UNIVERSIDADE FEDERAL FLUMINENSE

Ricardo Campanha Carrano

Schedule-based asynchronous duty cycling with Nested Block Designs

NITERÓI 2013

UNIVERSIDADE FEDERAL FLUMINENSE

Ricardo Campanha Carrano

Schedule-based asynchronous duty cycling with Nested Block Designs

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor em Computação. Área de concentração: Redes e Sistemas Distribuídos e Paralelos.

Orientador: Prof. Célio Vinicius Neves de Albuquerque, Ph.D.

Co-orientador: Prof. Luiz Claudio Schara Magalhães, Ph.D.

NITERÓI

2013

Schedule-based asynchronous duty cycling with Nested Block Designs

Ricardo Campanha Carrano

Tese de Doutorado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Doutor em Computação. Área de concentração: Redes e Sistemas Distribuídos e Paralelos.

Aprovada por:

Prof. Dr. Célio Vinícius Neves de Albuquerque / IC-UFF (Orientador)

Prof. Dr. Luiz Claudio Schara Magalhães / TET-UFF (Co-orientador)

Prof. Dr. Julius Leite / IC-UFF

Prof. Dr. Michael Anthony Stanton / IC-UFF

Profa. Dra. Luci Pirmez / NCE-UFRJ

Prof. Dr. Nelson Luis Saldanha da Fonseca / IC-Unicamp

Niterói, 23 de maio de 2013.

"Dilegua, o notte! Tramontate, stelle! Tramontate, stelle!

All'alba vincerò! Vincerò! Vincerò!"

— Giacomo Puccini, Giuseppe Adami and Renato Simoni, *Turandot*, Act III.

To my sweet Erika.

Agradecimentos

Começo agradecendo ao Professor Célio Albuquerque, por sua orientação constante, fundamentada e positiva. Como seu orientado, estive sempre bem acompanhado e amparado.

Ao Professor Claudio Schara, agradeço pelo imenso incentivo. Esta tese é, em boa medida, o fruto de sua visão presciente de uma carreira acadêmica que parecia improvável.

A todos no Laboratório MídiaCom, representados na pessoa sempre amável da Professora Débora Saade, muitos agradecimentos são devidos. E são particularmente devidos à Sra. Marister Outão, pelo carinho e constância com que ajuda a todos.

Um agradecimento especial, que reforça o de numeroso coro de colegas, é devido a Diego Passos, colaborador brilhante e generoso.

No âmbito pessoal, a gratidão a meus pais, Jorge e Wanda, não cabem no papel. A valorização do trabalho e do estudo, e todos os outros exemplos inestimáveis, foram e serão sempre fundamentais.

Finalmente, e principalmente, meus agradecimentos eternos a minha querida Erika, que enriquece e alegra minha existência, diariamente.

Resumo

Nos ciclos de trabalho assíncronos baseados em padrões de escalonamento, os nós de uma rede ativam e desativam suas interfaces de rádio, de acordo com um padrão especial de atividade, que garante que esses mesmos nós terão intervalos de tempos de atividade em comum, independente de seus desvios de sincronização. Quando comparada aos métodos síncronos, esta abordagem tem a vantagem de ser simples de implementar, eliminando a necessidade de protocolos de sincronização, cálculos complexos ou hardware adicional. No entanto, entre as propostas já publicadas, não há um padrão de escalonamento que resulte na menor latência para todos os cenários, quando consideradas a simetria dos ciclos de trabalho, a probabilidade de entrega dos quadros, ou a própria taxa de atividade do ciclo de trabalho. Após estudar em detalhes e modelar o comportamento dos mecanismos baseados em padrões de escalonamento, esta tese propõe o uso de configurações de bloco aninhadas — um novo padrão de escalonamento, que estende o uso dos esquemas baseados em configurações de bloco para aplicações onde estes não poderiam ser utilizados, ou o seriam de forma ineficiente. As configurações de bloco aninhadas apresentam a menor latência entre todos os padrões de escalonamento conhecidos em uma ampla gama de cenários, conforme demonstrado por modelos analíticos e implementações em dispositivos sensores reais.

Abstract

In schedule-based asynchronous duty cycling, nodes activate and deactivate their radio interfaces according to a specially designed wakeup schedule, which guarantees overlapping active time between nodes, irrespective of their synchronization offsets. When compared to synchronous duty cycling, such an approach has the advantage of being simple to implement, eliminating the need for synchronization protocols, complex computations or extra hardware. However, among published proposals, there is no single schedulebased mechanism that provides the lowest latency in all scenarios, when considering duty cycling symmetry, frame delivery probability and duty cycle rate. This thesis presents an in-depth analysis and modelling of schedule-based mechanisms and proposes *Nested Block Designs*, a new schedule that extends the use of block designs to application scenarios for which they were not possible or not as efficient to implement as other schedules. Nested block designs provide the lowest latency among known schedule-based asynchronous duty cycling mechanisms for a wide range of applications, as confirmed by analytical models and real implementations on WSN motes.

Keywords

- 1. Duty cycling
- 2. Schedule design
- 3. Latency modelling
- 4. Wireless networks
- 5. Wireless Sensor Networks
- 6. Block designs

Abbreviations

AM	:	Amplitude Modulation
DC	:	Duty Cycle
GPS	:	Global Positioning System
LCM	:	Least Common Multiple
LEACH	:	Low-Energy Adaptive Clustering Hierarchy
LPL	:	Low Power Listening
LPP	:	Low Power Probing
MAC	:	Media Access Control
NAMA	:	Node Activation Multiple Access
NDT	:	Neighbor Discovery Time
RAW	:	Random Asynchronous Wakeup Protocol
RFID	:	Radio Frequency Identification
S-MAC	:	Sensor-MAC
STEM	:	Sparse Topology and Energy Management
TDMA	:	Time Division Multiple Access
T-MAC	:	Timeout-MAC
TRAMA	:	Traffic-Adaptive MAC protocol
WSN	:	Wireless Sensor Network

Contents

Li	st of :	Figures	5		xxi
Li	st of '	Tables			xxiii
1	Intr	oductio	n		1
	1.1	Contra	ibutions o	of this thesis	2
	1.2	How t	his thesis	is organized	2
2	Dut	y cyclii	ıg		5
	2.1	The cl	hallenge o	of duty cycling	6
	2.2	Duty	cycling ta	xonomy	7
		2.2.1	Synchro	nous Schemes	8
			2.2.1.1	Rendezvous or Strictly Synchronous Schemes	9
			2.2.1.2	Skewed/Staggered Schemes	10
		2.2.2	Semi-Sy	nchronous Schemes	11
			2.2.2.1	Spontaneous Clustering	12
			2.2.2.2	Elected Cluster-head	13
		2.2.3	Asynchr	ronous Schemes	14
			2.2.3.1	Preamble Sampling	15
			2.2.3.2	Receiver-Initiated	15
			2.2.3.3	On-demand wakeup	16
			2.2.3.4	Random duty cycling	17
			2.2.3.5	Schedule-based	18

3	Sche	edule-ba	ased asynchronous duty cycling	19					
	3.1	Schedu	les, co-schedules and other definitions	20					
	3.2	A met	hod for finding the NDT of asynchronous schedule-based mechanisms	26					
		3.2.1	Discussion	30					
	3.3	Quoru	m Systems	31					
		3.3.1	Grid Quorum	31					
		3.3.2	Torus Quorum	33					
		3.3.3	Discussion	34					
			3.3.3.1 Impact of active slots selection	34					
			3.3.3.2 Torus versus Grid	37					
	3.4	Prime	numbers-based mechanisms	38					
		3.4.1	Discussion	40					
	3.5	Block	Designs	42					
		3.5.1	Discussion	45					
			3.5.1.1 Qualitative understanding of the NDT in Block Designs .	45					
			3.5.1.2 Projective Planes versus (other) Block Designs	46					
	3.6	Why a	a new schedule?	48					
		3.6.1	Closed-form expressions for the $E[NDT]$	49					
			3.6.1.1 Closed-form expression for the $E[NDT]$ of Block Designs .	49					
			3.6.1.2 Closed-form expression for the $E[NDT]$ of Grid \ldots	51					
			3.6.1.3 Closed-form expression for the $E[NDT]$ of Torus	52					
			3.6.1.4 Closed-form expression for the $E[NDT]$ of Disco	52					
		3.6.2	2 Comparisons and the relative-latency metric						
		3.6.3	The need for a new schedule	57					
4	Nest	ted Blo	ck Designs	61					
-	4.1	Design	choices	62					
	-· -								

		4.1.1 Inner design rotation	63
		4.1.2 Nesting order	63
	4.2	NDT is comparable to Block Designs	65
		4.2.1 Closed-form expression for the NDT of Nested Block Designs	67
	4.3	Duty cycle may be arbitrarily low	68
	4.4	Duty cycle granularity is significantly improved	69
	4.5	Asymmetric operation is possible	70
	4.6	Implementing Nested Block Designs	72
		4.6.1 Symmetric operation	73
		4.6.2 Asymmetric operation	74
	4.7	Nested Block Designs and other schedules	74
5	Con	clusion and Future Directions	77
0	5 1		
	5.1	Future directions	78
A	ppend	lix A - Computation of the NDT	81
٨	ppopc	liv B Closed form approximative expressions for the NDT of monotonia	
A	sche	emes	83
	B.1	Block Design schemes	83
	B.2	Grid schemes	84
	B.3	Torus schemes	86
	B.4	Disco schemes	88
		B.4.1 Case 1: θ is a Multiple of q_1 but not of q_2	89
		B.4.2 Case 2: θ is a Multiple of q_2 but not of $q_1 \ldots \ldots \ldots \ldots \ldots$	90
		B.4.3 Case 3: θ is not a Multiple of either q_1 or q_2	90
		B.4.4 Case 4: θ is 0	93
		B.4.5 Averaging the Cases	93

B.5 Nested Block Designs	94
Appendix C – Nested Block Designs: A closer look at the NDT	97
References	101

List of Figures

2.1	The proposed taxonomy for duty cycling techniques.	8
2.2	Synchronous Schemes.	9
2.3	Semi-synchronous Schemes.	12
2.4	Asynchronous Schemes	14
3.1	An example of a wakeup schedule	19
3.2	Slot border alignment.	22
3.3	An example of a co-schedule and its delta-set and Phi-coefficients	28
3.4	Two nodes operating under a 4×4 Grid Quorum	32
3.5	Two nodes operating under a 4×4 Torus Quorum	33
3.6	Equivalent and non-equivalent schedules for Grid and Torus schedules. $\ . \ .$	35
3.7	Best and worst columns in Grid	35
3.8	The NDT for three Torus schedules of same dimension	37
3.9	Torus versus Grid: comparing NDT	38
3.10	Two nodes operating under Disco	40
3.11	NDT for 15 Disco schedules of same approximate duty cycle	41
3.12	Difference in NDT as unbalancing level increases in Disco	42
3.13	Example of a Block Design	43
3.14	The behavior of the NDT in Block Designs.	47
3.15	NDT and duty cycle for known Block Designs	47
3.16	The NDT of Projective Planes and other Block Designs	48
3.17	Gap for the closed-form expression for the NDT of Block Designs	51
3.18	Gap for the closed-form expression for the NDT of Grid	52

3.19	Gap for the closed-form expression for the NDT of Torus	53
3.20	Gap for the closed-form expression for the NDT of Disco	54
3.21	Comparison with relative-latency metric.	55
3.22	Comparison between the four mechanisms	57
3.23	Experimental NDT in real sensor motes	58
3.24	A comparison between our proposed models and current literature. \ldots .	58
3.25	Decision Tree for schedule-based mechanisms.	59
4.1	An example of a Nested Block Design.	61
4.2	Rotations of the inner schedule and the NDT	64
4.3	Nesting order and the NDT	65
4.4	Comparing the NDT of Nested Block Designs and Block Designs	66
4.5	Comparing the NDT of Nested Block Designs and other schedules	67
4.6	Gap for the closed-form expression for the NDT of Nested schedules. $\ . \ .$	69
4.7	Nested schedules from Projective Planes	70
4.8	A Nested Block Design for asynchronous operation	71
4.9	Asymmetric operation with Nested Block Designs	72
4.10	New Decision Tree	75
B.1	Example of the parameters used in the proof	84
B.2	Sum of the times until the first and the i_{th} opportunity $\ldots \ldots \ldots$	85
B.3	Possible co-schedules for a Disco $(2,5)$ monotonic scheme	89
B.4	Possible values of e_1 as θ_B varies	96

List of Tables

2.1	Popular WSN motes hardware (from manufacturers data sheets)	6
2.2	Duty cycling downsides and their causes	7
3.1	Summary of the main properties of Grid schedules.	32
3.2	Summary of the main properties of Torus schedules	34
3.3	Summary of the main properties of Disco.	40
3.4	Summary of the main properties of Block Design schedules	43
3.5	Examples of Block Designs with their forming sets and duty cycles	44
3.6	Block Designs with $\lambda = 1$ (Projective Planes)	45
3.7	Block Designs with $\lambda > 1$	46
3.8	Three special cases derived from the model	50
3.9	Power-latency product as it appears in the literature	54
3.10	The relative-latency metric.	55
3.11	Synoptic table — comparing Block Designs, Grid, Torus and Disco	57
4.1	The granularity of Nested Block Designs	70
4.2	Implementation of Nested Designs on sensor motes $(p = 1.0)$	73
4.3	Implementation of Nested Designs on sensor motes $(p = 0.8)$	74
4.4	Implementation of Nested Designs on sensor motes: asymmetric operation .	75
B.1	Values of constants $a \dots g$ on Equation B.5. $\dots \dots \dots \dots \dots \dots \dots \dots$	85
B.2	Values of constants $a \dots f$ on Equation B.10	87
B.3	Values of constants $a \dots f$ on Equation B.13	87

Chapter 1

Introduction

As an engineering practice, duty cycling has long been used in a wide variety of devices to save energy and extend the lifetime of equipments, such as air-conditioning, pumps and electric motors [1]. Its application to the radio of wireless communicating devices developed in tandem with the popularization of mobile telephony and mobile computing. Such battery-operated devices are severely power constrained, and their radio interface, which is seldom used continuously, accounts for a great part of its power consumption [2], [3].

Therefore, duty cycling is a fundamental mechanism in most wireless networks and a necessity if nodes are required to operate for more than a few days before recharging the battery — a typical requirement of Wireless Sensor Networks (WSNs). In fact, duty cycling mechanisms have been proposed for most categories of multi-hop wireless networks, but it was taken to its state-of-the-art by the demands of WSNs. Sensor nodes (motes) need to be small and inexpensive, while the application data they have to transmit is often just a few bytes per second or even per day. This combination calls for small batteries and favors duty cycling.

Duty cycling mechanisms comes in many forms and impose different requirements. There is, however, a category of duty cycling in which nodes are not required to run synchronization protocols or to rely on additional hardware (as GPS or extra precision clocks). This is the category of the *schedule-based asynchronous duty cycling*. All that is required from the nodes is that they activate and deactivate their radios according to a periodic and fixed schedule, with no regard whatsoever to the synchronization offsets to their neighbors. This category also does not require any modification to MAC or physical layers and pose no extra burden to the limited computation, memory or storage of the minute devices that populate WSNs. The only downside is the longer latency due to *sleep waiting*, that happens because nodes need to wait for their neighbors to become active.

A comparison between current proposals of schedule-based asynchronous duty cycling mechanisms establishes no clear winner. Each mechanism could be a better choice, depending on the average quality of links, the target duty cycle, the tolerable latency, or the capability to change to lower or higher duty cycles, if necessary, i.e. asymmetric operation. With *Nested Block Designs* schedules, we propose a mechanism that can, as demonstrated, outperform current alternatives in any of these metrics.

1.1 Contributions of this thesis

The two main contributions of this thesis are:

- 1. A method for finding the neighbor discovery time (NDT) for any asynchronous schedule-based mechanism. This method is the first in literature to include link quality as a parameter and also the first to yield exact calculation of the expected NDT (pre-existing models are grossly inaccurate). Moreover, the proposed method unveils important aspects on the design of asynchronous schedules and helps addressing open questions on each of the studied mechanisms, making for a cluster of additional contributions.
- 2. Nested Block Design schedules, which are based on Block Designs, and address their limitations. Nested Block Designs are capable of achieving arbitrarily low duty cycles and of asymmetric operation. They also result in NDT near the optimal NDT of Block Designs, and significantly shorter than that of other schedules.

Other contributions include closed-form expressions for the NDT of all studied mechanisms, that facilitate general conclusions and comparisons among them, and also permit the creation of a new metric for the assessment of schedules — the relative-latency metric. Finally, a more detailed taxonomy of duty cycling mechanisms followed by a comprehensive study and characterization of the schedule-based asynchronous category are also among our contributions.

1.2 How this thesis is organized

The rest of this thesis is organized as follows:

• Chapter 2 discusses duty cycling and presents the most prominent proposals organized into a two-level taxonomy.

- Chapter 3 provides a comprehensive analysis of the mechanisms in the category of schedule-based asynchronous duty cycling. It includes our method and closeform models for the expected NDT, and ends with the relative-latency metric and comparisons between the proposals.
- Chapter 4 is dedicated to Nested Block Design schedules. After their description and discussion on how they should be designed, each of their features is presented. Test results, including implementation on sensor motes, close the chapter.
- Chapter 5 presents our conclusions and future directions.
- Appendix A presents an example of source code that implements our exact method to compute the NDT for any asynchronous schedule.
- Appendix B demonstrates how each of the contributed closed-form expressions for the NDT, presented for all studied schedules, was obtained.
- Appendix C provides additional detail on the analysis of the NDT of Nested Block Designs, useful for some of the suggested future work.

Chapter 2

Duty cycling

The primary objective of duty cycling in WSNs is to reduce the energy consumption of motes and to increase the overall network longevity as a consequence. More specifically, duty cycling aims at reducing idle listening, i.e. having the radio transceiver waiting in vain for a frame. The difficulty of keeping the radio on only when necessary is that, for most applications, motes do not know beforehand when data is coming.

Idle listening is not the only source of energy waste. Overhearing (when a node wastes power in listening to uninteresting frames), control packet overhead and collisions also waste power. These causes are important to identify since, while attempting to reduce idle listening, duty cycling can increase the collision rates and introduce more control traffic — side effects that can increase the very energy consumption duty cycling is trying to reduce.

Given the current state of the art in batteries, low power solid state technology and radios, duty cycling is a necessity. As an example, a TI CC2420 radio (found in many models of motes) will deplete two size AA batteries in four days of continuous activity [4]. Hence, in order to operate for one year, it should operate on a duty cycle of approximately 1%. Table 2.1 presents the current drain of typical modern radios found in sensor motes, and shows that idle listening will drain as much as tens to thousands times the current consumed during the sleeping state. Data on power consumption of microcontrollers commonly found in motes are also found in Table 2.1. Generally, the microprocessor accounts for the second highest energy consumption of a mote and, as seen in the table, active radios consume at least twice as much power as active CPUs (except for the Imote2).

Mote			RADIO		С		Power Source	
	Model		current drai	current drain		current drain		
		sleep *	idle †	active (TX/RX) ‡		sleep	active	
Mica2	TI CC1000	0.2 to $1\mu A$	0.030 to 0.105 mA	10 to 27mA	ATmega128L	$15\mu A$	8mA	2xAA batteries
Mica2dot	TI CC1000	0.2 to $1\mu A$	0.030 to $0.105~\mathrm{mA}$	10 to 27mA	ATmega128L	$15 \mu A$	8mA	3V coin cell
MicaZ	TI CC2420	$20\mu A$	0.426 mA	11 to 20mA	ATmega128L	$15 \mu A$	8mA	2xAA batteries
TelosB	TI CC2420	$20\mu A$	0.426 mA	11 to 20mA	TI MSP430	$5.1 \mu A$	$1.8 \mathrm{mA}$	2xAA batteries
Imote2	TI CC2420	$20\mu A$	0.426 mA	11 to 20mA	Marvell PXA271	$390 \mu A$	66mA	3xAAA batteries
Iris	Atmel AT86RF230	$0.02\mu A$	1.5 mA	10 to $17mA$	ATmega1281	$8\mu \dot{A}$	8mA	2xAA batteries
* Under slee	ep state, the voltage r	egulator is p	owered up but the c	rystal oscillator is not	enabled. For CC10)00 and (CC2420, th	is corresponds to the PD

Table 2.1: Popular WSN motes hardware (from manufacturers data sheets)

(Power Down) state. For AT86RF230, this state is called SLEEP state.

[†] For our purposes, a transition from *sleep* to *idle* is finished when the crystal oscillator is enabled. In the resulting *idle* state, the radio is ready for a transition either to receiving or transmitting states, or back to *sleep*. This *idle* state is called IDLE in CC2420 documentation and TRX_OFF in AT86RF230's. It has no specific name for CC1000.

‡ Current drain during transmission depends on transmission power and, for CC1000, also on the frequency used (433 or 868 MHz). Current drain during frame reception is typically of the same order of the transmission states.

2.1 The challenge of duty cycling

Duty cycling is currently a major trend in the design of Wireless Sensor Networks (WSNs), but it is also still an evolving research area. Designers aim at very low duty cycles (< 1%, meaning that the radio will be active less than 1% of the time), but to achieve that they will have to compromise on other network performance goals. Also, sensor applications have diverse requirements in terms of end-to-end delay, throughput or robustness, and the duty cycling mechanism employed for one scenario may not work efficiently in another. The downsides of duty cycling are summarized in Table 2.2 and briefly discussed next.

End-to-end message delay — Data traversing a duty cycling multi-hop network will occasionally have to wait for the next hop to wake up. This is called sleep waiting and may add significantly to end-to-end latency. Some applications will not be able to cope with that. Surveillance applications, for instance, need guarantees that a given event will be communicated timely. In fact, for hard real time applications the problem is not only the added delay but also the nondeterministic way in which that delay occurs (for more information on Real-Time MAC protocols for WSNs the reader is referred to [5]).

Collision rates — Another side effect of duty cycling is the shortening of transmission and reception time windows. If a contention-based medium access control (MAC) protocol is used, these smaller time windows will increase the probability of collisions. If TDMA is used instead, a better synchronization will be required and synchronization means an increase in control traffic, which costs energy to transmit.

Control packet overhead — Duty cycling may need extra control traffic. The most common source of this overhead is synchronization. Fine-grained synchronization requires frequent resynchronization to deal with clock skews. Designers must check if the added power drain caused by the extra control traffic overhead is compensated by the savings

ISSUE	CAUSE
Increase in end-to-end message delay	Sleep waiting — nodes have to wait other nodes to become active
Increase in collision rates	Transmissions are concentrated in shorter intervals
Control packet overhead	Duty cycling may require control messages, particularly when synchronization is needed

Tab	le :	2.2	::]	Duty	cy cy	cling	dov	vnsid	les	and	tl	heir	cause	S
-----	------	-----	------	------	-------	-------	-----	-------	-----	-----	----	------	-------	---

from duty cycling.

Notwithstanding these shortcomings, duty cycling may be the most important of all the "green" techniques. It is the most effective [3] and may be implemented in conjunction with other techniques as power control or energy aware routing, to cite two. It is also thoroughly implemented in WSNs, being included in the two main operating systems for WSNs: TinyOS [6] and Contiki [7].

2.2 Duty cycling taxonomy

Many different proposals of duty cycling mechanisms have emerged during the last decade and new ones are still being published. The usual way to classify them is in accordance with the synchronization between nodes. However, simply dividing the proposals into synchronous and asynchronous is not sufficient to provide enough understanding of the relevant literature. Two given asynchronous mechanisms may be extremely different in aspects such as topology dependency or network density requirements, resulting in different end-to-end delay or reliability, for example. In, this section, we organize the most important proposals into a taxonomy that starts on their synchronization requirements and adds another level, in order to provide useful insights into the strengths and weaknesses of the many mechanisms. The last category presented, the schedule-based asynchronous duty cycling will deserve a more detailed analysis, since nested block designs, our proposed mechanism, is a representative of this category. Therefore, schedule-based asynchronous duty cycling will be quickly introduced here and thoroughly studied in Chapter 3.

The content presented here differs from other published surveys in that it is neither concerned with the whole spectrum of techniques for energy conservation (as in [8], [3] and [9]), nor is it focused solely on some specific category of duty cycling (as in [10]) or application scenario (as in [5]). It is also more comprehensive than [11] and [12]. Our taxonomy organizes proposals in more categories and subcategories than any other taxonomy of our knowledge, henceforth providing a more detailed analysis of duty cycling proposal features.

Duty cycling schemes are usually classified into synchronous and asynchronous in



Figure 2.1: The proposed taxonomy for duty cycling techniques.

relation to the mechanism used to coordinate motes' schedules. Synchronous schemes are those where nodes are time synchronized, although the degree of synchronization varies greatly. In asynchronous methods, nodes are not required to keep a common clock.

Since pair-wise synchronization is easier to achieve than global synchronization [13], it makes sense to group neighbors into synchronized clusters and have the clusters interact with each other asynchronously, resulting in a semi-synchronous scheme (also called hybrid or semi-asynchronous). This taxonomy is summarized in Figure 2.1 and discussed in the following subsections.

2.2.1 Synchronous Schemes

In this category of duty cycling, nodes are supposed to keep common time references. This does not necessarily mean that they agree on a universal wall time, but it will imply the exchange of synchronization information to achieve and to keep the necessary degree of synchronization throughout the network. In this proposed taxonomy, synchronous mechanisms will fall into the rendezvous (or strictly synchronous) or the skewed/staggered approaches. Figure 2.2 summarizes the synchronous category.


Figure 2.2: Synchronous Schemes.

2.2.1.1 Rendezvous or Strictly Synchronous Schemes

Clearly, the most intuitive synchronous scheme is the global rendezvous scheme, where all nodes will turn their radios on and off at the same time. In a multi-hop environment, nevertheless, global rendezvous is hard to achieve since pair-wise synchronization errors tend to accumulate. Moreover, even between neighbors, strict synchronization is not easy to attain and, in general, schemes must cope with a certain degree of synchronization error. In order to keep the synchronization error tolerable, the use of synchronization protocols and occasionally of extra hardware is a prerequisite to many proposals. Either way, *guard times* are usually necessary, i.e. nodes will turn their radio on slightly before the reference time.

Most of the duty cycling proposals are actually designed to be incorporated into new MAC layers [2, 14]. MAC layers can be categorized into TDMA (Time Division Medium Access), also called contention-free, where time is divided into slots assigned to different nodes and contention-based where nodes contend for the medium, generally by listening before transmitting (carrier sensing).

RT-Link [15] is a TDMA-based link layer protocol designed to run on FireFly Sensor Nodes [16] which incorporate a GPS receiver for clock synchronization and, optionally, an AM receiver to listen to WWVB radio clock broadcasts [17] for indoor nodes. Such extra hardware will add to cost and power consumption of motes.

Another TDMA-based mechanism, TRAMA (Traffic-Adaptive MAC protocol) [18] intends to save energy by eliminating collisions and also by putting in sleep mode nodes that will not participate in the communication (as transmitters or receivers). It is assumed that global time synchronization is addressed by other mechanisms. As most of the TDMA-based approaches, both TRAMA and RT-Link include contention-based slots for node admission.

TRAMA resembles the early NAMA (Node Activation Multiple Access) [19] as it uses

a distributed election mechanism to assign time slots to nodes, avoiding collisions, but the latter was not proposed with energy savings in mind. It has been pointed out in [11] that TRAMA will result in duty cycling of at least 12.5% — high by current standards.

If application data is predictable, then data-flow information can be used as a basis for node scheduling. This is the idea behind μ -MAC [20], another TDMA-based approach. However, as pointed out in [21], μ -MAC requires long contention-based slots, strict synchronization and it is not resilient to topology changes.

If global time synchronization is available, rendezvous schemes may be advantageous, because of their high capability to coordinate transmissions, reducing idle listening and collisions at the same time. Rendezvous schemes are also inherently good for broadcast traffic. On the other hand, global synchronization is rarely available, and usually depends on extra hardware and/or significant amounts of control messages.

2.2.1.2 Skewed/Staggered Schemes

Interestingly, rendezvous schemes are not necessarily the optimal solution in terms of minimizing end-to-end delay. It has been observed that when a frame traverses the network hop by hop, from a sensor node to a sink, it may suffer what the authors of [22] called the Data Forwarding Interruption Problem. This happens when the data-flow is interrupted because an upstream node, unaware that a frame is due to come, goes to sleep before it arrives. This is, in fact, a special case of the aforementioned sleep waiting.

In order to solve the Data Forwarding Interruption Problem and reduce sleep waiting, staggered or skewed schemes have been proposed. In [22, 23, 24, 25, 26] a topology tree is formed, rooted at the sink node, and each node will schedule its wakeup in a ladder pattern, according to its depth in the topology tree.

One of the first uses of such scheme was DMAC [22]. Besides offsetting the nodes according to its hop count to the sink node, DMAC is also an adaptive mechanism in that the number of active slots will vary with the traffic load offered to a node. The authors of PELLMAC [23] suggested that DMAC suffers from intense interference, since nodes at the same offset (at the same hop distance from the sink) would contend for the medium at the same time. Therefore they proposed the use of different schedules for different branches on the topology tree. Assigning nodes to such branch trees however adds extra complexity, book-keeping and additional control traffic. Another enhancement to DMAC is LEEMAC [24], but this mechanism is focused on improving the adaptation to traffic of

DMAC, considered inefficient by the authors.

The expression *staggered schemes* is used in reference to other differentiation mechanisms for the wakeup scheduling. In [27] a graph coloring mechanism is proposed in which nodes schedule wakeups according to their assigned colors. A protocol would guarantee that a node always has at least one neighbor operating under the same color (or schedule). Although the authors call their mechanism Asynchronous Wakeup Schedule, it is debatable whether the scheme is actually asynchronous since it relies and depends upon an external synchronization protocol.

More recently [25] and [26] include proposals for a third generation of staggered synchronization mechanisms. Avoiding collisions was again the main concern in SPEED-MAC [25], but this time the idea is the introduction of a signalling period used to notify the occurrence of an event and prepare nodes downstream to forward the incoming data. The signalling would also be used as a collision detection mechanism — an unusual feature in wireless networks — in which nodes would be able to detect and notify collisions based on the difference between signal strength and background noise. The authors show their collision detection mechanism would be effective 95% of the time, but their experiments were performed in a very specific and controlled environment.

CUPID [26] proposes a bidirectional ladder scheme, pointing out that upstream traffic from nodes to sink (convergecast traffic) is not necessarily the only traffic pattern present in a WSN and a unidirectional ladder would not be a good fit for downstream traffic (configuration traffic).

Staggered or skewed schemes are topology-dependent and rely on topology discovery and maintenance. Some assumptions may not hold over a wide range of scenarios. PELLMAC, for instance, relies on disjoint paths and SPEED-MAC on a collision detection mechanism. They are also dependent on synchronization between nodes and may require guard times but, as most proposals in this category are contention-based, the synchronization requirement is less strict than that of rendezvous proposals.

2.2.2 Semi-Synchronous Schemes

In semi-synchronous proposals, summarized in Figure 2.3, neighbors are grouped into synchronized clusters and clusters interact with each other asynchronously. These schemes try to take the best of synchronous and asynchronous mechanisms. As mentioned, the main advantage of clustering is that synchronization between neighbors is easier to achieve than



Figure 2.3: Semi-synchronous Schemes.

global synchronization. On the other hand, cluster maintenance may need election mechanisms [28], [29] that require control traffic and may also be inadequate for dynamic topologies. However, there are also schemes that will form clusters spontaneously [30, 31, 32] as a consequence of synchronization among neighbors. In this proposed taxonomy, cluster coordination will be used to further categorize semi-synchronous schemes. *Spontaneous clustering* will refer to mechanisms where nodes coordinate themselves without the need of a cluster-head, while *Elected Cluster-heads* will include the mechanisms where one of the nodes in each cluster (the cluster-head) receives the special assignment of (temporarily, in most cases) coordinating cluster activity.

2.2.2.1 Spontaneous Clustering

The most seminal of the spontaneous cluster-forming proposals was S-MAC (Sensor-MAC [30]). S-MAC was inspired by PAMAS [33] and is a contention-based mechanism often used as an example of a synchronous mechanism, though it does not try to achieve global synchronization, and also does not require strict synchronization.

In S-MAC nodes form loosely synchronized virtual clusters. Synchronization is considered loose since it is performed solely with the exchange of timestamps between neighbors. Because S-MAC slot resolution is in the order of 0.5 seconds, synchronization requirements are less strict than in the proposals discussed so far. The virtual clusters form spontaneously as each node broadcasts its schedule to the neighbors. If node B listens to node A's schedule before deciding on its own, B becomes a follower of A, meaning that B adopts the schedule of A. If node B receives a different schedule after that it will adopt both schedules, i.e. nodes that participate in more than one neighborhood will have to follow schedules for all its neighborhoods, a feature frequently criticized as energy inefficient. Though influential, S-MAC may be considered outdated — its fixed and long (hundreds of milliseconds) sleep and listen periods and the overall high duty cycling (about 20%, typically) falls short of the current needs. Moreover, nodes are required to keep track of synchronization information of neighbors, and sleep waiting is considerably high when a frame traverses different clusters.

T-MAC (Timeout-MAC [31]) was proposed as an improvement over S-MAC. It adds an adaptive edge to it, by making nodes switch off dynamically whenever traffic activity in the neighborhood ceases. The resulting scheme is more energy efficient but sacrifices listen synchronization among members of a virtual cluster, resulting in the *early sleep problem* (a node goes to sleep when a neighbor still has traffic for it), identified and addressed by the authors at the cost of added complexity. DSMAC [32] is also based on SMAC but its main objective is to reduce the latency imposed by the long sleep intervals of the latter. To achieve that, DSMAC proposes an adaptive mechanism that shortens the sleep interval as traffic increases.

2.2.2.2 Elected Cluster-head

LEACH (Low-Energy Adaptive Clustering Hierarchy [28]) is a pioneer clustering-based protocol in which cluster-heads are randomly rotated to guarantee energy consumption fairness. A cluster-head will coordinate the activity of the cluster members and perform traffic aggregation to improve energy saving. In LEACH, cluster-heads are assumed to be one hop away from the sink, and hence capable of direct transmission. However, this assumption poses a limit to network scalability.

Many of the proposals in this subcategory are improvements over LEACH, such as multihop-LEACH [34], that introduces multi-hop communications between the clusterhead and the sink, and Energy-LEACH, also proposed in [34], that advocates that the selection of cluster-head should take the residual energy of candidate nodes into account.

Cluster forming in election-based mechanisms may be complex and result in significant control traffic. Position-based clustering propositions, such as GAF [29], may simplify cluster-forming by the use of a coordinate system as the one provided by a GPS device. GAF was not specifically designed for WSNs but the concept still applies — the sensorcovered area is divided into quadrants and each node will know its quadrant based on GPS readings. The problem of head election, though, is still present and a protocol is needed to solve it. Clearly, GPS will add cost and energy consumption, and is not feasible in indoor deployments.



Figure 2.4: Asynchronous Schemes.

Challenges in this subcategory include finding efficient cluster-head election and intracluster coordination mechanisms, as well as efficient ways for inter-cluster traffic relaying. These issues, along with the presentation of other cluster-based routing protocols for WSNs are discussed in [10]. In terms of duty cycling, elected cluster-head proposals tend to adopt a form of TDMA for intra-cluster communications, suggesting the use of rendezvous schemes in which nodes synchronize their duty cycles with the schedule of their cluster-heads. However, inter-cluster communication is hard to be achieved with synchronous schemes and, usually, this implies that cluster-heads will operate under higher duty cycles as, in most cases, they are also responsible for inter-cluster relaying.

2.2.3 Asynchronous Schemes

It is generally accepted that synchronizing nodes in a multi-hop wireless network is hard and costly, requiring extra hardware or processing capacities that may be high for typical sensor motes and adding frequent control traffic that takes airtime and drains precious energy to transmit [13]. In response to that, the asynchronous branch of proposals — in which nodes will not need to agree on time references — is prolific and diverse. Asynchronous proposals are illustrated in Figure 2.4.

2.2.3.1 Preamble Sampling

One important technique, first incorporated into WSNs in 2004 by B-MAC [35] and WiseMAC [36], is preamble sampling, sometimes referred to as LPL (low power listening) in the context of sensors networks. The idea is to reduce idle listening by transferring the burden of energy expenditure to the sender (just one) and removing it from the receiver (possibly many). The mechanism goes as follows. Every node goes to sleep asynchronously and wakes up periodically to check for channel activity. Since every frame is preceded by a long preamble — longer than the duration of active and sleep times together — any node will have time to wake up, detect the preamble transmission, and stay awake to receive the incoming frame, if necessary. Though influential and actually implemented in TinyOS, the disadvantages of this mechanism were soon exposed. Firstly, there is the long appropriation of the channel by the preamble that will not only be wasteful, but also prevent other nodes from transmitting. Secondly, the end-to-end latency may be too large and, finally, there is excessive overhearing, since the uninterested nodes will also remain active while hearing the preamble. In fact, TinyOS supports LPL only for the CC2420 and CC1000 radios and it is not currently (TinyOS 2.1.1) compiled in by default because of the extra memory footprint [37].

The scheme has been improved by the introduction of short preamble techniques in X-MAC [38], which substitutes an intermittent train of short frames (strobe) for the long preamble. The short frames, with short intervals between them, have the advantage of being interruptible, giving the receiver an early opportunity to acknowledge the transmitter's intention and diminish the signaling time. Also, instead of a meaningless preamble, these short frames may carry the receiver address permitting that uninterested nodes go back to sleep, minimizing unnecessary overhearing. On the downside, inserting the target receiver address in the probe frames for acknowledgement makes broadcasting difficult.

2.2.3.2 Receiver-Initiated

Another asynchronous method is the receiver-initiated transmission. Differently from the preamble sampling technique, instead of signaling that it has data to transfer, the willing sender will wait for a periodic beacon from the receiver, and transmit the frame only after that beacon is heard. This substitutes the periodic beacons for the preamble, with the advantage that the receiver beacon does not occupy the medium for as long as the sender preamble.

The idea of receiver-initiated transmissions was first introduced mainly as a collision avoidance mechanism by [39] and later applied to infrastructure WSNs (where all nodes are one-hop away from the sink) in PTIP [40]. Low Power Probing (LPP) was described in [41] as a means to distribute data from sinks to nodes. In the proposal, every node will periodically wake up, send a beacon and go back to sleep unless an acknowledgement, meant to work as a *stay awake* signal, is received shortly after. While LPP was devised as a means to wake up the network for configuration traffic, RI-MAC [42] proposed receiverinitiated communication as a more generic solution, operating over a wide range of traffic rates and patterns.

As in preamble sampling, the burden of extra energy expenditure lies on the transmitter that has to stay active until receiving the beacon from the intended receiver, but the authors of RI-MAC reported significant improvements over X-MAC, particularly when multiple data flows are present. However, Receiver-Initiated still incurs in high end-to-end delay.

2.2.3.3 On-demand wakeup

On-demand wakeup is based on the idea that a node may be removed from the sleep state when necessary. The mechanism usually relies on another communication interface, generally called wakeup radio — a low power radio that would listen to a wakeup signal and send an interruption to the CPU that would activate the primary (or data) radio in response.

As a concept on-demand wakeup is clearly advantageous, the question being whether keeping the wakeup radio active all the time would not consume more power than that saved from reducing the active time of the data radio. According to [43] that extra device must consume no more than tens of microwatts for the on-demand scheme to truly save energy. The same authors presented preliminary results with a device that would consume about $20\mu W$. Consumptions even lower (470nW in [44]) are reported, the problem being the short range of operation that would make its use on WSNs difficult.

Instead of a full wakeup radio, the authors of [45] propose radio-triggered circuits that would wake up the radio device upon detection of the radio signal, much like RFID tags do, but with the main difference that the circuit would be activated by weaker signals than those involved in RFID. In this case the question is whether the signal would be strong enough when tests are performed at typical WSN distances. Even before such hardware-based approaches became popular, proposals of on-demand mechanisms were already found in literature. One influential mechanism is STEM (Sparse Topology and Energy Management [46]). Because different radio types will have different transmission ranges, a coverage mismatch between the wakeup signal and the data transmitted by the primary interface limits the use of the so-called ultra-low radio interfaces as a wakeup device. STEM addresses that by using a regular radio as a wakeup radio. It applies duty cycling to the wakeup radio and keeps the data radio off unless demanded. When a node has data to send, it will notify that by sending a sequence of beacons through the wakeup interface and wait for the intended receiver to activate its radio and respond with a wakeup acknowledgement. Then the transmitter sends the data via the primary radio. Such scheme resembles the preamble-sampling approach and could as well be implemented with a single radio, but the advantage of that second radio, according to the authors, is the fact that ongoing transmission will not prevent or postpone the wakeup signaling. The authors argue that the added cost for the extra interface would be tolerable.

There are also proposals that claim to be on demand but are actually synchronous mechanisms. An example is DW-MAC [47] which reserves periods in which nodes signal their intention to transmit to other nodes upstream.

2.2.3.4 Random duty cycling

Another asynchronous category is the random duty cycling. The idea is that, in sufficiently dense deployments, nodes can go to sleep and wakeup randomly, since there is a high probability that there will be enough active nodes anytime. RAW (Random Asynchronous Wakeup Protocol [48]) draws on this idea. It proposes a random wakeup scheme in which the activity time of a node would be inversely proportional to the number of its neighbors and data forwarding would follow a geographical routing mechanism. In such a scheme, if a relatively high duty cycle of 5% is chosen, and supposing that a node has five neighbors capable of forwarding a frame, there will be less than 23% of chance that one of these neighbors is active when such frame is transmitted. For such probability to reach 80%, the number of neighbors should increase to 31. These numbers illustrate how dense a WSN should be for RAW to work properly.

Randomness combined with duty cycling is also found in [49]. In the proposed pseudorandom duty cycling scheme, the random seeds used to generate the duty cycling schedules are exchanged between neighbors so that each node is capable of predicting when a given neighbor will be active. The idea is to reduce latency and the authors compare their mechanism with pure random walk forwarding where nodes select the next hop randomly among all neighbors, independently of their duty cycle schedules. However, random walk forwarding results in extremely high latencies. Even with the improvements proposed in the paper, the resulting latency and duty cycling are both still too high considering the typical maximum delay demands and energy constraints of a WSN.

The random approach is restricted to very dense scenarios and the duty cycle must be carefully adjusted to the quantity of available nodes, otherwise low delivery rates are to be expected. That means that the random schemes call for adaptation to topology, what undermines the simplicity that its proponents often indicate as one of the strengths of random mechanisms. Advantages of this category include a fair distribution of traffic load (due to randomness) and low end-to-end delay (due to the elimination of sleep waiting), both subject to enough graph density.

2.2.3.5 Schedule-based

Finally, some early proposals of asynchronous duty cycling were based solely on the design of the wakeup/sleep schedule. In this category, nodes will divide time into cycles and each cycle will have active and inactive slots. These active slots will be distributed in such a way that a common active time per cycle is ensured for any two nodes, with no need for synchronization.

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 least demanding of the asynchronous techniques, not requiring extra hardware, or modifications to the MAC layer. They are also topology-independent.

Our main contribution — nested block designs — belong to this category. Therefore, schedule-based mechanisms and their more relevant representatives will be studied in detail in the next chapter.

Chapter 3

Schedule-based asynchronous duty cycling

It is not difficult to see that if two nodes are active for more than half the time, at least a portion of their active times will overlap [50]. Such simple approach, that would clearly result in high duty cycles, is an incipient example of a schedule-based mechanism. Fortunately, this line of research evolved significantly in the last decade, borrowing from other areas as Quorum systems [51] and Block Designs [52] and achieving much lower duty cycles, of 1% or less.

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. Figure 3.1 illustrates the idea.



Figure 3.1: An example of wakeup schedule. Time is divided into cycles, in this example consisting of 7 slots, either active (dark) or inactive (white).

To be used in asynchronous duty cycling, a schedule must guarantee that two nodes will have overlapping active time irrespective of their offset (difference in their slot counting). In [51] this is referred to as the *rotation closure property*. We define this property and other entities referred throughout this thesis in Section 3.1.

As noted in [51] and [50], some *Quorum systems*, such as the Grid and the Torus [51] present the rotation closure property. Quorum systems consist of a set system (set of sets) where any two elements present non-null intersections. Quorum systems have been studied in the context of distributed systems since at least 1985 [53], though initially as a solution to the mutual exclusion problem, i.e. avoiding the concurrent use of a resource.

In another category of schedule-based mechanisms, schedules are built from prime numbers, where the rotation closure property is guaranteed by the Chinese Remainder Theorem [54]. An example in this category is Disco [55].

However, optimum schedules, in terms of duty cycling, as demonstrated in [53], come from *Block Designs*. An asynchronous mechanism based on Block Designs is presented in [52], and recently the idea was revisited in [56], in which the authors advocate its use in an energy-efficient mechanism for neighbor discovery.

We start this chapter by presenting a set of definitions (Section 3.1), that are necessary for understanding schedule-based duty cycling. Then, using these definitions, we present our method for finding the neighbor discovery time (NDT) for *any* schedule in Section 3.2. Finally, the main propositions in the area are presented: Section 3.3 is dedicated to Quorum systems, Section 3.4 discusses proposals based on prime numbers, and Section 3.5 covers Block Designs. Each category will be thoroughly discussed and, at the end of the chapter (Section 3.6), they will be compared, with the identification of their major strengths and shortcomings, based on which our own mechanism will be proposed in Chapter 4.

3.1 Schedules, co-schedules and other definitions

Before discussing schedule-based propositions and our own mechanism in depth, we will first provide a set of definitions that will help clarifying important characteristics of these mechanisms. We start with the fundamental definitions of *schedule* and *co-schedule*, and then move to the definition of the operation of *rotation of a schedule* and to the concept of *scheme*. Finally, we define the *rotation closure property* and provide a more formal definition for *neighbor discovery time*, NDT — one of the most important metrics used to analyze and compare the mechanisms.

Definition 1 Schedule — The schedule of a node A, S_A , is an infinite sequence of time slots of fixed duration, that can be either active or inactive, following a pattern that repeats every w slots, the schedule cycle length. The schedule starts once the node boots up, and the number of the active slots within each cycle is q, the schedule order. The set of active slots within the first w slots (i.e. the first cycle) of a schedule defines its forming set $\{s_0, s_1, ..., s_{q-1} | s_i < w\}$. Because of its cyclic nature, a schedule may be completely defined by its cycle length, order and forming set: $S_A = \varsigma[w, q, \{s_0, s_1, ..., s_{q-1}\}]$. For shortness, we omit the parameter q, as it may be obtained from the cardinality of the forming set, and end with the notation $S_A = \varsigma[w, \{s_0, s_1, ..., s_{q-1}\}].$

• Example: $S_A = \varsigma[5, \{0, 1, 2\}]$ is a schedule formed by cycles of 5 slots, where the 3 first slots in each cycle are active: $S_A = \varsigma[5, \{0, 1, 2\}] = \{0, 1, 2, 5, 6, 7, 10, 11, 12, ...\}$.

Definition 2 Co-schedule of two schedules — The co-schedule of two schedules $S_A = \varsigma[w_A, \{a_0, a_1, ..., a_{q_A-1}\}]$ and $S_B = \varsigma[w_B, \{b_0, b_1, ..., b_{q_B-1}\}]$, noted S_{AB} , is their intersection. The resulting co-schedule is, in itself, a schedule, of length $w' = LCM(w_A, w_B)$ — the least common multiple of w_A and w_B . The forming set and the order of the co-schedule may be determined by finding the common active slots in the first w' slots of both schedules.

• Example: The co-schedule of the schedules $S_A = \varsigma[3, \{0\}]$ and $S_B = \varsigma[8, \{5, 6, 7\}]$ is the schedule $S_{AB} = \varsigma[24, \{6, 15, 21\}].$

Clearly, if the schedules of A and B represent the time slots when these nodes are active, their co-schedule determines the communication opportunities between these two nodes.

It is important to notice that the presented definitions and the following analyses assume that the time slots of A and B are border-aligned (starting and ending together), which would, actually, demand synchronization. However, as shown in [51] and [53], if a schedule satisfies the rotation closure property, slot border alignment is not a requirement for discovery opportunities to happen. Figure 3.2 illustrates this property with an example schedule — a minimum overlapping time equivalent to the duration of one slot is guaranteed for any offset (three are shown in the figure). Moreover, we assume the lack of alignment may alter the NDT negatively or positively with equal probability and, hence, it does not affect the average values of NDT. In summary, we treat the problem as if slots were aligned and argue that all conclusions and formulations presented with regard to the NDT still hold when they are not. Our claim is supported by experimental data, partially presented in Chapters 3 and 4, in which the NDT, measured from implementations in real sensor motes, is always remarkably close to the predictions from our models.

Note also that, even when programmed to operate under the same schedules, because nodes are not expected to be synchronized in their *slot counting*, they may end up operating under different schedules in respect to an external time reference. To better characterize this, we describe the idea of *rotation of a schedule*.



Figure 3.2: Three different offsets (0.5, 1.0 and 1.5 slots) between neighbor nodes operating under the same schedule illustrate that slot border alignment is not a requirement for common active time, given the schedule presents rotation closure.

Definition 3 Rotation of a schedule — The rotation of $S = \varsigma[w, \{s_0, s_1, ..., s_{q-1}\}]$ by $r, (r \in \mathbb{Z})$, noted as \overrightarrow{S}^r , is another schedule of same cycle length and order, $\overrightarrow{S}^r = \varsigma[w, \{(s_i + r) \text{ modulo } w \mid i = 0, ..., q - 1\}].$

- Example: The rotation of the schedule $S = \varsigma[5, \{0, 1, 2\}]$ by 3 is $\overrightarrow{S}^3 = \varsigma[5, \{0, 3, 4\}] = \{0, 3, 4, 5, 8, 9, 10, 13, ...\}.$
- Note: Since co-schedules are, in themselves, schedules, we may as well apply the rotation operation to co-schedules.

As already noted, in asynchronous duty cycling, nodes may operate under different schedules that may or may not be rotations of each other's schedules. Again, if two nodes are programmed to operate under the same schedule, because they are not synchronized, they will end up operating in rotations of each other schedules, dictated by their synchronization offsets. However, nodes may also operate under schedules that are not rotations of each other. For example, two nodes A and B might operate under schedules, as in the above example, where A operates under $S_A = \varsigma[3, \{0\}]$, and B under $S_B = \varsigma[8, \{5, 6, 7\}]$ and they would still be able to communicate, as their co-schedule $S_{AB} = \varsigma[24, \{6, 15, 21\}]$ is non-null. This points out the need to define another entity which captures this idea of a set of schedules that may be found in a network. We will call this entity a *scheme*.

Definition 4 Scheme — A scheme, χ , is a set of schedules: $\chi = \{S_0, S_1, ..., S_n\} = \{\varsigma_0[w_0, \{s0_0, s0_1, ..., s0_{q_0-1}\}], \varsigma_1[w_1, \{s1_0, s1_1, ..., s1_{q_1-1}\}], ..., \varsigma_n[w_n, \{sn_0, sn_1, ..., sn_{q_n-1}\}]\}.$

We find some particular schemes to be of interest and classify them as:

- Uniform scheme If all schedules in the scheme χ have the same cycle length w, we say χ is a uniform scheme in w slots. Otherwise, the scheme is non-uniform.
- Example: χ = {ς[5, {0, 1, 2}], ς[5, {1}], ς[5, {0, 1, 4}]} is an example of a uniform scheme in 5 slots, consisting of 3 schedules. The co-schedule of any two schedules in this scheme will also have the same cycle length, of 5 slots.
- Symmetric scheme If all schedules in the scheme χ have the same proportion
 of order to cycle length, (q/w which translates to the same duty cycle), we say
 that χ is a symmetric scheme. Otherwise, we say the scheme is asymmetric.
- Example: χ = {ς[6, {0, 1, 2}], ς[4, {1, 2}], ς[6, {0, 1, 4}}] is a symmetric scheme with duty cycle of 50%.
- Monotonic scheme A scheme that contains all possible different rotations of a given schedule, and no other schedule, is a monotonic scheme. As monotonic schemes are important to our analysis, we shall note them compactly as *\overline{\chi}*[S], where S is a schedule that may be used to generate (by successive rotations) all other schedules in the scheme, or alternatively: *\overline{\chi}*[w, {s₀, s₁, ..., s_{q-1}}]. We call schemes that does not hold this property non-monotonic.
- Example: χ = {ς[4, {0, 1, 2}], ς[4, {1, 2, 3}], ς[4, {0, 2, 3}], ς[4, {0, 1, 3}]} is an example of a monotonic scheme in 4 slots. Using our compact notation this scheme could be represented as χ[4, {0, 1, 2}]
- Note: A monotonic scheme will be always symmetric and uniform, but the opposite may not hold.
- Example: χ = {ς[5, {0, 1, 2}], ς[5, {1, 2, 3}], ς[5, {0, 1, 4}]} is an example of a symmetric uniform scheme in 5 slots, consisting of 3 schedules, all with duty cycle of 60%. However, χ is non-monotonic.

When all nodes in an unsynchronized network are programmed to operate under the same schedule, we have a monotonic scheme. Monotonic schemes are, in fact, the most common schemes to be found since, in most cases, there is no reason to program different schedules to each node, a procedure that would, in fact, be more laborious than simply programming the same schedule. In some instances, however, a network may need asymmetric schemes in order to accommodate different battery conditions, that may come as a result of different workloads for identical nodes, or because nodes with different capacities were mixed. This capability for *asymmetric operation* will be relevant to our future discussions.

One advantage of the schedule approach is its simplicity of implementation, since it is sufficient that nodes operate under the proper wakeup schedule. A mechanism may rely on a monotonic scheme (where all nodes operate under the rotations of a schedule), or in a non-monotonic scheme. In both cases, the scheme must be designed in a way that any two nodes will have overlapping active time, irrespective of their time offsets.

What ensures that a scheme provides these opportunities of discovery is the property of *rotation closure* [51]. Without it, 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. The rotation closure property may now be easily defined within our framework of definitions.

Definition 5 Rotation closure property — A scheme χ presents the rotation closure property if, and only if, any rotations of any two of its schedules have non-null intersection, i.e. a non-null co-schedule: $\forall S_i, S_j \in \chi : S_i \cap \overrightarrow{S_j}^k \neq \emptyset, k = 0, 1, ...$

- Example: The scheme $\chi_a = \{\varsigma[4, \{0, 1, 2\}], \varsigma[4, \{1, 2, 3\}], \varsigma[4, \{0, 2, 3\}]\}$ presents the rotation closure property, while $\chi_b = \{\varsigma[4, \{0, 1\}], \varsigma[4, \{0, 2\}], \varsigma[4, \{1, 2\}]\}$ does not, since $\{0, 1\} \cap \{(1+1) \text{ modulo } 4, (2+1) \text{ modulo } 4\} = \emptyset$.
- Note: Non-null intersection between any two schedules is a necessary condition for the rotation closure. However, as seen in the example of χ_b above, this condition is not sufficient — rotations must also be considered.

In a practical asynchronous network, all rotations of any schedule in a scheme may occur, because of the random slot counting offsets between nodes. Therefore, a practical scheme will be the union of one or more monotonic schemes, and should present rotation closure. As a consequence, all schedules may be designed in a way that a monotonic scheme built from it presents the rotation closure property. As we will see, this is the case for the schedules studied in the following sections (Quorum systems, Block Designs and Disco) and also a requisite for our own proposition (Nested Block Designs).

We are now able to define the Neighbor Discovery Time, the quantity extensively used throughout the text to analyze and compare the main asynchronous schemes. **Definition 6** Neighbor Discovery Time — Say A and B are two nodes operating under duty cycling schedules from a scheme with rotation closure, and that each node broadcasts a beacon during each of its active slots. Define t_0 as the moment when A and B are placed within communication range of each other. If p > 0 is the probability of A receiving a beacon from B, the neighbor discovery time NDT is the time, measured in number of time slots, that will elapse from t_0 until A hears the first message from B.

Definition 7 Expected Neighbor Discovery Time — The expected NDT, E[NDT] is the average NDT, as in Definition 6, for all possible slot counting offsets θ and for a fixed message delivery probability p.

We are particularly interested in obtaining the expected value for the NDT (E[NDT]), for a monotonic scheme, where all nodes operate under the same schedule, but with random offsets. Although we do not exclude non-monotonic schemes from our analysis, we will be soon comparing the NDT of the monotonic schemes obtained from the main propositions in the literature. Therefore, we refine our definition of the NDT for this more specific scenario:

Definition 8 Expected Neighbor Discovery Time for Monotonic Schemes — Say A and B are two nodes operating under rotations of the same duty cycling schedule from a monotonic scheme with rotation closure. Assume such schedule has cycle length w and order q and that the schedule of A is θ time slots ahead of the schedule of B $(S_A = \vec{S}_B^{\theta})$, where θ may be any number of slots, with equal probability. Assume also that each node broadcasts a beacon during each of its active slots, and define t_0 as the moment when A and B are placed within communication range of each other. If p > 0 is the probability of A receiving a beacon from B, the neighbor discovery time (NDT) is the expected number of time slots that will elapse from t_0 until A hears the first message from B, weighting all possible offsets θ .

To propose a final and useful definition, we note that, if S_B is a rotation of S_A $(S_B = \overrightarrow{S}_A^{\theta})$, it follows that they will form equivalent monotonic schemes $(\overline{\chi}[S_A] \equiv \overline{\chi}[\overrightarrow{S}_A^{\theta}])$ and, therefore, result in the same E[NDT]. We will define this latency-equivalence simply as equivalence.

Definition 9 Equivalent schedules — We say two schedules S_1 and S_2 are equivalent if they are rotations of each other ($\exists r: S_1 = \overrightarrow{S}_2^r$).

Once equipped with clear definitions of schedule, scheme and NDT, we move to our contributed method for finding the NDT for any asynchronous scheme.

3.2 A method for finding the NDT of asynchronous schedule-based mechanisms

In order to find E[NDT] between A and B, given their schedules S_A and S_B and subject to beacon reception probability p, we need to consider all possible co-schedules that these two schedules may result in, as a function of their offset θ , $S_{AB}(\theta)$, and then, all possible starting slots of each of these co-schedules, in respect to t_0 (the moment we start counting the time). Algorithm 1 summarizes this procedure. In the outer loop, all possible coschedules that two schedules may result in (as a function of their offset θ) are calculated and then, in the inner loop, all possible starting positions of each of these co-schedules, with respect to t_0 are considered.

Computing all co-schedules (outer loop) is a straightforward algorithm to find the intersection between two vectors, with complexity O(w'), where w' is given by the cycle length of the schedules of both nodes, w_A and w_B^1 . As for the inner loop, we now present a method with complexity $O(q^2)$, where q is the average number of discovery opportunities in the co-schedules. Therefore, the complete procedure results in an algorithm of complexity² $O(w' \times q^2)$.

```
1 E[NDT] \leftarrow 0;

2 for \theta \leftarrow 0 to w - 1 do

3 \begin{vmatrix} S_{AB}(\theta) \leftarrow S[w] \cap \rho(S[w], \theta); \\ 4 & \text{for } t_0 \leftarrow 0 \text{ to } w - 1 \text{ do} \\ 5 & | Compute E[NDT]^{\theta} \text{ for } S_{AB}(\theta); \\ 6 & \text{end} \\ 7 & E[NDT] \leftarrow E[NDT] + E[NDT]^{\theta}; \\ 8 & \text{end} \\ 9 & E[NDT] \leftarrow E[NDT]/w \\ Algorithm 1: Finding E[NDT] \text{ for a given schedule.} \end{cases}
```

To find the E[NDT] for a given $S_{AB}(\theta)$, we define Φ_i as the average number of slots until the i_{th} discovery opportunity, and p as the probability of message delivery. Clearly,

¹Remember that $w' = LCM(w_A, w_B)$ and, for a monotonic scheme, $w' = w_A = w_B$

²This algorithm is not intended to be executed on motes, but on the computer of the network designer and, given the typical values of w and q, and based on our own computations, it runs in a few seconds or less.

$$E[NDT] = p\Phi_0 + p(1-p)\Phi_1 + p(1-p)^2\Phi_2 + \dots$$
(3.1)

As $S_{AB}(\theta)$ is cyclic, if q is the number of discovery opportunities per cycle, then $\Phi_{n+q} = \Phi_n + w'$, and Equation 3.1 can be rewritten as:

$$\begin{split} E[\text{NDT}] &= p\Phi_0 + p(1-p)\Phi_1 + \dots + p(1-p)^{q-1}\Phi_{q-1} \\ &+ p(1-p)^q(\Phi_0 + w') + p(1-p)^{q+1}(\Phi_1 + w') + \dots + p(1-p)^{2q-1}(\Phi_{q-1} + w') \\ &+ p(1-p)^{2q}(\Phi_0 + 2w') + p(1-p)^{2q+1}(\Phi_1 + 2w') + \dots + p(1-p)^{3q-1}(\Phi_{q-1} + 2w') \\ &+ \dots \end{split}$$

Define c as the number of cycles (of w' slots) until discovery, then:

$$E[NDT] = p \sum_{c=0}^{\infty} \sum_{i=0}^{q-1} (\Phi_i + w'c)(1-p)^{cq+i}$$
(3.3)

The problem now is reduced to finding these Phi-coefficients, Φ_i , that represent each of the discovery opportunities, and that we define more formally as:

Definition 10 *Phi-coefficients of a co-schedule* — *Given a co-schedule* $S_{AB}(\theta)$, the i_{th} *Phi-coefficient of* $S_{AB}(\theta)$, Φ_i , is the sum of the i_{th} elements of all rotations of $S_{AB}(\theta)$ divided by w', the co-schedule cycle length.

• Note: Because $\Phi_{n+q} = \Phi_n + w'$, there are only $q = |S_{AB}(\theta)|$ different Phi-coefficients for $S_{AB}(\theta)$: $\Phi_0, \Phi_1, ..., \Phi_{q-1}$, i.e. there are as many Phi-coefficients as there are discovery opportunities within each cycle.

We use Figure 3.3 to illustrate our method of obtaining the Phi-coefficients of a coschedule. The figure shows all rotations of a co-schedule $S_{AB}(\theta) = \varsigma[7, \{2, 3, 5\}]$. As expected, there are 7 (w') rotations of $S_{AB}(\theta)$, and they may be divided in three groups such that each group starts with a rotation of $S_{AB}(\theta)$ that contains the element zero (an opportunity of discovery in the first slot) and contains all other successive rotations that does not contain slot zero.

Within each group, the three opportunities of discovery follow a fixed relation that is determined by the *delta-set* of the co-schedule — the difference between successive discovery opportunities, defined as follows:

(3.2)



Figure 3.3: An example of a co-schedule $S_{AB}(\theta) = \varsigma[7, \{2, 3, 5\}]$, with the elements of its delta-set $\{1,2,4\}$ and the Phi-coefficients $\{1,3,5\}$. Partial phi-coefficients are computed for each of the three groups of rotations of the co-schedule and then summed up to find the Phi-coefficients. Note that any rotation of this co-schedule results in the same Phi-coefficients.

Definition 11 Delta-set of a co-schedule — The delta-set of a co-schedule $S_{AB}(\theta) = \varsigma[w, \{o_0, o_1, ..., o_{q-1}\}]$ is the sequence $\Delta\{S_{AB}(\theta)\} = \{\delta_0, \delta_1, ..., \delta_{q-1} | \delta_i = o_i - o_{(i-1) \mod w}\}.$

• Note: The delta-set of the rotation of a co-schedule $\Delta\{\vec{S}_{AB}^r(\theta)\}$ is a cyclic permutation of the delta-set of the schedule $\Delta\{S_{AB}(\theta)\}$.

Key to our method is the fact that all relations between discovery opportunities may be defined in terms of the delta-set of a co-schedule. In the topmost group in Figure 3.3, for instance, the second opportunity (mid-gray) always occurs exactly one slot (δ_0) after the first opportunity (light gray). Likewise, the third opportunity (dark gray) happens two slots (δ_1) after the second. As for the first opportunity, it occurs in all slots from zero to three, i.e. from 0 to $\delta_2 - 1$, successively.

In fact, similar relations may be observed in all three groups. The only difference being the order that each δ should be applied. The very size of each group also comes from the delta-set (1, 2 and 4 rotations of $S_{AB}(\theta)$). In short, all Phi-coefficients may be obtained entirely from the delta-set and are given by the following expressions:

$$\Phi_{0} = \frac{1}{2w'} \sum_{i=0}^{q-1} \delta_{i} (\delta_{i} - 1)$$

$$\Phi_{1} = \Phi_{0} + \frac{1}{w'} \sum_{i=0}^{q-1} \delta_{i} (\delta_{(i-1) \mod u \log q})$$

$$\Phi_{i} = \Phi_{0} + \frac{1}{w'} \sum_{r=1}^{i} \sum_{j=0}^{q-1} \delta_{i} (\delta_{(j-r) \mod u \log q})$$

Finally, Equation 3.4 gives the NDT of a co-schedule $S_{AB}(\theta)$.

$$E[NDT] = p \sum_{c=0}^{\infty} \sum_{i=0}^{q-1} (\Phi_i + w'c)(1-p)^{cq+i}$$
(3.4)

where,

$$\Phi_{i} = \Phi_{0} + \frac{1}{w'} \sum_{r=1}^{i} \sum_{j=0}^{q-1} \delta_{i}(\delta_{(j-r) \ modulo \ q})$$
(3.5)

and,

$$\Phi_0 = \frac{1}{2w'} \sum_{i=0}^{q-1} \delta_i (\delta_i - 1)$$
(3.6)

A more intuitive form for Equation 3.4 may be found by solving the infinite summations, as follows:

$$E[NDT] = \frac{p}{w'} \sum_{i=0}^{q-1} (1-p)^i \left\{ \Phi_i \sum_{c=0}^{\infty} (1-p)^{cq} + w'^2 \sum_{c=0}^{\infty} c(1-p)^{cq} \right\}$$
$$E[NDT] = \frac{p}{w'} \sum_{i=0}^{q-1} (1-p)^i \left\{ \Phi_i \frac{1}{1-(1-p)^q} + w'^2 \frac{(1-p)^q}{[1-(1-p)^q]^2} \right\}$$
$$E[NDT] = \left[\frac{1}{1-(1-p)^q} - 1 \right] \cdot w' + \sum_{i=0}^{q-1} \frac{p(1-p)^i}{1-(1-p)^q} \frac{\Phi_i}{w'}$$
(3.7)

Equation 3.7, which is equivalent to Equation 3.4, is formed by two terms with clear meanings. The first term is the expected number of cycles (given the success probability for each cycle $[1 - (1 - p)^q]$) times the size of a cycle (w'). The second term is the average of the expected distances to each of the q discovery opportunities (Φ_i/w') weighted by their respective success probabilities, given that the encounter will happen within a given cycle $\frac{p(1-p)^i}{1-(1-p)^q}$.

Finally, Equation 3.2 provides an alternative recursive formula for finding the Phicoefficients that can be easily computed (see Appendix A for code example).

$$\Phi_{i} = \begin{cases} \frac{1}{2w'} \sum_{j=0}^{q-1} \delta_{j}(\delta_{j}-1), & \text{if } i = 0\\ \Phi_{i-1} + \frac{1}{w'} \sum_{j=0}^{k-1} \delta_{j} \delta_{(j-i) \mod ulo \ k}, & \text{otherwise.} \end{cases}$$

3.2.1 Discussion

Some interesting conclusions on the design of asynchronous schedules can be drawn from the proposed model.

- 1. Two monotonic schemes formed from two schedules of same length and same duty cycle do not necessarily result in the same NDT (unless the two schedules are equivalent, in which case the schemes would be the same). Hence, there is merit in designing schedules to minimize the NDT. However, minimizing the NDT without considering the delivery probability p (as in [53] or [52]) is not enough. A schedule that performs better when the link quality is high, may be surpassed when this quality drops.
- 2. The way active slots are distributed within the schedule is important, for it affects the way discovery opportunities are distributed within the co-schedule.
- 3. Not only the number of discovery opportunities counts. The interval between these opportunities, i.e. the way they occur within a cycle, also affects the NDT. If two schedules present the same duty cycle and result in monotonic schemes with the same average order (number of discovery opportunities per cycle averaged for all offsets), the one where these opportunities are more regularly distributed will produce the shorter NDT for good link quality. This happens because Φ_0 , which is the most significant coefficient for $p \sim 1$ (and the only one, if p = 1), increases with the standard deviation of the delta-set of the co-schedule.

In short, the design of asynchronous wakeup schedules is the quest for low duty cycle schedules, which result in schemes that satisfy the rotation closure property, while providing acceptable E[NDT]. There are, however, a few more desirable characteristics in the resulting scheme. Asymmetric schemes, for instance, may be useful in deployments

where nodes need to operate under different duty cycles, and since asymmetric schemes are necessarily non-monotonic, these schemes are harder to obtain. Also, the duty cycle range of a mechanism — the range of duty cycles that are possible under the mechanism — is clearly important, as also is the duty cycle granularity of a mechanism, i.e. the availability of duty cycle values within a given duty cycle sub-range. We now discuss the most prominent asynchronous schemes and highlight each of these properties.

3.3 Quorum Systems

A Quorum is a set system where the intersection of any two sets, its elements, is never null. At least three systems may be found throughout the networking literature: the Grid [51, 50], the Torus [51, 50] and the e-Torus [51]. The last one was proposed as a basis for an adaptive mechanism where nodes are capable of determining their level of mobility — an added complexity that is not assumed in other mechanisms. Because of that, the e-Torus departs from the category of schedule-based mechanisms and we will limit our analysis to the first two propositions — the Grid and the Torus.

3.3.1 Grid Quorum

To define whether a slot is active or not in an asynchronous schedule based on a Grid Quorum, each cycle is equated to an $n \times n$ matrix where each cell represents a time slot. The first slot is represented by the cell in (1,1) (line=1, column=1), the second slot by cell (1,2). Cell (2,1) represents time slot n+1, and so on, line after line of the matrix. In a Grid Quorum schedule, a node will activate only the slots corresponding to a single column and to a single line of the matrix.

Figure 3.4 depicts an example of two Grid schedules based on a 4×4 matrix. As the figure illustrates, nodes A and B are not demanded to select the same combination of lines and columns. The scheme suggested in the figure is uniform and symmetric, but not monotonic³, since $S_A = \varsigma[16, \{1, 5, 9, 12, 13, 14, 15\}]$ is not a rotation of $S_B =$ $\varsigma[16, \{2, 4, 5, 6, 7, 10, 14\}]$. In this example, there is an offset of four slots between nodes A and B, and the resulting co-schedule is the intersection between S_A and \overrightarrow{S}_B^4 , which is $S_{AB} = \varsigma[16, \{2, 6, 10, 14\}]$.

Table 3.1 summarizes important characteristics of the Grid Quorum schedules. A

 $^{^{3}}$ The capacity of creating both monotonic and non-monotonic schemes with rotation closure is, as we will see, an advantage of Grid.



Figure 3.4: Two nodes operating under a 4×4 Grid Quorum.

Parameters	n	grid dimension
	line	active line
	column	active column
Cycle length		n^2
Duty cycle		$\frac{2n-1}{n^2}$
Duty cycle ra	ange	(0,1]
Granularity	1 to 10%	180
	0.1 to 1%	1800
	0.01 to $0.1%$	18000
Asymmetric operation?		yes

Table 3.1: Summary of the main properties of Grid schedules.

Grid schedule is completely characterized by its dimension, n, and the line and column selected as active. The cycle length for a schedule generated from an $n \times n$ grid is n^2 and the resulting duty cycle is $\frac{2n-1}{n^2}$. The range refers to the range of duty cycles achievable by the mechanism. As n can be arbitrarily large, there is no lower limit to the duty cycle of a Grid schedule, and if we consider dimension n = 1, the resulting duty cycle will be 1 (or 100%).

The granularity is the number of different possible configurations in three given duty cycle ranges, selected as to characterize (1) future demands for ultra-low duty cycle (0.01 to 0.1%), (2) most of the current mainstream propositions (0.1 to 1%) and (3) early propositions (1 to 10%) of duty cycling. There are, for example, 1800 different possible duty cycles in the second interval.

The line Asymmetric operation refers to the possibility of having different nodes operating under different duty cycles coexisting as neighbors. In fact, Grid schedules of different dimensions will always have non-null intersections. To understand that, suppose two nodes, A and B, operate under schedules based on the grids $n_A \times n_A$ and $n_B \times n_B$, $n_A > n_B$. Because there will be a sequence of n_A consecutive active slots in the schedule of A (due to the active line in the schedule matrix of A) and, since there is at least one active slot every n_B slots in the schedule of B (due to the active column in the schedule matrix of B), and considering $n_A > n_B$, it follows necessarily that there will be at least one opportunity of discovery per cycle in the co-schedule of A and B. This means that any grid schedule is able to interoperate with any other grid schedule, of any dimension. Therefore, grid schemes intrinsically adhere to the rotation closure property and support asymmetric operation, a fact that has been neglected by current literature (See Table 2, in [55]).

3.3.2 Torus Quorum

A Torus Quorum system was proposed as an improvement over the Grid in the sense that it requires fewer active slots to ensure the rotation closure property. In a Torus, schedule a cycle is again equated to a matrix and each node should again activate a complete column of slots. However, instead of selecting a complete line, each node needs only to select one slot in each column c+i, $i = 1, ..., \lfloor n/2 \rfloor$ (in a $n \times n$ matrix, where c is the active column). We will call this set of slots not in the selected column, the *horizontal slots*, as calling them a line would be incorrect.

Figure 3.5 shows an example of two Torus schedules based on a 4×4 matrix, whereas only two additional slots are activated as horizontal slots, instead of a complete line. As in the case of the Grid Quorum, nodes are not required to synchronize their slot counting, nor to select the same columns or horizontal slots.



Figure 3.5: Two nodes operating under a 4×4 Torus Quorum.

Table 3.2 shows the properties of schedules obtained from Torus. Its parameters are n, the torus *dimension*, the *active column* and the set of *horizontal slots*. Cycle length is n^2 , and it is not difficult to see that the duty cycle for a Torus schedule is $\frac{3}{2n}$, if n is even, or $\frac{3n-1}{2n^2}$, if n is odd, and duty cycle may be arbitrarily small. Note that a Torus quorum will present a duty cycle 25% smaller in relation to a Grid of the same length (a little less, if n is odd).

Parameters	n	torus dimension
	column	active column
	horizontal slots	active slots not in column
Cycle length		n^2
Duty cycle	if n is even	$\frac{3}{2n}$
	if n is odd	$\frac{3n-1}{2n^2}$
Duty cycle ra	ange	(0,1]
Granularity	1 to 10%	136
	0.1 to 1%	1351
	0.01 to 0.1%	13501
Asymmetric	operation?	no

Table 3.2: Summary of the main properties of Torus schedules.

Granularity is smaller, if compared to Grid, but of the same order of magnitude. However, while in the Grid Quorum, nodes may select matrices of different order and still preserve rotation closure, this is not the case for the Torus. Hence, the Torus Quorum does not support asymmetric operation, even though non-monotonic schemes, as exemplified in Figure 3.5, are possible.

3.3.3 Discussion

Due to the absence of a precise model to understand the NDT of Quorum schedules, there are some unanswered questions, which we are now, based on our method, able to address. This section discusses two of them: (1) what is the impact of the selection of active line (or horizontal slots for Torus) and column on the NDT, and is it possible to determine which should be selected for reducing the NDT, and (2) to what extent is Torus a practical improvement over Grid.

3.3.3.1 Impact of active slots selection

Proposers of either the Grid or the Torus Quorum did not indicate which column or line/horizontal slots should be selected as active to reduce the NDT. Actually, initial and subsequent papers did not even raise the question. Neglecting these selections, however, is consistent with the simplified approach used in current literature studies of the NDT of these schedules, and also with the lack of tools to calculate it. We have used our method to investigate this selection, as explained in this section.

We start by analyzing the case with Grids. Firstly, we notice that because of the cyclic repetition of the schedule, the selection of the active line does not have an impact



Figure 3.6: Equivalent and non-equivalent schedules for Grid and Torus schedules. Because line selection does not change the NDT, Grid schedules (a) and (b) are equivalent. Column selection impacts the NDT, and Grids (a) and (c) are not equivalent. For Torus, column selection is irrelevant, and equivalence is preserved as long as the relative position of the horizontal slots is not changed, as in (d), (e) and (f), but opposite to (d), (g) and (h).



Figure 3.7: The difference in the NDT between best and worst column selection for Grid dimensions n = 20, 40, ..., 400. Plotted values are for p = 1, when the difference is bigger.

on the NDT. Two grid schedules of same dimension and same active column, but with different active lines, are simply rotations of each other and, henceforth, *equivalent* in terms of NDT (Definition 9). This equivalence is represented in Figure 3.6 — schedules (a) and (b) differ only in active line and would produce the same monotonic scheme.

However, we found that the selection of the active column does have an impact on the expected NDT. To understand that, we used our model to compare column selections for grid dimensions in the intervals n=3,4,...,100 and n=120,140,...,400. For all these schedules, the E[NDT]s were computed for probabilities p varying from 0.05 to 1, with increments of 0.05.

Figure 3.7 shows the difference between the worst and the best case for some of the tested grids. Results indicate that, as n increases, the impact of column selection decreases. Also, the difference is more significant as p increases, and reaches its maximum at p = 1. For better legibility we plotted the difference only for n = 20, 40, ..., 400 and for p = 1 (for lower values of p the difference is even smaller). As we see, the difference is always below 0.005 (or 0.5%) for all depicted values.

As expected, the most severe impact of column selection occurs in a 3×3 Grid (middle column versus first or last column). In this case, the difference grows from less than 0.03% to 4.8%, as p goes from 0.05 to 1. In comparison, this difference is less than 0.000013 (or 0.0013%) for all values of p in the 400 × 400 Grid.

The difference is more accentuated for higher values of p because the schedule pattern tends to make a bigger difference in these cases. For poor quality links, there is a tendency for the NDT to be dominated by the number of elapsed cycles until discovery, and the wakeup pattern has a diminished impact. In conclusion, the difference is small for schedules with low duty cycle (less than 1% for n > 12, less than 0.1% for n > 43 and less than 0.01% for n > 140).

Another regular pattern that emerges is that the column selections c and n - c (n being the dimension of the grid) result in the same NDT. Also, the worst selections are always the first and the last columns, while the best columns (averaging all values of p) are around 3n/8 or 5n/8 (least squares fitting of the obtained results indicates the best column as $0.3564935 \cdot n + 1.1434984$). Therefore, we recommend one of these columns to be selected, as there is nothing to lose, and there may be a small gain in some scenarios. For n = 100, for example, the best columns are 37 and 63. For n = 400, the best columns would be 143 and 257. Note, however, that the best column selection for a given p, actually depends on p.

For the analysis of Torus, we recall that the horizontal slots do not need to belong to the same line and, because of that, there is a great number of possible schedules for a Torus of same dimension — $n^{\frac{n}{2}-1}$, if n is even and $n^{\frac{n}{2}}$, if n is odd.

Figure 3.6 illustrates some possibilities for a 6×6 Torus schedule. Note that schedules (d), (e) and (f) are equivalent — any of them may be obtained as a rotation of either of the other two schedules. Contrary to what happens with Grid, column selection has no impact on the NDT in a Torus. What does impact on the NDT is the relative positions of the horizontal active slots. This explains why (a) is equivalent to (e), but not to (g) or (h).



Figure 3.8: Three Torus schedules with dimension n = 100 but different horizontal slots selection and the resulting NDT for each of them.

For measuring the impact or determining the best horizontal selection of slots, for any sufficiently big Torus schedule (i.e. one with low duty cycle), a huge number of computations would be necessary. Nonetheless, for illustration, we compared the results for the three 100×100 Torus schedules depicted in Figure 3.8a. The results of the comparison are shown in Figure 3.8b — the observed difference was always below 0.1%, indicating that the impact of the selection of the horizontal slots may be of little significance.

3.3.3.2 Torus versus Grid

Another interesting question regarding Quorum-based mechanisms is determining to which extension Torus schemes are a substitute for Grid schemes. In one hand, the resulting duty cycle, for the same dimension n, will be smaller in the case of the Torus. On the other hand, if the NDT is considered, the comparison is not as simple.

Again, we resort to our method for an answer. We compared three Grid and Torus schedules of the same approximate duty cycles: a 400×400 Grid with a 300×300 Torus; a 200×200 Grid with a 150×150 Torus; and a 100×100 Grid with a 75×75 Torus. The best columns were selected for the Grid schedules (150, 75 and 37, respectively) and, for the Torus, all horizontal slots were selected in the same line. The results are presented



Figure 3.9: A comparison between Torus and Grid schedules of same duty cycle, reveals that Torus will result in shorter NDT only for good link quality.

in Figure 3.9 and reveal that Torus will result in shorter NDT for good link quality. The results are similar for all three comparisons — a delivery probability around 0.75 marks the transition from Grid to Torus as the schedule resulting in shorter NDT. Therefore, Torus does not always surpass Grid. The best choice depends, once again, on p.

3.4 Prime numbers-based mechanisms

The second category of asynchronous schedule-based duty cycling mechanisms relies on properties of prime numbers. For example, if two nodes select different prime numbers, m and n, and activate every m_{th} and the n_{th} slots, 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 [57], 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. Nevertheless, a closer look reveals that a U-connect schedule is actually a Torus schedule⁴. Therefore, we will not analyze this proposal.

Moreover, making a node select one prime number severely limit the choices in terms of duty cycle. The authors of Disco [55] 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 [54] guarantees the rotation closure property irrespective of the selection [55].

Figure 3.10 illustrates the use of Disco. For conciseness and clarity, nodes A and B select small primes: Node A selects the pair (5,7) resulting in the schedule $S_A = \varsigma[35, \{0, 5, 7, 10, 14, 15, 20, 21, 25, 28, 30\}]$ and Node B selects the pair (3,13) for a schedule $S_B = \varsigma[39, \{0, 3, 6, 9, 12, 13, 15, 18, 21, 24, 26, 27, 30, 33, 36\}]$. Evidently, 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).

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 asynchronous 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. We will discuss this assertive in the next section, but before proceeding we find useful to define a measure for the balancing or unbalancing of the two selected prime numbers. We will call it *level of unbalancing*. A level of unbalancing of 1 (u = 1) means that only consecutive primes are selected (11 and 13, for instance). For u = 2, 11 could be combined with 17 or 5. In other words, if \mathbb{P} is the set of prime numbers and $q_i \in \mathbb{P}$ is the i_{th} element of \mathbb{P} , for a given u, q_i may be paired with all primes q_{i-u} or q_{i+u} .

Table 3.3 summarizes the main characteristics of a Disco schedule. It is defined completely by the selected primes, q_1 and q_2 , from which it is easy to calculate the cycle

 $^{^{4}}$ With the unnecessary drawback of being restricted to prime dimensions, which severely limits its granularity

Parameters	q_1, q_2	selec	ted pi	rime n	umbers	3
Cycle length		$q_1 \cdot q$	2			
Duty cycle		$1/q_1$	+1/q	2		
Duty cycle range		(0,2)	/3]			
Granularity		level of unbalancing (u)				
		1	2	3	4	5
	1% to $10%$	7	13	19	25	31
	0.1% to $1%$	45	89	132	174	216
	0.01% to $0.1%$	302	603	904	1204	1504
Asymmetric operation?		yes				
∠ initial slot						
Node A (m_A =5, n_A =7)						
∩ _ <u></u>						
Node B (m _B =3,n _B =13)						
Common active slots						

Table 3.3: Summary of the main properties of Disco.

Figure 3.10: 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.

length $(q_1 \cdot q_2)$ and the duty cycle $(1/q_1 + 1/q_2)$. Duty cycle can be arbitrarily small or reach 2/3 $(q_1 = 2, q_2 = 3)$. Granularity strongly depends on the admitted level of unbalancing. If any pair of prime numbers is admitted, the granularity of Disco is infinite. However, as balanced pairs are preferred, granularity was provided for five different *levels* of unbalancing (u). Finally, Disco supports asymmetric operation, since different pairs of primes will result in different duty cycles.

3.4.1 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. However, the delivery probability p was left outside of such analysis and, once again, we will employ our model to confirm the empirical observations of the proposers of Disco, and to check it over the whole spectrum of p. For this assessment, we tested 15 pairs of primes which result in duty cycles of approximately 1% (all within the interval [0.9969%,1.0072%]): (109, 1201), (113,863), (127,467), (131,421), (137,367), (139,359), (149,307), (151,293), (157,277), (163,257),



Figure 3.11: The NDT for 15 Disco schedules with the same approximate duty cycle of 1%. More balanced pairs result in shorter NDT.

(167,251), (173,233), (179,227), (181,223) and (191,211). The resulting NDT is plotted in Figure 3.11. As shown by the figure, as the level of unbalancing increases, the NDT also increases, and although the difference between the first 10 pairs is small, the difference between pairs (109, 1201) and (191,211) is really significant.

Figure 3.12 complements the information by showing the increase in the NDT of the less balanced pairs in respect to the more balanced pair (191, 211, level of unbalancing 4). The NDT for pair (109,1201, level of unbalancing 143) is approximately 3.25 times longer, and this relation seems to be independent of p for all pairs and values of p observed. This data is consonant with the principle that balanced pairs are superior than unbalanced pairs in symmetric operation, and it also demonstrated that this holds irrespective of link quality.



Figure 3.12: The NDT of the most unbalanced fourteen pairs in relation to the NDT of (191,211). The NDT of pair (109,1201) is approximately 3.25 times as high.

3.5 Block Designs

Block Designs come from the Mathematics Area of Combinatorics, and were first introduced as a technique to design experiments in agriculture [58]. A Block Design⁵ is a set system (a set of sets called *blocks*) that can be defined as follows:

Definition 12 Block Design — Given a finite set V of v elements and integers $k, \lambda \ge 1$, a Block Design, represented as $\{v, k, \lambda\}$, will have exactly v blocks $(B_0 \dots B_{v-1} \subset V)$ of k elements and the following properties:

- 1. Each and every element of V occurs in exactly k blocks; and
- 2. Any two blocks will have exactly λ elements in common.
- Example: Suppose, for instance, that V={0,1,2,3,4,5,6}. A Block Design {7,3,1} (exemplified in Figure 3.13) 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}}.

 $^{^{5}}$ We should actually refer to a *Symmetric* Block Design. However, 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 [59] for a formal definition.



Figure 3.13: 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.

Parameters	v	block length
	k	number of active slots
	λ	number of overlapping slots
	forming set	active slots in the first cycle
Cycle length		v
Duty cycle		k/v
Duty cycle r	ange	[0.0103, 0.43]
Granularity	1 to 10%	118 (35 Projective Planes)
	0.1 to 1%	0
	0.01 to 0.1%	0
Asymmetric	operation?	no

Table 3.4: Summary of the main properties of Block Design schedules.

In relation to our previous definitions, a *block* is equivalent to a *schedule*, while a *block design* is equivalent to a *monotonic scheme* formed by all blocks in the design. The rotation closure property is ensured by the Property 2 in Definition 12. In the notation introduced earlier in Section 3.1, the scheme resulting from the Block Design $\{7,3,1\}$ would be denoted as $\overline{\chi}[7, \{0, 1, 3\}]$.

Table 3.4 summarizes the properties of schedules based on Block Designs. A Block Design schedule is characterized by its three parameters, $\{v, k, \lambda\}$, plus its forming set⁶. The duty cycle is k/v, which ranges from 1.03%, for a {9507,98,1} design, and 43%, for {7,3,1}. Block designs outside this interval are still unknown [60], limiting both range and granularity. Asymmetric operation is not possible under Block Designs.

As an illustration, Table 3.5 presents some examples of Block Designs, providing the resulting duty cycles and their elements, i.e. their forming sets. A complete list of Block Designs with their forming sets is found in [60].

⁶There exist cases of Block Designs of same parameters and different forming sets.

Table 3.5: Some examples of Block Designs, with the resulting duty cycles and the list of active slots (forming sets). 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 forming set $\{0,1,3\}$ is completely equivalent to $\{1,2,4\}$ or any other rotation of the schedule.

design	duty cycle	forming set
$\{7,3,1\}$	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%	$\{0, 2, 3, 4, 8\}$

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*⁷. 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* (which differs in one unit to our definition of the order of a schedule — the number of active slots per cycle). It is conjectured that there only exist Projective Planes for values of s which are powers of primes [59]. 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 and 27 are variations with same parameters $\{v, k, \lambda\}$ and different forming sets [60]. All known Block Designs were included in our analysis and are presented in Tables 3.6 (Projective Planes) and 3.7 (other Block Designs). The forming sets were omitted to save space and are found in [60].

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 [53]. 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. As we will see, this property places Block Designs in an unique position among all asynchronous schedules.

⁷Actually, 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.
$\{v, k, \lambda\}$	duty cycle	$\{v,k,\lambda\}$	duty cycle	$\{v\}$	$\{k, \lambda\}$	duty cycle
$\{7,3,1\}$	42.86%	$\{553,24,1\}$	4.34%	{3	$541,\!60,\!1\}$	1.69%
$\{13,4,1\}$	30.77%	$\{651, 26, 1\}$	3.99%	${3'}$	783,62,1}	1.64%
$\{21,5,1\}$	23.81%	$\{757, 28, 1\}$	3.70%	{4	$161,\!65,\!1\}$	1.56%
$\{31, 6, 1\}$	19.35%	$\{871, 30, 1\}$	3.44%	{4	$557,\!68,\!1\}$	1.49%
$\{57, 8, 1\}$	14.04%	$\{993, 32, 1\}$	3.22%	{5	$113,72,1\}$	1.41%
$\{73,9,1\}$	12.33%	$\{1057, 33, 1\}$	3.12%	${5}$	$403,74,1\}$	1.37%
$\{91,10,1\}$	10.99%	$\{1407, 38, 1\}$	2.70%	{6	321,80,1	1.27%
$\{133, 12, 1\}$	9.02%	$\{1723, 42, 1\}$	2.44%	{6	$543, 82, 1\}$	1.23%
$\{183, 14, 1\}$	7.65%	$\{1893, 44, 1\}$	2.32%	{69	973,84,1}	1.20%
$\{273, 17, 1\}$	6.23%	$\{2257, 48, 1\}$	2.13%	{80	$)11,90,1\}$	1.12%
${307,18,1}$	5.86%	${2451,50,1}$	2.04%	{9	507, 98, 1	1.03%
$\{381, 20, 1\}$	5.25%	$\{2863, 54, 1\}$	1.89%			

Table 3.6: Known Block Designs with $\lambda = 1$ (Projective Planes), sorted by duty cycle.

The literature sometimes refers to *Perfect Difference Sets* [56] 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 [59].

Before moving on to the discussion section, is it worth noting the fact that no known Block Design provides duty cycles of less than 1% which is the major drawback of the Block Design schemes.

3.5.1 Discussion

We will now use our model for an in-depth analysis of two aspects regarding the NDT of Block Designs. Firstly, we will build a qualitative understanding of how the NDT varies with the parameters of Block Designs, and with the delivery probability p. Secondly, we will determine whether Block Designs with $\lambda > 1$ may or not provide useful schedules.

3.5.1.1 Qualitative understanding of the NDT in Block Designs

The three graphs in Figure 3.14 provide a qualitative understanding on the way E[NDT] grows with the variation of the three parameters $(p, \lambda \text{ and } v)$ in a monotonic scheme. Figure 3.14a 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

$\{v,k,\lambda\}$	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%	$\{31,15,7\}$ (2)	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 3.7: 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 forming sets for the {1023,511,255} design).

of E[NDT] is observed.

In Figure 3.14b, 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 Figure 3.14c — 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).

3.5.1.2 Projective Planes versus (other) Block Designs

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



Figure 3.14: The behavior of Equation 3.8 — linear growth of E[NDT] with v and nonlinear decrease, with both p and λ . As the NDT is infinite for p = 0, all graphs are plotted for $p \in [0.1, 1]$.



Figure 3.15: 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.

link quality deteriorates (low values of p). Figure 3.15a plots all known Block Designs in the NDT × 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 3.7).

Figure 3.15b 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 Figures 3.15a and 3.15b. 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.

Figures 3.16a and 3.16b continue this analysis and provide a direct comparison be-



Figure 3.16: Comparisons between a Projective Plane and another Block Design ($\lambda > 1$) with similar duty cycle show that the best NDT depends on the link quality p.

tween two pairs of Block Designs, always opposing a Projective Plane with another Block Design with bigger λ and same approximate duty cycle. Figure 3.16a 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 shortest. Likewise, as we can see in Figure 3.16b, 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.

3.6 Why a new schedule?

After presenting the most important schedule-based asynchronous mechanisms in the literature, one should question if a new mechanism is possible, that would present clear advantages over all present propositions. To answer to that question, we need first to select a set of metrics for comparison. The natural candidates are: duty cycle range, duty cycle granularity, support for asymmetric operation and the resulting NDT for a given duty cycle. Though we already possess these first three metrics for all studied mechanisms, we still need to find the fourth in a way that permits a direct comparison.

Our method, though able to compare any two schedules, fails to give us means for a general comparison among the mechanisms, that would be possible with the aid of closed-form expressions. Therefore, we devote this section to finding such closed-form expressions for the mechanisms under study. After that, and based on these expressions, we will propose the last metric — the relative-latency. With this four metrics (duty cycle range and granularity, support for asymmetric operation and relative-latency), we will show that there is room for improvement, once there is not a single mechanism that is superior to the others in all metrics, in all scenarios. This will lead us to our own proposition of a mechanism — Nested Block Design schedules, in Chapter 4.

3.6.1 Closed-form expressions for the E[NDT]

Although our exact model provides the NDT with complexity $O(w' \times q^2)$, a closed-form expression for the NDT for each of the studied schemes would be useful. Firstly, it would provide a more intuitive understanding of the NDT, without summations and coefficients with non-obvious meanings. Secondly, a closed-form expression may be used for general and direct analytical comparison between the many existing schemes.

All closed-form expressions presented in here will be approximate. As it will become clear, removing the Phi-coefficients from the expressions is only achievable through the removal of some of the possible co-schedules. In some cases, the choice of which simplifications to use will lead to a more or less accurate expression for a given set of parameters. We decided to be more precise in schedules that result in duty cycles of 2% or less⁸ and to sacrifice accuracy for the other, less useful, schedules that result in higher duty cycles. The specifics of each simplifying assumption will be clarified as an expression is obtained for each of the four studied schedules (Block Designs, Grid, Torus and Disco). The presentation of each expression is followed by a validation, where the estimated NDT of the expression is compared to the NDT given by our exact model.

3.6.1.1 Closed-form expression for the E[NDT] of Block Designs

In this section we present a closed-form expression that was obtained by combinatorial analysis of all possible co-schedules between the schedules of two unsynchronized nodes operating with random offsets in a monotonic scheme based on Block Designs. In order to build such a model, we analyzed every possible distribution of discovery opportunities that may occur for two unsynchronized nodes, accounting for the discovery probability for each opportunity.

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 of

 $^{^8 \}rm We$ would prefer to set this limit to 1%. However, that would make a comparison with Block Designs impossible.

the co-schedule, 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.

We argue that our model, presented in Equation 3.8, provides an estimate of the NDT in asynchronous duty cycling monotonic schemes based on Block Designs. Equation 3.8, which is derived in Appendix B, Section B.1, takes the parameters v (number of slots per cycle) and λ (number of discovery opportunities per cycle) from the Block Design and the discovery probability, p, to estimate the expectancy for the neighbor discovery time, E[NDT].

$$E[NDT]_{blockdesign} = \frac{v+1}{p(\lambda+1)} - \frac{(v+1)(1-p)^{\lambda} - (\lambda+1)}{(\lambda+1)[(1-p)^{\lambda} - 1]}$$
(3.8)

Some special cases can be derived from Equation 3.8 and provide more practical and insightful expressions for the NDT. Three of these cases are presented on Table 3.8. 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 $\left(\frac{v-1}{2}\right)$. Case 2 is a sub-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.

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

Table 3.8: Three special cases derived from the model.

For validation, the closed-form expression provided in Equation 3.8 was compared to the results from our exact method, for all known Block Designs for the delivery probabilities from 0.05 to 1.0, with increments of 0.05. Figure 3.17, shows the gap for all Block Designs with duty cycles of less than 2% as p grows. The gap is smaller than 0.3% in all cases. As all Block Designs with duty cycle of less than 2% are Projective Planes, we can use the expression for Case 1, at Table 3.8, for very good accuracy.



Figure 3.17: The gap between the simplified closed-form and the exact model for all Block Designs with duty cycle of less than 2%.

3.6.1.2 Closed-form expression for the E[NDT] of Grid

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 intersect as the offsets change. By empirically analyzing the behavior of this intersection patterns with the R language [61] and applying methods of interpolation and curve fitting, we were able to find the expression for the NDT, presented in Equation 3.9.

As detailed in Appendix B, Section B.2, we obtained a 6^{th} degree polynomial expression, and tested it with many interesting values of n to reassure its accuracy and, then, we derived a compact quadratic version that approximates the complete 6^{th} degree expression with accuracy better than 99% for all schedules with duty cycles lower than 2% (n > 100). Equation 3.9 is our expression for the expected NDT, when two nodes operate under a monotonic scheme obtained from a $n \times n$ Grid schedule, for n > 100. It takes the parameters p, the probability of discovery, and n, the grid dimension.

$$E[NDT]_{grid} = \frac{(3-p)n^2}{6p}$$
 (3.9)

To measure the gap for the expression in Equation 3.9 we selected 16 Grid schedules with n = 100, 120, ..., 400, i.e. duty cycles between 0.5% and 2%, always selecting the column which gives the shortest NDT. Figure 3.18 shows the gap to the exact model for all these Grid schedules. The gap decreases as n increases and is more accentuated for mid-range values of p. Although significantly higher than the gap found for the simplified expression for Block Designs, a gap of less than 10% still fits our purposes, as we will see in the end of this chapter.



Figure 3.18: The gap between the simplified closed-form and the exact model for Grids with n = 100, 120, ..., 400.

3.6.1.3 Closed-form expression for the E[NDT] of Torus

In order to obtain a model for the estimation of the NDT for monotonic schemes based on Torus schedules, 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 (Equation 3.10). Because intersection patterns in Torus are better behaved than in Grid, the resulting expression for Torus is more accurate. The specifics of the methodology are detailed in Appendix B, Section B.3. Equation 3.10 is our NDT model for Torus schemes. It takes the parameters p, the probability of discovery, and n, the grid dimension.

$$E[NDT]_{torus} = \frac{(2-p)n^2}{2p}$$
(3.10)

To measure the gap for the expression in Equation 3.10, we selected 10 Torus schedules with n = 75, 100, 125, ..., 300, i.e. duty cycles between 0.5% and 2%. Figure 3.19 shows the gap to the exact values for all these Torus schedules. The gap tends to decrease as n increases, although not monotonically, and is more accentuated for lower values of p. The maximum gap observed for the tested schedules was of 2.54%.

3.6.1.4 Closed-form expression for the E[NDT] of Disco

In order to obtain a model of the NDT for monotonic duty cycling schemes out of 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



Figure 3.19: The gap between the simplified closed-form and the exact model for Torus with n = 75, 100, 125, ..., 300.

— a method similar to that employed for Block Designs.

Although, as in the case of Block Designs, a combinatorial analysis is feasible, such analysis is more complex, since a change in the offset between neighbors affects the number of discovery opportunities per cycle. Therefore, the derivation of this model, presented in Appendix B, Section B.4 is the longest between the four schedules, and the resulting expression, even in its compact quadratic version, presented in Equation 3.11, is also slightly more cumbersome.

Equation 3.11 gives an approximation of the expected NDT with an error proportional to $\frac{1}{q_1q_2}$ (see Section B.4). It takes the parameters q_1 and q_2 (the prime numbers) and the probability of discovery p.

$$E[NDT]_{disco} = \frac{q_1 q_2 (p^2 - 3p + 3)}{3p(2 - p)}$$
(3.11)

To measure the gap for the expression in Equation 3.11 we selected 10 Disco schedules formed with balanced pairs (consecutive primes) and duty cycles between 0.5% and 2%. Figure 3.20 shows the gap to the exact model for these Disco schedules. The gap tends to decrease as the duty cycle decreases, and does not change appreciably with p. The maximum gap observed for the tested schedules was of 2.01%. For schedules with duty cycles of less than 1% the gap was never superior to 1%.



Figure 3.20: The gap between the simplified closed-form and the exact model for 10 selected Disco schedules.

Table 3.9: Power-latency product as it appears in the literature.

Mechanism	Latency	Duty cycle	Power-latency product
Block Designs (Projective Plane) $\{s^2+s+1,s+1,1\}$	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)	$N = q_1 \cdot q_2$	$DC = \frac{q_1+q_2}{q_1q_2} \sim \frac{2}{\sqrt{N}}, (q_1 \sim q_2 \sim \sqrt{N})$	$2\sqrt{N}$

3.6.2 Comparisons and the relative-latency metric

Schedule-based asynchronous mechanisms are usually compared through the *power-latency metric* [57, 56], which is calculated as the product of the duty cycle and the latency. In this computation, the latency used 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 between two mechanisms, one will always be better or worse for all link qualities, which we already proved to be erroneous by comparing Grid and Torus schedules, and also by comparing Block Designs with different lambdas.

The power-latency metric values for the studied schemes are presented in Table 3.9. As already mentioned, 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 cycle⁹. The power-latency product is also misleading since the N in the formulae, which is actually 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 delivery

 $^{^9\}mathrm{For}$ all values of N>3

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)	$L = q_1 \cdot q_2$	$\frac{L(p^2-3p+3)}{3p(2-p)}$	$DC \sim \frac{2}{\sqrt{L}}$	$\frac{4(p^2-3p+3)}{3p(2-p)}\cdot\frac{1}{DC^2}$

Table 3.10: The relative-latency metric.



Figure 3.21: Comparison between the four mechanisms based on the relative-latency metric. This graph applies for all duty cycles.

probability, p. The relative-latency values for the four studied schedules are presented in Table 3.10. They were obtained from the closed-form expression provided earlier in this chapter. The third column in Table 3.10 shows the latency as a function of the cycle length L, as given by the closed-form NDT expressions. The relative-latency is obtained by replacing the value of the duty cycle DC, provided in the fourth column, in these expressions for latency.

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. It is important to note that we do not need the relative-latency metric to compare two given schedules, since we have an exact method for such analysis. However, the new metric permits general conclusions that, though approximative (since the relative metric is obtained from the approximative closed-form expressions), provide a clear picture of how the schedules behave comparatively. This behavior is captured in Figure 3.21 where each mechanism is represented by an area that contains all curves that represent the NDT as a function of p, normalized in relation to the NDT of Block Design schedules.

Torus schedules, for example, will always result in approximately 2.25 times the NDT of a Block Design, for the same duty cycle and delivery probability. This is easily observed by dividing the relative-latency metric of both schedules. We may also conclude that, compared to Torus, Grid and Disco will result in shorter NDT as link quality deteriorates.

The areas represent the imprecision of the closed-form expressions. However, for all mechanisms, as the duty cycle decreases, the results converge to the upper limit of each area. That means that Grid will result in shorter NDT for higher duty cycles, but as the duty cycles decrease, Disco will eventually be more advantageous. Another conclusion is that, for delivery probabilities around 0.7, the difference between Disco, Grid and Torus may be irrelevant.

To reinforce the results presented in Figure 3.21, we used our model to compute the NDT of four schedules of same approximate duty cycle: the $\{9507,98,1\}$ Block Design, the 193×193 Grid Quorum, the 145×145 Torus Quorum and Disco with the primes (193,197). The selected schedules result in the duty cycle of 1.03% (selected, because it is the lowest achievable by using Block Designs). The results are presented in Figure 3.22 and clearly fall within the prediction areas of the relative-latency metric.

Additionally, we implemented the same four schedules on TelosB motes [62]. The results after 400 trials for each method are presented in Figure 3.23. The motes were tested in conditions where practically no frames were lost, hence p = 1 was assumed. The results confirm 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 presented similar results. The results also indicate the good prediction of the closed-form expression, which, in fact, provides predictions not very different from the exact values taken from our model. In all cases, a gap of less than 4% was observed between the three bars.

Finally, we compare the literature assumption that the cycle length may be used as an indicator of the NDT with our models. Figure 3.24 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%) were selected and, in each case, the results 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 accurate estimations. Since the accuracy of our models



Figure 3.22: The results from our exact model for four schedules of same approximate duty cycle of 1.03%.

Table 3.11: Synoptic table — final comparison between Block Designs, Grid Quorum, Torus Quorum and Disco.

Scheme	Relative	Minimum	Granularity	Asymmetric
	latency	duty cycle	(order of magnitude)	operation?
Block Design	$\frac{2-p}{2p}\cdot\frac{1}{DC^2}$	1.03%	low	no
Torus Quorum	$\frac{9(2-p)}{8p} \cdot \frac{1}{DC^2}$	any	medium	no
Grid Quorum	$\frac{2(3-p)}{3p} \cdot \frac{1}{DC^2}$	any	medium	yes
Disco, balanced	$\frac{4(p^2-3p+3)}{3p(2-p)}\cdot\frac{1}{DC^2}$	any	low to medium ^{\dagger}	yes

 † depending on level of unbalancing

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.

3.6.3 The need for a new schedule

We are now in position to decide in which way a new schedule could improve the performance of schedule-based asynchronous duty cycling. Table 3.11 summarizes our comparison between the four schedule-based asynchronous duty cycling schemes. As demon-



Figure 3.23: 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.



Figure 3.24: A comparison between our proposed models and current literature.

strated, Block Designs are indisputable if the same duty cycle can be applied to all nodes in the network, i.e. if asymmetric operation is not required. However, there is currently no known Block Design that provides less than 1% of duty cycle — which is a significant disadvantage.



Figure 3.25: Decision Tree for selecting a schedule-based mechanism, considering asymmetric operation, duty cycle and link quality. All studied schemes have application scenarios.

For duty cycles inferior to 1% the decision is more complex. For good quality links Torus outperforms Grid or Disco, but it does not support asymmetric operation. For poor quality links, or when asymmetric operation is required, Grid and Disco surpass Torus. Between Disco and Grid, there is no clear winner. As already mentioned, Disco tends to be superior for lower duty cycles. With the help of our model, we determined that Grid is slightly better for duty cycles of 0.78% or higher, and Disco is better for lower duty cycles. But the difference is small and restricted to mid-range delivery probabilities.

Granularity proved to be a poor metric. Actually, no mechanism provides high granularity (except highly unbalanced Disco, which is not recommended for its high latency), although it tends to increase if low duty cycles are targeted. In terms of order of magnitude, granularity is in the order of 1/DC for the studied mechanisms.

Based on all this considerations, a decision tree like the one portrayed in Figure 3.25 summarizes the options and motivates our own proposition — to design a schedule that achieves NDTs as low as Block Designs, while achieving arbitrarily low duty cycles and supporting asymmetric operation. We introduce such a schedule in the next chapter.

Chapter 4

Nested Block Designs

This chapter presents Nested Block Designs, our proposal to extend the application domain of Block Designs, by removing its aforementioned shortcomings: (1) the lack of a schedule that provides duty cycles lower than 1%, (2) the impossibility of asymmetric operation and (3) its low duty cycle granularity.

As in the other schedules, under Nested Block Design schedules, time is also divided into cycles. However, each cycle is divided into *superslots* that are further divided into slots. A superslot is either active or inactive, according to a given Block Design, referred to as *outer design*. If a superslot is inactive, all its slots will also be inactive. If, on the other hand, the superslot is active, its constituting slots will be either active or inactive according to a second given Block Design, referred to as *inner design*. The inner and outer designs may be the same Block Design, or different Block Designs. To denote a Nested Block Design we will use $\{v_o, k_o, \lambda_o\} \#\{v_i, k_i, \lambda_i\}$, where $\{v_o, k_o, \lambda_o\}$ indicates the outer schedule or design, and $\{v_i, k_i, \lambda_i\}$ the inner schedule or design.



Figure 4.1: A Nested Block Design. The outer design is $\{7,3,1\}$. Each one of the active superslots in the outer design is further divided into active and inactive slots according to the inner design $\{13,4,1\}$.

Figure 4.1 presents an example of a Nested Block Design. The depicted Nested Block Design is $\{7, 3, 1\}$ # $\{13, 4, 1\}$, meaning that each cycle is formed by 7 superslots, consisting

of 13 slots each. The three active superslots will have four active slots according to the $\{13,4,1\}$ design. The resulting duty cycle is $\frac{k_o k_i}{v_o v_i}$. In this example, there will be 12 active slots in a cycle of 91 slots.

It should be noted that a Nested Block Design is no longer adherent to the definition of Block Design given in Section 3.5, and its optimality in terms of duty cycle is lost. This is easy to see if we compare the schedules $\{91,10,1\}$ and $\{7,3,1\}\#\{13,4,1\}$. While both present the same length, the Projective Plane has only 10 active slots, against 12 of the Nested Design. However, as we will see, this increase in duty cycle will be followed by a decrease in the NDT, due to the increased number of discovery opportunities per cycle.

The rest of this chapter is organized as follows. We start by presenting some of the design choices taken for Nested Block Designs, in Section 4.1. Sections 4.2 to 4.5 are organized to demonstrate the desirable qualities of Nested Block Designs:

- Neighbor discovery time for Nested Block Designs is shorter than that of Grid and Torus Quora or Disco, and only marginally longer than that of Block Designs (Section 4.2).
- Nested Block Designs may achieve arbitrarily low duty cycles (Section 4.3).
- Nested Block Designs have duty cycle granularity superior than that of Block Designs and at least comparable to that of other mechanisms (Section 4.4).
- With Nested Block Designs it is possible to devise a scheme for asymmetric operation (Section 4.5).

Section 4.6 presents the results of tests performed with Nested Block Designs in real sensor motes, and Section 4.7 presents our final remarks on Nested Block Designs, concluding the chapter.

4.1 Design choices

Nested Block Designs may be formed from any Block Designs, i.e. Projective Planes or other Block Designs. Because of their optimality, we will use only Projective Planes in most of our analyses, using Block Designs with $\lambda > 1$ only when necessary, such as to illustrate a given point. However, as we did for all other schedules, we may use our method to help finding answers to other specific questions regarding the design of Nested Block Design schedules.

4.1.1 Inner design rotation

Given a $\{v_o, k_o, 1\} # \{v_i, k_i, 1\}$ schedule, there is the question of which rotation of the inner design $\{v_i, k_i, 1\}$ should be selected for lower NDT. Note that any rotation of the outer design $\{v_o, k_o, 1\}$ will result in an equivalent schedule. However, rotating $\{v_i, k_i, 1\}$ results in an non-equivalent Nested Design.

Figure 4.2 shows the impact on the NDT as the inner schedule $\{7,3,1\}$ rotates in a $\{7,3,1\}\#\{7,3,1\}$ Nested schedule. As demonstrated in the Appendix C, the smaller the last element of the delta-set of the inner schedule (δ_{k_i-1}) , the smaller the NDT. In the case of the $\{7,3,1\}$ schedule, there is one rotation with $\delta_2 = 1$ (forming set $\{0,2,6\}$), two rotations with $\delta_2 = 2$ ($\{0,4,5\}$ and $\{1,4,6\}$) and four rotations with $\delta_2 = 4$ ($\{0,1,3\}$, $\{1,2,4\}$), $\{2,3,5\}$ and $\{3,4,6\}$) (see Figure 3.13). Figure 4.2 shows the percentual increase in the NDT in relation to the best case ($\delta_2 = 1$). The difference increases with p but never reaches 1.65%. The impact is proportional to MAX($\Delta\{S_i\}$)/ v_i — the maximum delta in the delta-set of the inner schedule divided by the cycle length of the inner schedule (see Appendix C), which, by testing all known Projective Planes, we determined to be maximum for $\{7,3,1\}$. Therefore, the percentual difference is always less than 2%, falling inside the gray area in Figure 4.2 for all inner rotations.

Although the difference is small, there is no reason not to select the inner schedule with minimum δ_{k_i-1} . In a Projective Plane, there will always be exactly one rotation with $\delta_{k_i-1} = 1$. This can be explained by the relation between Projective Planes and Perfect Difference Sets (see [59]) — two consecutive active slots will always occur in Projective Planes, and there is always one rotation with the first and the last slots active, as a consequence.

4.1.2 Nesting order

Another question regarding the design of Nested schedules is the order in which the forming schedules should be nested. Given two distinct Projective Planes, $\{v_a, k_a, 1\}$ and $\{v_b, k_b, 1\}$, it is possible to select either of them as inner or outer design: the resulting duty cycle will be the same. However, we may ask which nesting order, $\{v_a, k_a, 1\} \#\{v_b, k_b, 1\}$ or $\{v_b, k_b, 1\} \#\{v_a, k_a, 1\}$, would result in the shortest NDT.

An analytical answer to the question above would require the solving of the maximization problem presented in Appendix C. Nevertheless, as the number of Projective Planes are finite, we may actually test all possibilities and avoid the maximization prob-



Figure 4.2: The percentual difference in the NDT given rotations of the inner schedule $\{7,3,1\}$. The minimum NDT happens when $\delta_{k_i-1} = 1$.

lem. We proceeded likewise, and concluded that one should select the longest schedule as the inner schedule. Therefore, if $v_b > v_a$: $\{v_a, k_a, 1\} \# \{v_b, k_b, 1\}$ results in shorter NDT than $\{v_b, k_b, 1\} \# \{v_a, k_a, 1\}$.

Figure 4.3 illustrates that by showing a comparison between two pairs of Nested schedules. The first pair, $\{7,3,1\}$ # $\{9507,98,1\}$ and $\{9507,98,1\}$ # $\{7,3,1\}$, is extreme, in the sense that it is the Nested schedule where the difference between inner and outer designs is maximized. It is, in fact, the pair where the observed difference in NDT is the greatest. When p, the delivery probability, approaches 1, the difference between the NDT of $\{7,3,1\}$ # $\{9507,98,1\}$ vs $\{9507,98,1\}$ # $\{7,3,1\}$ reaches 8%, showing that the first schedule is significantly faster than the second in neighbor discovery.

For the second pair in Figure 4.3, $\{7,3,1\}\#\{13,4,1\}$ and $\{13,4,1\}\#\{7,3,1\}$, the difference is smaller. As the cycle lengths of inner and outer designs approach each other, the impact of the nesting order decreases. For schedules $\{1057,33,1\}\#\{993,32,1\}$ and $\{993,32,1\}\#\{1057,33,1\}$, for example, the difference is always smaller than 0.01%. Actually, for all combinations the difference in nesting order will always fall within the gray area in Figure 4.3.



Figure 4.3: The difference in the NDT as a function of the nesting order. Two specific cases are shown in the graph, and all cases fall within the gray area.

4.2 NDT is comparable to Block Designs

As shown in Section 3.6.2, for the same duty cycle and discovery probability, Block Designs require less than half the time for neighbor discovery than any other studied schedule. Therefore, if we demonstrate that the E[NDT] of Nested Block Designs are only slightly longer than that of Block Designs, this will also prove that Nested Block Designs perform better than the other mechanisms, with respect to NDT.

We start by using our model for a series of comparisons between Block Designs and Nested Block Designs. Since there are no combinations of these schedules that provide the same exact duty cycle, we paired schedules that present less than 1% of relative difference in duty cycle. Because it is already difficult to match these duty cycles for a fair comparison, we will use Nested Designs formed by any Block Designs, not only Projective Planes.

Figure 4.4 shows a comparison between three pairs of Block Designs and Nested Block Designs of approximately same duty cycle: $\{19,9,4\}\#\{553,24,1\}$ (DC = 2.056%) and $\{2451,50,1\}$ (DC = 2.040%), $\{91,10,1\}\#\{91,10,1\}$ (DC = 1.208%) and $\{6973,84,1\}$ (DC = 1.205%), and $\{121,40,13\}\#\{1057,33,1\}$ (DC = 1.032%) and $\{9507,98,1\}$ (DC = 1.031%). Though, as expected, Block Designs present lower NDT, the difference is not as marked as it is for other schedules (Grid, Torus and Disco). As seen in the next section,



Figure 4.4: Comparison between Nested Block Designs and Block Designs of similar duty cycle.

the difference peaks at 10.6% and it is typically much smaller.

To demonstrate better the difference between Nested Block Designs and other schedules, another set of direct comparisons, including Grid, Torus and Disco, is provided in Figure 4.5. The graphs clearly demonstrate that Nested Block Designs provide lower NDT than all of the other schedules (except Block Designs, when they exist — see Figure 4.5d). Figure 4.5a shows a comparison between Disco, Grid, Torus and Nested Block Designs with duty cycles of approximately 0.0975%. Though the difference between the first three is hard to spot, the superiority of Nested Block Designs is clear for all values of p. Figures 4.5b and 4.5c demonstrate the same for other selected duty cycles (0.2% and 0.5%, respectively).

For a relatively high duty cycle of 1.2%, it is possible to include regular Block Designs in the comparison. Figure 4.5d shows that, as expected, the latter results in shorter *NDT*. However, Nested Block Designs perform noticeably closer to Block Designs than the other schedules. In fact, the choice between nested and regular designs, could tend towards the former because they provide the capability for asymmetric operation. In short, Nested Block Designs not only extend the applicability of Block Designs to new domains (duty cycles of less than 1% and asymmetric operation), but it may also achieve comparable performance in terms of NDT.



Figure 4.5: Comparing Nested Block Designs with other schedules of approximately the same duty cycle.

4.2.1 Closed-form expression for the NDT of Nested Block Designs

For the same reasons exposed in Chapter 3, it is useful to have a closed-form expression for the NDT of Nested Block Designs. Although Nested Block Designs depart from Block Designs in some aspects, it is possible to obtain a reasonable approximation for the NDT of the former, based on the expression for the latter. We will, therefore, use the closed-form expression for NDT of Projective Planes, replacing v with $v_o v_i$ as the cycle length. Assuming both nodes operate asynchronously under the Nested Block Design $\{v_o, k_o, 1\} \#\{v_i, k_i, 1\}$, we declare that Equation 4.1 provides the E[NDT] with a gap proportional to $(1/v_o + 1/v_i)$. A proof is provided in Appendix B, Section B.5.

$$E[NDT]_{nested} = \frac{v_i v_o}{p} - \frac{v_i v_o + 1}{2}$$
(4.1)

By computing Nested schedules $\{v_o, k_o, 1\} \# \{v_i, k_i, 1\}, v_i \geq v_o$, that result in duty cycles of less than duty cycle of $\{9507,98,1\}$ (1.03%), we found that the worst gap (for the $\{7,3,1\} \# \{1893,44,1\}$ design) is 8.9% and, for any schedule with duty cycle of less than 0.1%, the biggest gap, that happens for $\{133,12,1\} \# \{8011,90,1\}$, is 0.6%. The gap decreases with the decrease in duty cycle, as expected¹.

Figure 4.6 shows the percentual difference from the closed-form expression results to the exact values obtained with our method. Each line represents a family of schedules formed by all Nested Designs with the same outer design — within a family all schedules present the same approximate gap, with relative difference of less than 0.5% between any two, which results in less than 0.05% of difference in the gap to the exact model, in the worst case. Therefore, the gap is approximately the same for all schedules in the same family, but drops noticeably as the outer design lengthens. For all values of p, the gap is less than 2% for all Nested Designs with outer schedules longer than 57 slots, which means that the simplification is reasonable.

Moreover, as the gap in the closed-form expression for Block Designs is so low (less than 0.05% for all designs with duty cycle of less than 1%), the results in Figure 4.6 are also an indicator of the increase in the NDT of Nested Block Designs in relation to the NDT of Block Designs. This difference will always remain under 10.6%, and will be of less than 2% for outer schedules longer than 57 slots.

4.3 Duty cycle may be arbitrarily low

With Nested Block Designs, a duty cycle as low as 0.01% can be achieved (for example, with the schedule $\{9507, 98, 1\}\#\{9507, 98, 1\}$). In fact, if more than one *level of nesting* is employed, the duty cycle can be arbitrarily low. A second level of nesting would be

¹We have not computed all 596 schedules, for some have more than 10^7 slots. There is clear indication that the gap decreases monotonically as the inner schedule increases in cycle length. Therefore, we computed all combinations of designs with length < 1457.



Figure 4.6: The gap between the simplified closed-form and the exact model for Nested Block Designs.

achieved by subdividing the slots of the inner design into superslots, further divided into slots. In practice, nesting can proceed in as many levels as desired and achieve arbitrarily low duty cycles.

Figure 4.7 presents the resulting duty cycle for all Nested Block Designs formed with Projective Planes in four ranges. Projective Planes are represented by their length (v). The combinations in white are of limited use, since there are Projective Planes with the same duty cycle.

4.4 Duty cycle granularity is significantly improved

There are many ways to measure the duty cycle granularity of Nested Block Designs. Obviously, we should not count schedules that result in the same duty cycle. Therefore, $\{v_a, k_a, 1\} \#\{v_b, k_b, 1\}$ and $\{v_b, k_b, 1\} \#\{v_a, k_a, 1\}$ count as the same. Likewise, two variants of the same schedule should also count as the same (there are, for example, 10 different forming sets for $\{1023, 511, 255\}$).

However, there are also different criteria that will result in different values for the granularity of Nested Block Designs. First, we may or may not consider Block Designs with $\lambda > 1$. Secondly, there is the question of the level of nesting $(\{v_o, k_o, \lambda_o\} \# \{v_i, k_i, \lambda_i\})$ has one level, $\{v_o, k_o, \lambda_o\} \# \{v_i, k_i, \lambda_i\} \# \{v_t, k_t, \lambda_t\}$ has two levels of nesting). When the level of nesting increases, the granularity also increases. Table 4.1 presents the granularity of Nested Block Designs considering all Block Designs or only Projective Planes, for levels of nesting 1 to 3. Granularity increases rapidly with the level of nesting and, if all Block Designs are considered, it surpasses all other schedules for level of nesting 2



Figure 4.7: All combinations of Projective Planes (each represented by their corresponding length v) and their approximate duty cycles. Duty cycles as low as 0.01% are possible.

Table 4.1: The granularity of Nested Block Designs.

Level of nesting	Projective Planes only			A	all Block De	signs
	1 to 10%	0.1 to 1%	0.01 to 0.1%	1 to 10%	0.1 to 1%	0.01 to 0.1%
1	61	276	290	2229	1502	305
2	90	1295	15500	256517	292563	129540
3	38	2108	560555	25574405	33239617	32218860

(considering Disco with level of unbalancing 5), as can be seen by comparing Table 4.1 with Tables 3.1, 3.2, 3.3 and 3.4.

4.5 Asymmetric operation is possible

For achieving asymmetric operation, we need designs with different duty cycles that are still able to interoperate, i.e. a scheme with rotation closure. In order to achieve rotation closure with Nested Block Designs, we introduce a new variant of these schedules and extend our notation. We will refer to a $\{v, k, \lambda\} \# N$ schedule as a $\{v, k, \lambda\}$ schedule where each superslot is divided into N slots that will be either all active, for an active superslot, or all inactive, for an inactive superslot. Figure 4.8 represents the concept for a $\{7,3,1\} \# 13$ Nested Design. Note that the duty cycle of the schedule $\{v, k, \lambda\} \# N$ is the same as the duty cycle of the schedule $\{v, k, \lambda\}$.



Figure 4.8: A $\{7,3,1\}$ #13 Nested Block Design. The outer design is $\{7,3,1\}$. But in this case, each slot of an active superslot is also active and, likewise, each inactive superslot consists only of inactive slots.

These new Nested Designs permit asymmetric operation since a scheme formed by rotations of $\{v_o, k_o, \lambda_o\} \# v_i$ and $\{v_o, k_o, \lambda_o\} \# \{v_i, k_i, \lambda_i\}$ present rotation closure. This means that a node operating under the $\{91,10,1\} \# 183$ nested design can change its schedule to a $\{91,10,1\} \# \{183,14,1\}$, thus reducing its duty cycle from 11% to 0.84% and still be able to communicate with nodes operating under the former schedule, as well as with other neighbors that also made the switch.

It is worth reminding what is the advantage of operating under $\{91,10,1\}\#183$, in comparison to the more power efficient $\{91,10,1\}\#\{183,14,1\}$. The answer lies in the decreased latency, since an increase in the number of overlapping slots translates into more discovery opportunities and a corresponding decrease on the NDT. The intuition behind the NDT in this case can be easily illustrated by considering all possibilities in our example:

- 1. Both nodes operate under $\{91,10,1\}\#\{183,14,1\}$. In this case, the modal number of discovery opportunities per cycle will be one.
- 2. One node operates under {91,10,1}#{183,14,1}, while the other operates under {91,10,1}#183. In this case, the modal number of opportunities of discovery per cycle will be 10, instead of one in Case 1. These opportunities will be concentrated within a superslot (of 183 slots, in this case).
- Both nodes operate under {91,10,1}#183. The modal number of opportunities of discovery per cycle will be 183 all of the slots of the overlapping superlot.

It is not hard to see that the probability of discovery within the first cycle increases significantly from Case 1 to Case 3, as the typical number of opportunities increases from 1 to 10 and then to 183. A similar example is illustrated in Figure 4.9, which shows the



Figure 4.9: An example of how Nested Block Designs can be combined for asymmetric operation and their resulting duty cyle as a function of p

resulting NDT for the three combinations of schedules $\{7,3,1\}\#\{13,4,1\}$ and $\{7,3,1\}\#13$, calculated with our method. There is a clear trade off between the most power efficient scheme, where both nodes operate under $\{7,3,1\}\#\{13,4,1\}$, and the most robust scheme, where both nodes operate under $\{7,3,1\}\#13$. The intermediary case may be a good compromise if one node's battery is low.

One obvious example of use of these interoperable Nested Designs could be an application where nodes start with the higher duty cycle schedule until a certain event happens as, for instance, the node finds a given number of neighbors. After this event, the node could switch to the less energy consuming schedule. The node may also switch back to the more aggressive schedule, if necessary.

With additional levels of nesting, it is also possible to achieve increased degrees of asymmetry in nodes' duty cycles. A schedule $\{91,10,1\}\#\{91,10,1\}\#\{91,10,1\}$ (duty cycle = 0.13%) is compatible with both $\{91,10,1\}\#\{91,10,1\}\#91$ (duty cycle = 1.2%) and $\{91,10,1\}\#8281$ (duty cycle = 11%).

4.6 Implementing Nested Block Designs

This section provides experimental results that show the feasibility of Nested Block Designs implemented in real sensor motes. Moreover, the experiments reinforce the claim that slot alignment is not a requirement for asynchronous mechanisms to work, and that our method to find the NDT is applicable to the general case, as the experiments with other schedules in Chapter 3 also indicated.

4.6.1 Symmetric operation

As a first test, the Nested Block Design $\{91,10,1\}\#\{183,14,1\}$ was implemented in MicaZ sensor motes, manufactured by Memsic [62]. Because neighbor discovery is a phenomenon involving only two peers, the tests used two sensor motes (A and B), with no loss of generality, since interference caused by other nodes is incorporated as affecting p. The experiment worked as follows. Mote A operates continually on an activity pattern taken from the schedule 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 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 100% during all measurements, indicating that p was, at least, very close to 1 during the entire experiment.

Tested schedule	${91,10,1}\#{183,14,1}$
Slot duration	$9.77 \mathrm{ms}$ (10 ticks of a 1024Hz clock)
Number of measurements	1,000
Link quality	p = 1.0
E[NDT] for $p = 1.0$	80.6s
Experimental NDT	79.8s
95% confidence interval	78.2 to 81.4 s

Table 4.2: Implementation of Nested Block Designs on sensor motes — experimental results for p = 1.0.

In order to assess the model in an error prone scenario, a second battery of tests was performed with the motes placed inside metallic boxes separated by 75cm and transmitter power was reduced to -20dBm. In this controlled setup, the delivery probability was measured during an entire day and determined to be of 0.8. Because a higher variance is to be expected with lower values of p, the test was performed 1,447 times over three days (instead of the 1,000 times in tests with p = 1). Results for this second series of tests are presented in Table 4.3. The same schedule and slot length was used. This time, the expected NDT would be of 122 seconds, and the experimental average NDT was 123.3 seconds, with a 95% confidence interval between 118 and 128 seconds. Once more, an accuracy better than 98% was achieved.

Table 4.3: Implementation of Nested Block Designs on sensor motes — experimental results for p = 0.8.

Tested schedule	${91,10,1}\#{183,14,1}$
Slot duration	9.77 ms (10 ticks of a 1024Hz clock)
Number of measures	1,447
Link quality	p = 0.8
E[NDT] for $p = 0.8$	122s
Experimental NDT	123.3s
95% confidence interval	118 to 128s

Table 4.2 summarizes the experiment and its results. Slots were 9.77 ms long (10 ticks of a 1024 Hz clock). According to our method, for p = 1, the NDT for the schedule $\{91,10,1\}\#\{183,14,1\}$ would be equivalent to 8,248.23 slots, or 80.6 seconds. As shown in the table, the experimental NDT after 1,000 measures was 79.8 seconds. Therefore, the NDT predicted by the model falls within the 95% confidence interval for the experiment. Also, the difference between the average experimental NDT and the model prediction was less than 1%.

4.6.2 Asymmetric operation

To test asymmetric operation two Iris motes [62], A and B, where programmed to operate under the schedules $\{7,3,1\}\#\{13,4,1\}$ and $\{7,3,1\}\#13$, respectively. All the other parameters where kept identical of the tests performed with symmetric operation (Section 4.6.1), except for slot duration —100 ticks for this test.

The results are presented in Table 4.4. The experimental average NDT of 2.80s deviates only 1.45% from the expected value of 2.76s, which is within the confidence interval of the experiment.

4.7 Nested Block Designs and other schedules

With Nested Block Designs we addressed the shortcomings of Block Designs and proposed a schedule that may operate under arbitrarily low duty cycles in asymmetric mode, and yields NDT marginally longer than Block Designs, and therefore significantly shorter than Disco, Grid Quorum or Torus Quorum. Duty cycle granularity was also much increased

Tested schedules	$\{7,3,1\}\#13$ in A and $\{7,3,1\}\#\{13,4,1\}$ in B
Slot duration	$97.7 \mathrm{ms} \ (100 \ \mathrm{ticks} \ \mathrm{of} \ \mathrm{a} \ 1024 \mathrm{Hz} \ \mathrm{clock})$
Number of measurements	1,000
Link quality	p = 1.00
E[NDT] for $p = 1$	2.76s
Experimental NDT	2.80s
95% confidence interval	2.75 to 2.85

Table 4.4: Implementation of Nested Block Designs on sensor motes for asymmetric operation

and it is at least comparable to that of other schedules. As a result, we may simplify the decision tree presented in Figure 3.25 to include only Block Designs and Nested Block Designs, where the former would be advantageous only for duty cycles of more than 1%, and only useful for symmetric operation. For all other scenarios, we demonstrated that Nested Block Designs would out-perform all studied schedules. The new decision tree is presented in Figure 4.10.



Figure 4.10: Decision Tree for selecting a schedule-based mechanism, considering asymmetric operation, duty cycle and link quality, after the introduction of Nested Block Designs.

Chapter 5

Conclusion and Future Directions

This thesis introduced Nested Block Designs — a new schedule for asynchronous duty cycling that presents the advantages of Block Designs (lowest latency for a given duty cycle and rotation closure) while addressing its shortcomings (low granularity, incapability of achieving low duty cycles and of operating in asymmetric mode). These features of Nested Block Designs were demonstrated or supported by analytical and experimental results.

We also provided a method for finding the expected neighbor discovery time for any schedule-based scheme, and a set of closed-form expressions with same purpose, for the most prominent schedules in the literature. The method and a new set of definitions compose a framework that may be used to analyze any future schedule and better understand how they should be designed to hasten neighbor discovery without sacrificing duty cycle. Moreover, we used our method to better understand each of these schedules. A complete list of contributions follows:

- A survey and taxonomy of duty cycling mechanisms, which divides proposals not only into synchronous and asynchronous, but into nine subcategories (Chapter 2).
- Full characterization of schedule-based asynchronous duty cycling, with formal definition of basic concepts, such as schedule, schemes, etc (Section 3.1).
- An exact method to obtain the E[NDT] of any schedule-based mechanisms, with analysis of important design aspects of the schedules (Section 3.2).
- Detailed analysis of Quorum Systems, including the use of the exact method to address pending questions related to column selection, comparison between Grid and Torus (Section 3.3).

- Detailed analysis of Disco, including the use of the exact method to understand the impact of pair balancing on the NDT (Section 3.4).
- Detailed analysis of Block Designs, with comparison between Projective Planes and other Designs. (Section 3.5).
- Closed-form expressions for the E[NDT] of all mechanisms presented in Chapter 3 (Section 3.6.1 and Appendix B).
- Proposal of the relative-latency metric, for better comparison of the mechanisms (Section 3.6.2).
- A comparison between the existing mechanisms and a decision tree to summarize the process of selecting a mechanism that is adequate to a given scenario (Section 3.6.3).
- Introduction of Nested Block Designs, a new schedule-based mechanism, that presents the aforementioned advantages and changes the decision tree (Chapter 4). Nested Block Designs should be used in all cases where a duty cycle of less than 1% is required or when asymmetric operation is desired.

5.1 Future directions

As demonstrated, Nested Block Design schedules are close to optimal in terms of NDT and equal or surpass all other schedules in all usual metrics. There are, however, aspects of the new schedule that deserve further investigation or enhancements.

In relation to the schedule design, the most important aspect is the study of the mixed schemes used for asymmetric operation. The resulting NDT needs better characterization. Also, the scheme could be enhanced to allow for a higher number of different duty cycles. In its current form, the number of possible duty cycles is limited to the level of nesting. Therefore, new and more flexible forms of combining schedules should be investigated.

Still regarding design issues, the use of Block Designs with $\lambda > 1$ may also render useful and interesting Nested Designs that have not yet been studied. In terms of modelling, the complete characterization of the NDT as a random variable, defined by its probability distribution, seems the natural sequence after the determination of its expectation.

If we venture outside the scope of this thesis — schedule design and modeling — other questions may be of interest for further investigation. For example, there is the question of designing protocols that employ such schedules, and the important study of

the interaction between schedule-based asynchronous duty cycling and other techniques of energy conservation and, even more importantly, its interaction with sensory applications.
APPENDIX A – Computation of the NDT

The Python code (for version 3) below computes the NDT for a monotonic scheme based on a generic schedule, for all probabilities in the range 0.05 to 1, with increments of 0.05:

```
# Input form: entrada="schedule description; length; active slots (list, comma separated)"
entrada="{7,3,1};7;0,1,3"
description=entrada.split(";")[0]
cyclelength=int(entrada.split(";")[1])
activeslots_str=(entrada.split(";")[2]).split(",")
activeslots = [int(slot) for slot in activeslots_str]
numactive=len(activeslots)
# ''opportunities'' is a list of lists for all discovery opportunities ordered by the offset
opportunities = []
opportunities.append(activeslots)
for offset in range(1,cyclelength):
    opportunities_offset = []
    for i in range(0,numactive):
        if (activeslots[i] + offset)%cyclelength in activeslots:
            opportunities_offset.append((activeslots[i] + offset)%cyclelength)
    opportunities.append(opportunities_offset)
# ''phi'' is a list of lists of the phi coefficients ordered by the offset
phi = []
off = 0
for ops in opportunities:
    # ''phi_off'' keeps the phi coefficients for the current offset
   phi_off = []
    # Find the delta set of the co-schedule
    delta = []
    for j in range(0,len(ops)):
```

```
delta.append(ops[(j + 1) % len(ops)] - ops[j])
        if (delta[j] <= 0):
            delta[j] += cyclelength
    # Find phi_0
    phi_off_0 = 0;
    for d in range(0,len(delta)):
        phi_off_0 += delta[d] * (delta[d] - 1)
    phi_off_0 = phi_off_0 / (2 * cyclelength)
    phi_off.append(phi_off_0)
    # Find the other phi coefficients
    phi_prev = phi_off_0
    for d in range(1,len(delta)):
        phi_k = 0
        for k in range (0,len(delta)):
            prev = k - d
            if (prev < 0):
                prev += len(delta)
            phi_k += delta[k] * delta[prev]
        phi_k = phi_prev + phi_k/cyclelength
        phi_prev = phi_k
        phi_off.append(phi_k)
    phi.append(phi_off)
    off+=1
# Calculate NDT for p in (0.05, 1, 0.05)
for prob in range (5,105,5):
    p = prob / 100
    NDT = 0
    for off in range(0,cyclelength):
        NDT_off = 0
        numops = len(phi[off])
        # The part which depends on the expected number of cycles
        termcycles = cyclelength * ((1 / (1 - (1 - p) * numops)) - 1)
        # The part which depends on the phi coefficients
        termphi = 0
        for i in range(0, numops):
           termphi += (phi[off][i] * p * (1 - p)**i) / (1 - (1 - p)**numops)
        NDT_off = termcycles + termphi
        NDT += NDT_off
    NDT = NDT / cyclelength
    print (p, NDT)
```

APPENDIX B – Closed-form approximative expressions for the NDT of monotonic schemes

This Appendix provides details on the techniques used to obtain closed-form expressions for monotonic schemes out of Block Designs, Grid Quorum, Torus Quorum, Disco and Nested Block Designs.

B.1 Block Design schemes

This section presents a derivation of Equation 3.8 — a closed-form expression for Block Designs.

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 $e_i, 1 \leq i \leq \lambda$, as the i_{th} common active slot between both nodes within a given cycle (i.e. the i_{th} opportunity of discovery in a cycle). Figure B.1 exemplifies such definitions with an instance where two nodes operate under a $\{15,7,3\}$ design. Clearly, for $0 < i < \lambda$:

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

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

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



Figure B.1: 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.

$$t_i^c = t_1^c + \sum_{j=1}^{i-1} \{e_{i+1} - e_i\}$$
$$E[t_i^c] = \left[v(c-1) + \left(\frac{v+1}{\lambda+1} - 1\right)\right] + (i-1)\frac{v+1}{\lambda+1}$$
(B.2)

and,

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

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

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

B.2 Grid schemes

In order to derive a model of the NDT for the Grid Quorum, we developed a script using the R language [61] to compute the sum of the times needed until the first opportunity of encounter for all n^4 combinations of offset and starting slots, for n = 1, ..., 60. Figure B.2a 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 6^{th} degree polynomial. Using a simple interpolation method, we find that the mean time until the first opportunity of encounter is given by:

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



Figure B.2: 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.1: Values of constants	$a \dots g$	on	Equation	B.5.
--------------------------------	-------------	----	----------	------

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

We reinforced the validity of Equation B.4 by additionally testing the values n = 1000and n = 10000, to find that the curve also fits perfectly to these values. We then proceeded our analysis by executing the same computation, but for the *i*-th opportunity (notice that for the first opportunity, we consider i = 0). Figure B.2b 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 Equation B.4 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 = 10000. Interestingly, the obtained coefficients behave approximately as a 6th degree polynomial function of n. By applying the least squares method, we found the following approximation for the value of the angular coefficients:

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

where the values of $a \dots g$ are given on Table B.1.

Therefore, by combining Equations B.4 and B.5, we can find an approximation for the mean time until the i_{th} opportunity of encounter:

$$E[e_i] = E[e_1] + i \times AngularCoefficient(n)$$
(B.6)

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

$$E[NDT] = -\{[(6a-2)n^{6} + (2-6b)n^{5} + 6cn^{4} + (3-6d)n^{3} + (6e-4)n^{2} + (1-6f)n + 6g]p - 6an^{6} + 6bn^{5} - 6cn^{4} + 6dn^{3} - 6en^{2} + 6fn - 6g\} \cdot \frac{1}{6n^{4}p}$$
(B.7)

However, for values of n > 100, Equation B.7 can be approximated, with less than 1% of error, by Equation B.8:

$$E[NDT] = \frac{(3-p)n^2}{6p}$$
(B.8)

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.

B.3 Torus schemes

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:

$$e_1 = \frac{12n^6 - 16n^5 - 6n^4 + 11n^3 - 9n^2 + 2n}{24n^4} \tag{B.9}$$

And the i_{th} opportunity:

$$e_{i} = \frac{12n^{6} - 16n^{5} - 6n^{4} + 11n^{3} - 9n^{2} + 2n}{24} + i \cdot \frac{an^{6} - bn^{5} + cn^{4} - dn^{3} + en^{2} - fn + g}{n^{4}},$$
 (B.10)

where the values for constants $a \dots g$ are given in Table B.2.

abcdefg0.99942.41876.434122.608574.7884148.5750118.8220

Table B.2: Values of constants $a \dots f$ on Equation B.10.

Table B.3: Values of constants $a \dots f$ on Equation B.13.

a	b	с	d	е	f	g
0.9992	1.9079	7.1622	21.2087	54.0977	80.9621	43.0001

Finally, for even values of n:

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

Similarly, for odd values of n:

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

$$e_{i} = \frac{12n^{6} - 12n^{5} - 6n^{4} + 25n^{3} - 30n^{2} + 11n}{24} + i \cdot \frac{an^{6} - 1.bn^{5} + cn^{4} - dn^{3} + en^{2} - fn + g}{n^{4}},$$
 (B.13)

where the values for constants a to g are given in Table B.3.

$$E[NDT] = -\{[(24a - 12)n^{6} + (12 - 24b)n^{5} + (24c + 6)n^{4} + (-24d - 25)n^{3} + (24e + 30)n^{2} + (-24f - 11)n + 24g]p - 24(an^{6} + bn^{5} - cn^{4} + dn^{3} - en^{2} + fn - g)\} \cdot \frac{1}{24n^{4}p}$$
(B.14)

A simplification similar to that used in the case of the Grid model can also be used for the Torus — for n > 100. Equations B.11 and B.14 can be unified and simplified to the more manageable form presented in Equation B.15, with less than 1% of error.

$$E[NDT] = \frac{(2-p)n^2}{2p}$$
 (B.15)

B.4 Disco schemes

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 θ , $0 \le \theta < q_1q_2$, denote the offset between the current slots of A and B. The co-schedule of A and B can be computed based on the solutions for the following modular equations (where $x_a \dots x_d$ are the variables):

- $\theta + q_1 \cdot x_a \equiv 0 \pmod{q_1} \tag{B.16a}$
- $\theta + q_1 \cdot x_b \equiv 0 \pmod{q_2} \tag{B.16b}$
- $\theta + q_2 \cdot x_c \equiv 0 \pmod{q_1} \tag{B.16c}$

$$\theta + q_2 \cdot x_d \equiv 0 \pmod{q_2} \tag{B.16d}$$

In each equation, the left-hand side represents a set of active slots for B, while the right-hand side represents a set of active slots for A. For example, the left-hand side of Equation B.16a 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 $x_a \dots x_d$ on the left-hand sides, one can obtain all discovery opportunities.

Consider, for example, the case $q_1 = 2$ and $q_2 = 5$. Figure B.3 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, while the gray boxes represent discovery opportunities and the boxes with diagonal lines are active slots of B which are not in the co-schedule. The figure also relates each offset with the solutions for Equations B.16a to B.16d. Consider, for instance, the co-schedule for offset 2. Every $x_a \in \mathbb{Z}$ is a solution for Equation B.16a, resulting in slots 0, 2, 4, 6, 8 of the co-schedule¹. Likewise, Equation B.16b has a solution for slot 0 and Equation B.16c has a solution for slot 2. However, Equation B.16d has no solution.

In order to compute the NDT, we need to analyze the delta-set in the resulting co-

¹We 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, ...

							_												
Schedule of A												Solutions to the modular equations							
							$\Theta + 2x_a \equiv 0 \pmod{2}$		$\Theta + 2x_b \equiv 0 \pmod{5}$		$\Theta + 5x_c \equiv 0 \pmod{2}$		$\Theta + 5x_a \equiv 0 \pmod{5}$						
	0 1	2	3	4	5	6	7	8	9	Case	θ	xa	$\Theta + 2x_a$	Хb	$\Theta + 2x_b$	x _c	$\Theta + 5x_{C}$	xd	$\Theta + 5x_d$
										4	0	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
m										1	2	Z	0,2,4,6,8	,-1,4,9,	0	,-2,0,2,	2	none	none
of										3	3	none	none	,-4,1,6,	5	,-1,1,3,	8	none	none
le										1	4	Z	0,2,4,6,8	,-2,3,8,	0	,-2,0,2,	4	none	none
np										2	5	none	none	,-5,0,5,	5	,-1,1,3,	0	Z	0,5
Che										1	6	Z	0,2,4,6,8	,-3,2,7,	0	,-2,0,2,	6	none	none
Ň										3	7	none	none	,-1,4,9,	5	,-1,1,3,	2	none	none
										1	8	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

Figure B.3: Possible co-schedules if nodes operate under a monotonic scheme generated by Disco using prime numbers 2 and 5. White slots are inactive, slots with diagonal lines are active only in B, and gray slots belong to the co-schedule.

schedule. Based on the offset, there are four distinct cases, also indicated in Figure B.3. Case 1 happens when the offset is a 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.

B.4.1 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 Equation B.16a. Therefore, all slots which are multiples of q_1 belong to the co-schedule. Equation B.16b 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 $q_1q_2(k_b + 1)$ are solutions. Notice, however, that these slots are also multiples of q_1 . For Equation B.16c, 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 $\theta + q_1q_2 \cdot k_c$. Again, all slots are multiples of q_1 . Finally, Equation B.16d has no feasible solutions.

The analysis of these equations shows that the only discovery opportunities in the co-schedule of A and B are the multiples of q_1 . Thus, every two consecutive active slots in the co-schedule 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:

$$E[NDT]_1 = \frac{q_1 - 1}{2} + \left(\frac{1}{p} - 1\right) \cdot q_1 = \frac{2q_1 - p(q_1 + 1)}{2p}$$
(B.17)

B.4.2 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 co-schedule are the multiples of q_2 . Therefore, every two consecutive active slots in the co-schedule are separated by $q_2 - 1$ inactive slots. Hence, the expected NDT for this case, given the success probability of a discovery p, is:

$$E[NDT]_2 = \frac{q_2 - 1}{2} + \left(\frac{1}{p} - 1\right) \cdot q_2 = \frac{2q_2 - p(q_2 + 1)}{2p}$$
(B.18)

B.4.3 Case 3: θ is not a Multiple of either q_1 or q_2

In this case, Equations B.16a and B.16d have no solutions. According to the Chinese Remainder Theorem [54], Equations B.16b and B.16c 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 discovery opportunities 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 discovery opportunities in the co-schedule (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 discovery opportunities in the co-schedule, 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_1q_2 that are not divisible by either 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:

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

$$\Delta_{\theta} = q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 - (q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2)$$
(B.19a)

We can rewrite this equation in two equivalent forms:

$$q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 = \Delta_\theta + q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2$$
(B.20a)

$$q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2 = q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 - \Delta_\theta$$
(B.20b)

Since $[x_b]_0$ and $[x_c]_0$ are solutions for Equations B.16b and B.16c:

$$\theta + \Delta_{\theta} + q_1 \cdot [x_b]_0 + k_b \cdot q_1 q_2 \equiv 0 \pmod{q_1} \tag{B.21a}$$

$$\theta + q_2 \cdot [x_c]_0 + k_c \cdot q_1 q_2 - \Delta_\theta \equiv 0 \pmod{q_2}$$
(B.21b)

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

$$\Delta_{\theta} \equiv -\theta \pmod{q_1} \tag{B.22a}$$

$$\Delta_{\theta} \equiv \theta \pmod{q_2} \tag{B.22b}$$

By definition, θ is not divisible by either q_1 or 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 a co-schedule, we can substitute Δ_{θ} for Δ_i in Equations B.22a and B.22b:

$$\Delta_i \equiv -\theta \; (\mathrm{mod} \, q_1) \tag{B.23a}$$

$$\Delta_i \equiv \theta \pmod{q_2} \tag{B.23b}$$

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_1q_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 $\frac{q_1q_2-\Delta_{\theta}-1}{2}$. Hence, on average, the time until the first opportunity is given by:

$$\left[\Delta_{\theta} \frac{\Delta_{\theta} - 1}{2} + (q_1 q_2 - \Delta_{\theta}) \frac{q_1 q_2 - \Delta_{\theta} - 1}{2}\right] \cdot \frac{1}{q_1 q_2} = \frac{(q_1 q_2)^2 + (-2\Delta_{\theta} - 1)q_1 q_2 + 2\Delta_{\theta}^2}{2q_1 q_2}$$
(B.24)

If the first opportunity fails, the nodes have to wait, respectively, $q_1q_2 - 1$ or Δ_{θ} more slots. On average, the total time until the second opportunity is:

$$\left[\Delta_{\theta} \frac{2q_1q_2 - \Delta_{\theta} - 1}{2} + (q_1q_2 - \Delta_{\theta})\frac{q_1q_2 + \Delta_{\theta} - 1}{2}\right] \cdot \frac{1}{q_1q_2} = \frac{(q_1q_2)^2 + (2\Delta_{\theta} - 1)q_1q_2 - 2\Delta_{\theta}^2}{2q_1q_2} \tag{B.25}$$

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

$$E[NDT]_{3}(\theta) = q_{1}q_{2}\left(\frac{1}{1-(1-p)^{2}}-1\right) + \frac{p}{1-(1-p)^{2}} \cdot \frac{(q_{1}q_{2})^{2}+(-2\Delta_{\theta}-1)q_{1}q_{2}+2\Delta_{\theta}^{2}}{2q_{1}q_{2}} + \frac{p(1-p)}{1-(1-p)^{2}} \cdot \frac{(q_{1}q_{2})^{2}+(2\Delta_{\theta}-1)q_{1}q_{2}-\Delta_{\theta}^{2}}{2q_{1}q_{2}}$$
(B.26)

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 $|ND_{q_1,q_2}|$. This results in an expected NDT for this case of:

$$E[NDT]_{3} = \left[\frac{(q_{1}-1)\left(2\,q_{1}\,q_{2}\,p^{2}-q_{2}\,p^{2}+3\,p^{2}-6\,q_{1}\,q_{2}\,p-6\,p+6\,q_{1}\,q_{2}\right)}{6\,(p-2)\,p} + \frac{(q_{2}-1)\left(2\,q_{1}\,q_{2}\,p^{2}-q_{1}\,p^{2}+3\,p^{2}-6\,q_{1}\,q_{2}\,p-6\,p+6\,q_{1}\,q_{2}\right)}{6\,(p-2)\,p} - \frac{(q_{1}\,q_{2}-1)\left(q_{1}\,q_{2}\,p^{2}+p^{2}-3\,q_{1}\,q_{2}\,p-3\,p+3\,q_{1}\,q_{2}\right)}{3\,(p-2)\,p}\right] \cdot \frac{1}{q_{1}q_{2}-q_{1}-q_{2}+1} (B.27)$$

B.4.4 Case 4: θ is 0

This case is simple from the point of view of finding out the co-schedule, 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 $\frac{1}{q_1q_2}$ 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 $\frac{1}{q_1q_2}$. Therefore, for balanced pairs $(q_1 \sim q_2)$, the error drops quadratically with the decrease of the duty cycle.

B.4.5 Averaging the Cases

The previously analyzed cases do not happen with the same frequency. Cases 1 and 2 happen $\frac{q_2-1}{q_1q_2-1}$ and $\frac{q_1-1}{q_1q_2-1}$ of the times, respectively. Case 3 happens $\frac{q_1q_2-q_1-q_2+1}{q_1q_2-1}$. By weighting all three cases, we obtain the final expression for the NDT for Disco:

$$E[NDT] = \frac{1}{3(q_1q_2 - 1)(p - 2)p} \left[-\left(\left(q_1^2 - q_1 \right) q_2^2 + \left(-q_1^2 + 6q_1 - 2 \right) q_2 - 2q_1 - 1 \right) p^2 - \left(\left(3q_1 - 3q_1^2 \right) q_2^2 + \left(3q_1^2 - 18q_1 + 6 \right) q_2 + 6q_1 + 3 \right) p - \left(3q_1^2 - 3q_1 \right) q_2^2 - \left(-3q_1^2 + 15q_1 - 6 \right) q_2 + 6q_1 \right]$$
(B.28)

Similarly to the other models, a simplification of the model presented in Equation B.28 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 Equation B.28 to Equation B.29.

$$E[NDT] = \frac{q_1 q_2 (p^2 - 3p + 3)}{3p(2 - p)}$$
(B.29)

B.5 Nested Block Designs

As already mentioned, the approximative closed-form expression for the NDT of Nested Block Designs was obtained from the approximative closed-form expression for Block Designs, already presented in Section B.1 of this appendix. This section is devoted to showing why this formula presents a gap proportional to $(1/v_o + 1/v_i)$ as stated in Section 4.2.1 and supported by experimental data.

The gap comes from a simplification used in the model regarding the possible offsets between the two neighbor nodes. We should confirm that this simplification results in a gap proportional to $(1/v_o + 1/v_i)$. In order to do so, we define θ as the schedule offset of node X in relation to node Y and determine all four cases for θ , that define the four groups of co-schedules bellow:

- Group I: When $\theta = 0$, the schedules of X and Y are perfectly aligned and there are exactly $k_o k_i$ opportunities of discovery per cycle. This case happens with probability $1/(v_o v_i)$ and the resulting co-schedule is the schedule itself.
- Group II: When $\theta = nv_i$, n = 1, 2, ... there will be k_i opportunities of discovery per cycle. This case happens with probability $(v_o 1)/(v_o v_i)$ and the discovery opportunities will be concentrated in a given superslot.
- Group III: In this Group, there are $(v_i 1)$ cases with k_o opportunities of discovery per cycle. These cases occur within two intervals: $1 < \theta < v_i - 1$ and $v_o v_i - v_i < \theta < v_o v_i$ (half the cases in each interval)². These cases happen with probability $(v_i - 1)/(v_o v_i)$ and the discovery opportunities will be distributed throughout the co-schedule, one at each of the active superslots.
- **Group IV**: For all the other cases, the number of opportunities per cycle is 1. This is the simplest and more abundant case.

Group IV is of particular interest for it contains the majority of the cases, and also because its cases entail only one opportunity of discovery per cycle, therefore being the worst in terms of NDT. Our simplification consists of only considering cases in this group. The relative weight of the cases not in Group IV in the NDT is low:

²The exact values of θ in which this case occurs depend on the way the active slots are distributed in the block designs and are irrelevant to our analysis.

$$\overline{G}_{\text{weight}}^{IV} = \frac{1 + (v_o - 1) + (v_i - 1)}{v_i v_o} \approx \frac{1}{v_i} + \frac{1}{v_o}$$
(B.30)

For example, the ratio of cases not in Group IV, $\overline{G}_{weight}^{IV}$, for the Nested schedule {91,10,1}#{183,14,1} (duty cycle = 0.0975%) is only 0.016, while for the design {9507,98,1}#{9507,98,1} (duty cycle = 0.01%) it is as low as 0.00021. Because it does not include this small number of cases, our model is based only on the parameters v_i and v_o of the Nested Block Design, eliminating complex parameters, such as the delta-sets of the many possible co-schedules.

Proof — Let A and B be two nodes operating under a scheme of asynchronous duty cycling based on a Nested Block Design from Projective Planes, $\{v_o, k_o, 1\} \# \{v_i, k_i, 1\}$, and define $v = v_o v_i$ as the resulting cycle length. Consider that the NDT will be calculated starting from a given moment t_0 when nodes A and B become able to communicate with each other. Define θ_{AB} as the schedule offset of A in relation to B and θ_B as the offset of B in relation to t_0 . Since, at moment t_0 , nodes A and B may be in any slot with equal probability, it follows that θ_{AB} and θ_B are both uniformly distributed in the interval [0,...,v-1] (the cycle length).

As a simplification, consider that all co-schedules fall within Group IV. In this case, there will be only one discovery opportunity per cycle. Define e_1 as the slot in which the first opportunity occurs during the first cycle. Clearly, if the discovery does not happen in e_1 (due to a message loss, for example), the next opportunity will happen in $e_1 + v$, then in $e_1 + 2v$, and so on. As a result, if p is the message reception probability, the expected time, in slots, for the discovery, i.e. the E[NDT], is:

$$\mathbf{E}[\mathrm{NDT}]_{nested} = \mathbf{E}[e_1] + \frac{1-p}{p}v$$

In order to find $E[e_1]$ we shall determine e_1 for all possible combinations of θ_B and θ_{AB} . To do so, we take each value of θ_{AB} and make θ_B vary from 0 to v - 1. As illustrated by Figure B.4, for any^3 value of θ_{AB} , if we take all possible values of θ_B (from 0 to v - 1 with unitary increments), all values of e_1 in the interval [0,...,v - 1] will be produced exactly once. In other words for each increment of θ_B the sole discovery opportunity will also advance by one in relation to t_0 (increment 1 modulo v - 1). As a result:

³Because of the simplification of eliminating all cases from Groups I, II and III.



Figure B.4: For a fixed value of θ_{AB} (12, in this example), if θ_B varies from 0 to v - 1, e_1 will assume all values from 0 to v - 1.

$$E[e_1] = \frac{v-1}{2}$$

and,

$$E[NDT]_{nested} = \frac{v-1}{2} + \frac{1-p}{p}v = \frac{v_i v_o}{p} - \frac{v_i v_o + 1}{2}$$

As for the impact of ignoring Groups I to III, note that the relative occurrence of such cases is low (given by Equation B.30) and diminishes as we increase v_i or v_o . But the duty cycle also decreases with v_i and v_o , since:

$$DC = \frac{k_o k_i}{v_o v_i}; \quad k_i \sim \sqrt{v_i}; \quad k_o \sim \sqrt{v_o} \Rightarrow DC \sim \frac{1}{\sqrt{v_o v_i}}$$

Therefore, the schedules with lower duty cycle are less affected by the simplification. Moreover, considering that cases in Groups I to III are those who present a higher density of discovery opportunities, our model works as an upper bound for the NDT.

APPENDIX C – Nested Block Designs: A closer look at the NDT

This appendix is dedicated to the formulation of some parameters of Nested Block Design schedules, such as its delta-sets and Phi-coefficients, that may serve for building and solving optimization problems in the design of these schedules. Some of the results presented in Chapter 4 are empirical. However, it is possible that they could be subject to analytical methods. This appendix provides a better understanding of the resulting co-schedules of Nested Block Designs, which are, in fact, considerably more complex than the co-schedules formed by Block Designs in general, and Projective Planes in particular.

We start this analysis by repeating the four different classes of co-schedules of Nested Block Designs composed with Projective Planes (which were already presented in Appendix B, Section B.5). These four classes of co-schedules are mapped to four classes of offsets, θ , between the two neighbor nodes X and Y:

- Group I: When $\theta = 0$, the schedules of X and Y are perfectly aligned and there are exactly $k_o k_i$ opportunities of discovery per cycle. This case happens with probability $1/(v_o v_i)$ and the resulting co-schedule is the schedule itself.
- Group II: When $\theta = nv_i$, n = 1, 2, ... there will be k_i opportunities of discovery per cycle. This case happens with probability $(v_o 1)/(v_o v_i)$ and the discovery opportunities will be concentrated in a given superslot.
- Group III: In this Group, there are $(v_i 1)$ cases with k_o opportunities of discovery per cycle. These cases occur within two intervals: $1 < \theta < v_i - 1$ and $v_o v_i - v_i < \theta < v_o v_i$ (half the cases in each interval). These cases happen with probability $(v_i - 1)/(v_o v_i)$ and the discovery opportunities will be distributed throughout the co-schedule, one at each of the active superslots.
- **Group IV**: For all the other cases, the number of opportunities per cycle is 1. This is the simplest and more abundant case.

In order to find the expected NDT for a Nested schedule, we need to find the NDT for all of these cases, and average them, weighted by their frequency of occurrence. Note also that the NDT may be determined by the cycle length and order of the co-schedules and from their Phi-coefficients, which, in their turn, are calculated from the delta-sets. Delta-sets are given by the difference between successive discovery opportunities in a co-schedule.

Note that the Phi-coefficients within a group are the same, since all co-schedules within a group have the same delta-set. Moreover, all four different delta-sets (call them Δ_I , Δ_{II} , Δ_{III} and Δ_{IV}) for a schedule $\{v_o, k_o, 1\} \#\{v_i, k_i, 1\}$ may be obtained from the delta-sets of their forming Block Designs, $\{v_o, k_o, 1\}$ and $\{v_i, k_i, 1\}$. Suppose these delta-sets are respectively $\{\delta_0^o, \delta_1^o, ..., \delta_{k_o-1}^o\}$ and $\{\delta_0^i, \delta_1^i, ..., \delta_{k_i-1}^i\}$. Then:

$$\begin{split} \Delta_{I} &= \{\delta_{0}^{i}, \delta_{1}^{i}, ..., \delta_{k_{i}-1}^{i} + (\delta_{0}^{o}-1)v_{i}, \delta_{0}^{i}, \delta_{1}^{i}, ..., \delta_{k_{i}-1}^{i} + (\delta_{1}^{o}-1)v_{i}, ..., \delta_{0}^{i}, \delta_{1}^{i}, ..., \delta_{k_{i}-1}^{i} + (\delta_{k_{o}-1}^{o}-1)v_{i}\}\\ \Delta_{II} &= \{\delta_{0}^{i}, \delta_{1}^{i}, ..., \delta_{k_{i}-1}^{i} + (v_{o}-1)v_{i}\}\\ \Delta_{III} &= \{\delta_{0}^{o}v_{i}, \delta_{1}^{o}v_{i}, ..., \delta_{k_{o}-1}^{o}v_{i}\}\\ \Delta_{IV} &= \{v_{o}v_{i}-1\} \end{split}$$
(C.1)

With these delta-sets, one could obtain all Phi-coefficients for all Groups. The first Phi-coefficient is particularly interesting and meaningful. It represents the NDT when the delivery probability, p, is 1. The four Φ_0 coefficients for the four groups (Φ_0^I to Φ_0^{IV}) are:

$$\Phi_{0}^{I} = \frac{1}{2v_{o}v_{i}} [k_{o}(\delta_{0}^{i^{2}} + \delta_{1}^{i^{2}} + \dots + \delta_{k_{i}-1}^{i^{2}}) + v_{i}^{2}(\delta_{0}^{o^{2}} + \delta_{1}^{o^{2}} + \dots + \delta_{k_{o}-1}^{o^{2}})
+ 2v_{i}(v_{o} - k_{o})\delta_{k_{i}-1}^{i} - v_{i}^{2}(2v_{o} - k_{o}) - v_{i}v_{o}]
\Phi_{0}^{II} = \frac{1}{2v_{o}v_{i}} [(\delta_{0}^{i^{2}} + \delta_{1}^{i^{2}} + \dots + \delta_{k_{i}-1}^{i^{2}}) + 2v_{i}(v_{o} - 1)\delta_{k_{i}-1}^{i} - v_{i}^{2}(v_{o} - 1)^{2} - v_{i}v_{o}]
\Phi_{0}^{III} = \frac{1}{2v_{o}v_{i}} [v_{i}^{2}(\delta_{0}^{o^{2}} + \delta_{1}^{o^{2}} + \dots + \delta_{k_{o}-1}^{o^{2}}) - v_{o}v_{i}]
\Phi_{0}^{IV} = \frac{v_{o}v_{i} - 1}{2}$$
(C.2)

A recurring expression in the set of equations above is the sum of the squares of all deltas in the delta-set of the inner and outer schedules, $(\delta_0^{i^2} + \delta_1^{i^2} + ... + \delta_{k_i-1}^{i})$ and $(\delta_0^{o^2} + \delta_1^{o^2} + ... + \delta_{k_o-1}^{o^2})$, which we abbreviate as φ_o and φ_i in order to obtain a more manageable set of expressions:

$$\Phi_{0}^{I} = \frac{1}{2v_{o}v_{i}} [k_{o}\varphi_{i} + v_{i}^{2}\varphi_{o} + 2v_{i}(v_{o} - k_{o})\delta_{k_{i}-1}^{i} - v_{i}^{2}(2v_{o} - k_{o}) - v_{i}v_{o}]
\Phi_{0}^{II} = \frac{1}{2v_{o}v_{i}} [\varphi_{i} + 2v_{i}(v_{o} - 1)\delta_{k_{i}-1}^{i} - v_{i}^{2}(v_{o} - 1)^{2} - v_{i}v_{o}]
\Phi_{0}^{III} = \frac{1}{2v_{o}v_{i}} [v_{i}^{2}\varphi_{o} - v_{o}v_{i}]
\Phi_{0}^{IV} = \frac{v_{o}v_{i} - 1}{2}$$
(C.3)

It is interesting to note that φ_o and φ_i do not change when the inner or the outer schedules are rotated. From Equations C.3, it is clear that rotating the outer schedule does not affect the NDT, while rotating the inner schedule does affect the NDT, as indicated by the presence of $\delta_{k_i-1}^i$ in the expressions of Φ_0^I and Φ_0^{II} . The smallest $\delta_{k_i-1}^i$ will result in the smallest NDT, what confirms the experimental results presented in Section 4.1.1.

Finally, it may be possible to build an optimization problem from the expressions here presented in order to demonstrate another experimental result — that if the two Projective Planes in a Nested schedule are of different cycle lengths, the longest should be used as the inner schedule (Section 4.1.2). The expression to minimize the NDT for p = 1 is given bellow:

$$E[NDT]_{p=1} = \frac{1}{v_o v_i} \left[\Phi_0^I + (v_o - 1)\Phi_0^{II} + (v_i - 1)\Phi_0^{III} + (v_i v_o - v_o - v_i + 1)\Phi_0^{IV} \right]$$
(C.4)

By analyzing the convexity and monotonicity of the complete expressions for the NDT, it may be possible to demonstrate that, for Nested schedules, if a schedule results in minimum NDT for p = 1, it may also result in minimum NDT for all values of p, for the same duty cycle. We leave this for future work.

References

- [1] A. Thumann and D.P. Mehta, *Handbook of energy engineering*, Fairmont Press, 2008.
- [2] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks", *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.
- [3] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey", *Ad Hoc Netw.*, vol. 7, pp. 537–568, May 2009.
- [4] P. Levis and D. Gay, *TinyOS Programming*, Cambridge University Press, 2009.
- [5] Z. Teng and K. Kim, "A Survey on Real-Time MAC Protocols in Wireless Sensor Networks", *Communications and Network*, vol. 02, pp. 104–112, 2010.
- [6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors", *SIGPLAN Not.*, vol. 35, pp. 93–104, November 2000.
- [7] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki a lightweight and flexible operating system for tiny networked sensors", in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, November 2004, pp. 455–462.
- [8] A. P. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, "A survey of green networking research", *IEEE Communications Surveys and Tutorials*, vol. 14, no. 1, pp. 3–20, 2012.
- [9] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks", *Wirel. Netw.*, vol. 7, no. 4, pp. 343–358, September 2001.
- [10] A. Martirosyan, A. Boukerche, and R. W. N. Pazzi, "A taxonomy of cluster-based routing protocols for wireless sensor networks", in *Proceedings of the International* Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN 2008), May 2008, pp. 247–253.
- [11] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey", *Communications Magazine*, *IEEE*, vol. 44, no. 4, pp. 115–121, April 2006.
- [12] A. Roy and N. Sarma, "Energy saving in MAC layer of wireless sensor networks: a survey", Analysis, 2010.
- [13] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey", *IEEE Network*, vol. 18, no. 4, pp. 45–50, July-August 2004.

- [14] R. C. Carrano, L. C. S. Magalhaes, D. C. M. Saade, and C. V. N. Albuquerque, "IEEE 802.11s multihop MAC: A tutorial", *IEEE Communications Surveys and Tutorials*, vol. 13, no. 1, pp. 52–67, quarter 2011.
- [15] A. Rowe, R. Mangharam, and R. Rajkumar, "RT-Link: A global time-synchronized link protocol for sensor networks", Ad Hoc Netw., vol. 6, no. 8, pp. 1201–1220, November 2008.
- [16] R. Mangharam, A. Rowe, and R. Rajkumar, "FireFly: a cross-layer platform for real-time embedded wireless networks.", *Real-Time Systems*, pp. 183–231, 2007.
- [17] NIST National Institute of Standards and Technology USA, "WWVB radio controlled clocks", http://www.nist.gov/pml/div688/grp40/radioclocks.cfm, February 2012.
- [18] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks", in *Proceedings* of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys '03), New York, NY, USA, 2003, pp. 181–192, ACM.
- [19] L. Bao and J. J. Garcia-Luna-Aceves, "A new approach to channel access scheduling for ad hoc networks", in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, New York, NY, USA, 2001, pp. 210–221, ACM.
- [20] A. Barroso, U. Roedig, and C. Sreenan, "μMAC: an energy-efficient medium access control for wireless sensor networks", in *Proceedings of the Second European Workshop on Wireless Sensor Networks.*, January. 2005, pp. 70–80.
- [21] P. P. Czapski, "A survey: MAC protocols for applications of wireless sensor networks", in *Proceedings of the IEEE Region 10 Conference TENCON 2006*, November. 2006, pp. 1–4.
- [22] G. Lu, B. Krishnamachari, and C. S. 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, April 2004, p. 224.
- [23] S. N. Parmar, S. Nandi, and A. R. Chowdhury, "Power efficient and low latency MAC for wireless sensor networks", in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON* '06), September. 2006, vol. 3, pp. 940–944.
- [24] S. W. Hussain, T. Khan, and S. M. H. Zaidi, "Latency and energy efficient MAC (LEEMAC) protocol for event critical applications in wsns", in *Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS 2006)*, May 2006, pp. 370–378.
- [25] L. Choi, S. H. Lee, and J. A. Jun, "SPEED-MAC: Speedy and energy efficient data delivery MAC protocol for real-time sensor network applications", in *Proceedings of* the IEEE International Conference on Communications (ICC 2010), May 2010, pp. 1–6.

- [26] D. Krüandger, D. Pfisterer, and S. Fischer, "CUPID communication pattern informed duty cycling in sensor networks", in *Proceedings of the Fifth International Conference on Systems and Networks Communications (ICSNC 2010)*, August 2010, pp. 70–75.
- [27] N. A. Vasanthi and S. Annadurai, "AWS: asynchronous wakeup schedule to minimize latency in wireless sensor networks", in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, June 2006, vol. 1, pp. 144–151.
- [28] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", in *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00)*, Washington, DC, USA, 2000, pp. 8020–8029, IEEE Computer Society.
- [29] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing", in *Proceedings of the 7th annual international conference on Mobile* computing and networking (MobiCom '01), New York, NY, USA, 2001, pp. 70–84, ACM.
- [30] W. Ye, J. Heidemann, and 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)*, 2002, vol. 3, pp. 1567–1576.
- [31] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks", in *Proceedings of the 1st ACM International Conference* on Embedded Networked Sensor Systems (SenSys '03), New York, NY, USA, 2003, pp. 171–180, ACM.
- [32] C. Yin, Y. Li, D. Zhang, Y. Cheng, and M. Yin, "DSMAC: An energy-efficient MAC protocol in event-driven sensor networks", in *Proceedings of the 2nd International Conference on Advanced Computer Control (ICACC 2010)*, March 2010, vol. 1, pp. 422–425.
- [33] S. Singh and C. S. Raghavendra, "PAMAS power aware multi-access protocol with signalling for ad hoc networks", SIGCOMM Comput. Commun. Rev., vol. 28, no. 3, pp. 5–26, July 1998.
- [34] F. Xiangning and S. Yulin, "Improvement on leach protocol of wireless sensor network", in Proceedings of the International Conference on Sensor Technologies and Applications, (SensorComm 2007), October 2007, vol. 1, pp. 260 –264.
- [35] J. Polastre, J. Hill, and 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)*, New York, NY, USA, 2004, pp. 95–107, ACM.
- [36] A. El-Hoiydi and J. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for Multi-hop Wireless Sensor Networks", in *Algorithmic Aspects of Wireless Sensor Networks - Lecture Notes in Computer Science*, Sotiris Nikoletseas and José Rolim, Eds., vol. 3121, pp. 18–31. Springer Berlin / Heidelberg, 2004.

- [37] TinyOS, "TinyOS wiki page on low power implementation over the CC2420 radio", http://docs.tinyos.net/index.php/CC2420_Asynchronous_Low_Power_ Listening_Implementation, February 2012.
- [38] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks", in *Proceedings of the 4th ACM International Conference on Embedded Networked Sensor Systems (SenSys '06)*, New York, NY, USA, 2006, pp. 307–320, ACM.
- [39] J. J. Garcia-Luna-Aceves and A. Tzamaloukas, "Reversing the collision-avoidance handshake in wireless networks", in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '99)*, New York, NY, USA, 1999, pp. 120–131, ACM.
- [40] A. El-Hoiydi and J. Decotignie, "Low power downlink MAC protocols for infrastructure wireless sensor networks", *Mob. Netw. Appl.*, vol. 10, no. 5, pp. 675–690, October 2005.
- [41] R. Musaloiu-E., C. M. Liang, and A. Terzis, "Koala: Ultra-low power data retrieval in wireless sensor networks", in *Proceedings of the 7th International Conference* on Information Processing in Sensor Networks (IPSN '08), Washington, DC, USA, 2008, pp. 421–432, IEEE Computer Society.
- [42] Y. Sun, O. Gurewitz, and 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), New York, NY, USA, 2008, pp. 1–14, ACM.
- [43] P. Le-Huy and S. Roy, "Low-power wake-up radio for wireless sensor networks", Mob. Netw. Appl., vol. 15, no. 2, pp. 226–236, April 2010.
- [44] S. Marinkovic and E. Popovici, "Nano-power wake-up radio circuit for wireless body area networks", in *Proceedings of the IEEE Radio and Wireless Symposium (RWS 2011)*, January 2011, pp. 398–401.
- [45] L. Gu and J. A. Stankovic, "Radio-triggered wake-up for wireless sensor networks", *Real-Time Syst.*, vol. 29, no. 2-3, pp. 157–182, March 2005.
- [46] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space", *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, January-March 2002.
- [47] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks", in *Proceedings* of the 9th ACM International symposium on Mobile ad hoc networking and computing (MobiHoc '08), New York, NY, USA, 2008, pp. 53–62, ACM.
- [48] V. Paruchuri, S. Basavaraju, A. Durresi, R. Kannan, and S. S. Iyengar, "Random asynchronous wakeup protocol for sensor networks", in *Proceedings of First International Conference on Broadband Networks (BroadNets 2004).*, October 2004, pp. 710–717.

- [49] P. Basu and C. Chau, "Opportunistic forwarding in wireless networks with duty cycling", in *Proceedings of the third ACM workshop on Challenged Networks (CHANTS* '08), New York, NY, USA, 2008, pp. 19–26, ACM.
- [50] Y. Tseng, C. Hsu, and T. 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)*, 2002, vol. 1, pp. 200–209 vol.1.
- [51] J. Jiang, Y. Tseng, C. Hsu, and T. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks", *Mobile Networks and Applications*, vol. 10, pp. 169–181, 2005.
- [52] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks", in Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '03), New York, NY, USA, 2003, pp. 35–45, ACM.
- [53] M. Maekawa, "A \sqrt{N} algorithm for mutual exclusion in decentralized systems", ACM Trans. Comput. Syst., vol. 3, pp. 145–159, May 1985.
- [54] C. Ding, D. Pei, and A. Salomaa, *Chinese remainder theorem: applications in computing, coding, cryptography*, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996.
- [55] P. Dutta and 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)*, New York, NY, USA, 2008, pp. 71–84, ACM.
- [56] J. Link, C. Wollgarten, S. Schupp, and K. Wehrle, "Perfect difference sets for neighbor discovery", in *Proceedings of the 3rd Extreme Conference of Communication (ExtremeCom 2011)*, September 2011.
- [57] A. Kandhalu, K. Lakshmanan, and 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), New York, NY, USA, 2010, pp. 350–361, ACM.
- [58] F. Yates, "Incomplete randomized blocks", Annals of Eugenics, vol. 7, no. 2, pp. 121–140, 1936.
- [59] C. J. Colbourn and J. H. Dinitz, Handbook of Combinatorial Designs, Second Edition (Discrete Mathematics and Its Applications), Chapman & Hall/CRC, 2006.
- [60] D. Gordon, "La Jolla cyclic difference set repository", http://www.ccrwest.org/ diffsets.html, February 2012.
- [61] R. Gentleman and R. Ihaka, "The R project for statistical computing", http: //www.r-project.org/, February 2012.
- [62] Memsic Corporation, "MEMSIC", http://www.memsic.com/, February 2012.