If the first opportunity fails, nodes have to wait, respectively, $q_1 q_2 - 1$ or Δ_θ more slots. On average, the total time until the second opportunity is:

$$\left[\Delta_\theta \frac{2 q_1 q_2 - \Delta_\theta - 1}{2} + (q_1 q_2 - \Delta_\theta) \frac{q_1 q_2 + \Delta_\theta - 1}{2}\right]$$
$$\cdot \frac{1}{q_1 q_2}$$
$$= \frac{(q_1 q_2)^2 + (2\Delta_\theta - 1) q_1 q_2 - 2\Delta_\theta^2}{2 q_1 q_2} \tag{D.10}$$

Since every cycle has exactly two opportunities, the expected NDT for a given offset θ that falls into this case is:

$$\begin{aligned} E[NDT]_3(\theta) = {} & q_1 q_2 \left(\frac{1}{1 - (1-p)^2} - 1\right) \\ & + \frac{p}{1 - (1-p)^2} \\ & \cdot \frac{(q_1 q_2)^2 + (-2\Delta_\theta - 1) q_1 q_2 + 2\Delta_\theta^2}{2 q_1 q_2} \\ & + \frac{p(1-p)}{1 - (1-p)^2} \\ & \cdot \frac{(q_1 q_2)^2 + (2\Delta_\theta - 1) q_1 q_2 - 2\Delta_\theta^2}{2 q_1 q_2} \end{aligned} \tag{D.11}$$

By Lemmas 1 and 2, we can compute the expected NDT for all values of θ that fall into this case by computing the summation of $E[NDT]_3(\theta)$ for all values of $\Delta_\theta \in ND_{q_1,q_2}$ and dividing the result by $\mid ND_{q_1,q_2} \mid$. This results in a expected NDT for this case of:

$$\begin{aligned} E[NDT]_3 = {} & \left[\frac{(q_1 - 1)(2 q_1 q_2 p^2 - q_2 p^2 + 3p^2 - 6 q_1 q_2 p - 6p + 6 q_1 q_2)}{6(p-2)p}\right. \\ & + \frac{(q_2 - 1)(2 q_1 q_2 p^2 - q_1 p^2 + 3p^2 - 6 q_1 q_2 p - 6p + 6 q_1 q_2)}{6(p-2)p} \\ & \left. - \frac{(q_1 q_2 - 1)(q_1 q_2 p^2 + p^2 - 3 q_1 q_2 p - 3p + 3 q_1 q_2)}{3(p-2)p}\right] \\ & \cdot \frac{1}{q_1 q_2 - q_1 - q_2 + 1} \end{aligned} \tag{D.12}$$

D.4. Case 4: θ is 0

This case is simple from the point of view of finding out the intersections, since both A and B use the exact same schedule. However, the analysis of the expected NDT is much more complex than with the previous cases. Since the probability of occurrence of this case is $\frac{1}{q_1 q_2}$ and the most interesting schedules for energy conservation are the ones with larger values of q_1 and q_2, as an approximation we disregard this case for computing the expected NDT for Disco. Notice that the duty cycle of Disco is proportional to $1/q_1 + 1/q_2$, while the error caused by this approximation is proportional to $\frac{1}{q_1 q_2}$. Therefore, for balanced pairs $(q_1 \sim q_2)$, the error drops quadratically with the decrease of the duty cycle.

D.5. Averaging the cases

The previously analyzed cases do not happen with the same frequency. Cases 1 and 2 happen $\frac{q_2-1}{q_1 q_2-1}$ and $\frac{q_1-1}{q_1 q_2-1}$ of the times, respectively. Case 3 happens $\frac{q_1 q_2-q_1-q_2+1}{q_1 q_2-1}$. By weighting all three cases, we obtain the final expression for the NDT for Disco:

$$\begin{aligned} E[NDT] = \big[&-\big(\big(q_1{}^2 - q_1\big)q_2^2 + \big(-q_1^2 + 6q_1 - 2\big)q_2 - 2q_1 - 1\big) \\ &p^2 - \big(\big(3q_1 - 3q_1^2\big)q_2^2 + \big(3q_1^2 - 18q_1 + 6\big)q_2 + 6q_1 + 3\big) \\ &p - \big(3q_1^2 - 3q_1\big)q_2^2 - \big(-3q_1^2 + 15q_1 - 6\big)q_2 + 6q_1\big] \\ &\times \frac{1}{3(q_1 q_2 - 1)(p-2)p} \end{aligned} \tag{D.13}$$

Similarly to the other models, a simplification of the model presented in Eq. (D.13) is possible under certain assumptions. For example, for reasonably balanced pairs and duty cycles of less than 2%, $(q_1 \sim q_2 > 100)$, we can reduce Eqs. (D.13) and (D.14).

$$E[NDT] = \frac{q_1 q_2 (p^2 - 3p + 3)}{3p(2-p)} \tag{D.14}$$

References

[1] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Netw. 7 (2009) 537–568.

[2] A. Bachir, M. Dohler, T. Watteyne, K. Leung, MAC essentials for wireless sensor networks, IEEE Commun. Surv. Tut. 12 (2) (2010) 222–248.

[3] W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in: Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), vol. 3, 2002, pp. 1567–1576.

[4] G. Lu, B. Krishnamachari, C. Raghavendra, An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks, in: Proceedings of the 18th International Parallel and Distributed Processing Symposium, 2004, p. 224.

[5] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, T.-H. Lai, Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks, Mobile Networks Appl. 10 (2005) 169–181.

[6] R. Zheng, J.C. Hou, L. Sha, Asynchronous wakeup for ad hoc networks, in: Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '03), ACM, New York, NY, USA, 2003, pp. 35–45.

[7] Y.-C. Tseng, C.-S. Hsu, T.-Y. Hsieh, Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks, in: Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), vol. 1, 2002, pp. 200–209.

[8] P. Dutta, D. Culler, Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications, in: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08), ACM, New York, NY, USA, 2008, pp. 71–84.

[9] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd ACM International Conference on Embedded Networked Sensor Systems (SenSys '04), ACM, New York, NY, USA, 2004, pp. 95–107.

[10] Y. Sun, O. Gurewitz, D.B. Johnson, RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks, in: Proceedings of the 6th ACM International Conference on Embedded Networked Sensor Systems (SenSys '08), ACM, New York, NY, USA, 2008, pp. 1–14.

[11] C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, Optimizing sensor networks in the energy-latency-density design space, IEEE Trans. Mob. Comput. 1 (1) (2002) 70–80.

[12] V. Paruchuri, S. Basavaraju, A. Durresi, R. Kannan, S. Iyengar, Random asynchronous wakeup protocol for sensor networks, in: Proceedings of First International Conference on Broadband Networks (BroadNets 2004), 2004, pp. 710–717.

[13] A. Kandhalu, K. Lakshmanan, 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 (IPSN '10), ACM, New York, NY, USA, 2010, pp. 350–361.

[14] M. Maekawa, A N algorithm for mutual exclusion in decentralized systems, ACM Trans. Comput. Syst. 3 (1985) 145–159.

[15] C. Ding, D. Pei, A. Salomaa, Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996.

[16] J. Link, C. Wollgarten, S. Schupp, K. Wehrle, Perfect difference sets for neighbor discovery, in: Proceedings of the 3rd Extreme Conference of Communication (ExtremeCom 2011), 2011.

[17] F. Yates, Incomplete randomized blocks, Ann. Eugen. 7 (2) (1936) 121–140.

[18] C.J. Colbourn, J.H. Dinitz, Handbook of Combinatorial Designs, second ed., Discrete Mathematics and its Applications, Chapman & Hall/CRC, 2006.

[19] D. Gordon, La Jolla Cyclic Difference Set Repository (February 2012) <http://www.ccrwest.org/diffsets.html>.

[20] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister, System architecture directions for networked sensors, SIGPLAN Not. 35 (2000) 93–104.

[21] R. Gentleman, R. Ihaka, The R Project for Statistical Computing, February 2012 <http://www.r-project.org/>.

[22] P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: accurate and scalable simulation of entire tinyos applications, in: Proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys '03), ACM, New York, NY, USA, 2003, pp. 126–137.

[23] M. Corporation, MEMSIC, February 2012 <http://www.memsic.com/>.

Ricardo Campanha Carrano received his D.Sc. degree in Computing from Universidade Federal Fluminense (UFF), Brazil, in 2013, and his B.S and M.S. degrees in Telecommunications Engineering from the same university, in 1995 and 2008. He is a Professor at the Telecommunications Engineering Department at UFF. His current research interests include energy-efficient networking and ad hoc networks.

Diego Passos is currently a postdoctoral fellow at Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil. He received his B.Sc., M.Sc, and D.Sc degrees in computer science from the same university in 2007, 2009, and 2013. His research interests include multihop wireless networks, network coding, and wireless routing.

Luiz Claudio Schara Magalhães has a Ph.D. in Computer Science from the University of Illinois at Urbana–Champaign. He has been a professor at the Telecommunications Department at Universidade Federal Fluminense since 1994, instructor at UIUC, Visiting Scholar at HP-Labs in Palo Alto and is currently working on the creation of wired and wireless testbeds for Future Internet research. His main research interests lie in the area of mobility, large scale infrastructure to support mobile nodes, hyperconnectivity (high-redundancy last mile access) and device environment awareness and cooperation.

Célio Albuquerque (S'94-M'00) received the B.S. and M.S. degrees in electrical and electronics engineering from Universidade Federal do Rio de Janeiro, Brazil, in 1993 and 1995, and the M.S. and Ph.D. degrees in information and computer science from the University of California at Irvine in 1997 and 2000, respectively. From 2000 to 2003, he served as the networking architect for Magis Networks, designing high-speed wireless medium access control. Since 2004 he has been an Associate Professor at the Computer Science Department of Universidade Federal Fluminense, Brazil. His research interests include wireless networks, Internet architectures and protocols, multicast and multimedia services, and traffic control mechanisms.