**Number-Theoretic Sequence Design for Uncoordinated
Resource Block Assignments in Relay-Assisted
Machine-Type Communication Systems**

by

Yaser Fouad, B. Sc., M. A. Sc.

A DISSERTATION SUBMITTED TO THE
FACULTY OF GRADUATE AND POSTDOCTORAL AFFAIRS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
Doctor of Philosophy in Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario
June 2018

Abstract

Terminal relaying offers effective means for realizing machine-type communication (MTC) systems. However, coordinating these relaying terminals (RTs) becomes a cumbersome task as the number of RTs increases. In the absence of channel state information, the efficient utilization of RTs requires a mechanism by which RTs can autonomously assign available resource blocks (RBs) to large numbers of uncoordinated MTC devices with minimal conflicts. Unlike random RB assignments, using prescribed assignment sequences provides an opportunity for obtaining performance gains. However, realizing these gains requires optimizing RB assignments over a large set of sequences. One technique for selecting assignment sequences is based on an exhaustive search of exponential complexity over sequences generated by multiplicative cyclic groups. This technique restricts the RB assignment sequence length and does not consider sequences generated using other group operations.

In this thesis, we use group isomorphism to eliminate the constraint on the sequence length and to show that the optimal assignment sequences generated by a specific cyclic group are globally-optimal over the set of all cyclically-generated sequences. We also develop a greedy algorithm with polynomial complexity for the sequential selection of assignment sequences in systems with large numbers of RTs. This algorithm is further simplified by invoking the graphical representation of cyclic groups to enable its application in massive MTC systems. To further improve performance, we extend the RB assignment sequences set by considering multi-generator sequences. In particular, the Chinese remainder theorem (CRT) is used to combine cyclically-generated sequences into longer ones. This combining process introduces additional degrees of freedom in sequence generation, thereby enriching the set of assignment sequences. We then use group isomorphism to show that the set of pseudonoise maximal length sequences (M-sequences) constitutes a subset of the cyclically-generated sequences set and that their performance is upper bounded by

that of cyclically-generated and CRT-based sequences. This isomorphism is also used to extend the ease of generation property of M-sequences to their corresponding cyclically-generated supersets. Finally, we compare the correlation properties of CRT-based, Zadoff-Chu (ZC), and M-sequences and discuss the applicability of the CRT-based sequences for synchronization signalling in practical systems with large carrier frequency offsets.

Acknowledgements

Before beginning, I am extremely grateful to Allah for allowing me to reach this point in pursuing my goal. I am also extremely thankful for my supervisors Prof. Halim Yanikomeroglu and Prof. Ramy Gohary for their abundant support and encouragement during the course of my Ph.D. degree. I am also thankful for my colleagues in our research group for their encouragement and insightful suggestions for improving this work along the long journey.

I would also like to acknowledge Huawei Technologies Ltd. for financially supporting this research and for their insightful comments that helped in improving this work.

I would also like to thank Dr. Sivanesan Kathiravetpillai for his insightful comments and fruitful discussions.

Finally, I would like to dedicate this work to my beloved parents, brother, wife and daughter for their generous support, understanding, and unconditional love in every single moment of the journey. Without their words and smiles, I would have never been able to continue.

Table of Contents

Abstract	ii
Acknowledgements	iv
Table of Contents	v
List of Figures	viii
List of Acronyms	xi
List of Symbols	xii
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Thesis contributions	10
1.3 Published peer-reviewed papers	12
1.4 Issued patents	13
1.5 Thesis summary	14
1.5.1 Chapter 2	14
1.5.2 Chapter 3	15
1.5.3 Chapter 4	15
1.5.4 Chapter 5	15
1.5.5 Chapter 6	16
1.5.6 Chapter 7	17
1.5.7 Chapter 8	17
Chapter 2: Number Theory Preliminaries	18
2.1 Cyclic groups	18
2.1.1 Additive cyclic groups (ACGs)	20
2.1.2 Multiplicative cyclic groups (MCGs)	21
2.1.3 Effect of cyclic shifts on cyclically-generated sequences	22
2.1.4 The arithmetic inverse	22
2.2 The Chinese remainder theorem (CRT)	23
2.2.1 A systematic approach for solving a set of linear congruences	24

Chapter 3:	System Model	26
3.1	System model	26
3.2	Cyclic-structured RB assignment sequences	30
3.3	CRT-based RB assignment sequences	32
Chapter 4:	Globally Optimal Cyclically-Generated Sequences	33
4.1	Global optimality over the set of cyclically-generated sequences	33
4.1.1	Application of group isomorphism in RB assignment	34
Chapter 5:	Greedy-Based Sequences Design for Autonomous Resource Block Assignments in Relay-Assisted Systems	39
5.1	A greedy-based approach	39
5.1.1	Pairing of group generators	41
5.1.2	Sequential selection of assignment sequences	42
5.1.3	Performance evaluation of cyclic assignment sequences	45
5.1.4	Computational complexity of the greedy algorithm	49
5.2	The graphical greedy algorithm	51
5.2.1	Computational complexity of the graphical greedy algorithm .	57
5.3	Numerical examples	60
Chapter 6:	Chinese Remainder Theorem-Based Sequence Design for Autonomous Resource Block Assignments in Relay-Assisted MTC Systems	74
6.1	A CRT-based RB assignment scheme	75
6.1.1	Association between sequence elements and RBs	75
6.1.2	Signalling protocol for RB assignments	76
6.1.3	CRT-based sequence generation for RB assignments	78
6.1.4	The cardinality of the CRT-based sequences set	81
6.1.5	Bounds on the cardinality of CRT-based sequences	87
6.1.6	Computational complexity of the CRT-based scheme	91
6.1.7	A greedy-based approach for sequences selection	94
6.2	Simulation results	98
6.2.1	Simulation environment	98
6.2.2	Numerical examples	99
Chapter 7:	Pseudonoise-Based Resource Block Assignments in Relay-Assisted MTC Systems	114
7.1	Motivation	114
7.2	Preliminaries of PN-based sequences	116
7.3	Isomorphism between the cyclically-generated and the M-sequences .	117
7.3.1	Performance of maximal length RB assignment sequences . . .	118
7.3.2	Generation of binary M-sequences using the ACG structure . .	118
7.3.3	Generation of cyclic sequences using shift registers	119
7.4	Simulation results	121

7.4.1	Simulation environment	122
7.4.2	Numerical examples	124
Chapter 8:	Summary and Future Work	134
8.1	Thesis summary	134
8.2	Proposed future work	136
Appendix A:	Proof of Theorem 2	139
A.1	Proof of Theorem 2	139
Appendix B:	Proof and Discussion on The Upper Bound on $q(N)$ in	
	Lemma 4	142
B.1	Proof of Lemma 4: upper bound on $q(N)$	142
B.2	Discussion on the tightness of the bound on $q(N)$	143
References		144

List of Figures

1.1	A terminal relay assisting communication between a machine-type communication device and a base station.	2
2.1	The Euler Totient function for the first 500 integers.	20
3.1	A cellular machine-type communication system assisted by relaying terminals.	28
5.1	The greedy algorithm.	44
5.2	A graphical representation of a cyclic group of order 10. This cyclic group has 2 group generator pairs that are represented by the outer circle and the inner pattern, respectively.	54
5.3	Rotation of the inner pattern of $\mathbb{Z}_{12}$ to minimize the number of hits between the dominant resource blocks (the first element of the inner pattern is marked by an arrow for ease of exposition).	56
5.4	Comparison between the computational complexities of the proposed greedy algorithms and exhaustive search for systems with $M = 6$ and $M = 10$ relaying terminals.	59
5.5	Comparison between randomly-generated assignment sequences and the cyclically-generated sequences obtained by exhaustive search and the greedy algorithms.	62
5.6	Comparison between randomly-generated assignment sequences and the cyclically-generated sequences obtained by exhaustive search and the greedy algorithms with the hit identification and avoidance technique.	63
5.7	Comparison between the random and the graphical greedy-based resource block assignment sequences with and without the hit identification and avoidance technique for $N = 520$ and $N = 750$ resource blocks.	65
5.8	Comparison between the random and the graphical greedy-based resource block assignment sequences with and without the hit identification and avoidance technique for $M = 30$ and $M = 60$ relaying terminals.	67
5.9	An example of an association between relaying terminals and machine-type communication devices.	69
5.10	Aggregate data rates of random, greedy-based and exhaustive search-based resource block assignment sequences.	70

5.11	CDF of the number of resource block assignment attempts needed for successful assignment.	72
6.1	A mapping example between sequence elements and resource blocks. .	76
6.2	Sequence generation using the Chinese remainder theorem.	79
6.3	Comparison between the number of cyclically-generated sequences, $\bar{N}$, and $N!$	85
6.4	Comparison between the Chinese remainder theorem-based sequences, the cyclically-generated sequences, and the random assignments. . . .	100
6.5	Effect of combining order on the performance of the Chinese remainder theorem-based sequences for $N = 30$	102
6.6	Comparison between the random assignments, the shuffled pseudonoise-based sequences, and the proposed Chinese remainder theorem-based sequences for $N = 63$ resource blocks.	103
6.7	Comparison between the random assignments, the shuffled pseudonoise-based sequences, and the proposed Chinese remainder theorem-based sequences for $N = 511$ resource blocks.	104
6.8	Comparison between randomly-generated sequences, the sequences obtained by the multiplicative cyclic group structure, and the proposed Chinese remainder theorem-based scheme when $100\% < \frac{L}{N} \leq 200\%$. .	106
6.9	A performance comparison between the Chinese remainder theorem, the soft fractional frequency reuse, the cyclic, and the random based resource block assignment schemes for $N = 30$ resource blocks.	107
6.10	Comparison between the aggregate data rates and the average SINR achieved by the cyclically-generated and the Chinese remainder theorem-based resource block assignment sequences for $N = 30$ resource blocks.	110
6.11	CDF of the number of attempts needed for successful resource block assignment for $N = 30$ resource blocks.	112
7.1	Multiplicative shift register-based generation of the M-sequence corresponding to the primitive polynomial, $x^4 + x + 1$	120
7.2	A graphical representation of a cyclic group of order 7. This cyclic group has 3 group generator pairs, each yielding a unique pattern (the two inner patterns and the outer circle).	122
7.3	A performance comparison between M-sequences, Chinese remainder theorem-based sequences, cyclically-generated sequences, and random assignments.	125
7.4	A comparison between the autocorrelation and cross-correlation properties of the Chinese remainder theorem-based sequences and the M-sequences, when $f_o = 0$	126
7.5	A comparison between the ambiguity functions of the Chinese remainder theorem-based sequences of Chapter 6 and the M-sequences for $N = 15$	128

7.6	A comparison between the autocorrelation and the cross-correlation properties of the Chinese remainder theorem-based sequences and the Zadoff-Chu ones, when $f_o = 0$	131
7.7	A comparison between the ambiguity functions of the Chinese remainder theorem-based sequences of Chapter 6 and the Zadoff-Chu sequences for $N = 15$	132
8.1	Relationship between the set of cyclically-generated sequences, the set of CRT-based sequences and the set of all available sequences of a given length.	137

List of Acronyms

3GPP	3rd Generation Partnership Project
ACG	Additive Cyclic Group
BS	Base Station
CRT	Chinese Remainder Theorem
CSI	Channel State Information
CSMA	Carrier Sense Multiple Access
dB	Decibel
DL	Downlink
DSRC	Dedicated Short Range Communications
FDD	Frequency Division Duplexing
FH	Frequency Hopping
HIA	Hit Identification and Avoidance
IoT	Internet-of-Things
IRSA	Irregular Repetition Slotted ALOHA
LTE	Long Term Evolution
MCG	Multiplicative Cyclic Group
M-sequences	Maximal Length Sequences
MTC	Machine-Type Communications
NR	New Radio
OFDMA	Orthogonal Frequency Division Multiple Access
PN	Pseudonoise
RB	Resource Block
RT	Relaying Terminal
SIC	Successive Interference Cancellation
SINR	Signal-to-Noise-Plus-Interference-Ratio
UL	Uplink
ZC	Zadoff-Chu

List of Symbols

$\phi(\cdot)$	The Euler Totient function
B_j	The j -th assigned RB
BW	The assigned bandwidth
c_i	The i -th element of the CRT-based sequence
d	The transmission delay
D	The number of dominant RBs
e	The group identity element
f	The CRT function
f_o	The carrier frequency offset
f_s	The sampling frequency
F	The isomorphism function
$\mathcal{F}$	A field
g	A group generator of a cyclic group
g^{-1}	The multiplicative inverse of the group generator g
$\mathbb{G}$	A group
$\mathbb{G}_n$	The multiplicative cyclic group of order n
$H_{i,j}$	The pairwise hit matrix between RTs i and j
H_{i,j,B_m}	The channel coefficient between RT i and MTC device j on RB m
I_i	The performance metric of the pair (g_i, s_i)
L	The total system load
$\mathcal{L}$	The set of all possible load distributions
m	The number of seed cyclic sequences
M	The number of relaying terminals
N	The total number of resource blocks
$\bar{N}$	The cardinality of the set of CRT-based sequences
$\mathcal{O}(\cdot)$	The Big O notation
p_i	The i -th prime factor of N
P	A prime number

P_n	The noise power
P_w	The power wastage due to collision
P_t	The transmit power
$q(N)$	The number of unique prime factors of N
Q	The total number of MTC devices
R	The aggregate data rate
s	The cyclic shift applied to the group generator g
s^{-1}	The inverse of a cyclic shift
$\bar{s}$	The scheduling interval duration
S	A sequence
t_i	The repetitions of p_i
$\mathcal{T}$	The look-up table to be constructed by the greedy algorithm
$\mathcal{T}_g$	The look-up table to be constructed by the graphical greedy algorithm
$\mathcal{T}_{CRT}$	The look-up table to be constructed by the CRT greedy algorithm
$\mathcal{U}$	The set containing the group generator pairs that have yet to be selected by the greedy algorithm
$\mathcal{U}_g$	The set containing the group generator pairs that have yet to be selected by the graphical greedy algorithm
$\mathcal{U}_{CRT}$	The set containing the group generators of the seed cyclic sequences that have yet to be selected by the CRT greedy algorithm
W	Watt
X_i	The load matrix of an RT i that is assigned the pair (g_i, s_i)
$\mathbb{Z}$	The set of natural numbers
$\mathbb{Z}_n$	The additive cyclic group of order n
$\{^n_k\}$	Stirling number of the second kind

Chapter 1

Introduction

In this chapter, we begin by introducing the motivation behind the proposed work. We then provide a list of the thesis contributions. This is then followed by the list of publications from this thesis. Finally, we conclude this chapter with a summary of the thesis.

1.1 Motivation

Machine-type communications (MTCs) are prospected to be the engine that will underlie a wide range of Internet-of-Things-based (IoT-based) applications, including smart grids and intelligent transportation systems [1–3]. Such applications will involve communication between large numbers of low-power geographically dispersed MTC devices [4, 5]. Communication between these devices can be facilitated by exploiting the currently available cellular infrastructure [6, 7]. However, this structure has two major drawbacks: first, the transmission power of the MTC devices may not be sufficient to communicate directly with the base station (BS); and second, the prospective large number of MTC devices communicating simultaneously with the BS may result in heavy interference [8, 9]. One potential candidate for solving this problem is a system architecture that exploits wireless user terminals to opportunistically act as relaying terminals (RTs) to assist the communications between their

neighboring MTC devices and the BSs. In other words, such wireless terminals are not dedicated for relaying. This approach is desirable because the number of wireless terminals tends to naturally increase with the number of MTC devices and thus these terminals can play a key role in providing connectivity to neighboring MTC devices with no additional cost to the system.

Using terminal relaying, the MTC devices in each cell are grouped into clusters, where each cluster is served with an RT that acts as the cluster head [10–15]. Each of these RTs aggregates data from the MTC devices within its cluster and subsequently relays the aggregated data to the BS as depicted in Figure 1.1.

Figure 1.1: A terminal relay assisting communication between a machine-type communication device and a base station.

This relaying-based approach has been proposed by the 3rd Generation Partner-

ship Project (3GPP) for the Long Term Evolution (LTE) system. In particular, this approach was first initiated in 3GPP for Release 12 of LTE under proximity-based services [16]. However, in this release it was only limited to providing coverage extensions to terminals with unfavorable channel conditions [17]; e.g., cell edge users. This approach was further extended in Release 14 to provide support for vehicle to infrastructure applications [18]. In these applications one vehicle will act as a cluster head and accordingly relay the signals of its neighboring vehicles to the BS. Another 3GPP work stream; i.e., the 3GPP 5th Generation New Radio (5G NR), envisions that wireless terminals can act as relays and accordingly collect data from their neighboring MTC devices over an operator-controlled side-link in the licensed cellular spectrum [19–21]. Since each MTC device communicates only with its neighboring RT, terminal relaying offers the following favorable features:

- The MTC devices can utilize low power communications and low complexity transmitters, thereby reducing their cost and power requirements.
- The data collected by the wireless terminals can be locally processed thus reducing the required resources on the expensive long range link; i.e., the link between the wireless terminal and the BS.
- The massive number of MTC devices attempting simultaneous communication with the BS can be significantly reduced, thereby reducing the interference at the BS and the probability of collision on the random access channel.

Despite its potential advantages, the efficient implementation of terminal relaying in an MTC framework requires a mechanism by which the RTs can allocate the available (time-frequency) resource blocks (RBs) to their assisted devices.

In the presence of global instantaneous channel state information (CSI), the RBs can be centrally allocated by the BS to both the BS-RT and the RT-MTC links; e.g., [22–26]. Despite the advantages of centralized RB allocation, the prospected large number of RTs and their incidental operation suggest that the signalling overhead for centrally coordinating their RB assignments may not be practically feasible. In particular, the network topology of an RT-assisted system is considered to be highly dynamic due to the mobility and the large numbers of RTs and MTC devices. Hence, the signalling required for coordination between neighboring RTs can significantly infringe on the available resources. This is because coordinating RB assignments between neighboring RTs requires two types of signalling: 1) frequent discovery and handshaking to obtain an updated list of neighboring RTs; and 2) RB reservation messages to coordinate RB assignments [27]. To address this drawback, semi-distributed RB assignment schemes were discussed in [28, 29]. In these schemes, the role of the BS is limited to propagating the estimated interference (price) of each link and the devices independently compete for the available resources using game theoretic approaches. Alternatively, when only local instantaneous CSI is available, the RBs can be independently allocated by the RTs to their MTC devices in a distributed manner; e.g., [30–32].

The availability of CSI also enables the implementation of another class of medium access schemes, which does not require explicit RB allocations. One example of such schemes is the coded random one, i.e., the irregular repetition slotted ALOHA (IRSA) scheme, wherein MTC devices would choose their RBs randomly and the RTs use the available CSI to cancel the effect of interference [33, 34]. The underlying concept of this scheme is that each device transmits a random number of replicas of each of its packets over multiple randomly selected RBs with each packet containing

a pointer to the RBs over which its replicas are transmitted. Using these pointers, along with the CSI, an RT can resolve packet collisions by applying a successive interference cancellation (SIC)-based decoding scheme. This work was further extended in [35–37] by optimizing the distribution according to which the devices select the number of replicas as well as the RBs over which the replicas are transmitted. The performance of these schemes is generally good. However, acquiring the CSI necessary for their operation in practice incurs a significant overhead and can seriously infringe on the resources available for communication. In addition, performing the SIC process increases the complexity of the receiver and the packet decoding delay. These drawbacks were partially mitigated in another class of medium access schemes by relaxing the requirement of the instantaneous CSI. In particular, this class uses a statistical model of channel uncertainties to perform the RB assignments based on imperfect CSI; e.g., [38–40]. This deviation from using the instantaneous CSI is mainly motivated by the fact that obtaining statistical CSI requires much less communication overhead compared to its instantaneous counterpart [41]. To further extend the work, MTC devices with different quality of service requirements were considered in [42, 43]. Although this statistical CSI approach reduces the signalling overhead in fixed relaying systems, the prospected high mobility of both the RTs and the MTC devices might result in substantial changes in the actual RT-MTC channels, thereby resulting in highly suboptimal RB assignments. Hence, to extract the potential gains of terminal relaying in MTC systems, it is desirable to design a computationally efficient scheme by which the RBs can be allocated blindly, i.e., without CSI, and autonomously, i.e., without centralized coordination.

In the absence of CSI, autonomous RB allocation schemes that use random RB assignments are available, e.g., [44]. Other examples of random access schemes in-

clude ALOHA and Slotted-ALOHA in which wireless devices transmit their data over randomly selected RBs [45–47]. These schemes dispense with the overhead necessary for centralized coordination and CSI acquisition and thus are implemented in many practical communication systems [48] such as Sigfox [49, 50] and LoRa [51]. Similar to the ALOHA-based scheme, another category of medium access schemes dispenses with the CSI requirement, but relies on a sense-before-transmit strategy when accessing the available RBs. This category of medium access schemes is usually referred to as carrier sensing multiple access (CSMA) [52–54]. Some practical systems that employ such a medium access strategy include the Dedicated Short Range Communications (DSRC) system, which is designed for vehicle-to-vehicle communications [55, 56]. A variation of the CSMA scheme, namely truncated CSMA, was also proposed in [57] to enable distributed RB reservations by MTC devices in a network with stringent quality of service requirements. In [58], the authors proposed a dynamic RB assignment scheme for wireless sensor networks that relies on a CSMA-based approach and a tool from game theory, i.e., regret matching, for RB assignments. Similarly, the work in [59] introduced a dynamic resource allocation scheme for cluster-based Mobile Ad-Hoc Networks (MANETS). In this scheme, devices randomly contend for resources and rely on a sensing-based approach to avoid collisions with their neighbors. This approach was also used by the authors of [60] to develop a distributed RB assignment strategy for wireless sensor networks (WSNs), whereby devices select their resources randomly from a predefined pool that is continuously updated based on the observed collisions with their neighbors. Despite the simplicity of these schemes, the lack of structure in their RB assignment strategy results in, otherwise avoidable, RB allocation conflicts. This drawback of the random schemes is alleviated by the one proposed in [61], wherein the RB assignment

sequences are endowed with a multiplicative cyclic group structure. In this work, the proposed structure was used to obtain cyclically-generated RB assignment sequences and an exhaustive search was performed to select the sequences that minimize the number of RB allocation conflicts between the RTs. Despite its advantages, when the number of RTs increases, the computational cost of exhaustive search over cyclically-generated sequences becomes practically infeasible. As such, the exhaustive search cannot be used for admission control in massive MTC systems with large numbers of RTs and RBs.

In this thesis, we consider a distributed relay-assisted cellular system that supports MTC. In this system, CSI is assumed to be neither available at the RTs nor at the BS. Each RT locally assigns RBs to its incoming MTC devices according to a prescribed RB assignment sequence; i.e., without centralized coordination by the BS. These sequences are generated at the RTs by using the properties of cyclic groups with a particular group operation. However, some group operations impose constraints on the number of RBs that would prevent the direct application of cyclic-structured RB assignments in IoT applications; e.g., the multiplicative group operation constrains the number of system RBs to be a prime number minus one. To alleviate this constraint, we use a property of cyclic groups, namely, group isomorphism, to show that the performance of the cyclically-generated assignment sequences is independent of the group operation, thereby waiving the constraint on the number of system RBs in [61]. A direct consequence of this observation is that the optimal assignment sequences obtained by searching over the set of sequences generated by a specific cyclic group are globally optimal over the set of all cyclically-generated sequences of the same number of elements. We also provide a computationally efficient technique that, in contrast with the scheme in [61], enables the design of RB assignment se-

quences for networks with large numbers of RTs and RBs. In particular, we develop a greedy algorithm, whereby the RB assignment sequences are cyclically-generated in an efficient manner that relies on the structure of cyclic groups and the sequential assessment of the number of assignment conflicts between the sequences. Since the assignment sequences generated by the proposed algorithm are obtained from distinct group generators and cyclic shifts, the quest for designing sequences is reduced to the quest for finding group generators and cyclic shifts that result in minimal assignment conflicts. In each iteration of the proposed algorithm, a pair of group generators is chosen such that the number of assignment conflicts between the sequences that they generate and the ones generated by the previously selected group generators is minimized. The selected group generators are then ranked in a look-up table in an ascending order of their observed number of assignment conflicts. To further simplify the design, we invoke a graphical representation of cyclic groups to gain insight into their structure, and to subsequently develop a graphical greedy algorithm for selecting the group generators and cyclic shifts.

Unlike exhaustive search, the proposed greedy algorithms yield RB assignment sequences with a significantly less computational cost. In particular, the complexity of the proposed greedy algorithms is polynomial, whereas that of the exhaustive search proposed in [61] is exponential. As such, the greedy algorithms enable the application of cyclic-structured RB assignments in MTC systems, which are expected to comprise tens of thousands of RBs and potentially hundreds of RTs. Cyclic RB assignment sequences for systems with that many RTs and RBs are computationally prohibitive when designed by the exhaustive search technique of [61]. Furthermore, the proposed greedy algorithms do not require active RTs to update their RB assignment sequences when new RTs access the system. In particular, for a network with a given number of

RBs, the look-up tables provided by the algorithms contains all the generators of the corresponding cyclic group. When a new RT enters the system at any given instant, it uses the first unutilized group generator for obtaining its RB assignment sequence. This is in contrast with the exhaustive search of [61], which necessitates that all active RTs update their assignment sequences when new RTs enter the system.

Despite the advantages of the cyclically-generated sequences and the simplicity of their generation, restricting the RTs' RB assignment strategy to such sequences might limit their performance. To overcome this limitation, one possibility is to consider sequences that can be generated by multiple group generators. In particular, these multi-generator sequences can be obtained by combining multiple cyclic sequences using a technique such as the Chinese remainder theorem (CRT). Using this technique, each RT can obtain its RB assignment sequence by combining the cyclic sequences generated by multiple group generators belonging to two or more cyclic group structures. In particular, each RT utilizes the CRT to solve a set of linear congruences to obtain each element of its RB assignment sequence. The number of these congruences solely depends on the number of group generators assigned to the RT. Unlike its single group generator counterpart, the proposed CRT-based approach offers an enriched set of structured and easily generated sequences that can be used for RB assignments. Subsequently, this creates an opportunity to further enhance the performance of the autonomous RB assignment scheme employed by the RTs.

Pseudonoise (PN)-based sequences such as the maximal length sequences (M-sequences) play a key role in the realization of a wide spectrum of applications. Examples of such applications include frequency hopping spread spectrum [62] and synchronization signalling in practical systems such as the LTE [63] and the 3GPP 5G NR [64]. Given their wide adoption and the simplicity of their generation, one

might be tempted to use the PN-based M-sequences for autonomous RB assignments in relay-assisted MTC systems. Hence, in this work, we focus attention on the performance of M-sequences, when used for autonomous RB assignments. In particular, we use the concept of group isomorphism to show that the set of M-sequences constitutes a subset of that of the sequences cyclically-generated by the additive cyclic group structure; i.e., each M-sequence can be obtained from a group generator of the additive cyclic group. Using this isomorphism, we conclude that the performance of M-sequences, when used for autonomous RB assignments in relay-assisted MTC systems, will be upper bounded by that of the cyclic and the CRT-based sequences. We then use numerical examples to evaluate the performance gap between the set of M-sequences and their cyclic and CRT-based supersets. In addition, we use M-sequences isomorphism to reduce the computational complexity of generating the cyclic sequences by RTs. In particular, we use this isomorphism and the properties of cyclic groups to develop a technique that extends the simplified generation property of M-sequences to their corresponding cyclically-generated supersets. We then compare the autocorrelation and cross-correlation properties of CRT-based sequences with those of Zadoff-Chu (ZC) and M-sequences. Finally, we conclude this thesis with a comparison between the ambiguity function of CRT-based, ZC, and M-sequences and a discussion on the applicability of CRT-based sequences for synchronization signalling in practical systems with large carrier frequency offsets.

1.2 Thesis contributions

- In Chapter 4, we invoke group isomorphism to show that exhaustive search over sequences cyclically-generated by a given group operation suffices to guarantee

optimality over all cyclically-generated sequences. This results in: 1) exposing the independence between the performance of the cyclic-structured RB assignment sequences and the group operation underlying their generation, thereby reducing the computational complexity of the sequences selection process; and 2) eliminating the constraint imposed by the multiplicative cyclic group structure on the number of RBs to be a prime minus one.

- In Chapter 5, we develop a flexible greedy algorithm that offers a tradeoff between the performance of the cyclically-generated sequences and the computational complexity of their selection process. We then invoke the graphical representation of cyclic groups to develop an efficient greedy algorithm that enables the application of cyclic-structured RB assignments in MTC systems with hundreds of RTs and thousands of RBs. Unlike the exhaustive search technique of [61], the proposed greedy algorithms reduce the signalling overhead in the network by allowing active RTs to maintain their RB assignment sequences when new RTs enter the system.
- In Chapter 6, we use the CRT to combine the cyclic sequences generated by the simple cyclic group structure into longer ones. The combining process offers additional degrees of freedom for sequence generation by allowing the RTs to combine multiple cyclic sequences, thereby yielding a rich set of sequences for RB assignments. In addition, we introduce a greedy-based algorithm to simplify the CRT-based sequences selection process, thereby enabling the application of these sequences in MTC systems with large numbers of RTs and RBs.
- In Chapter 7, we use group isomorphism to show that the set of PN-based

M-sequences constitutes a subset of that of the sequences cyclically-generated by the additive cyclic group structure. Subsequently, we conclude that the performance of M-sequences, when used for autonomous RB assignments, will be upper bounded by that of the cyclic and the CRT-based sequences. In addition, we develop a technique that uses the properties of cyclic groups and the isomorphism of M-sequences to extend the simple shift register-based generation property of M-sequences to their corresponding cyclically-generated supersets. Finally, we compare the correlation properties of CRT-based, ZC, and M-sequences and discuss the applicability of the CRT-based sequences for synchronization signalling in practical systems that suffer from high carrier frequency offsets.

1.3 Published peer-reviewed papers

- **Yaser Fouad**, Ramy Gohary, and Halim Yanikomeroglu, “Chinese remainder theorem based sequence design for resource block assignment in relay-assisted Internet-of-Things communications”, IEEE Transactions on Wireless Communications, vol. 17, no. 5, pp. 3401-3416, May 2018.
- **Yaser Fouad**, Ramy Gohary, and Halim Yanikomeroglu, “Number-theoretic sequence design for uncoordinated autonomous multiple access in relay-assisted machine-type communications”, IEEE Transactions on Vehicular Technology, vol. 66, no. 10, pp. 9018-9034, October 2017.
- Jorge Cabrejas, Sandra Rogers, Daniel Calabuig, **Yaser Fouad**, Ramy H. Gohary, Jose F. Monserrat, and Halim Yanikomeroglu, “Non-coherent open-loop

MIMO communications over temporally-correlated channels”, IEEE Access, vol. 4, pp. 6161-6170, June 2016.

- **Yaser Fouad**, Ramy Gohary, Jorge Cabrejas, Halim Yanikomeroglu, Daniel Calabuig, Sandra Roger, and Jose F. Monserrat, “Time-frequency Grassmannian signalling for MIMO multi-channel-frequency-flat systems”, IEEE Communications Letters, vol. 19, no. 3, pp. 475-478, March 2015.
- **Yaser Fouad**, Ramy Gohary, and Halim Yanikomeroglu, “An efficient greedy-based autonomous resource block assignment scheme for beyond 4G cellular networks with self-organizing relaying terminal”, 15th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) 2014, June 2014, Toronto, Canada.
- Sandra Rogers, Jorge Cabrejas, Daniel Calabuig, Jose F. Monserrat, **Yaser Fouad**, Ramy Gohary, and Halim Yanikomeroglu, “Non-coherent MIMO communication for the 5-th generation mobile: overview and practical aspects”, Waves Journal, no. 6, pp. 5-13, 2014.

1.4 Issued patents

- **Yaser Fouad**, Ramy Gohary, Halim Yanikomeroglu, and Hosein Nikopour, “System and method for Grassmannian signaling in a broadband network”, US patent in collaboration with Huawei Technologies Ltd., granted May 2017, US patent No. 9654324.
- **Yaser Fouad**, Ramy Gohary, Halim Yanikomeroglu, and Gamini Senarath, “System and method for greedy-based autonomous resource block assignment

scheme for cellular networks with self-organizing relaying terminals”, US patent in collaboration with Huawei Technologies Ltd., granted March 2016, US patent No. 9288679.

1.5 Thesis summary

We begin by introducing the preliminaries pertaining to the proposed work in Chapter 2. This is then followed by the underlying system model in Chapter 3. We then introduce the concept of group isomorphism in Chapter 4 and use it to show that the performance of cyclically-generated sequences is independent of the group operation used for their generation. In Chapter 5, we introduce greedy-based algorithms for selecting the cyclically-generated sequences for RB assignments. We then expand the set of available sequences for RB assignment by using the CRT in Chapter 6 and provide a detailed comparison between the performance of the CRT-based, the cyclic and the PN-based M-sequences in Chapter 7. The thesis ends with a summary of the contributions and future work in Chapter 8.

1.5.1 Chapter 2

In the proposed work, we develop sequence-based autonomous RB assignment schemes for MTC systems with self-organizing relaying terminals. The sequences underlying the proposed schemes are either obtained directly by using the cyclic group structure, or by combining multiple cyclically-generated sequences. Hence, in this chapter, we introduce the preliminaries pertaining to the generation of the cyclically-generated sequences. In addition, we discuss the preliminaries underlying the proposed CRT-based technique, which is used in Chapter 6 to expand the set of available sequences for RB assignments.

1.5.2 Chapter 3

In this chapter, we discuss the system model underlying the proposed work. In addition, to highlight the advantages of the proposed RB assignment schemes, we discuss their potential application in relay-assisted MTC systems. In particular, we show how the sequences yielded by the proposed schemes can be used for autonomous RB assignments by the RTs with minimal assignment conflicts.

1.5.3 Chapter 4

To obtain a cyclically-generated sequence, an RT repeatedly applies a group operation on its assigned group generator. Hence, to obtain the optimal RB assignment sequences, one might be tempted to search exhaustively over sequences generated by all possible group operations. This is because, restricting exhaustive search to a specific group operation would guarantee optimality over its particular cyclic group. To reduce the complexity of this exhaustive search, in this chapter, we use a technique from number theory, namely group isomorphism, to show that all cyclic group structures are isomorphic to the additive cyclic group structure. In other words, performing an exhaustive search over the sequences obtained by the additive cyclic group structure is sufficient to guarantee optimality over all cyclically-generated sequences irrespective of their underlying group operation.

1.5.4 Chapter 5

Following up in Chapter 4, it will be shown that performing an exhaustive search over the sequences cyclically-generated by the additive cyclic group structure yields the optimal cyclically-generated sequences. However, this exhaustive search becomes computationally prohibitive for practical relay-assisted MTC systems with large num-

bers of RTs and RBs. To address this problem, in this chapter, we develop an efficient greedy algorithm that enables sequential selection of the RB assignment sequences, thereby reducing the complexity of the sequences selection process. This algorithm is further simplified by invoking the graphical representation of cyclic groups. Unlike the exhaustive search technique, the greedy algorithms proposed in this chapter are computationally efficient and thus applicable in MTC systems with medium and large numbers of RBs and RTs.

1.5.5 Chapter 6

Despite the favorable features of cyclically-generated sequences, restricting the proposed RB assignment scheme to the set of sequences that can be obtained by a single generator can result in limiting its performance. To address this concern, in this chapter, we utilize a technique from number theory, namely the CRT, to obtain additional degrees of freedom for sequence generation. In particular, we use the CRT to combine two or more cyclically-generated sequences, thereby yielding an enriched set of sequences for RB assignments. We then show that the set of sequences obtained by the CRT can be considered as a superset of the set of cyclically-generated sequences. Hence, when used for autonomous RB assignments, the performance of the sequences obtained by the CRT constitutes an upper bound to that of the cyclically-generated sequences. Finally, we develop a simplified greedy-based algorithm that enables the efficient selection of CRT-based sequences for RB assignments. In particular, the computational complexity of this algorithm is significantly lower than exhaustive search, thereby enabling the application of the CRT-based sequences in MTC systems with large numbers of RTs and MTC devices.

1.5.6 Chapter 7

In this chapter, we investigate the relationship between the cyclically-generated sequences and PN-based M-sequences. In particular, we use the concept of group isomorphism to show that the set of M-sequences constitutes a subset of the set of sequences cyclically-generated by the additive cyclic group structure. Using this isomorphism, we conclude that the performance of M-sequences is upper bounded by that of the cyclic and the CRT-based sequences, when used for autonomous RB assignments in relay-assisted MTC systems and use numerical examples to evaluate the performance gap. In addition, we use this isomorphism along with the properties of cyclic groups to develop a technique that extends the ease of generation property of M-sequences to the cyclically-generated sequences of length, $N = 2^m - 1$, $m \geq 1$. We also investigate the autocorrelation and cross-correlation properties of CRT-based sequences and compare them to those of ZC and M-sequences. Finally, we compare the ambiguity functions of CRT-based, ZC, and M-sequences and accordingly discuss the applicability of the CRT-based sequences for synchronization signalling in practical systems that suffer from high carrier frequency offsets.

1.5.7 Chapter 8

In this chapter, we conclude the thesis with a brief summary and a list of the thesis contributions. We also provide directions for future research.

Chapter 2

Number Theory Preliminaries

In this thesis, we use several number theoretic techniques for the generation and selection of easily generated sequences for autonomous RB assignments in relay-assisted MTC systems. In this chapter, we will focus attention on reviewing the preliminaries pertaining to these techniques. In particular, we will begin by reviewing the cyclically-generated sequences and their properties. This will then be followed by a brief introduction of the CRT. Finally, we will conclude this chapter by introducing a systematic approach for solving the set of linear congruences that is formed by the CRT.

2.1 Cyclic groups

We now review the preliminaries pertaining to the generation of cyclic RB assignment sequences. We begin by recalling the following definitions.

Definition 1 (Groups [65]). *A group $\mathbb{G}$ is a set on which a group operation (denoted by juxtaposition) is defined such that: the operation underlying $\mathbb{G}$ is associative; for all $(x, y) \in \mathbb{G} \times \mathbb{G}$, $xy \in \mathbb{G}$; and for each element $x \in \mathbb{G}$, the inverse x^{-1} lies in $\mathbb{G}$, where $xx^{-1} = e$ and e is the group identity. The order of $\mathbb{G}$ is the number of its elements.* □

Cyclic sequences are generated from a specific subset of groups; i.e., the cyclic groups which are defined next.

Definition 2 (Cyclic Groups [65]). *A group $\mathbb{G}$ is cyclic if all its elements can be generated by repeated application of the group operation on at least one element in $\mathbb{G}$. Such an element is called a group generator.* $\square$

For a cyclic group with N elements, the number of group generators is given by the Euler Totient function $\phi(N)$.

Definition 3 (Euler Totient Function [66]). *Given a positive integer N , the Euler Totient function, $\phi(N)$, is the cardinality of the set of integers that are coprime with N ; that is, $\phi(N)$ is the cardinality of the set $\{q < N \mid \gcd(q, N) = 1\}$, where $\gcd(q, N)$ is the greatest common divisor of q and N .* $\square$

From Definition 3, it follows that for any N , the value of $\phi(N)$ can be easily calculated by counting the integers less than N that are coprime with N . Although the Euler Totient function, $\phi(N)$, is not monotonically increasing, it is known that its value is large for large values of N , e.g., $\phi(100) = 40$, $\phi(310) = 120$, and $\phi(498) = 164$. For illustration, a plot of $\phi(N)$ is given in Figure 2.1 for $N \leq 500$.

From Definition 2, it can be concluded that each of the generators of a group $\mathbb{G}$ with order N yields a unique sequence with N entries. This sequence can be cyclically obtained by repeatedly applying the group operation on the generator according to a predefined structure.

Cyclic groups can be categorized based on their underlying group operation. Without loss of generality, in this work, we will focus attention on additive and multiplicative cyclic groups.

Figure 2.1: The Euler Totient function for the first 500 integers.

2.1.1 Additive cyclic groups (ACGs)

An additive cyclic group is a cyclic group with a modulo- N addition as the group operation. In this case, based on the group definition stated above, for every $x, y \in G$, $x + y \pmod{N} \in G$. For example, the set $\{0, \dots, N - 1\}$ forms an additive cyclic group under modulo- N addition. The number of group generators is given by $\phi(N)$ and for each group generator g the ordered set $(gk_1 \pmod{N}, \dots, gk_N \pmod{N}) = (0, \dots, N - 1)$, for some integers $k_i \in \{1, \dots, N\}$, $i = 1, \dots, N$. In other words, for an additive cyclic group structure of order N , the set of cyclically-generated sequences is obtained by

$$\{kg_i \pmod{N}\}_{k=1}^N, \quad (2.1)$$

where g_i is the i -th group generator of the cyclic group and $i \in \{1, \dots, \phi(N)\}$. For example, for the ACG of order, $N = 6$, there exists $\phi(6) = 2$ group generators;

i.e., $g_1 = 1$ and $g_2 = 5$. The sequences generated by these group generators are $(1, 2, 3, 4, 5, 0)$ and $(5, 4, 3, 2, 1, 0)$, respectively.

2.1.2 Multiplicative cyclic groups (MCGs)

A multiplicative cyclic group is a cyclic group in which the group operation is modulo- N multiplication. That is, for every $x, y \in G$, $xy \pmod{N} \in G$. A classic result in number theory asserts that the set $\{1, \dots, N-1\}$ forms a multiplicative cyclic group under modulo- N multiplication if and only if N is a prime number [67].

Let N be a prime number. A primitive root modulo N is defined to be an integer $g \in \{1, \dots, N-1\}$ such that the ordered set $(g^{k_1} \pmod{N}, \dots, g^{k_{N-1}} \pmod{N}) = (1, \dots, N-1)$, for some integers $k_i \in \{1, \dots, N-1\}$, $i = 1, \dots, N-1$. The number of primitive roots is given by $\phi(N-1)$ and each primitive root by definition generates the set of integers $\{1, \dots, N-1\}$. Hence, it can be concluded that the group generators of the group $G = \{1, \dots, N-1\}$ are the primitive roots modulo N . Subsequently, the set of cyclically-generated sequences of a multiplicative cyclic group of order $N-1$ is obtained by

$$\{g_i^k \pmod{N}\}_{k=1}^{N-1} \quad (2.2)$$

where N is prime, g_i is the i -th primitive root modulo N , and $i \in \{1, \dots, \phi(N-1)\}$.

There is no specific formula for computing the primitive roots modulo a prime number N . However they can be found by efficient techniques. For example, in the case of relatively small primes, primitive roots can be obtained by searching for $g \in \{1, \dots, N-1\}$ such that the sequence $\{g^k \pmod{N}\}_{k=1}^{N-1}$ spans all integers from 1 to $N-1$. The primitive roots are also tabulated for the first few thousand prime numbers. For example, for the MCG of order $N = 7$, there exists $\phi(6) = 2$ primitive roots (group generators); i.e., $g_1 = 3$ and $g_2 = 5$. The sequences generated by these

group generators are $(3, 2, 6, 4, 5, 1)$ and $(5, 4, 6, 2, 3, 1)$, respectively.

2.1.3 Effect of cyclic shifts on cyclically-generated sequences

In this section, we begin by introducing the following Lemma:

Lemma 1 (Effect of cyclic shifts [61]). *Let $\mathbb{G}$ be a cyclic group and let g_1 and g_2 be two distinct generators of $\mathbb{G}$. Then, there is no cyclic shift for which the sequence generated by g_1 coincides with the sequence generated by g_2 .* $\square$

This Lemma implies that cyclic shifts enrich the design of cyclically-generated sequences by an additional degree of freedom. Hence, it follows that for the multiplicative and the additive cyclic groups, enriched sets of cyclically-generated sequences can be obtained by

$$\{g_i^{(k+s_j)} \pmod{N+1}\}_{k=1}^N, \text{ and} \quad (2.3)$$

$$\{g_i(k+s_j) \pmod{N}\}_{k=1}^N, \text{ respectively,} \quad (2.4)$$

where s_j is the j -th cyclic shift, $j \in \{1, \dots, N\}$, $i \in \{1, \dots, \phi(N)\}$, and $N+1$ is prime for the MCG structure.

2.1.4 The arithmetic inverse

In modular arithmetic, the modular multiplicative inverse of an integer x modulo N is represented by x^{-1} and defined as follows:

$$xx^{-1} \equiv 1 \pmod{N}. \quad (2.5)$$

This inverse, x^{-1} , exists if and only if x and N are coprime. For instance, if we consider the values of x and N to be 3 and 17, respectively, then the arithmetic inverse of x , $x^{-1} = 6$, since $3 \times 6 \equiv 1 \pmod{17}$.

For a multiplicative cyclic group of N elements, there exists a unique inverse for each group generator g and its value is given by:

$$g^{-1} = g^{N-2} \pmod{N}. \quad (2.6)$$

This can be easily proven from the properties of multiplicative cyclic groups. In particular, Fermat's little theorem states that for any group generator g of the multiplicative cyclic group of order N , $g^{N-1} \equiv 1 \pmod{N}$ [68]. Now, let g^z be the modular arithmetic inverse of the group generator g , then it follows that $gg^z \equiv g^{N-1} \equiv 1 \pmod{N}$, thereby yielding $z = N - 2$.

2.2 The Chinese remainder theorem (CRT)

To further enrich the set of easily generated sequences for RB assignments, one might consider a technique that yields multi-generator sequences. One example is the CRT, which can be used to combine the cyclic sequences obtained by the group generators of two or more cyclic groups. In particular, the CRT can be used to solve a set of congruences for the sequence elements of cyclically-generated sequences belonging to different cyclic groups, thereby yielding sequences of longer lengths. Hence, in this section, we begin by introducing the CRT.

Theorem 1 (CRT [69]). *Let $\{N_1, \dots, N_m\}$ be a set of pairwise coprime numbers. There exists an integer n that satisfies the system of congruences*

$$\begin{aligned} n &\equiv a_1 \pmod{N_1}, \\ &\vdots \\ n &\equiv a_m \pmod{N_m}, \end{aligned} \quad (2.7)$$

where $a_i \in \mathbb{Z}, i = 1, \dots, m$ and n is unique modulo $\prod_{i=1}^m N_i$. $\square$

In the following lemma, we will show that the solution obtained by the CRT for the congruences system is also unique for each ordered set of $(a_i, \dots, a_m)$.

Lemma 2. *Let n be a solution for a set of congruences $\{n \equiv a_1 \pmod{N_1}, \dots, n \equiv a_m \pmod{N_m}\}$. This solution is unique for all $0 \leq a_i < N_i, i = 1, \dots, m$.* $\square$

To prove Lemma 2, we will use a contradiction as follows:

Proof. Let n be a solution for the two unique sets of congruences given by $\{n \equiv a_1 \pmod{N_1}, \dots, n \equiv a_m \pmod{N_m}\}$ and $\{n \equiv b_1 \pmod{N_1}, \dots, n \equiv b_m \pmod{N_m}\}$, respectively, where $0 \leq a_i < N_i$ and $0 \leq b_i < N_i$. This yields that $n \equiv a_i \pmod{N_i} \equiv b_i \pmod{N_i}$ and hence $a_i = b_i, i = 1, \dots, m$. $\square$

In the following section, we will describe a technique that can be used for obtaining a solution for the set of linear congruences described in (2.7).

2.2.1 A systematic approach for solving a set of linear congruences

We will begin by briefly describing a simple technique for solving a linear congruence and then we will show how this technique can be extended to solve the set of linear congruences described in (2.7).

Consider a linear congruence $ax \equiv 1 \pmod{N}$, where a and N are relatively prime constants and x is the congruence solution. Equivalently, this congruence can be solved by finding a constant c and the value of x such that

$$1 = xa + Nc. \quad (2.8)$$

The values of x and c can be obtained by using a systematic approach in number theory referred to as the Euclidean algorithm. In particular, consider the two integers

ℓ_1 and ℓ_2 with a given greatest common divisor $\gcd(\ell_1, \ell_2)$. The Euclidean algorithm can be used to find the integers c_1 and c_2 such that $\gcd(\ell_1, \ell_2) = \ell_1 c_1 + \ell_2 c_2$. This is done by performing the repeated divisions in the procedure described in [70]. Since $\gcd(a, N) = 1$, applying the Euclidean algorithm on (2.8) yields x and c . This algorithm solves one linear congruence, but cannot be used to solve a set of linear congruences; e.g., the one in (2.7). However, using the pairwise coprime property, the CRT furnishes a systematic approach for solving (2.7) by repeated applications of the Euclidean algorithm. To illustrate, consider the m congruences in (2.7) and let $M_k = (\prod_{j=1}^m N_j)/N_k, k = 1, \dots, m$. From the definition of M_k , it follows that

$$M_k \equiv 0 \pmod{N_j} \text{ if } j \neq k. \quad (2.9)$$

Let x_k be the solution for the following linear congruence:

$$M_k x_k \equiv 1 \pmod{N_k}, \quad k = 1, \dots, m. \quad (2.10)$$

Since $\{N_1, \dots, N_m\}$ is a set of coprime numbers, it follows that M_k and $N_k, k = 1, \dots, m$, are coprime and thus the value of x_k can be obtained using the Euclidean algorithm, cf. (2.8). Multiplying both sides of (2.10) by a_k yields that x_k is a solution for

$$M_k x_k a_k \equiv a_k \pmod{N_k}. \quad (2.11)$$

Subsequently, using (2.9) and (2.11), the solution n , for the set of congruences described in (2.7) can be obtained from the values of $x_k, k = 1, \dots, m$ as follows:

$$n = a_1 M_1 x_1 + a_2 M_2 x_2 + \dots + a_m M_m x_m \pmod{(N_1 N_2 \dots N_m)} \quad (2.12)$$

Chapter 3

System Model

In this chapter, we provide a detailed discussion on the system model underlying the proposed work. In particular, we begin by discussing the envisioned relay-assisted MTC system architecture. We then focus attention on the distributed RB assignment problem in the absence of coordination and CSI information. In particular, we provide a brief introduction on the possible utilization of cyclically-generated sequences for RB assignments at each RT. Finally, we conclude this chapter by briefly introducing how the Chinese remainder theorem can be used to enrich the set of available sequences for RB assignments.

3.1 System model

We consider an autonomous uncoordinated relay-assisted orthogonal frequency division multiple access (OFDMA)-based frequency division duplexing (FDD) MTC system. In this system, wireless user terminals switch to the relaying mode and assist the communication between the BS and the MTC devices within their vicinity. In particular, each of these terminals aggregates data from its neighboring MTC devices and subsequently relays the aggregated data to the BS as depicted in Figure 3.1. MTC devices are not required to use the assistance of the same RT for both uplink (UL) and downlink (DL). In particular, in the considered FDD system, UL

and DL transmissions are performed on separate bands and thus the resources available for UL transmissions will be separate from those available for DL transmissions. Hence, an MTC device can have a decoupled DL/UL association with two RTs, each of which assisting the communication in one direction, thereby enabling MTC devices to realize the gains of decoupled DL/UL associations [71]. In this work, we assume that there exists a dedicated spectrum for communications between RTs and MTC devices. This spectrum is independent from that used for communications between RTs and BSs. This means that RT-BS transmissions do not cause any interference to the MTC devices thus rendering the proposed work readily extensible to multi-cell MTC systems. In other words, both single-cell and multi-cell systems will experience the same performance since BS-RT transmissions do not interfere with those between RTs and MTC devices.

An RT is selected by an MTC device according to the observed signal-to-noise-plus-interference-ratio (SINR). MTC devices are divided into categories prior to system startup and RTs can allocate one or multiple RBs based on the category of the incoming MTC device; i.e., each category is associated with a predefined number of RBs per assignment. For instance, when an MTC device requests an RT's assistance, it sends an assignment request that includes its ID. Using this ID, the RT identifies the device category and accordingly assigns the corresponding number of RBs. However, if an MTC device still requires more RBs, it can send more requests to the RT for additional RB assignments. RTs relay the data of their neighboring MTC devices to their respective BSs over a different set of RBs. Hence, an RT can always aggregate its own data and communicate it along with the relayed MTC data to the BS. RTs communicate the RB allocations to individual MTC devices over a dedicated control channel. This channel can also be used by an RT to notify its assisted

Figure 3.1: A cellular machine-type communication system assisted by relaying terminals.

MTC devices that it is no longer available for relaying. Once notified of the RT's updated status, the MTC devices will select one of the remaining RTs depending on the observed SINR. The selected RT will then assign its available RBs to fulfil the data rate requirements of these MTC devices.

To avoid the excessive signalling required to coordinate the RB assignments of neighboring RTs, each RT is assumed to autonomously assign the available resources to its assisted devices. This assumption also helps in reducing the signalling overhead

over the links between the BS and the RTs since the RB assignments are performed in a distributed fashion; i.e., without centralized coordination by the BS. However, in the absence of coordination, the potential interference between neighboring RTs can have a significant impact on the performance, thereby yielding an interference limited system. To maintain autonomy, two methodologies can be considered for RB assignment. The first methodology uses unstructured random RB allocations [44], whereas the second one performs the RB assignments based on prescribed sequences [61]. In both methodologies, the absence of coordination between the RTs might result in instances at which one RB is assigned to multiple MTC devices simultaneously. Occurrences of such an instance, referred to as a ‘hit’, are likely to result in high interference levels, thereby significantly deteriorating the quality of the signals observed by the MTC devices. In particular, hit occurrences between neighboring RTs result in reducing their observed SINRs, which subsequently impede the average spectral efficiency and the system aggregate data rate; cf. Examples 7 and 15. In addition, these hit occurrences also increase the transmission delay and result in wasting the scarce power of MTC devices since a payload retransmission is required after each hit occurrence; cf. Examples 8 and 16. Hence, to improve the quality of the RT-MTC links, it is desirable for the RTs to reduce the number of hit occurrences. Unfortunately, achieving this goal is not guaranteed if the RBs were to be assigned randomly. This is because random RB assignments do not possess a known structure that would enable the RTs to control the number of hit occurrences. In contrast, RB assignments that are based on prescribed sequences provide an opportunity for optimizing the number of hits, thereby reducing the interference levels observed by the MTC devices. Hence, in the following sections, we will focus attention on the latter approach.

3.2 Cyclic-structured RB assignment sequences

In the absence of coordination, a relay-assisted MTC system can rely on a structured RB assignment scheme to reduce the interference between neighboring RTs. In such a scheme, each RT performs the RB assignments to its assisted MTC devices according to a prescribed assignment sequence with entries corresponding to available RBs. Each of these RB assignment sequences spans all the available RBs in the system and thus each RT has access to all the RBs in the system. This is unlike other RB assignment schemes, e.g., fractional frequency reuse schemes [72–74], that restrict each RT to a specific subset of the available resources. Despite the advantages of this sequence-based RB assignment approach, optimizing the RB assignments over all possible sequences constitutes a formidable task that, for a network with N RBs and M RTs, requires exhaustive search over $\binom{N!}{M}$ combinations of sequences.

When the number of RTs is small, e.g., $M = 4$ RTs, a practical approach is to restrict the exhaustive search to cyclically-generated sequences [61], that is, sequences generated from a single element in the sequence. This is because such sequences possess the following favorable features:

1. Can be easily generated locally by the RTs by repeated applications of a simple group operation, e.g. addition, on a single seed.
2. In contrast with random sequences, the cyclic structure of these sequences can be utilized to further reduce the number of assignment conflicts between neighboring RTs; e.g., by using the hit identification and avoidance algorithm as discussed in Section 5.3.

Instances of cyclically-generated sequences are the PN-based M-sequences, which

are commonly used in Frequency Hopping (FH) systems [75–77]. However, the design of these sequences is based on communication applications that differ significantly from the RB assignment one considered herein. For instance, the design of sequences that are ‘good’ for FH systems is based on minimizing the number of collisions for an application in which each wireless device occupies a particular frequency slot during one dwell interval, and relinquishes it after that. Being generated to meet other design criteria, these sequences do not necessarily perform well when used for assigning RBs in the relay-assisted MTC system considered herein; cf. the discussion on the performance of PN-based RB assignment sequences in Chapter 7. To see this, we note that in MTC systems no spreading is needed. This is in contrast with applications for which PN-based sequences are preferable.

When designing cyclically-generated RB assignment sequences for a network with N RBs, it suffices to consider the order- N cyclic group that generates them. In particular, given a distinct group generator, each RT can generate a cyclic sequence that spans all the RBs available to the network, and the quest for optimal cyclic sequences is distilled to the quest for optimal generators and group operation. In [61], cyclically-generated sequences were obtained using the MCG structure. These sequences were enriched by considering cyclically shifted versions thereof. In particular, applying a cyclic shift, s , to a sequence, cyclically rotates its entries by s slots, where $s \in \{0, \dots, N - 1\}$. In [61], the group generators and cyclic shifts were chosen by exhaustive search to ensure a minimal number of hit occurrences. Despite its advantages in minimizing hit occurrences, performing an exhaustive search over all cyclically-generated sequences by the MCG structure incurs a computational cost that prohibits its utilization in systems with practical numbers of RTs; e.g., exhaustive search cannot be implemented in systems with more than 4 RTs. Moreover,

restricting RB assignments in [61] to sequences generated by the MCG structure imposes a constraint on the number of RBs and guarantees the optimality of the selected sequences only with respect to the MCG structure. These drawbacks will be alleviated by the techniques proposed in Chapters 4 and 5.

3.3 CRT-based RB assignment sequences

To further reduce the number of hits in the absence of coordination, it is desirable to expand the set of sequences for RB assignment while maintaining a structure that facilitates their generation. To meet this objective, one might consider number theoretic techniques to combine the readily available cyclically-generated sequences by the cyclic group structure. This combining process yields an enlarged set of sequences that can potentially reduce the number of hits when used for RB assignments by the RTs. One example on the combining techniques is the CRT, which combines the cyclically-generated sequences by solving a set of linear congruences. In particular, to obtain its sequence, each RT combines the sequences generated by multiple group generators belonging to two or more cyclic group structures. In this case, for each sequence element, a set of simultaneous linear congruences is solved by the RT using the CRT. The number of these congruences solely depends on the number of group generators assigned to the RT. Unlike the single group generator approach discussed in the previous section, the proposed CRT-based approach offers additional degrees of freedom for sequence generation by allowing the RTs to combine multiple cyclic sequences thus yielding a rich set of sequences for RB assignments. The details of the CRT-based approach will be discussed in Chapter 6.

Chapter 4

Globally Optimal Cyclically-Generated Sequences

In the previous chapter, we proposed an autonomous RB assignment scheme for relay-assisted MTC systems. In this scheme, each RT assigns its available RBs to its assisted MTC devices based on a locally-generated cyclic RB assignment sequence. Hence, the proposed scheme creates an opportunity for performance enhancement by reducing the number of assignment conflicts between neighboring RTs. However, for this gain to be achieved, both the cyclic group operation and the cyclically-generated sequences must be carefully selected. To reduce the complexity of this selection process, in this chapter, we use a property of cyclic groups to show that the performance of the cyclically-generated RB assignment sequences is independent of their underlying group operation, thereby reducing the complexity of the search process for the globally-optimal cyclically-generated sequences.

4.1 Global optimality over the set of cyclically-generated sequences

The design of optimal RB assignment sequences for systems with large numbers of RBs and RTs involves a computationally infeasible exhaustive search over a significantly large set of sequence combinations. To address this drawback, a possible

approach is to restrict this search to the set of sequences that can be easily generated from a single generator; i.e., cyclically-generated sequences. This reduces the number of candidate RB assignment sequences for a system with N RBs from $N!$ to $\phi(N)$. However, the entries of the $\phi(N)$ cyclic sequences depend on the group operation used for their generation. Based on this observation, one might be tempted to search exhaustively over sequences generated by all possible group operations; restricting oneself to a specific group operation would guarantee optimality over that particular group.

Fortunately, optimizing the sequences over all group operations is not necessary. Indeed, in this section, we show that the set of cyclically-generated sequences that are optimal with respect to a given group operation are globally optimal over the set of all cyclically-generated sequences of the same order. This follows from the observations that the performance of an RB assignment sequence depends solely on the sequence structure, but not on its actual entities, and that this structure is independent of the group operation. The latter observation follows from a fundamental property of cyclic-groups known as group isomorphism, which is described next.

4.1.1 Application of group isomorphism in RB assignment

For any two cyclic groups of the same order, there exists a structure-preserving function that maps their entities. This function is referred to as group isomorphism, which is formally defined as follows.

Definition 4 (Isomorphism of Cyclic Groups [78]). *Let $\mathbb{G}$ and $\bar{\mathbb{G}}$ be two cyclic groups of the same order. An isomorphism $F : \mathbb{G} \rightarrow \bar{\mathbb{G}}$ is a one-to-one and onto map that respects the group operation, that is, if $x, y \in \mathbb{G}$, then $F(x), F(y) \in \bar{\mathbb{G}}$ and $F(xy) = F(x)F(y)$. $\square$*

Although juxtaposition is used to denote the group operations in both $\mathbb{G}$ and $\bar{\mathbb{G}}$, these operations are generally different.

Definition 4 implies that the isomorphism F can be considered as a relabelling function that maps every element in $\mathbb{G}$ to an element in $\bar{\mathbb{G}}$. In addition, since both groups, $\mathbb{G}$ and $\bar{\mathbb{G}}$, have the same order, N , they also have the same number of group generators, $\phi(N)$, irrespective of their underlying group operations. Using these implications, we will show that, when designing cyclic sequences for RB assignment, the selection of the underlying group operation is immaterial. The key step to show that follows from the following lemma.

Lemma 3 (Isomorphism with Additive Groups [67, 79]). *Any cyclic group, $\mathbb{G}_N$, of order N is isomorphic to the additive cyclic group, $(\mathbb{Z}_N, +) = \{0, 1, 2, \dots, N - 1\}$, irrespective of the group operation that underlies $\mathbb{G}_N$.* $\square$

Although this lemma asserts the existence of an isomorphism between any two cyclic groups of order N , $\mathbb{G}_N$ and $\bar{\mathbb{G}}_N$, it does not provide information on the sequences generated by these groups. In the following theorem, we will show that not only $\mathbb{G}$ and $\bar{\mathbb{G}}$ are isomorphic, but also their cyclically-generated sequences, whereby two cyclic sequences are considered isomorphic if there exists a one-to-one and onto structure preserving function that maps their elements.

Theorem 2. *For any cyclic group $\mathbb{G}_N$ of order N , there exists an isomorphism F such that the set of sequences that are cyclically-generated by the group operation underlying $\mathbb{G}_N$ is isomorphic to the set of sequences that are cyclically-generated by the additive group operation underlying $\mathbb{Z}_N$.*

Proof. See Appendix A.1. $\square$

We note that the isomorphism, F , in this theorem is not unique. In other words, there are multiple isomorphisms that map the sequences of $\mathbb{G}_N$ to those of $\mathbb{Z}_N$. What is common between these isomorphisms is that they all map the set of sequences generated by $\mathbb{G}_N$ to that of those generated by $\mathbb{Z}_N$.

Before discussing the implications of Theorem 2 on the proposed technique, we verify it with an illustrative example.

Example 1. Let $\mathbb{G}_{10}$ and $\mathbb{Z}_{10}$ be the multiplicative and additive cyclic groups of order $N = 10$, respectively. The number of cyclically-generated sequences by both groups is given by $\phi(10) = 4$; cf. Definition 3. For the MCG structure, the set of cyclically-generated sequences is obtained by $\{g_i^{(k+s_i)} \pmod{N+1}\}_{k=1}^N$, cf. Chapter 2.1.2, where g_i is the i^{th} group generator of $\mathbb{G}_N$, s_i is the cyclic shift and $\pmod{N+1}$ is the modular operation that yields the integer remainder of dividing $g_i^{(k+s_i)}$ by $(N+1)$. For ease of exposition, in this example we set all cyclic shifts to 0. Doing so, yields the following sequences for $\mathbb{G}_{10}$:

$(g_1 = 2)$	2	4	8	5	10	9	7	3	6	1
$(g_2 = 6)$	6	3	7	9	10	5	8	4	2	1
$(g_3 = 7)$	7	5	2	3	10	4	6	9	8	1
$(g_4 = 8)$	8	9	6	4	10	3	2	5	7	1

Similarly, the set of cyclically-generated sequences by the ACG structure can be obtained by $\{g_i(k + s_i) \pmod{N}\}_{k=1}^N$, cf. Chapter 2.1.1, where g_i is the i -th group generator of $\mathbb{Z}_N$ and s_i is the cyclic shift. Applying this equation yields the following sequences for $\mathbb{Z}_{10}$:

$(g_1 = 1)$	1	2	3	4	5	6	7	8	9	0
$(g_2 = 3)$	3	6	9	2	5	8	1	4	7	0
$(g_3 = 7)$	7	4	1	8	5	2	9	6	3	0
$(g_4 = 9)$	9	8	7	6	5	4	3	2	1	0

Considering the isomorphism mapping function $\{F : \mathbb{G}_{10} \rightarrow \mathbb{Z}_{10} : F((g_1 = 2)^k \pmod{N+1}) = k \pmod{N}\}$, the elements of the two group structures can be mapped as:

MCG	2	4	8	5	10	9	7	3	6	1
ACG	1	2	3	4	5	6	7	8	9	0

Applying this mapping to the set of cyclically-generated sequences by the MCG structure yields the one generated by the ACG structure. We note that the sequence-to-sequence mapping is not unique. For instance, another mapping function is $\{F : \mathbb{G}_{10} \rightarrow \mathbb{Z}_{10} : F((g_2 = 3)^k \pmod{N+1}) = k \pmod{N}\}$. However, under any isomorphism, the complete set of cyclically-generated sequences by $\mathbb{Z}_{10}$ will be obtained from the one generated by $\mathbb{G}_{10}$. $\square$

To illustrate the implications of Theorem 2 on RB assignment, we note that this theorem asserts that all cyclic group operations will provide sequences with the same structure, but with different labels. In other words, the relative locations between any two elements of a cyclically-generated sequence will be preserved under an isomorphism between any two cyclic groups of the same order. Now, consider the process of RB assignment in a relay-assisted MTC system. A hit will occur between any two RTs when they concurrently assign the same RB irrespective of its label; i.e., when they both use the same element in their respective RB assignment sequences.

Hence, only the relative locations of elements in the RB assignment sequences, i.e., the structures of the sequences, will affect the number of hit occurrences. Since isomorphism preserves the sequence structure across all group operations, performing an exhaustive search over the set of cyclic sequences that is generated by a cyclic group of order N guarantees optimality over the set of all cyclically-generated sequences of the same order. This implies that, without loss of generality, the cyclically-generated sequences by the ACG structure can be utilized for RB assignments in any MTC system with an arbitrary number of RBs, N . More specifically, this observation enables us to eliminate the restriction imposed by the MCG structure on N to be a prime minus one; cf. Chapter 2.1.2.

Chapter 5

Greedy-Based Sequences Design for Autonomous Resource Block Assignments in Relay-Assisted Systems

In this chapter, we provide computationally efficient techniques that enable the design of RB assignment sequences for relay-assisted MTC systems with large numbers of RTs and RBs. In particular, we begin by developing a greedy algorithm, whereby the RB assignment sequences are cyclically-generated in an efficient manner that relies on the structure of cyclic groups and the sequential assessment of the number of assignment conflicts between the cyclic sequences. To further simplify the proposed algorithm, we then invoke the graphical representation of cyclic groups to gain insight into their structure, and to subsequently develop a graphical greedy algorithm for selecting the group generators and cyclic shifts.

5.1 A greedy-based approach

The goal of this section is to develop an efficient technique for designing assignment sequences that reduces assignment conflicts in relay-assisted MTC systems. In these systems, the number of RBs and RTs is expected to be large, which renders RB assignment schemes that rely on exhaustive search rather impractical. To address this

difficulty, the assignment scheme developed herein will rely on a greedy algorithm that sequentially selects the RB assignment sequences from the set of cyclically-generated ones. The performance of this algorithm will be shown to be comparable to that of exhaustive search, but its computational complexity is significantly less.

Consider a group of order N , corresponding to a system with N RBs. For such a system, the greedy algorithm begins by pairing the $\phi(N)$ group generators, so that the two entries of each pair yield reversed cyclic sequences. The algorithm will later associate with each pair a cyclic shift that determines the starting point of each sequence. The optimization of these shifts and their association with the group generator pairs will be clarified below. In the first iteration of the greedy algorithm, a pair is arbitrarily selected, along with a cyclic shift $s = 0$, to be the initial basis of the algorithm. In each of the subsequent iterations, the algorithm augments its basis with one group generator pair and its associated cyclic shift, thereby exhausting all group generator pairs in $\frac{\phi(N)}{2} - 1$ iterations. The group generator pair and the cyclic shift selected in each iteration are tabulated in a look-up table, which is constructed once offline for every value of N supported by the system. For example, a system in which N_1 and N_2 RBs are used in different coverage areas would require two look-up tables. Once constructed for a coverage area with N RBs, the table can be used to generate assignment sequences for any number of RTs less than $\phi(N)$; for M RTs, only the first $\frac{M}{2}$ entries of the look-up table are used to generate the M RB assignment sequences. Note that when an RT becomes unable to assist its neighboring MTC devices, it immediately notifies the BS of its updated status and releases the group generator and cyclic shift pair that it used to generate its RB assignment sequence. The released pair can then be acquired by any new RT that joins the system. However, the procedure through which this pair is acquired depends

on the admission protocol between the RTs and the BSs. In particular, for systems in which the RTs notify BSs of their joining, the look-up table is administrated by BSs and accordingly they can assign the released pair to the new RT. In contrast, for completely autonomous systems, the RTs are self-admitted once they are capable of relaying and thus will need to locally select their pair. In such systems, a new RT will sequentially attempt to utilize the pairs in the same order of its local look-up table until it reaches a released one.

In the remainder of this section, we will elaborate on the details of the pairing technique and the selection methodology of the RB assignment sequences.

5.1.1 Pairing of group generators

Our pairing methodology is based on finding the optimal generators for the two-relay case. In that case, the quest for the optimal cyclically-generated sequences can be distilled to the quest for two assignment sequences that proceed in reversed orders. In particular, using these sequences for RB assignment results in no hits as long as the number of assigned RBs is less than the number of available ones. To obtain sequences that proceed in reversed orders, we begin by noting that for a group generator g and cyclic shift s , there exists a unique inverse (g^{-1}, s^{-1}) , where g^{-1} is the modular arithmetic inverse of g satisfying

$$gg^{-1} \equiv e \pmod{N+1}, \quad (5.1)$$

where in writing (5.1) we used e to denote the group identity element; cf. Chapter 2. To determine s^{-1} , we note that the sequences generated by (g, s) and (g^{-1}, s^{-1}) should proceed in reversed orders, which implies that

$$s^{-1} = N - s - 1. \quad (5.2)$$

For example, consider the pair $(g = 2, s = 0)$ of the MCG of order $N = 10$. The inverse of this pair is $(g^{-1} = 6, s^{-1} = 9)$ and from (2.3), its cyclically-generated sequence becomes $(1, 6, 3, 7, 9, 10, 5, 8, 4, 2)$, which is the reversed version of the sequence generated by $(g = 2, s = 0)$, cf. Example 1 in Section 4.1.

5.1.2 Sequential selection of assignment sequences

Let $\mathcal{T}$ be the look-up table to be constructed by the algorithm and let $\mathcal{U}$ be the set containing the group generator pairs that have yet to be selected. At first, $\mathcal{T}$ is empty and $\mathcal{U}$ contains the $\frac{\phi(N)}{2}$ group generator pairs. The algorithm arbitrarily selects a group generator pair (g, g^{-1}) from $\mathcal{U}$ and, without loss of generality, sets its cyclic shift s to 0, which, using (5.2), yields $s^{-1} = N - 1$. The quadruple (g, g^{-1}, s, s^{-1}) is taken to be the initial basis of the algorithm and the pair (g, g^{-1}) is removed from $\mathcal{U}$. In each of the subsequent $\frac{\phi(N)}{2} - 1$ iterations, the algorithm examines the group generator pairs in $\mathcal{U}$ against a particular performance metric for all possible cyclic shifts $s \in \{0, \dots, N - 1\}$. The quadruple that yields the best performance is then used to augment $\mathcal{T}$ and its group generators are removed from $\mathcal{U}$. The algorithm continues to iterate until the exhaustion of all the pairs in $\mathcal{U}$.

When determining an appropriate performance metric for choosing the group generator pairs and cyclic shifts, it is desirable that this metric enables the algorithm to adapt to both uniform and non-uniform distributions of MTC devices. As such, this metric should not depend on the instantaneous loads of the RTs. One such metric is the average number of pairwise hits over all possible system load distributions. Another metric is based on the graphical representation of cyclic groups, which will be discussed in the following section.

We note that considering the performance metric to be the average number

of pairwise hits over all possible system load distributions facilitates the sequence-selection process. To see this, let (g, g^{-1}, s, s^{-1}) be the quadruple examined at the Q -th iteration of the greedy algorithm. Now, the sequences generated by (g, s) and (g^{-1}, s^{-1}) proceed in reversed orders and thus these sequences result in identical average number of pairwise hits with the sequences generated by the $Q - 1$ quadruples previously selected, $(g_q, g_q^{-1}, s_q, s_q^{-1}) \in \mathcal{T}$, $q = 1, \dots, Q - 1$. Hence, when evaluating the performance of a quadruple (g, g^{-1}, s, s^{-1}) , it suffices to evaluate the metric for the sequence generated by either (g, s) or (g^{-1}, s^{-1}) .

We conclude this section by pointing out a key advantage of using the look-up table yielded by the greedy algorithm. In particular, the order of the quadruples in this table depends solely on the number of pairwise hits, but not on the number of RTs in the system. Thus, it suffices for the system to have only one look-up table to support any number of RTs less than $\phi(N)$. Subsequently, by using the greedy algorithm, active RTs do not need to update their sequences when new ones join the system. In contrast, the exhaustive search scheme in [61], requires having one look-up table for each possible number of RTs. Thus, in the exhaustive search scheme, when a new RT joins the system, the ones already active must switch to the look-up table that corresponds to the new number of RTs.

The greedy algorithm is summarized in the flowchart in Figure 5.1. In this flowchart, we will use I_i to denote the performance metric for a candidate group generator and cyclic shift (g_i, s_i) . This metric will be discussed in detail in the following section.

Figure 5.1: The greedy algorithm.

5.1.3 Performance evaluation of cyclic assignment sequences

In the greedy algorithm, the sequence selection mechanism is based on evaluating the performance of a candidate sequence with respect to the ones previously selected in the look-up table $\mathcal{T}$ only. This table is then augmented by the quadruple corresponding to the sequence yielding the best performance; i.e., in each iteration $|\mathcal{T}|$ increases by 1. In contrast, in the exhaustive search in [61], the sequence-selection mechanism is based on the computationally intensive task of the joint performance evaluation of all cyclically-generated sequences and the look-up table is constructed all at once. In this section, we adapt the average number of pairwise hits metric in [61] to the selection methodology of the greedy algorithm developed herein.

5.1.3.1 The performance metric

Let $X_i \in \{0,1\}^{N \times N}$ be what we refer to as the load matrix of the i -th RT, $i = 1, \dots, M$. The row and column indices of X_i represent the RB index and the load level of the i -th RT, respectively. The (r_1, r_2) -th entry of X_i represents the binary state of RB r_1 when the load level of the i -th RT, is r_2 . This entry is 1 if RB r_1 is assigned by the i -th RT when loaded with r_2 devices, and is 0 otherwise, that is,

$$X_i(r_1, r_2) = \begin{cases} 1 & \text{if } r_1 \in S_i(r_2), \quad \text{and} \\ 0 & \text{otherwise,} \end{cases}$$

where $S_i(r_2) = \{g_i^{(q+s_i)} \pmod{N+1}\}_{q=1}^{r_2}$ is the assignment sequence of the i -th RT and (g_i, s_i) is its group generator and cyclic shift pair. As an illustrative example,

suppose that the pairs (g_i, s_i) and (g_j, s_j) yield the assignment sequences $(1, 3, 2, 4)$ and $(3, 2, 4, 1)$, respectively. The load matrices, X_i and X_j , corresponding to these sequences are

$$X_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad X_j = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

By using these matrices, we can evaluate the performance of two sequences, say S_i and S_j , that are assigned to the i -th and j -th RTs, respectively. To see this, let ℓ_i and ℓ_j denote the loads, and k_i and k_j denote the indices of the last RBs assigned by these RTs. Note that, in general $k_i \geq \ell_i$ because when the i -th RT detects a hit at an assigned RB, it relinquishes this RB and assigns the following one in its RB assignment sequence. Unfortunately, there is no direct relation that maps ℓ_i and ℓ_j to k_i and k_j . However, an iterative method for evaluating k_i and k_j for given RT loads will be provided in the following section. For now, we will assume that the values of k_i and k_j are given. To evaluate the performance of the sequences S_i and S_j for these values, we observe that the number of pairwise hits detected by the RTs is given by the inner product of the k_i -th column of X_i and the k_j -th column of X_j . This product is given by the (k_i, k_j) -th entry of the pairwise hit matrix

$$H_{i,j} = X_i^T X_j. \tag{5.3}$$

These $\{H_{i,j}\}$ matrices will enable us to evaluate the number of pairwise hits between the sequences generated by a candidate group-generator-cyclic-shift pair,

(g_i, s_i) , and any of the $\{(g_j, s_j, g_j^{-1}, s_j^{-1})\}$ quadruples in the look-up table, $\mathcal{T}$. To do so, for each $H_{i,j}$ matrix, we define a complement matrix $\bar{H}_{i,j} \triangleq X_i^T \bar{X}_j$, where $\bar{X}_j$ is the load matrix corresponding to the sequence generated by (g_j^{-1}, s_j^{-1}) . In essence, $\bar{H}_{i,j}$ characterizes the number of pairwise hits between the sequence generated by (g_i, s_i) and that generated by (g_j^{-1}, s_j^{-1}) . By denoting the index of the last assigned RB in the sequence generated by (g_j^{-1}, s_j^{-1}) as $\bar{k}_j$, the number of pairwise hits between the sequences corresponding to the pairs (g_i, s_i) , (g_j, s_j) and (g_j^{-1}, s_j^{-1}) can be expressed as

$$H_{i,j}(k_i, k_j) + \bar{H}_{i,j}(k_i, \bar{k}_j). \quad (5.4)$$

Averaging over all possible load combinations of a system with N MTC devices, the total number of pairwise hits between the sequence corresponding to (g_i, s_i) and those corresponding to each of the quadruples in $\mathcal{T}$ can be expressed as

$$I_i = \frac{1}{2|\mathcal{T}|} \sum_{j=1, j \neq i}^{|\mathcal{T}|} \sum_{(k_i, k_j, \bar{k}_j) \in \mathcal{Y}} H_{i,j}(k_i, k_j) + \bar{H}_{i,j}(k_i, \bar{k}_j),$$

where $i = 1, \dots, |\mathcal{U}|$. (5.5)

where $\mathcal{Y}$ is the set of all the indices $(k_i, k_j, \bar{k}_j)$ such that the sum of their corresponding loads is less than or equal to N .

5.1.3.2 Evaluating the index of the last assigned RB in a sequence

The use of the $\{H_{i,j}\}$ matrices in the evaluation of the number of pairwise hits between two RTs requires invoking the dynamics of the RB assignment process. In this process, the index of the last assigned RB in the assignment sequences of the two RTs might be higher than their actual number of assigned RBs. This is because when an RT detects a hit on assigning a particular RB, it relinquishes this RB and

assigns the following one in its sequence. This process is illustrated by the following example.

Example 2. Let the cyclically-generated RB assignment sequences of the i -th and j -th RTs be $(1, 3, 5, 4, 6, 2, 7)$ and $(3, 5, 4, 7, 6, 1, 2)$, respectively. Let the load of the j -th RT be two MTC devices, which are assigned RBs 3 and 5. Now, suppose that the i -th RT has two incoming devices each requesting an RB. Following its sequence, the i -th RT will assign RBs 1 and 3 to the incoming devices, which will result in a hit at RB 3. Subsequently, it will relinquish this RB and assign the following one in its sequence; i.e., RB 5. Since this RB has also been assigned by the j -th RT, the i -th RT again relinquishes this RB and assigns RB 4 to the second device. $\square$

Careful inspection of this example yields two observations: first, k_i is in general greater than ℓ_i ; e.g., for the i -th RT, $k_i = 4$, whereas $\ell_i = 2$; and second, the relinquishment of a particular RB and the assignment of a subsequent one can result in additional hits and subsequently an increase in k_i .

Unfortunately, there is no direct relation that maps loads to sequence indices. However, these indices can be iteratively evaluated using the pairwise hit matrices in (5.3). In particular, consider the number of pairwise hits in (5.4) and let k_j and $\bar{k}_j$ be fixed. To determine k_i , we provide a function f_i that, at the μ -th iteration, evaluates a tentative index $k^{(\mu)}$ based on $k^{(\mu-1)}$ with the initial index $k^{(0)}$ being equal to the load of the i -th RT, ℓ_i . In particular, at the μ -th iteration, f_i evaluates the number of hit occurrences between the sequences corresponding to the three pairs (g_i, s_i) , (g_j, s_j) and (g_j^{-1}, s_j^{-1}) when $k = k^{(\mu-1)}$. If no hits are detected, then $k_i = k^{(\mu-1)}$; none of the assigned RBs will be relinquished and thus k_i will not increase. On the other hand, if hits are detected, f_i obtains a tentative index, $k^{(\mu)}$, by adding

the number of hit occurrences when $k_i = k^{(\mu-1)}$ to the value of $k^{(0)}$; i.e.,

$$\begin{aligned} k^{(\mu)} &= f_i(k^{(\mu-1)}) \\ &= k^{(0)} + H_{i,j}(k^{(\mu-1)}, k_j) + \bar{H}_{i,j}(k^{(\mu-1)}, \bar{k}_j), \end{aligned} \quad (5.6)$$

where $k^{(0)} = \ell_i$.

Since the assumption in this work is that the maximum number of MTC devices does not exceed N , there exists at least one index, k , at which no additional hits are detected; i.e., $f_i(k) = k$. This index will be equal to k_i , and using it with k_j and $\bar{k}_j$ in (5.4) yields the number of pairwise hits between the sequences corresponding to the pairs (g_i, s_i) , (g_j, s_j) and (g_j^{-1}, s_j^{-1}) . Note that k_i depends on the RT loads, and therefore must be reevaluated for each load distribution.

We conclude this section by noting that the evaluation of k_i in (5.6) is bounded by N iterations, each involving 3 additions. Hence, the computational complexity of evaluating k_i is upper bounded by $3N$ additions.

5.1.4 Computational complexity of the greedy algorithm

To assess the computational complexity of the greedy algorithm, we note that this algorithm relies on evaluating I_i in (5.5) for each of the group generator pairs in $\mathcal{U}$; i.e., $\forall (g_i, g_i^{-1}) \in \mathcal{U}$, $i = 1, \dots, \frac{\phi(N)}{2}$. This evaluation involves only addition operations, but no multiplications. In particular, computing I_i for the pair (g_i, g_i^{-1}) involves: 1) the evaluation of the $H_{i,j}$ and $\bar{H}_{i,j}$ matrices; 2) the index k_i ; and 3) the cardinality of the set $|\mathcal{Y}|$. The number of additions required to evaluate each of the pairwise hit matrices in (5.5) is bounded by N^3 , and the complexity of evaluating k_i for a given load distribution is bounded by $3N$ additions; see previous section for details. Finally, by construction, the cardinality of $\mathcal{Y}$ in (5.5) is equal to the number of ordered partitions of the integer N . This number is given by $|\mathcal{Y}| = \binom{N-1}{2}$ [80].

Using these elements, we can now provide a bound on the complexity of the proposed greedy algorithm.

In essence, the algorithm iteratively selects the group generator in $\mathcal{U}$ that yields the minimum number of pairwise hits with the quadruples in $\mathcal{T}$. We note that when computing the number of hits between a group generator g_i and a given quadruple in $\mathcal{T}$ over all possible load combinations, the pairwise hit matrices need to be evaluated once; these matrices do not depend on the RT loads. In addition, the number of possible load combinations for which k_i must be evaluated is given by $|\mathcal{Y}|$. Hence, the complexity of evaluating the pairwise number of hits between a group generator g_i and a given quadruple in $\mathcal{T}$ over all possible load combinations is bounded by

$$2N^3 + 3N \binom{N-1}{2} + \binom{N-1}{2} = 3.5N^3 - 4N^2 + 1.5N + 1. \quad (5.7)$$

In the left hand side of (5.7), the first term represents the number of additions required for evaluating the pairwise matrices, $H_{i,j}$ and $\bar{H}_{i,j}$. The second term represents the number of additions required to evaluate the index k_i for the $\binom{N-1}{2}$ possible load distributions. Finally, the last term represents the complexity involved in the summation of the number of pairwise hits for the $\binom{N-1}{2}$ load distributions.

Since s_i can take values from 0 to $N-1$, it follows that the complexity of evaluating the performance of a candidate group generator, g_i , for all possible cyclic shifts with respect to a quadruple in $\mathcal{T}$ is bounded by $3.5N^4 - 4N^3 + 1.5N^2 + N$. Hence, the number of additions required for evaluating the performance of g_i against all the quadruples in $\mathcal{T}$ can be bounded by the product of this number and the number of quadruples in $\mathcal{T}$; i.e., $(3.5N^4 - 4N^3 + 1.5N^2 + N) \left(\frac{\phi(N)}{2} - 1 \right)$. Finally, to determine the overall complexity of the greedy algorithm, we note that this algorithm exhausts the group generator pairs in $\mathcal{U}$ in $\frac{\phi(N)}{2} - 1$ iterations, each of which involves

all the entries in $\mathcal{U}$. Since $|\mathcal{U}| \leq \frac{\phi(N)}{2} - 1$, it follows that the overall computational complexity of the algorithm is bounded by

$$(3.5N^4 - 4N^3 + 1.5N^2 + N) \left(\frac{\phi(N)}{2} - 1 \right)^3. \quad (5.8)$$

We note that by definition, for any given integer N , $\phi(N) < N$. Hence, the computational complexity of the greedy algorithm in (5.8) can be bounded by $\mathcal{O}(N^7)$. This is in sharp contrast with the complexity of the exhaustive search in [61], which was shown to be bounded by $\frac{1}{2}M(M-1)N^M \left(N^2(N-1) + \binom{N-1}{M-1} \right)$; i.e., exponential in the number of RTs, M . Hence, in contrast with exhaustive search proposed in [61], the greedy algorithm is readily applicable in MTC systems with potentially large numbers of RTs.

5.2 The graphical greedy algorithm

In the previous section, we developed an efficient greedy algorithm that yields a look-up table of RB assignment sequences. The computational complexity of this algorithm was shown to be polynomial and to depend solely on N , but not on M . For example, using (5.8), the computational complexity of constructing the look-up table for a system with any number of RTs, M , and $N = 100$ RBs can be bounded by 2.5×10^{12} additions, which is manageable by widely available processors. However, for large MTC systems that serve thousands of devices concurrently, the greedy algorithm is not computationally feasible. For instance, for an MTC system with $N = 1000$ RBs, the computational complexity of the greedy algorithm is bounded by 10^{21} additions! For such systems, more efficient sequence selection mechanisms are required.

In this section, we propose an alternate greedy algorithm that significantly re-

duces the computational complexity required for selecting the assignment sequences. The philosophy of this algorithm is similar to that of the one proposed in the previous section, but it depends on a metric that is inspired by the graphical representation of cyclic groups. In particular, this metric is based on the observation that the RBs located at the beginning of each sequence, referred to as the dominating RBs, are the most significant contributors to the number of hit occurrences. This is because these RBs will always be assigned by the RTs to incoming MTC devices even when the system is lightly loaded. Hence, to reduce the number of hit occurrences, the cyclically-generated sequences should be selected such that their dominating RBs do not overlap.

Following analogous steps to those used in the greedy algorithm discussed hereinabove, we let $\mathcal{T}_g$ be the look-up table to be constructed by the graphical greedy algorithm and $\mathcal{U}_g$ be the set containing the group generator pairs that have yet to be selected. In each iteration, the algorithm uses a metric based on the graphical representation of cyclic groups to evaluate the number of pairwise hits between the dominating RBs of the sequences cyclically-generated by each of the group generator pairs in $\mathcal{U}_g$ and those cyclically-generated by the quadruples already selected in $\mathcal{T}_g$. This is in contrast with the greedy algorithm discussed in Section 5.1, which considers all the N RBs when evaluating the number of pairwise hits. The look-up table $\mathcal{T}_g$ is then augmented by the group generator pair and the cyclic shifts that correspond to the sequence resulting in the minimum number of hits between dominating RBs. The graphical representation of cyclic groups and the metric inspired thereby are discussed in more details in the following sections.

5.2.0.1 The graphical representation of cyclic groups [81]

A cyclic group with finite order can be represented graphically by connecting its elements in the same order as they appear in its cyclically-generated sequences. To do so, the group elements are first positioned on the circumference of a circle in the same order of one cyclic sequence and then connected according to the remaining ones. This results in a particular pattern that connects all the group elements for each cyclic sequence. An illustrative example of the graphical representation of the ACG of order 10 is presented in Figure 5.2. In this figure, the group has 2 pairs of group generators, one pair corresponds to the outer circle while the other corresponds to the inner pattern.

Since shifting a sequence cyclically rotates its entries, this shifting will rotate the pattern corresponding to the sequence. This rotation is clockwise when the cyclic shift is negative and anticlockwise when it is positive.

We conclude this section by the following remarks:

Remark 1. *Using the isomorphism property, it can be readily shown that the graphical representation of a cyclic group depends solely on the group order N ; i.e., it is independent of the group operation; cf. Definition 4. Hence, it is sufficient to study the properties of the graphical representation of only the ACG.* $\square$

Remark 2. *It was shown in [81] that the graphical representation of a cyclic group is independent of the cyclically-generated sequence that is selected to construct the outer circle. Hence, the inner patterns of this representation are unique and depend only on the group order.* $\square$

Using the graphical representation of cyclic groups, in the following section, we provide an alternate greedy algorithm that selects the cyclically-generated sequences

based on their patterns.

Figure 5.2: A graphical representation of a cyclic group of order 10. This cyclic group has 2 group generator pairs that are represented by the outer circle and the inner pattern, respectively.

5.2.0.2 *The graphical metric*

Let $\mathcal{T}_g$ be the look-up table to be constructed by the algorithm and let $\mathcal{U}_g$ be the set containing the group generator pairs that have yet to be selected. At first, $\mathcal{T}_g$ is empty and $\mathcal{U}_g$ contains the $\frac{\phi(N)}{2}$ group generator pairs. In the first iteration,

a pair (g, g^{-1}) is arbitrarily selected and removed from $\mathcal{U}_g$. By associating a cyclic shift $s = 0$ with this pair, we obtain the first quadruple (g, g^{-1}, s, s^{-1}) of the look-up table $\mathcal{T}_g$. This quadruple is then used to generate the first pattern in the graphical representation of the cyclic group; i.e., the outer circle. The dominating RBs in these two sequences are then marked. The number of these dominating RBs ranges from 1 to N and depends on the system load. However, since the contribution of an RB to the average number of hits decreases with its position in the sequence, we will consider only the few RBs at the beginning of each sequence to be the dominating RBs.

Following analogous steps to the greedy algorithm discussed hereinabove, in each of the subsequent $\frac{\phi(N)}{2} - 1$ iterations, the algorithm examines the group generator pairs in $\mathcal{U}_g$ for all possible cyclic shifts. However, unlike the greedy algorithm which considers the hits between the N RBs over all possible load combinations, the graphical algorithm considers only the hits between dominant RBs in the patterns corresponding to the pairs in $\mathcal{U}_g$ and the quadruples in $\mathcal{T}_g$. The look-up table $\mathcal{T}_g$ is then augmented by the quadruple (g, g^{-1}, s, s^{-1}) that yields the minimum hits between the dominant RBs. The group generators of this quadruple are then removed from $\mathcal{U}_g$ and the algorithm iterates until exhaustion of all pairs in $\mathcal{U}_g$. As an example of the graphical metric, consider the ACG, $\mathbb{Z}_{12}$, and consider the dominant RBs to be the first two in each sequence. This cyclic group has two pairs of group generators and its cyclically-generated sequences are as follows:

$(g_1, s_1) = (1, 0)$	1 2 3 4 5 6 7 8 9 10 11 0
$(g_1^{-1}, s_1^{-1}) = (11, 11)$	0 11 10 9 8 7 6 5 4 3 2 1
$(g_2, s_2) = (5, 0)$	5 10 3 8 1 6 11 4 9 2 7 0
$(g_2^{-1}, s_2^{-1}) = (7, 11)$	0 7 2 9 4 11 6 1 8 3 10 5

The corresponding patterns of these two pairs and their dominant RBs are depicted in Figure 5.3a. The outer circle corresponds to the group generator pair in $\mathcal{T}_g$, and the inner pattern corresponds to the examined pair in $\mathcal{U}_g$. The dominant RBs of the pair corresponding to the outer circle are marked with circles and the ones of the pair corresponding to the inner pattern are marked with squares. To avoid the hit at RB 0, the inner pattern is rotated anticlockwise by a cyclic shift $s = 2$. The rotated pattern is depicted in Figure 5.3b. The look-up table is then augmented by the quadruple $(g, g^{-1}, s, s^{-1}) = (5, 7, 2, 9)$, which corresponds to the rotated inner pattern.

Figure 5.3: Rotation of the inner pattern of $\mathbb{Z}_{12}$ to minimize the number of hits between the dominant resource blocks (the first element of the inner pattern is marked by an arrow for ease of exposition).

Finally, we note that the quadruples in the look-up table yielded by the graphical greedy algorithm depends solely on the number of pairwise hits, but not on the number of RTs in the system. Hence, active RTs do not need to update their sequences when new ones join the system. This is in contrast with the exhaustive search scheme, which requires each active RT to update its RB assignment sequence when a new RT enters the system thus resulting in a significant signalling overhead that might not be feasible for MTC systems with large numbers of RTs and MTC devices.

5.2.1 Computational complexity of the graphical greedy algorithm

The computational cost of the graphical greedy algorithm is mainly due to the evaluation of the number of hits between dominant RBs. This cost involves only addition operations and, similar to the algorithm in Section 5.1, is polynomial in N . However, the order of the computational complexity of the graphical greedy algorithm is significantly lower than that of the greedy algorithm. To see this, in this section we provide a bound on the complexity of the graphical greedy algorithm.

Denoting the number of dominant RBs by D , it can be seen that the number of required additions for evaluating the number of hits between dominant RBs of a pattern corresponding to an examined pair from $\mathcal{U}_g$ and all quadruples in $\mathcal{T}_g$ is bounded by $4D$. Since the cyclic shift can take values from 0 to $N - 1$, it follows that the complexity of evaluating the number of hits between the dominant RBs of the examined pattern over all possible cyclic shifts can be bounded by $4ND$. Finally, since $D \leq N$ and in each iteration $|\mathcal{U}_g| \leq \frac{\phi(N)}{2} - 1$, it follows that the computational complexity of the graphical greedy algorithm is bounded by

$$4ND \left(\frac{\phi(N)}{2} - 1 \right)^2 \leq N^4, \quad (5.9)$$

which shows that the graphical greedy algorithm offers a computational saving of N^3

over the greedy algorithm of Section 5.1; cf. Section 5.1.4. This is because, when selecting the RB assignment sequences, the graphical greedy algorithm considers only the number of hits between dominant RBs, which is independent of the system load. A numerical comparison between the complexity bounds on the greedy algorithms and the exhaustive search is provided in Table 5.1:

Search method	Exhaustive search	Greedy algorithm	Graphical greedy
$(M, N) = (6, 100)$	10^{21}	10^{14}	10^8
$(M, N) = (10, 100)$	10^{33}	10^{14}	10^8
$(M, N) = (10, 1000)$	10^{53}	10^{21}	10^{12}

Table 5.1: A comparative numerical example of the complexities of the greedy algorithms and the exhaustive search.

From Table 5.1, it can be seen that the computational complexity of exhaustive search is exponential in the number of RTs; cf. Section 5.1.4. Hence, it is computationally prohibitive even for MTC systems with small numbers of RTs; e.g., exhaustive search for a system with $M = 10$ and $N = 1000$ requires 10^{53} additions. In contrast with exhaustive search, the computational complexities of both greedy algorithms are polynomial and independent of the number of RTs. Hence, they are applicable in MTC systems, which comprise hundreds of RTs. The computational complexities of both greedy algorithms and exhaustive search are presented in Figure 5.4.

We conclude this section by noting that the proposed greedy algorithms offer a significant reduction in the computational complexity of the sequences selection process for autonomous RB assignments. This reduction is mainly achieved by it-

Figure 5.4: Comparison between the computational complexities of the proposed greedy algorithms and exhaustive search for systems with $M = 6$ and $M = 10$ relaying terminals.

eratively selecting the group generators and cyclic shifts that minimize the number of hit occurrences with respect to the ones already selected in previous iterations. This is unlike the exhaustive search technique which jointly optimizes the performance of all group generators and cyclic shifts to achieve the minimum number of hit occurrences. Despite the advantages of the proposed greedy algorithms, their computational savings are achieved at the expense of a suboptimal selection of RB assignment sequences. In other words, the performance of the sequences selected by the greedy algorithms will be upper bounded by those selected by exhaustive search.

In the following section, we will use numerical examples to evaluate the performance gap between the two sequences selection approaches, i.e., the greedy and exhaustive based approaches.

5.3 Numerical examples

In this section, we compare, in the first four examples, the performance of the RB assignment sequences obtained by the two greedy algorithms developed herein, the uniformly-distributed random assignment scheme developed in [44], and the exhaustive search developed in [61] for different relative loads $\frac{L}{N}$; i.e., different ratios of the currently assigned RBs, L , to the total number of RBs in the system, N . For the scheme in [61] and the ones proposed herein, the cyclic sequences are generated using the MCG structure of Section 5.1.1 with $N = P - 1$, where P is a prime number; cf. [61]. In all numerical examples, the MTC devices will select their assisting RTs from the pool of RTs that are capable of and willing to assist communication irrespective of their status; i.e., whether having its own data to transmit or not.

To be able to compare the performance of the greedy algorithms developed herein with that of the exhaustive search developed in [61], in the first two examples, we will restrict attention to the case of $M = 3$ RTs; exhaustive search is computationally prohibitive for $M > 3$. In these examples the performance is measured by the average number of hits over all possible load combinations; i.e., over all ordered triples $(\ell_1, \ell_2, \ell_3) \in \mathcal{L} \triangleq \{(\ell_1, \ell_2, \ell_3) | \sum_{i=1}^3 \ell_i = L, \ell_i \geq 0\}$, where ℓ_1, ℓ_2 and ℓ_3 are the load levels of RTs 1, 2 and 3, respectively. In these examples, the index of the last assigned RB by each of the 3 RTs, (k_1, k_2, k_3) , is evaluated for each load distribution $(\ell_1, \ell_2, \ell_3) \in \mathcal{L}$; cf. Section 5.1.3.2.

In the following two examples of this section, we will increase the number of RTs and RBs and compare the performance of the graphical greedy algorithm with the random assignment of [44]. In both examples, we will consider only 1000 MTC load distributions that are uniformly distributed over the M RTs. This is because the number of load combinations increases significantly with N and M ; e.g., for $N = 500$ and $M = 25$, this number is equal to $\binom{499}{24}$.

Example 3. In this example, we compare the performance of the cyclically-generated sequences obtained by the greedy algorithm developed in Section 5.1, the graphical greedy algorithm of Section 5.2, and the exhaustive search in [61] when the number of RBs $N = 16$ and $N = 40$. For both cases, the number of dominant RBs of the graphical greedy algorithm is set to $D = 3$. The performance of these algorithms is depicted in Figure 5.5.

From Figure 5.5, it can be seen that the cyclically-generated sequences obtained by exhaustive search and the greedy algorithms outperform randomly-generated ones. For example, for a system with $N = 40$ RBs at a relative load, $\frac{L}{N} = 90\%$, the average number of hits resulting from RB assignments based on exhaustive search [61], greedy algorithm of Section 5.1, graphical greedy algorithm of Section 5.2, and random sequences [44] are 4.4, 6.6, 7.4 and 9.1, respectively. This figure also shows that the greedy-based sequences provide comparable performance to the upper bound yielded by exhaustive search. The reason that exhaustive search constitutes an upper bound on the performance follows from the fact that it jointly optimizes the performance of the cyclically-generated sequences of all the M RTs over all possible group generators and cyclic shifts. In contrast, the greedy algorithms sequentially optimize the cyclically-generated sequences. In particular, these algorithms consider only the previously selected sequences when optimizing current ones. Despite their poten-

Figure 5.5: Comparison between randomly-generated assignment sequences and the cyclically-generated sequences obtained by exhaustive search and the greedy algorithms.

tial suboptimality, the greedy algorithms possess significantly lower computational complexities when compared with exhaustive search. This renders them suitable for systems with practical numbers of RTs and MTC devices; for such systems exhaustive search is computationally prohibitive, cf. Table 5.1. $\square$

Example 4. In this example we consider a setup similar to the one in the previous example, but with $N = 16$ and $N = 126$. For both cases, the number of dominant RBs of the graphical greedy algorithm is set to $D = 3$. In this example, the RTs apply the hit identification and avoidance (HIA) technique developed in [61]. In this technique, each RT is assumed to know the assignment sequences of the neighboring RTs and to be able to identify the RT with which it collided once a hit occurs. Using

this information, the RTs update their assignment sequences to avoid the RBs that are already assigned by neighboring RTs. Note that the HIA technique cannot be applied to the sequences that are randomly-generated because such sequences do not possess a specific structure. In Figure 5.6, we depict the performance improvement when the HIA technique is used with the MCG structured sequences generated by exhaustive search, the greedy algorithm in Section 5.1 and the graphical greedy algorithm in Section 5.2. For comparison, the performance of the randomly-generated sequences is also shown.

Figure 5.6: Comparison between randomly-generated assignment sequences and the cyclically-generated sequences obtained by exhaustive search and the greedy algorithms with the hit identification and avoidance technique.

From this figure, it can be seen that using the HIA algorithm significantly reduces the average number of hit occurrences. For example, for the case of $N = 16$ RBs,

the usage of the HIA reduces the average number of hits of the greedy algorithm from 2.3 to 1.5 when $\frac{L}{N} = 100\%$. The reason behind this reduction is that HIA enables the RTs to avoid future hits. In particular, when the HIA is implemented, the RTs are able to estimate the load levels of their neighbors based on their previous hits and subsequently avoid RB assignments that would result in future hits. Note that from this figure, it can be seen that the HIA also reduces the performance gap between sequences generated by exhaustive search and those generated by the greedy algorithms. $\square$

In the previous examples, we provided performance results for MTC systems with relatively small numbers of RTs and RBs. In the following two examples, systems with larger numbers of RTs and RBs will be considered. In these examples, the performance of the cyclically-generated sequences obtained by the graphical greedy algorithm will be compared only with the randomly-generated sequences [44] since the exhaustive search scheme in [61] and the greedy algorithm of Section 5.1 are computationally expensive in this case.

Example 5. In this example, we compare the performance of the RB assignment sequences generated by the graphical greedy algorithm of Section 5.2 and the randomly-generated assignment sequences [44], when the number of RBs $N = 520$ and $N = 750$. In both cases, the number of RTs is $M = 15$ and the number of dominant RBs is set to $D = 25$. The performance of the graphical greedy algorithm with and without the implementation of the HIA technique is compared with that of the random assignment scheme over 1000 load distributions of MTC devices for each relative load $\frac{L}{N}$. Note that the exhaustive search of [61] and the greedy algorithm of Section 5.1 are not considered in this example because of their large computational complexity. For example, the computational complexity of exhaustive search when $N = 520$ and

$M = 15$ is in the order of 10^{69} , whereas that of the greedy algorithm is in the order of 10^{17} .

Figure 5.7: Comparison between the random and the graphical greedy-based resource block assignment sequences with and without the hit identification and avoidance technique for $N = 520$ and $N = 750$ resource blocks.

From Figure 5.7, it can be seen that for both the $N = 520$ and $N = 750$ scenarios, the sequences that are obtained by the graphical greedy algorithm outperform those that are randomly-generated. This performance advantage increases significantly with the implementation of the HIA technique. For example, for a system with $N = 520$ RBs at a relative load, $\frac{L}{N} = 90\%$, the average number of hits resulting from using the randomly-generated sequences and that resulting from using the sequences obtained by the graphical greedy algorithm without HIA are 623 and 504, respectively. In contrast, the corresponding number of hits resulting from using the

ones obtained by the graphical greedy algorithm with HIA is only 194. In addition, it can be seen from the figure that the performance advantage of using the HIA algorithm increases with N at high loading scenarios. For example, the performance of the system with $N = 750$ and HIA was found to be superior to that of the system with $N = 520$, but without HIA, when $\frac{L}{N} \geq 80\%$. Similar to the instance examined in Example 4, this phenomenon results from the implementation of the HIA algorithm by the RTs, which enables them to estimate the load levels of their neighbors and accordingly avoid future hits. This performance advantage is more pronounced at highly loaded scenarios with large numbers of RBs since the accuracy of estimating the load levels of the neighbors of each RT improves with the increasing number of hits. In other words, although high loads and larger numbers of RBs usually result in more hits, they can have a positive impact on the performance when the HIA is implemented. Finally, we note that the increase in the average number of hits with respect to the previous examples is mainly due to the larger number of RTs and RBs and the consideration of only 1000 load distributions of MTC devices. $\square$

Example 6. In this example, we compare the performance of the cyclically-generated RB assignment sequences obtained by the graphical greedy algorithm and the randomly-generated assignment sequences [44] when the number of RTs $M = 30$ and $M = 60$. In both cases, the number of RBs is $N = 750$ and the number of dominant RBs is set to $D = 25$. Similar to the previous example, the performance of the graphical greedy algorithm is evaluated with and without the implementation of HIA.

From Figure 5.8, it can be seen that the sequences obtained by the graphical greedy algorithm outperforms those that are randomly-generated for $M = 30$ and $M = 60$. For example, for a system with $M = 30$ RTs and a relative load of $\frac{L}{N} = 90\%$ the average number of hits resulting from using the randomly-generated sequences

Figure 5.8: Comparison between the random and the graphical greedy-based resource block assignment sequences with and without the hit identification and avoidance technique for $M = 30$ and $M = 60$ relaying terminals.

and that resulting from using the sequences obtained by the graphical greedy algorithm without HIA are 977 and 552, respectively. In contrast, the corresponding number of hits resulting from using the sequences obtained by the graphical greedy algorithm with HIA is 356. In other words, the greedy algorithm without HIA has a 40% gain over the randomly-generated assignment sequences. However, this gain decreases as the number of RTs increases. For example, this gain reduces to 30% at the same relative load when $M = 60$. This is because the number of utilized assignment sequences increases with M , which results in the selection of assignment sequences with lower performance. Finally, it can be seen from the figure that at a relative load of $\frac{L}{N} = 100\%$, the system with $M = 60$ and HIA is superior to that with

$M = 30$, but without HIA. The reason behind this phenomenon is the hit avoidance feature provided by the HIA as described in the previous example. $\square$

Example 7. In this example, we evaluate the aggregate data rate when the proposed sequences are used in a system with $M = 3$ RTs and $N = 40$ RBs. Each RT assigns one RB to each incoming MTC device during a scheduling interval of 1 ms. In other words, in each scheduling interval, 3 RBs corresponding to $M=3$ RTs are assigned. Once RT i assigns an RB to an incoming MTC device, it increases the number of its assigned RBs, N_i , by one. At this point, two possibilities can occur, either successful assignment or collision. In case of a successful assignment, the assigned RB is retained and the RT proceeds during the next scheduling interval to assign the following RB in its sequence to the incoming MTC device. In case of collision, the RT which was first to assign the RB retains it, whereas the other conflicting RT releases the RB and reduces its N_i by one at the end of its scheduling interval. The impact of a collision will be reflected on the interference observed by the MTC devices. In particular, a collision on a given RB will result in an interference between the transmissions of their respective MTC devices for the duration of one scheduling interval. In this example, the system is initially unloaded and the number of MTC devices is increased in a round Robin fashion until two RTs are loaded with 13 MTC devices and the third RT is loaded with 14 MTC devices. In that case, the system is fully loaded; cf. Figure 5.9 for an illustration of the considered RT loads.

The channel gains between the RTs and the MTC devices are generated using the line-of-sight Hotspot pathloss model [82] with frequency-flat Rayleigh fading and standard Gaussian noise spectral density of -174 dBm/Hz. The RB bandwidth, and the receiver noise figure are set to 180 KHz and 5 dB, respectively [82]. The transmit power, P_t , of the MTC devices is set to 0 dBm and the locations of the MTC devices

Figure 5.9: An example of an association between relaying terminals and machine-type communication devices.

and the RTs are uniformly distributed within the cell boundaries. For a practical dense deployment scenario, the MTC devices are assumed to be less than 300 m away from their assisting RTs and the maximum inter-RT distance is set to 100 m. The RTs implement the HIA algorithm of [61] and the performance is averaged over 10000 channel realizations. Using the aforementioned parameters, the aggregate data rate, R , at a given scheduling interval can be calculated using:

$$R = BW \sum_{i=1}^M \sum_{j=1}^{N_i} \log_2 \left(1 + \frac{|H_{q,i,B_j}|^2 P_t}{P_n + \sum_{m=1, m \neq q}^Q |H_{m,i,B_j}|^2 P_t} \right), \quad (5.10)$$

where BW is the bandwidth, P_n is the noise power, B_j is the j -th RB assigned by

RT i , Q is the total number of MTC devices in the system, and H_{q,i,B_j} is the channel coefficient between RT i and MTC device q which is assigned B_j and is set to zero if q is not assigned B_j . Note that, the selected RB, B_j , by the i -th RT will depend on its sequence.

Figure 5.10: Aggregate data rates of random, greedy-based and exhaustive search-based resource block assignment sequences.

As shown in Figure 5.10, the sequences obtained by the greedy algorithms significantly outperform the random RB assignments and provide comparable performance to those obtained by exhaustive search. For example, at the 17th scheduling interval, the greedy-based sequences achieve an aggregate data rate of 10.3×10^7 bps, which is only 10% lower than that of its exhaustive search counterpart. In contrast, random RB assignments achieve a rate of only 6.86×10^7 bps. Figure 5.10 also illustrates the

effect of hit occurrences on the aggregate data rate. This effect is reflected in the jagged behavior of the aggregate data rate curves. In particular, during a scheduling interval, the three RTs perform RB assignments to their respective MTC devices. At this point, three possibilities can occur: 1) successful RB assignments with no collisions; 2) collisions between concurrent RB assignments; or 3) collisions between concurrent RB assignments and previous ones. In case one, the aggregate rate increases by the sum of the rates of the new transmissions. In case two, the aggregate rate will remain flat when all the concurrent RB assignments collide since all transmissions are unsuccessful, whereas when only two assignments are colliding, the rate will slightly increase since a successful transmission will be achieved on one RB. In case three, the aggregate rate can either: 1) significantly decrease when all the concurrent assignments collide; 2) slightly decrease when two assignments collide; or 3) slightly increase when one assignment collide. Hence, from these observations, it can be concluded that the number of hits plays a key role in constraining the aggregate data rate. $\square$

Example 8. In this example, we evaluate the number of MTC devices with successful RB assignments vs the number of RB assignment attempts. The setup of this simulation is similar to that of the previous one. However, unlike the previous example, the total system load is not uniformly distributed across the 3 RTs and the performance results are averaged over 10000 load distributions across the 3 RTs.

Consistent with our observations in the previous example, Figure 5.11 shows that the greedy-based sequences significantly outperform their random counterparts and provide comparable performance to those obtained by exhaustive search. For example, for two RB assignment attempts, the average numbers of MTC devices with successful assignments are 38.94, 38.64, 38.44, and 34.23 for RB assignments based

Figure 5.11: CDF of the number of resource block assignment attempts needed for successful assignment.

on exhaustive search, greedy, graphical greedy, and random sequences, respectively. From Figure 5.11, it can be seen that using the sequences selected by exhaustive search and greedy algorithms, the 99-th percentile of all MTC devices are successfully assigned with at most three RB assignment attempts, whereas in the case of random assignments the 99-th percentile is reached with eight attempts. Using the results reported in Figure 5.11, one can evaluate the power wastage incurred by collisions and the data transmission delay experienced by the MTC devices; i.e., the duration between the first RB assignment and the successful one. This can be done by invoking the scheduling interval and P_t .

Denoting the transmission delay by d and the number of RB assignment attempts by n , it can be seen that $d = \bar{s}(n - 1)$, where $\bar{s}$ is the scheduling interval duration. For

example, a device with 3 assignment attempts will incur a delay of 2 ms.

To evaluate the power wastage due to collisions, denoted by P_w , one can use $P_w = P_t(n - 1)$. Using Figure 5.11, it can be concluded that 96% and 85.5% of the MTC devices incurred less than 1 mW power wastage and 1 ms delay when using greedy-based and random sequences, respectively. $\square$

We conclude this section by noting that the proposed work offers a flexible trade-off between sequence generation complexity and performance by focusing attention on cyclically-generated sequences due to their favorable features; cf. Section 3.2. However, the underlying cyclic structure limits the number of sequences of length N that are available for RB assignments to $N\phi(N)$, which is much less than the total number of sequences, $N!$. Such a limitation might have an impact on performance. To overcome this limitation and potentially improve performance, it may be possible to enrich the set of available sequences by seeking number theoretic techniques for combining multiple cyclic ones. One example of such techniques, namely the CRT, will be discussed in the following section.

Chapter 6

Chinese Remainder Theorem-Based Sequence Design for Autonomous Resource Block Assignments in Relay-Assisted MTC Systems

In this chapter, we develop a CRT-based scheme for generating RB assignment sequences for relay-assisted MTC systems. Unlike the cyclic group-based scheme introduced in the previous section, the developed CRT-based scheme does not rely on a single group generator for sequence generation. In particular, this scheme offers multi-generator sequences, whereby a CRT-based sequence is obtained by combining cyclically-generated sequences belonging to two or more cyclic group structures. Hence, the proposed CRT-based scheme yields an enriched set of sequences that can be used for RB assignments. The combining process underlying the proposed scheme is based on: 1) constructing a set of linear congruences from a set of cyclically-generated sequences; and 2) solving the congruences set using the CRT to obtain an RB assignment sequence. Finally, in this chapter, we also develop an efficient greedy-based sequences selection algorithm that enables the application of CRT-based sequences in systems with large numbers of RTs and MTC devices.

6.1 A CRT-based RB assignment scheme

The philosophy of the proposed work is to distribute the available RBs among the RTs in the absence of coordination. In other words, the objective is to avoid the unintentional overutilization of particular RBs while others are being underutilized. To achieve this objective, in the proposed scheme, neighboring RTs assign the available RBs based on a set of prescribed CRT-based sequences. The sequences within this set are jointly designed such that their overlap is minimal. This joint design yields an implicit coordination between neighboring RTs when assigning their RBs. Hence, it reduces the number of RB assignment conflicts with minimal signalling overhead. This sequence structured RB assignment philosophy implies that each RT needs to: 1) associate each sequence element to a specific RB; 2) use a signalling protocol to convey the assigned RBs to the assisted MTC devices; and 3) locally generate a CRT-based RB assignment sequence that spans all the available RBs in the system. Next, we will discuss the details of these three steps.

6.1.1 Association between sequence elements and RBs

In the proposed scheme, each RB is assigned a label between 0 and $N - 1$, where N is the number of RBs. In other words, a one-to-one mapping is selected between the set of RBs and $\{0, \dots, N - 1\}$. This mapping is arbitrary and is announced to all BSs and RTs prior to system startup (any one-to-one mapping yields on average the same performance due to the absence of CSIs). This arbitrary mapping implies that consecutive sequence elements do not necessarily map to consecutive RBs; i.e., the RTs can assign non-consecutive RBs to a particular MTC device. For example, consider a 2-RT scenario, wherein the selected mapping function associates RBs A ,

B , and C to the sequence elements 0, 2, and 1, respectively and RTs 1 and 2 are assigned the sequences $\{0, 2, 1\}$ and $\{0, 1, 2\}$, respectively. Now, assuming that each RT assists two MTC devices, the resulting RB assignments for this scenario are depicted in Figure 6.1.

Figure 6.1: A mapping example between sequence elements and resource blocks.

From Figure 6.1, it can be seen that any mapping function would result in a hit between the two RTs. However, in the following sections, we will show how the CRT-based scheme can reduce the number of these hits. Finally, we note that the non-consecutive RB assignments approach is in line with LTE Release 10 and is chosen because it offers a greater flexibility in RB assignments and an increase in the potential for a frequency selective scheduling gain [83].

6.1.2 Signalling protocol for RB assignments

When an RT has an incoming MTC device, it uses a dedicated control channel to assign the next available RBs in its sequence to the MTC device. The number of these assigned RBs will depend on the category of the device; cf. Chapter 3. After receiving its RB assignment, the MTC device sends an acknowledgement to the RT. At this point, two cases can occur: 1) a response is received by the RT (either an acknowledgement or a negative acknowledgement); or 2) no response is received. If an acknowledgement is received, the RT identifies that this assignment is successful and starts an activity timer for the corresponding MTC device. This timer-based

approach is similar to that of LTE in which timers are used to monitor a user's inactivity duration [84]. If the activity timer expires before an RT receives data on an assigned RB, it identifies the RB as abandoned and infers that the corresponding MTC device is no longer in its vicinity. In this case, a gap will exist in the RT's RB assignment sequence. In other words, one or more of the previously assigned entries of the RB assignment sequence are currently not utilized. To use these entries, the RT can either allocate them to its first incoming MTC device, or it can shift the RB assignments of all the following MTC devices such that no gaps are present in the sequence. On the other hand, if a negative acknowledgement is received, the RT infers that a hit has occurred on this RB and assigns the following RB in the sequence to the MTC device. If no response is received, the RT infers that the RB assignment has failed and that this device is no longer in its vicinity. The RT then reassigns this RB to the following incoming MTC device. Finally, we note that, although this work focuses on terminals acting as RTs, it is readily extensible to networks with small cell deployments. In particular, each small cell can be considered as an RT with an RB assignment sequence and accordingly serve the MTC devices within its coverage area. However, unlike mobile RTs, small cells are fixed and thus can have a wired backhaul connection that can be exploited to further improve the performance by allowing nearby small cells to partially coordinate their RB assignments. For instance, small cells can obtain the load levels of their neighbors over the backhaul connection and accordingly avoid assigning the RBs that would result in hits.

Next, we will discuss the details of the CRT-based sequence generation.

6.1.3 CRT-based sequence generation for RB assignments

In this section, we will begin by noting that any cyclic group can be mapped to the additive cyclic group (ACG) structure [65]. In particular, there exists at least one one-to-one and onto function that maps all cyclic group structures to the additive one; cf. Chapter 4. Hence, without loss of generality, we will focus attention to the cyclically-generated sequences by the ACG. Using the ACG structure, an RT can locally generate its RB assignment sequences using only a group generator as follows:

$$\{kg_i \pmod{N}\}_{k=1}^N, \quad (6.1)$$

where N is the sequence length and g_i is the i^{th} group generator of the ACG $\mathbb{Z}_N$; cf. Chapter 2. To further enrich the set of sequences for RB assignments, cyclic shifts were applied on each of the N sequences of the ACG structure [61]. In particular, it was shown in [61] that cyclically-shifted sequences do not coincide with one another, thereby creating an additional degree of freedom for sequence generation; cf. Chapter 2. This implies that for a group $\mathbb{Z}_N$, the number of available sequences is $N\phi(N)$. Each of these sequences can be obtained by adding a cyclic shift, s_j , to (6.1) as follows:

$$\{(k + s_j)g_i \pmod{N}\}_{k=1}^N, j = 1, \dots, N. \quad (6.2)$$

Despite the simplicity of cyclic sequences generation, their limited numbers impose a serious restriction on the degrees of freedom available for avoiding collisions when used for autonomous RB assignments. This restriction becomes more pronounced in prospected autonomous systems with large numbers of RTs and MTC devices. To address this concern, we use the CRT to generate a superset of sequences that contains the cyclic ones and their shifted versions as a proper subset. In particular, the proposed CRT-based scheme is used to combine two or more cyclically-

generated sequences into longer ones, thereby offering additional degrees of freedom for sequence generation. This increases the cardinality of the sequences set available for RB assignments, while retaining the ease of generation property. In the proposed CRT-based scheme, each RT uses two or more generators and cyclic shifts of seed sequences, which are either selected autonomously by the RT or provided by the BS. The elements of these sequences are used to form a linear set of independent congruences. Such congruences have a unique solution that can be obtained by using the CRT. This solution is taken to be the element of the combined sequence; cf. Figure 6.2.

Figure 6.2: Sequence generation using the Chinese remainder theorem.

In this figure, c_i is the i -th element of the sequence generated by the CRT, m is the number of seed sequences, and $f(\cdot)$ is the CRT function that maps the elements of the seed sequences, $a_i^1, \dots, a_i^m$, to c_i . This mapping function represents the solution obtained by the CRT, cf. (2.12), for the set of congruences formed by the elements $a_i^1, \dots, a_i^m$. By repeatedly solving the set of congruences for all elements of the seed sequences, we obtain a sequence that is unique modulo the multiplication of the lengths of its seed sequences. We will illustrate the proposed methodology with a simple example. Consider an RT, which is assigned the group generators (g_1, g_2) of the cyclic groups $\mathbb{Z}_{N_1}$ and $\mathbb{Z}_{N_2}$. For simplicity we will assume zero cyclic shift in

this example. To obtain its RB assignment sequence, the RT will use the CRT to combine the seed sequences of lengths N_1 and N_2 , respectively. This sequence will be parsed into N_1 sections each of length N_2 elements. In particular, the RT solves the following set of congruences

$$c_k^i \equiv ig_1 \pmod{N_1}, \quad (6.3)$$

$$c_k^i \equiv kg_2 \pmod{N_2}, \quad (6.4)$$

with $i = 1$ and $k = 1, \dots, N_2$ to obtain the first section of N_2 elements of its RB assignment sequence; i.e., $(c_1^1, \dots, c_{N_2}^1)$. The following sequence section of N_2 elements is then obtained by the RT by setting $i = 2$ in (6.3). This procedure is then repeated until all the N_1 sections of the combined sequence are obtained; i.e., until $i = N_1$. To elaborate on the details of the proposed CRT-based sequence generation scheme, we will consider an example that combines two seed sequences of lengths $N_1 = 5$ and $N_2 = 6$. Assume that an RT is assigned two cyclic shifts, s_1 and s_2 , and two group generators, g_1 and g_2 , of two distinct ACGs, e.g., $\mathbb{Z}_5$ and $\mathbb{Z}_6$. For instance, for $\mathbb{Z}_5$ we choose $(g_1 = 4, s_1 = 0)$, whereas for $\mathbb{Z}_6$ we choose $(g_2 = 5, s_2 = 0)$. Using (6.2), the RT can generate its seed cyclic sequences from the assigned pairs. Using this expression, the RT generates two seed cyclic sequences, viz., $\mathbb{S}_1 = (4, 3, 2, 1, 0)$ and $\mathbb{S}_2 = (5, 4, 3, 2, 1, 0)$. Now, to obtain the first sequence element of the CRT-based sequence, the RT uses the lengths of the two seed cyclic sequences, i.e., 5 and 6, and their first elements, i.e., 4 and 5, to construct a set of joint linear congruences $c_1 \equiv 4 \pmod{5}$ and $c_1 \equiv 5 \pmod{6}$, where c_1 is the joint solution of the congruences. The RT then uses a systematic approach based on the CRT and the Euclidian algorithm to obtain the value of c_1 ; cf. Chapter 2. This value then becomes the first element of the CRT-based RB assignment sequence of the RT under consideration. For the

following elements, the RT fixes the first element of $\mathbb{S}_1$ and solves the constructed congruences for each of the remaining five elements of $\mathbb{S}_2$ until exhaustion. Next, this process is repeated with the following element of $\mathbb{S}_1$ until exhaustion of all its elements. Once all congruences are solved, the RT obtains a sequence of length $N_1 N_2 = 30$, and the resulting sequence becomes (29, 4, 9, 14, 19, 24, 23, 28, 3, 8, 13, 18, 17, 22, 27, 2, 7, 12, 11, 16, 21, 26, 1, 6, 5, 10, 15, 20, 25, 0).

Unlike the cyclic-based RB assignments in MTC systems, the proposed CRT-based scheme offers additional degrees of freedom for sequence generation. In particular, the proposed CRT-based scheme enables combining one or more seed cyclic sequences, thereby enriching the set of available sequences for RB assignments. The resulting combined sequences will depend on the number of combined seed sequences as well as the order in which they are combined. In other words, each ordered set of seed sequences will yield a unique set of sequences that can be used for RB assignments. Subsequently, the additional degrees of freedom increase the cardinality of the set of sequences available for RB assignments, which offers an opportunity for performance enhancement as will be shown by simulations in Section 6.2. To elaborate on the benefits of the proposed CRT-based scheme, we will discuss the cardinality of the CRT-based sequences and compare it to that of the cyclic ones discussed in [61] in the following section.

6.1.4 The cardinality of the CRT-based sequences set

In this section, we will evaluate the number of RB assignment sequences of length N that can be generated by the proposed CRT-based sequence generation scheme. This number will depend on the following four features of the proposed scheme:

1. *Length of component seed sequences:* The lengths of component seed sequences

determine the lengths of the combined sequences by the CRT. In other words, combining m seed sequences of lengths $N_1, \dots, N_m$ yields a sequence of length $N = \prod_{i=1}^m N_i$.

2. *Number of cyclically-generated sequences:* For a sequence of length N_i , the number of available cyclically-generated sequences is given by $N_i\phi(N_i)$; cf. Chapter 2.
3. *Coprimeness of $N_i, \dots, N_m$:* This condition is required for the congruences described in (2.7) to have a unique solution.
4. *Combining order of the seed sequences:* Lemma 2 implies that the CRT-based sequences directly depend on the order in which the seed cyclic sequences are combined. In other words, the sequence obtained by combining (S_1, S_2) is different from the one obtained by combining (S_2, S_1) , where S_1 and S_2 are two seed cyclic sequences.

Using these features, we will compute the cardinality of the set of length N sequences that can be generated by the proposed CRT-based scheme. In particular, these features imply that this cardinality will depend on the number of ordered sets (feature 4) of the pairwise coprime (feature 3) factors of N (feature 1) and their lengths (feature 2). Next, we will investigate the implications of each of these four features on this number.

We will begin by using the number of pairwise coprime factors of N to evaluate all the possible lengths of the seed cyclic sequences. For a given N , the number of pairwise coprime factors can be obtained by considering all possible combinations of its unique prime factors. In particular, let $N = \prod_{i=1}^{q(N)} p_i^{t_i}$, where $q(N)$ is the

number of unique prime factors of N , p_i is the i -th prime factor of N and t_i is its repetitions. Note that, each prime factor p_i with t_i repetitions will be considered as a single element to ensure the pairwise coprimeness of the obtained factors of N . Now, to obtain the m coprime factors of N , one will need to consider all possible combinations of the prime factors into m distinct numbers. For example, consider a sequence of length $N = p_1 p_2^2 p_3$. All the ordered sets of the coprime factors of this length can be represented by all the possible combinations of its prime factors as follows $\{(p_1, p_2^2, p_3), (p_1, p_3, p_2^2), (p_2^2, p_1, p_3), (p_2^2, p_3, p_1), (p_3, p_2^2, p_1), (p_3, p_1, p_2^2), (p_1, p_2^2 p_3), (p_2^2 p_3, p_1), (p_3, p_2^2 p_1), (p_2^2 p_1, p_3), (p_2^2, p_3 p_1), (p_3 p_1, p_2^2), p_1 p_2^2 p_3\}$. The number of these combinations is equivalent to the number of ways in which the $q(N)$ prime factors can be partitioned into m non-empty subsets. The number of possible partitions for the $q(N)$ prime factors into non empty subsets can be evaluated from the Stirling number of the second kind, which is defined as follows:

Definition 5 (Stirling number of the second kind [85]). *The number of ways to partition a set of n objects into k non-empty subsets is given by the Stirling number of the second kind and is denoted by $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$, where*

$$\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\} = \frac{1}{k!} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} j^n. \quad (6.5)$$

□

The expression in (6.5) yields the number of coprime factors of N by substituting $n = q(N)$ and $k = 1, \dots, q(N)$. However, it does not consider all possible permutations of these factors. Since, the output CRT-based sequences depend on the order in which the seed sequences are combined (feature 4), all ordered sets of the coprime factors of N must be considered when calculating the number of sequences. Using (6.5), the number of all possible ordered lengths of sequences that can be used as an input

to the CRT to obtain a sequence of length N can be evaluated by:

$$\sum_{k=1}^{q(N)} \left\{ \begin{matrix} q(N) \\ k \end{matrix} \right\} k! = \sum_{k=1}^{q(N)} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} j^{q(N)}. \quad (6.6)$$

So far, we have used features 1, 3, and 4 to compute the number of possible lengths of the seed sequences for the proposed CRT-based scheme. It remains to consider the effect of feature 2 on the number of available length N CRT-based sequences; i.e., the number of available cyclically-generated sequences for each of the lengths of the combined seed sequences. For a given N_i , there exists $N_i\phi(N_i)$ possible cyclic sequences and thus the total number of combinations of the possible seed cyclic sequences is given by $\prod_{i=1}^m N_i\phi(N_i)$. Since all values of N_i , $i = 1, \dots, m$, are relatively prime and $\prod_{i=1}^m N_i = N$, it follows that $\prod_{i=1}^m \phi(N_i) = \phi(N)$; cf. [86]. Subsequently, $\prod_{i=1}^m N_i\phi(N_i) = N\phi(N)$. Finally, using (6.6), the number of available length N sequences that can be generated by the CRT-based scheme, $\bar{N}$, can be calculated as follows:

$$\bar{N} = N\phi(N) \sum_{k=1}^{q(N)} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} j^{q(N)}. \quad (6.7)$$

To draw insights from the expression in (6.7), in Figure 6.3, we compare $\bar{N}$ with both the number of cyclically-generated sequences and the total number of available sequences.

From Figure 6.3, it can be seen that for a sequence with length N that is either prime, e.g., $N = 29$, or a power of a single prime, e.g., $N = 27$, the number of cyclically-generated sequences, i.e., $N\phi(N)$, is equal to that of the sequences generated by the proposed CRT-based scheme. This is because prime numbers and powers of a single prime yield only one coprime factor; i.e., for those numbers $q(N) = 1$ and $\left\{ \begin{matrix} 1 \\ 1 \end{matrix} \right\} = 1$. Hence, such values of N account for the trivial cases in which the input to the CRT sequence generator in Figure 6.2 is an element of the possible $N\phi(N)$ seed

Figure 6.3: Comparison between the number of cyclically-generated sequences, $\bar{N}$, and $N!$.

cyclic sequences. Figure 6.3 also shows that, apart from primes and powers of a single prime, the CRT-based scheme offers a larger number of sequences for RB assignment compared to their cyclic counterparts. Notice that the ratio between the number of CRT-based sequences in (6.7) and the cyclically-generated sequences is given by (6.6), which accounts for the additional degrees of freedom introduced by combining the cyclically-generated sequences using the CRT. From Figure 6.3, it can be seen that neither the number of cyclically-generated sequences nor the number of CRT-based

sequences is monotonically increasing with N . This is because neither $\phi(N)$, nor $q(N)$ is monotonic; e.g., $\phi(31) = 30 > \phi(32) = 16$, and $q(30) = 3 > q(31) = 1$. Finally, the figure shows that the number of sequences generated by the proposed CRT-based scheme is significantly smaller than the total number of available sequences; i.e., $N!$. For example, for a given sequence length $N = 30$, the number of available cyclically-generated sequences and the number of available sequences by the CRT-based scheme are 240 and 3120, respectively. This number is significantly smaller than the $30! = 2.65 \times 10^{32}$ available sequences.

Remark 3. *Although conventional CRT-based sequences will not be cyclic since they are obtained by combining multiple seeds, the proposed CRT-based scheme still yields cyclic sequences in the trivial case in which only a single seed sequence is used as an input to the CRT. In other words, the set of sequences obtained by the CRT can be considered as a superset to the cyclically-generated sequences set. Hence, when used for autonomous RB assignments, the performance of the sequences obtained by the CRT constitutes a lower bound on the number of RB assignment conflicts, when compared to the performance of their cyclically-generated counterparts.* $\square$

As illustrated in Figure 6.3, the proposed CRT-based scheme expands the set of RB assignment sequences, thereby offering an opportunity for performance enhancement. To better quantify this expansion, in the following section, we will derive upper and lower bounds on the number of CRT-based sequences and compare these bounds with the total number of sequences, $N!$.

6.1.5 Bounds on the cardinality of CRT-based sequences

6.1.5.1 Lower bound

We begin this section by evaluating the lower bound on the cardinality of the set of CRT-based sequences, which as per Remark 3, constitutes a superset of cyclically-generated sequences. Hence, the cardinality of CRT-based sequences is lower bounded by the number of cyclically-generated sequences; i.e., $N\phi(N)$. This bound is tight when N is either a prime or a power of a single prime; cf. Section 6.1.4.

6.1.5.2 Upper bound

To obtain an upper bound on the cardinality of the set of CRT-based sequences, we note that the gain in this cardinality with respect to length- N cyclically-generated sequences can be derived from (6.6). In particular, this equation shows that the gain is equivalent to the number of permutations of the coprime factors of N . In other words, this gain depends on all possible ordered sets of the seed cyclic sequences with coprime lengths that can be used as input to the CRT block in Figure 6.2; cf. Section 6.1.4. We are now ready to derive an upper bound on the gain in (6.6). This bound will then be used to compare the cardinality of length- N CRT-based sequences with the total number of available sequences of length N ; i.e., $N!$.

Before deriving the bound, we note that the gain in (6.6) can be divided into three components: 1) the Stirling number of the second kind which provides the cardinality of the coprime factors of N ; 2) the number of unique prime factors of N , $q(N)$; and 3) the number of possible ordered sets of the coprime factors of N . To obtain an upper bound on the gain in (6.6) due to the first and second components, we invoke an upper bound on the Stirling number of the second kind as a function of $q(N)$.

This bound will then be modified to include the gain due to the third component. A bound on the Stirling number of the second kind is given by [87, 88]:

$$\left\{ \begin{matrix} q(N) \\ k \end{matrix} \right\} \leq \frac{1}{2} \binom{q(N)}{k} k^{(q(N)-k)}, \quad k \leq q(N). \quad (6.8)$$

The right hand side of (6.8) gives an upper bound on the number of unordered sets of the coprime factors of N . Now, to compute an upper bound on the corresponding number of ordered sets (cf. Feature 4 in Section 6.1.4) both sides of (6.8) must be multiplied by $k!$ yielding

$$k! \left\{ \begin{matrix} q(N) \\ k \end{matrix} \right\} \leq \frac{1}{2} \frac{q(N)!}{(q(N)-k)!} k^{(q(N)-k)}. \quad (6.9)$$

To obtain better insight into the gain in the number of sequences in (6.6), we will further bound the right hand side of (6.9). Since $\frac{q(N)!}{(q(N)-k)!} \leq q^k(N)$, with equality when $k \in \{0, 1\}$, we have

$$k! \left\{ \begin{matrix} q(N) \\ k \end{matrix} \right\} \leq \frac{q^k(N)}{2} k^{(q(N)-k)}. \quad (6.10)$$

So far, we upper bounded the number of ordered sets of the coprime factors of N as a function of the summation index k in (6.6). Note that this index represents the number of seed sequences that are combined by the CRT. Using (6.10) to upper bound the left hand side of (6.6) yields:

$$\sum_{k=1}^{q(N)} \left\{ \begin{matrix} q(N) \\ k \end{matrix} \right\} k! \leq \max_{k \leq q(N)} \frac{1}{2} q^{k+1}(N) k^{(q(N)-k)} \quad (6.11)$$

$$\leq \frac{1}{2} (q(N))^{q(N)+1}, \quad (6.12)$$

where (6.12) is obtained by substituting $k = q(N)$. Using (6.12), the upper bound on the cardinality of the CRT-based sequences of length N , can be expressed as:

$$\bar{N} \leq \frac{N}{2} \phi(N) (q(N))^{q(N)+1}. \quad (6.13)$$

Now, to complete the derivation of an upper bound on (6.6), we need to bound $q(N)$. One possible bound can be given by the following Lemma:

Lemma 4. *For any value of N , $q(N)$ can be upper bounded by $\lceil \log_{10}(N) \rceil + 1$.*

Proof. See the Appendix B.1. □

Substituting this upper bound in (6.13) yields:

$$\bar{N} \leq \frac{N}{2} \phi(N) (\lceil \log_{10}(N) \rceil + 1)^{\lceil \log_{10}(N) \rceil + 2}. \quad (6.14)$$

Note that the upper bound in (6.14) is still significantly lower than the total number of N -length sequences; i.e., $N! \approx \sqrt{2\pi N} \left(\frac{N}{e}\right)^N$. This is because, the dominant multiplicative factor of $N!$ is N^N , which is a monotonically increasing function of N . On the other hand, the dominant multiplicative factor of (6.14) is a function of $\log_{10}(N)$, which is much less than N . Hence, the right hand side of (6.14) is much less than $N!$. Despite this significant reduction in the number of available sequences, (6.14) indicates that the proposed CRT-based scheme offers an enrichment to the set of easily generated sequences that can be used by the RTs for RB assignments. In particular, this scheme offers a multiplicative gain over its cyclic counterpart; i.e., a gain that is upper bounded by $\frac{1}{2} (\lceil \log_{10}(N) \rceil + 1)^{\lceil \log_{10}(N) \rceil + 2}$. For example, when $N = 30$, the numbers of the cyclically-generated sequences, the CRT-based sequences, and the sequences of length $N = 30$ are 240, 9720, 2.65×10^{32} , respectively. Hence, the proposed scheme offers an opportunity for performance enhancement with much less sequences than $N!$ while maintaining the ease of sequence generation property.

Finally, notice that the bound on $\bar{N}$ in (6.14) was obtained by using the upper bound on $q(N)$ derived in Appendix B.1. However, the gap between this bound and $q(N)$ is relatively small only for small and medium values of N ; cf. the discussion

in Appendix B.2. For asymptotically large values of N , a tighter bound on $q(N)$ is given in [88]. In particular, for asymptotically large values of N :

$$q(N) \leq \frac{\log_{10}(N)}{\log_{10}(\log_{10}(N))} (1 + \mathcal{O}(\log_{10}^{-1}(\log_{10}(N)))). \quad (6.15)$$

However, this bound is loose for small and medium values of N . To illustrate, we tabulated both, the bound in Lemma 4 and the bound in (6.15), for $N \leq 10^{50}$ in Table 6.1. Note that for simplicity, the bounds in Lemma 4 and in (6.15) are tabulated only for selected values of $N = 10^k$ where $k \in \{1, 3, 6, 9, 20, 50\}$ and the constant of the $\mathcal{O}(\cdot)$ is set to 1.

Table 6.1: Values of the bound in Lemma 4 and that given in (6.15) for different values of N .

N	$10^{\lceil \log_{10}(N) \rceil}$	Bound in Lemma 4	Bound in (6.15)
$1 < N \leq 10$	10	2	∞
$10^2 < N \leq 10^3$	10^3	4	19.47
$10^5 < N \leq 10^6$	10^6	7	17.62
$10^8 < N \leq 10^9$	10^9	10	19.32
$10^{19} < N \leq 10^{20}$	10^{20}	21	27.19
$10^{49} < N \leq 10^{50}$	10^{50}	51	46.75

From Table 6.1, it can be seen that the bound in Lemma 4 offers a smaller gap for small and medium values of N . However, for asymptotically large values of N , the bound in (6.15) is tighter than the one in Lemma 4. Since the lengths of the RB assignment sequences in practical relay-assisted scenarios are prospected to be less than 10^9 , we used the bound in Lemma 4 when evaluating (6.14). Next, we will

evaluate the complexity of sequence generation by the proposed CRT-based scheme.

6.1.6 Computational complexity of the CRT-based scheme

In this section, we begin by noting that three components contribute to the complexity of the proposed CRT-based RB assignment scheme, namely, association, search, and generation. The association and search are performed once prior to system startup and thus add negligible computational complexity. Generation, however, is performed at each RT and thus analyzing its computational complexity is essential. Next, we will discuss the three components in detail.

6.1.6.1 Association

Prior to system startup, a one-to-one mapping function is selected between the set of RBs and the sequence elements. We note that in the absence of CSIs, any one-to-one mapping yields the same average performance. Hence, the mapping function can be chosen arbitrarily, thereby adding negligible complexity.

6.1.6.2 Search

A metric-based search is performed to obtain a look-up table of CRT-based sequences with favorable performance. The computational complexity of this search depends on the underlying metric; e.g., exhaustive search or greedy. However, this search is performed only once prior to system startup and the resulting look-up table is stored at the BS. Hence, it does not add any computational burden on the RTs.

6.1.6.3 *Generation*

Before assigning its RBs, each RT uses the CRT to generate a sequence based on the parameters obtained from the BS; i.e., the group generators and the cyclic shifts of the seed cyclic sequences. Assessing the computational complexity of generating this CRT-based sequence requires calculating the computational complexity of two operations: 1) the generation of the elements of the seed cyclic sequences that will be combined by the CRT; and 2) the generation of the sequence elements using the CRT-based combining scheme. We will begin by evaluating the computational complexity of the first operation. Without loss of generality, we will focus on the sequences generated by the ACG structure; cf. Section 6.1.3. Using this structure, each element of the i -th sequence of length N_i can be obtained by performing a single addition of the group generator g_i modulo N_i as shown in (6.1). Hence, obtaining a cyclic sequence of length N_i involves N_i additions and N_i divisions. Now, feature 1 of the proposed CRT-based scheme asserts that combining m seed cyclic sequences yields a CRT-based sequence of length $N = \prod_{i=1}^m N_i$; cf. Section 6.1.4. The total number of elements of these seed sequences is $L_t = \sum_{i=1}^m N_i$. Since $\sum_{i=1}^m N_i \leq \prod_{i=1}^m N_i$, it follows that $L_t \leq N$; $L_t = N \iff m = 1$. Hence, obtaining the elements of the seed cyclic sequences of a CRT-based sequence of length N requires at most N additions and N divisions. For the division operation, we note that the computational complexity of dividing two numbers of j bits each requires $\mathcal{O}((j)^2)$ additions [89]. Since there are N divisions, the computational complexity of generating the seed sequences is bounded as follows:

$$\text{Number of additions} \leq \mathcal{O}(N(1 + (\log_2 N)^2)). \quad (6.16)$$

Next, we will evaluate the computational complexity of combining the seed cyclic sequences.

Feature 3 requires pairwise coprimeness of the lengths of the seed sequences. This coprimeness sets an upper limit on the number of seed sequences, m , to be equal to the number of unique prime factors of the length of the CRT-based sequence; i.e., $1 \leq m \leq q(N)$. In other words, generating an element of a CRT-based sequence involves solving at most $m = q(N)$ congruences; cf. (2.7). The computational complexity of this operation can be obtained by analyzing the complexity of evaluating the components of (2.12). In particular, obtaining an element of a CRT-based sequence using (2.12) involves calculating M_k and x_k , $k = 1, \dots, m$. First, $M_k = (\prod_{j=1}^m N_j)/N_k$, and thus obtaining the values of M_k , $k = 1, \dots, m$ involves m multiplications to calculate $\prod_{j=1}^m N_j$ and m divisions; one for each value of M . Second, to obtain x_k , $k = 1, \dots, m$ one needs to solve (2.8) using the Euclidean algorithm. In this algorithm, the value of x_k is obtained by sequentially dividing the value of N_k until the remainder is zero. Hence, the complexity of this algorithm can be upper bounded by $\frac{N_k}{2}$ divisions. From (2.12), the solution n on the left hand side that is provided by the CRT for the linear set of congruences described in (2.7) involves the addition of m terms; i.e., $\sum_{k=1}^m a_k M_k x_k$. Hence, it can be concluded that the computational complexity of combining m seed cyclic sequences by the CRT to obtain a sequence of N elements can be upper bounded by $N(m + \frac{1}{2} \sum_{k=1}^m N_k)$ divisions, $3mN$ multiplications and mN additions, where $m \leq (N)$. Combining the two operations necessary to obtain a CRT-based sequence of length N , the total complexity of the proposed CRT-based scheme can be bounded by $N(m + 1 + \frac{1}{2} \sum_{k=1}^m N_k)$ divisions, $3mN$ multiplications and $N(m + 1)$ additions. Since $\sum_{k=1}^m N_k \leq N$ and the complexity of a multiplication is identical to that of a division [89], it follows that

the computational complexity of the CRT-based scheme can be upper bounded by

$$\text{Number of additions} \leq \mathcal{O}((0.5N^2 + N(4m + 1))(\log_2 N)^2 + N(m + 1)). \quad (6.17)$$

Note that the complexity of generating a CRT-based sequence is generally affordable by the user terminals. This is because of two reasons. First, the CRT-based scheme has a low order polynomial complexity. Second, generating a CRT-based RB assignment sequence is performed only when the user terminal first switches to the relaying mode or when it is necessary for the RT to update its sequence.

Next, we will provide a greedy-based approach for CRT-based sequences selection in MTC systems with large numbers of RBs and RTs.

6.1.7 A greedy-based approach for sequences selection

As discussed in the previous sections, the number of CRT-based sequences for autonomous RB assignments, $\bar{N}$, tends to increase with the sequence length N ; cf. Figure 6.3. This is because as N increases, both the number of available cyclic sequences, $N\phi(N)$, and the number of unique prime factors, $q(N)$, tend to increase. Hence, for large values of N , it becomes computationally prohibitive to obtain the optimal CRT-based sequences for autonomous RB assignments through exhaustive search. To address this problem, in this section, we propose a greedy-based algorithm for sequences selection that is applicable for systems with large numbers of RBs and RTs. The philosophy of this algorithm is similar to that of the graphical greedy algorithm discussed in Chapter 5. In particular, the sequences selection metric of this greedy algorithm is based on the observation that the RBs located at the beginning of each sequence, i.e., the dominating RBs, are the most significant contributors to the number of hit occurrences since they will always be assigned by the RTs even in

lightly loaded scenarios. Hence, the CRT-based sequences must be selected such that the number of collisions between their dominating RBs is minimized. Next, we will discuss the details of the proposed greedy algorithm.

6.1.7.1 *The greedy algorithm*

Following analogous steps to those used in the graphical greedy algorithm discussed in Chapter 5, we let $\mathcal{T}_{CRT}$ be the look-up table to be constructed by the greedy algorithm and $\mathcal{U}_{CRT}$ be the set containing all the group generators of the seed cyclic sequences that have yet to be selected. At first, $\mathcal{T}_{CRT}$ is empty and $\mathcal{U}_{CRT}$ contains all the ordered group generator combinations of the seed cyclic sequences to be combined by the CRT. The cardinality of $\mathcal{U}_{CRT}$, is given by $\frac{\bar{N}}{N}$; cf. (6.7). We note that, unlike the cyclically-generated sequences, each element of this set does not necessarily contain a pair of group generators. In particular, the number of group generators in each element will depend on the number of seed cyclic sequences to be combined by the CRT-based technique. For instance, if a CRT-based sequence is obtained by combining three seed cyclic sequences, it follows that the corresponding generating element in $\mathcal{U}_{CRT}$ will be the group generator triplet (g_1, g_2, g_3) , where g_1 , g_2 , and g_3 , are the group generators of the three seed cyclic sequences, respectively. In the first iteration, an element, $(g_1, g_2, \dots, g_m)$, where m is the number of seed cyclic sequences corresponding to this generating element, is arbitrarily selected and removed from $\mathcal{U}_{CRT}$. This element is then associated with cyclic shifts $(s_1 = 0, s_2 = 0, \dots, s_m = 0)$, to obtain the first element of the look-up table $\mathcal{T}_{CRT}$. This element is then used to generate the corresponding CRT-based sequence and the dominating RBs in this sequence are then marked.

Following analogous steps to the graphical greedy algorithm discussed in Chap-

per 5, in each of the subsequent $\frac{\bar{N}}{N} - 1$ iterations, the algorithm evaluates the performance of the group generator elements in $\mathcal{U}_{CRT}$ for all possible cyclic shifts. In this evaluation, the greedy algorithm considers only the hits between the dominant RBs corresponding to the elements in $\mathcal{U}_{CRT}$ and the marked dominant RBs corresponding to the elements and cyclic shifts already selected in the look-up table $\mathcal{T}_{CRT}$. This look-up table is then augmented by the element $(g_1, g_2, \dots, g_m, s_1, s_2, \dots, s_m)$ corresponding to the CRT-based sequence that yields the minimum hits between the dominant RBs. The dominant RBs in this CRT-based sequence are then marked. The group generators of the seed cyclic sequences corresponding to the selected CRT-based sequence are then removed from $\mathcal{U}_{CRT}$ and the algorithm iterates until exhaustion of all elements in $\mathcal{U}_{CRT}$. The steps of the proposed greedy algorithm are summarized in Algorithm 1. We note that in this algorithm, we used H_D to refer to the number of hits between the dominating RBs corresponding to the examined CRT-based sequence and the marked dominating RBs corresponding to the selected elements in $\mathcal{T}_{CRT}$.

Next, we will focus attention on the computational complexity of the greedy algorithm.

6.1.7.2 Computational complexity of the greedy algorithm

The computational cost of the greedy algorithm is mainly due to the evaluation of the number of hits between dominant RBs. Similar to the graphical greedy algorithm in Chapter 5, this cost involves only addition operations and is polynomial in $\bar{N}$. To see this, in this section we provide a bound on the complexity of the greedy algorithm.

Denoting the number of dominant RBs by D , it can be seen that the number of required additions for evaluating the number of hits between dominant RBs of an

Algorithm 1 The proposed greedy algorithm

Input:
 $\mathcal{U}_{CRT}$

Output:
 $\mathcal{T}_{CRT}$

- 1: $\mathcal{U}_{CRT} = \mathcal{U}_{CRT} - \{(g_1, g_2, \dots, g_m)\}$
- 2: Set $s_1 = s_2 = \dots = s_m = 0$
- 3: $\mathcal{T}_{CRT} = \{(g_1, g_2, \dots, g_m, s_1, s_2, \dots, s_m)\}$
- 4: **while** $|\mathcal{T}_{CRT}| \leq \frac{\bar{N}}{N}$ **do**
- 5: **for** Each element in $\mathcal{U}_{CRT}$ **do**
- 6: **for** $(s_1, s_2, \dots, s_m)$, where $s_i < N_i, i = 1, \dots, m$ **do**
- 7: Calculate H_D
- 8: **End for**
- 9: **End for**
- 10: Select $(g_1, g_2, \dots, g_m, s_1, s_2, \dots, s_m)^* = \operatorname{argmin}_{(g_1, g_2, \dots, g_m) \in \mathcal{U}_{CRT}, s_i < N_i, i=1, \dots, m} H_D$
- 11: $\mathcal{U}_{CRT} \leftarrow \mathcal{U}_{CRT} - (g_1, g_2, \dots, g_m)^*$
- 12: $\mathcal{T}_{CRT} \leftarrow \mathcal{T}_{CRT} \cup \{(g_1, g_2, \dots, g_m, s_1, s_2, \dots, s_m)^*\}$
- 13: **End while**
- 14: return $\mathcal{T}_{CRT}$

element from $\mathcal{U}_{CRT}$ and all the ones in $\mathcal{T}_{CRT}$ is bounded by $4D$. We note that for a given CRT-based sequence, the number of possible cyclic shifts that are applicable on its seed cyclic sequences is equal to $\prod_{i=1}^m N_i = N$, where m is the number of seed cyclic sequences; cf. Section 6.1.4. Hence, it follows that the complexity of evaluating the number of hits between the dominant RBs of the examined element over all possible cyclic shifts can be bounded by $4ND$. Finally, since $D \leq N$ and in each iteration $|\mathcal{U}_{CRT}| \leq \frac{\bar{N}}{N} - 1$, it follows that the computational complexity of the graphical greedy algorithm is bounded by

$$4ND \left(\frac{\bar{N}}{N} - 1 \right)^2 \leq 4\bar{N}^2, \quad (6.18)$$

thereby enabling the application of the CRT-based sequences in MTC systems with large numbers of devices.

Next, we will provide numerical examples to show the performance gain obtained

by the CRT-based scheme in the context of autonomous RB assignments.

6.2 Simulation results

In this section, we begin by comparing the performance of the RB assignment sequences obtained by the proposed CRT-based scheme, the multiplicative cyclic group (MCG) structure proposed in [61], the shuffled PN sequences discussed in [62, 90–92], a soft fractional frequency reuse (SFFR) scheme similar to that in [93], and the uniformly-distributed random assignment scheme developed in [44]. We will begin by discussing the simulation environment.

6.2.1 Simulation environment

In this work, two simulation environments are considered. In the first environment, we numerically evaluate the performance of the proposed scheme in terms of hit occurrences; cf. Examples 9, 10, 11, 12, 13, and 14. In the second environment, the implications due to the channel effects and the RTs' deployment are considered when evaluating the performance; cf. Examples 15 and 16. In all examples, for the MCG-based scheme in [61], the cyclically-generated sequences are generated for $N = P - 1$, where P is a prime number; cf. [61]. In addition, the lengths of the seed cyclic sequences are selected such that they are relatively prime and their multiplication is equal to $P - 1$ to comply with the length requirement of the MCG structure. For ease of exposition, we will restrict attention to the case of $M = 3$ RTs and apply an exhaustive search based technique for the selection of the cyclic and the CRT-based sequences (except for Example 12 in which N is large). In Example 9, the interference incurred by the RTs is measured by the average number of hits over all possible load combinations for different relative loads $\frac{L}{N}$, where L is the number of assigned

RBs; i.e., over all ordered triples $(\ell_1, \ell_2, \ell_3) \in \mathcal{L} \triangleq \{(\ell_1, \ell_2, \ell_3) | \sum_{i=1}^3 \ell_i = L, \ell_i \geq 0\}$, where ℓ_1, ℓ_2 , and ℓ_3 are the load levels of RTs 1, 2 and 3, respectively. Note that when each MTC device is assigned an RB, the load ℓ_i represents the number of devices served by RT i and L becomes the number of MTC devices. In Examples 10, 11, and 12, we measure the performance by the probability of hit occurrences, which can be calculated by dividing the average number of hits by L ; assuming all loading combinations are equiprobable. In Example 13, we consider an overloaded scenario, i.e., $100\% < \frac{L}{N} \leq 200\%$, and use the number of overutilized RBs as a metric; an RB is overutilized if assigned by 3 RTs simultaneously thus resulting in high interference. In Example 14, we depict the performance of the CRT-based scheme under a non-uniform loading condition. In particular, we evaluate the hits incurred by the RTs in a non-uniform loading scenario in which ℓ_1 and ℓ_2 are fixed while ℓ_3 increases from 0 until the system is fully loaded.

In Examples 15 and 16, we evaluate the CRT-based scheme performance in a practical MTC system. In particular, we assume a fully loaded system, i.e., $\frac{L}{N} = 100\%$, and evaluate the aggregate data rate and the CDF of the successful RB assignments in an MTC system with 3 RTs.

6.2.2 Numerical examples

Example 9. In this example, we compare the performance of the sequences obtained by the proposed CRT-based sequence generation scheme and the MCG scheme proposed in [61] when the number of RBs $N = 30$ and $N = 40$. The RTs apply the hit identification and avoidance (HIA) technique developed in [61]. Note that the HIA technique cannot be applied to the randomly-generated sequences because they do not possess a specific structure.

Figure 6.4: Comparison between the Chinese remainder theorem-based sequences, the cyclically-generated sequences, and the random assignments.

From Figure 6.4, it can be seen that both, the CRT-based sequences and the cyclically-generated ones, outperform the randomly-generated sequences. For example, for a system with $N = 40$ RBs at a relative load, $\frac{L}{N} = 90\%$, the average number of hits resulting from using the CRT-based, the cyclic, and the random sequences are 1.4, 2.5, and 8.8, respectively. We also observe from the figure that the performance gain of the proposed CRT-based scheme over that of the cyclically-generated sequences of [61] increases with the number of RBs N at high relative loads; e.g., $\frac{L}{N} = 100\%$. This is mainly due to two reasons. First as N increases, the cardinality of the set of possible group generators increases, which increases the cardinality of the set of available seed cyclic sequences for the CRT-based scheme. Second, as N increases, the value of $q(N)$ tends to increase with N , cf. Figure 6.3, thereby increasing the set of coprime sequences' lengths that can be used as an input to the proposed

CRT-based scheme. Finally, we note that the sequences obtained by the MCG structure of [61] constitutes a subset of the set of CRT-based sequences and thus their performance is upper bounded by that of the CRT-based ones, cf. Remark 3. $\square$

Example 10. In this example, we evaluate the impact of the order in which the cyclic sequences are combined using the CRT on the performance. In particular, we consider two cases, both with $N = 30$ and $M = 3$, but with two different combining orders. In both cases, HIA was employed and the sequences were selected based on exhaustive search. The CRT-based scheme was used to combine the cyclically-generated sequences by the two additive cyclic groups $\mathbb{Z}_3$ and $\mathbb{Z}_{10}$ of lengths $N = 3$ and $N = 10$, respectively. However, in case one, the two combined seed sequences were selected from the cyclic groups pair $(\mathbb{Z}_{10}, \mathbb{Z}_3)$, respectively, whereas in case two, the two combined seed sequences were selected from $(\mathbb{Z}_3, \mathbb{Z}_{10})$, respectively.

From Figure 6.5, it can be seen that changing the combining order has an impact on the observed performance since different combining orders yield unique sets of assignment sequences each possessing a different structure. For example, at a relative load, $\frac{L}{N} = 90\%$, the probability of hit occurrences for cases one and two are 0.046 and 0.029, respectively. $\square$

In the following two examples, we compare the performance of the proposed CRT-based sequences with that of the shuffled PN-based RB assignment scheme discussed in [62, 90–92]. This sequence-based RB assignment scheme was specifically selected because its premises is similar to that of the CRT-based RB assignment scheme. In particular, the PN-based RB assignment scheme assumes the absence of the CSI and requires no coordination between neighboring devices. Hence, it is readily applicable in relay-assisted MTC systems with self-organizing RTs.

Figure 6.5: Effect of combining order on the performance of the Chinese remainder theorem-based sequences for $N = 30$.

Example 11. In this example, we compare the performance of the CRT-based sequences, the shuffled PN-based sequences of [62], and the random RB assignments of [44]. Similar to Example 9, the number of RTs, $M = 3$, and they are all assumed to use the HIA technique developed in [61]. All cyclic shifts are set to zero and the number of RBs, $N = 63$. The length of the corresponding shuffled PN-based sequences is $2^k - 1$, $k = 6$.

From Figure 6.6, it can be seen that both the shuffled PN-based sequences and the CRT-based ones outperform the random RB assignments. For example, at a relative load, $\frac{L}{N} = 90\%$, the probability of hit occurrences of the CRT-based, the shuffled PN-based, and the random sequences are 0.09, 0.17, and 0.24, respectively. We also observe that the proposed CRT-based sequences outperform their shuffled PN-based counterparts at all relative loads. This is mainly due to the fact that the

Figure 6.6: Comparison between the random assignments, the shuffled pseudonoise-based sequences, and the proposed Chinese remainder theorem-based sequences for $N = 63$ resource blocks.

cardinality of the set of CRT-based sequences, cf. Section 6.1.4, is much larger than that of the shuffled PN-based ones, which is limited to $\frac{\phi(2^k-1)}{k} = \frac{\phi(63)}{6} = 6$ [94]. $\square$

Example 12. In this example, we consider a setup similar to that of the previous example. We compare the performance of the CRT-based sequences with that of the shuffled PN-based ones when the number of RBs is large; i.e., $N = 511 = 2^k - 1, k = 9$. However, unlike the other examples, the RB assignment sequences are selected using the computationally-efficient greedy-based algorithm of Section 6.1.7. This is because obtaining the optimal RB assignment sequences through exhaustive search is computationally expensive for sequences with large numbers of RBs. In this example, the number of dominating RBs is set to 160.

From Figure 6.7, it can be seen that both, the shuffled PN-based sequences and

Figure 6.7: Comparison between the random assignments, the shuffled pseudonoise-based sequences, and the proposed Chinese remainder theorem-based sequences for $N = 511$ resource blocks.

the CRT-based ones, outperform the random RB assignments. For example, at a relative load, $\frac{L}{N} = 100\%$, the probability of hit occurrences of the CRT-based, the shuffled PN-based, and the random sequences are 0.022, 0.179, and 0.303, respectively. Figure 6.7 also shows that the computationally-efficient greedy-based approach, despite being potentially suboptimal, yields CRT-based sequences with performance significantly superior to that of the random RB assignments and the shuffled PN-based sequences. Hence, this greedy-based approach renders the CRT-based sequences appealing for MTC systems with large numbers of devices. $\square$

Example 13. In this example, we compare the performance of CRT-based sequences with that of cyclically-generated sequences of [61] and random sequences of [44] in an overloaded scenario, wherein the number of MTC devices is larger than the number

of RBs; i.e., the relative load $100\% < \frac{L}{N} \leq 200\%$. This comparison is depicted in Figures 6.8a and 6.8b when $N = 30$, $M = 3$, and without HIA. In this example, the performance metric is based on reducing the overutilization of particular RBs by limiting the load level of each RB to 2; an RB is overutilized if assigned by 3 RTs simultaneously. Note that an optimal RB assignment case that assumes perfect centralized coordination between neighboring RTs is also depicted for reference.

In Figure 6.8a, we show the reduction in the number of overutilized RBs resulting from using the CRT-based sequences. For instance, for $\frac{L}{N} = 180\%$, the number of overutilized RBs when using the CRT-based, the cyclic, and the random sequences are 1.67, 4.24, and 3.84, respectively; i.e., the CRT-based sequences yield a 56% gain over their cyclic counterparts. Note that an optimal RB assignment will yield 0 overutilized RBs at any given load (this optimal assignment cannot be achieved with any autonomous scheme since no sequences will yield such performance).

In Figure 6.8b, we show the increase in the number of RBs assigned by 2 RTs, i.e., RBs that are not overutilized, when using the CRT-based sequences. For instance, for $\frac{L}{N} = 180\%$, the number of RBs assigned by 2 RTs for the three schemes are 20.66, 15.52, and 16.33, respectively; i.e., the CRT-based sequences yield a 26% gain over their cyclic counterparts. For this metric, an optimal RB assignment would result in a linear increase in the number of RBs assigned by two RTs with L . Finally, we note that the performance gain shown in Figures 6.8a and 6.8b is mainly due to the structure of the jointly designed CRT-based sequences that enables implicit coordination between RTs, thereby reducing the number of overutilized RBs even in overloaded scenarios. $\square$

Example 14. In this example, we consider a 3-RT scenario in which $\ell_1 = \ell_2 = 4$, while ℓ_3 increases from 0 to 22; i.e., RT 3 status changes from not loaded to lightly

(a) resource blocks assigned simultaneously by 3 relaying terminals (overutilized).

(b) resource blocks assigned simultaneously by 2 relaying terminals.

Figure 6.8: Comparison between randomly-generated sequences, the sequences obtained by the multiplicative cyclic group structure, and the proposed Chinese remainder theorem-based scheme when $100\% < \frac{L}{N} \leq 200\%$.

loaded and finally to heavily loaded. Hence, this scenario captures both uniform and non-uniform load distributions. We compare the performance of the proposed CRT-based scheme with that of an SFFR scheme similar to that in [93], which provides optimal performance in uniformly loaded scenarios. In particular, in this scheme, each RT is given a subset of the available RBs and only attempts to assign the remaining system RBs if all the ones within its given subset are already assigned. For completeness, we also depict the performance of the cyclic and the random based schemes.

Figure 6.9: A performance comparison between the Chinese remainder theorem, the soft fractional frequency reuse, the cyclic, and the random based resource block assignment schemes for $N = 30$ resource blocks.

From Figure 6.9, it can be seen that the CRT-based scheme outperforms the SFFR, the cyclic, and the random based ones. For example, when $\ell_3 = 21$, the average number of hits for the CRT-based scheme is 1, whereas those of the cyclic,

the SFFR, and the random schemes are 2, 6.15, and 6.96, respectively; i.e., the CRT-based scheme offers a gain of at least 50% over the other schemes. It can also be seen from Figure 6.9 that no hits are observed by the SFFR scheme when $\ell_3 \leq 10$. This is because, each RT first attempts to assign from its given set of $\frac{N}{3}$ RBs before switching to random RB assignments. Hence, this scheme can provide optimal performance, when either the system load is uniform or when the loads of each of the 3 RTs is less than or equal $\frac{N}{3}$. Figure 6.9 also shows that at $\ell_3 = 22$, i.e., a fully loaded system, both the CRT and the cyclic-based scheme result in only two hits. This is because both schemes use the HIA algorithm to avoid potential future hits; a feature that is not applicable to SFFR and random RB assignments. $\square$

Example 15. In this example, we evaluate the performance of the CRT-based sequences in a practical relay-assisted MTC system. In particular, we evaluate the system's aggregate data rate when the CRT-based sequences are used in an MTC system with $M = 3$ RTs and $N = 30$ RBs and compare it with the aggregate data rate when cyclically-generated sequences of the same length are used. In the considered system, each RT assigns one RB to each incoming MTC device during a scheduling interval of 1 ms. In other words, in each scheduling interval, 3 RBs corresponding to $M = 3$ RTs are assigned. Once RT i assigns an RB to an incoming MTC device, it increases the number of its assigned RBs, N_i , by one. At this point, either the assignment will be successful or a collision will occur. In case of a successful assignment, the assigned RB is retained and the RT proceeds during the next scheduling interval to assign the following RB in its sequence. In case of collision, the RT which was first to assign the RB retains it, whereas the other conflicting RT releases the RB and reduces its N_i by one at the end of its scheduling interval. The impact of a collision will be reflected on the interference observed by MTC devices. In particular,

a collision on a given RB will result in an interference between the transmissions of their respective MTC devices for the duration of one scheduling interval. In this example, the system has an initial load of zero and the number of MTC devices is increased in a round Robin fashion until each of the RTs is loaded with 10 MTC devices. In that case, the system is fully loaded; i.e., $\frac{L}{N} = 100\%$.

The channel gains between the RTs and the MTC devices are generated using the line-of-sight Hotspot pathloss model [82] with frequency-flat Rayleigh fading and standard Gaussian noise spectral density of -174 dBm/Hz. The RB bandwidth, and the receiver noise figure are set to 180 KHz and 5 dB, respectively [82]. The transmit power, P_t , of all MTC devices is set to 0 dBm and the locations of these devices and the RTs are uniformly distributed within the cell boundaries. For a practical dense deployment scenario, MTC devices are assumed to be less than 300 m away from their assisting RTs and the maximum inter-RT distance is set to 100 m. The RTs implement the HIA algorithm of [61] and the performance is averaged over 10000 channel realizations. Using the aforementioned parameters, the aggregate data rate, R , at a given scheduling interval can be calculated using:

$$R = BW \sum_{i=1}^M \sum_{j=1}^{N_i} \log_2 \left(1 + \frac{|H_{l,i,B_j}|^2 P_t}{P_n + \sum_{m=1, m \neq l}^Q |H_{m,i,B_j}|^2 P_t} \right), \quad (6.19)$$

where BW is the assigned bandwidth and P_n is the noise power, and B_j is the j -th RB assigned by RT i and is selected by the RT based on its sequence; i.e., the j -th entry in the RT's assignment sequence. In (6.19), the parameter Q is the total number of MTC devices, and H_{l,i,B_j} is the channel coefficient between RT i and MTC device l on RB B_j ; H_{l,i,B_j} is set to zero if B_j is not assigned by RT i to the MTC device l .

As shown in Figure 6.10a, the aggregate data rate achieved by the CRT-based

(a) Aggregate data rates.

(b) Average SINR of RTs 1 and 2.

Figure 6.10: Comparison between the aggregate data rates and the average SINR achieved by the cyclically-generated and the Chinese remainder theorem-based resource block assignment sequences for $N = 30$ resource blocks.

sequences constitutes an upper bound on that achieved by the cyclically-generated sequences. This confirms the observation that the cyclically-generated sequences constitute a special case of the CRT-based sequences. For example, at the eighth scheduling interval, the aggregate data rate achieved by the CRT-based sequences is 17% higher than the one achieved by the cyclically-generated sequences. This performance gain is mainly due to the lower number of hit occurrences, i.e., the lower interference level, observed by the RTs due to the utilization of the CRT-based sequences. Figure 6.10a also illustrates that for the first few scheduling intervals, e.g., the third scheduling interval, both the cyclically-generated sequences and the CRT-based sequences achieve the same data rate. This is because for these scheduling intervals, all the RTs are lightly loaded, thereby no hits are observed between the RTs. Finally, at high scheduling interval indices the two sequence structures ultimately achieve the same aggregate data rate. This is because, at these scheduling intervals, all the N RBs are assigned by the RTs and the data rate becomes flat.

In Figure 6.10b, we consider a 3-RT setup similar to that of Figure 6.10a. However, in this setup, $P_t = -50$ dBm and the loads of the first two RTs are set to 3, i.e., $\ell_1 = \ell_2 = 3$, while ℓ_3 increases from 0 to 24; i.e., until $L = 100\%$. In addition, when a hit occurs, the RTs do not relinquish their assigned RBs. The average SINR of the first two RTs is depicted in Figure 6.10b to show the effect of hit occurrences on the system performance. In particular, as ℓ_3 increases, hits occur between the RBs assigned by RT 3 and the ones already assigned by the first two RTs, resulting in a sudden drop in the average SINR. From this figure, it can also be seen that the performance of the CRT-based sequences outperform their cyclic counterparts. For instance, when $\ell_3 = 20$, the average SINR achieved by the CRT-based sequences and their cyclic counterparts is 7 dB and 5.5 dB, respectively. This is mainly due to

the rich structure of the CRT-based sequences, which yields a lower number of hit occurrences. $\square$

Example 16. In this example, we evaluate the number of MTC devices with successful RB assignments versus the number of RB assignment attempts. The setup of this simulation is similar to that of the previous one. However, the total system load is not uniformly distributed across the 3 RTs and the performance results are averaged over 10000 load distributions.

Figure 6.11: CDF of the number of attempts needed for successful resource block assignment for $N = 30$ resource blocks.

From Figure 6.11, it can be seen that the CRT-based sequences outperform both, the random and the cyclically-generated sequences. For example, for the case of one RB assignment attempt per MTC device, i.e., no collisions, the numbers of successfully assigned MTC devices are 28.49, 25.09, and 19.9 when the RB assignments

are done using the CRT-based, the cyclically-generated, and the random sequences, respectively. In other words, the CRT-based sequences offer a 13.5% gain over the cyclically-generated sequences in the number of MTC devices not experiencing any collisions. This gain increases to 43.2% when compared to the performance of the random RB assignments. Note that the performance gain due to the CRT-based sequences is projected to increase for higher values of N . Figure 6.11 also shows that using the CRT, 99% of all MTC devices are successfully assigned with at most two RB assignment attempts, whereas in the case of cyclically-generated RB assignments, the 99-th percentile is reached with three assignment attempts. Using the results reported in Figure 6.11, one can evaluate the power wastage incurred by collisions and the data transmission delay experienced by MTC devices; i.e., the duration between the first RB assignment and the successful one. This can be done by invoking the scheduling interval and the transmit power, P_t .

Denoting the transmission delay by d and the number of RB assignment attempts by a , it can be seen that $d = \bar{s}(a-1)$, where $\bar{s} = 1$ ms is the scheduling interval duration. For example, a device with 3 assignment attempts will incur a delay, $d = 2$ ms.

To evaluate the power wastage due to hits, denoted by P_w , one can use $P_w = P_t(a-1)$. Using Figure 6.11, it can be concluded that 95% and 83.6% of the MTC devices incurred less than $P_w = 1$ mW power wastage and $d = 1$ ms delay when using the CRT-based sequences and the cyclically-generated sequences, respectively. $\square$

Chapter 7

Pseudonoise-Based Resource Block Assignments in Relay-Assisted MTC Systems

In this chapter, we begin by discussing the motivation behind investigating the potential application of PN-based sequences for autonomous RB assignments in MTC systems with self-organizing relaying terminals. We then focus attention on the relationship between a subset of the PN-based sequences, i.e., the M-sequences, and their cyclic counterparts. Subsequently, we use this relationship along with the properties of cyclic groups to develop a technique that simplifies the generation of cyclic sequences. This is followed by a performance comparison between PN-based, cyclic, and CRT-based sequences when used for autonomous RB assignments in relay-assisted MTC systems. Finally, we evaluate the autocorrelation and cross-correlation properties of CRT-based sequences and compare them to those of ZC and M-sequences.

7.1 Motivation

PN-based sequences play a key role in the realization of a wide spectrum of applications. One category of such sequences is the M-sequences, which are widely adopted in FH spread spectrum applications, wherein at any given dwell interval, the frequency used for transmission follows a locally-generated M-sequence [62, 90, 91].

These sequences are also widely used for achieving the initial synchronization in many cellular systems. For example, the secondary synchronization signals of LTE [63] and the primary synchronization signals of the 3GPP 5G NR [64] are both based on M-sequences. Other examples of the applications of M-sequences include ultra-wideband sensing [95] and image scrambling [96]. This wide range of applications is mainly sparked by the favorable properties of M-sequences [97]; e.g., their single peak autocorrelation property and the simplicity of their generation. In particular, each M-sequence can be easily constructed from a single seed, i.e., a primitive polynomial, using a set of linear shift registers. This feature indicates that there exists a relationship between the set of M-sequences and the set of cyclically-generated sequences since each of the sequences within these sets can be obtained by repeatedly applying a certain operation on a single seed. In fact, as will be discussed hereinbelow, the set of M-sequences is a subset of that of the cyclically-generated sequences and thus it follows that:

- there exists a group isomorphism that maps a subset of the set of sequences cyclically-generated by the ACG structure to that of the M-sequences;
- the performance of M-sequences, when used for RB assignments in our proposed relay-assisted MTC setup, will be upper-bounded by that of the cyclically-generated sequences and the CRT-based ones, cf. Example 17; and
- the generation of cyclic sequences can be simplified by following a shift register-based technique similar to the one used for generating M-sequences.

In the following sections, we will discuss the relationship between M-sequences and their cyclically-generated counterparts and highlight its implications in simpli-

fyng the generation of cyclic sequences.

7.2 Preliminaries of PN-based sequences

In this section, we begin by reviewing the preliminaries pertaining to the generation of M-sequences.

Definition 6 (Fields [67]). *A finite field is a set $\mathcal{F}$ of p^n elements, where p is prime and $n \geq 1$, that is defined with two operations; namely addition and multiplication. The elements of this set constitutes a group with respect to the addition operation. $\square$*

Each field, $\mathcal{F}(p^n)$, forms a cyclic group of order $p^n - 1$ under multiplication with the elements $\{1, a, a^2, \dots, a^{p^n-2}\}$, where the field element a is a group generator of the cyclic group.

Definition 7 (Primitive Elements [67]). *A field element a^s of a field $\mathcal{F}(p^n)$ is a primitive element iff $\gcd(s, p^n - 1) = 1$. This element is one of the group generators of the cyclic group $\mathcal{F}(p^n) \setminus \{0\}$. $\square$*

Each primitive element of a field, i.e., a group generator of the cyclic group $\mathcal{F}(p^n) \setminus \{0\}$, yields a cyclic sequence by repeated applications of the group operation. Next, we will provide a definition of the primitive polynomials of a field.

Definition 8 (Primitive Polynomials [98]). *A polynomial $g(x)$ of degree $n \geq 1$ is a primitive polynomial if it is the minimal polynomial of a primitive element of $\mathcal{F}(p^n)$, where a minimal polynomial of a field element $B \in \mathcal{F}(p^n)$ is the unique polynomial of lowest degree which has B as a root and whose coefficients lie in $\mathcal{F}(p)$. $\square$*

For any primitive polynomial of a field $\mathcal{F}(p^n)$, the polynomial roots are a subset of the set of primitive elements of the field. In particular, for any primitive polynomial, the roots consist of a primitive element and its conjugates, which are defined next.

Definition 9 (Conjugates of a Field Element [67]). *The conjugates of a field element $B \in \mathcal{F}(p^n)$ are the elements $\{B, B^p, B^{p^2}, \dots, B^{p^{r-1}}\}$ where $r \leq n$ is the smallest integer for which $B^{p^r} \equiv B$.* $\square$

Using these preliminaries and the properties of M-sequences, one can construct an isomorphism between the set of M-sequences and a subset of the cyclically-generated ones. Next, we will construct this isomorphism and provide a detailed discussion on its implications on the performance of M-sequences when used for autonomous RB assignments in relay-assisted MTC systems.

7.3 Isomorphism between the cyclically-generated and the M-sequences

We begin this section by introducing the following lemma:

Lemma 5. *For any given sequence length $N = 2^n - 1$, $n \geq 2$, the set of M-sequences is isomorphic to a subset of the set of cyclically-generated sequences by the ACG structure of order N .* $\square$

Proof. The proof of Lemma 5 follows from the properties of cyclically-generated and M-sequences. In particular, Theorem 2 states that for any cyclic group of order N , $\mathbb{G}_N$, the cyclically-generated sequences by the group operation underlying $\mathbb{G}_N$ are isomorphic to the ones that are cyclically-generated by the ACG operation underlying $\mathbb{Z}_N$. In other words, there exists a one-to-one and onto structure preserving function that maps the elements of each of the cyclically-generated sequences of $\mathbb{G}_N$ to those of a unique cyclically-generated sequence by the ACG structure.

Now, for the field of 2^n elements, $\mathcal{F}(p^n)$, $p = 2, n \geq 2$, that yields PN-based M-sequences of length $N = 2^n - 1$, the number of available generating primitive

polynomials can be given by $\frac{\phi(N)}{n}$ [94]. Since each M-sequence corresponds to a distinct primitive polynomial, it follows that the number of available M-sequences is equal to $\frac{\phi(N)}{n}$. In [97, 99], it was shown that each M-sequence can be represented exponentially as powers of a primitive element modulo p and the primitive polynomial. In other words, each M-sequence is a cyclically-generated sequence and its group generator is the primitive element of the corresponding primitive polynomial. Hence, it follows that the $\frac{\phi(N)}{n}$ cyclically-generated M-sequences are isomorphic to a subset of the $\phi(N)$ cyclically-generated sequences by the ACG structure. $\square$

Next, we will discuss the implications of Lemma 5 on M-sequences and cyclically-generated ones.

7.3.1 Performance of maximal length RB assignment sequences

As discussed in the previous section, the set of M-sequences constitutes a subset of that of the cyclically-generated ones. Subsequently, it follows that the set of M-sequences is also a subset of that of the CRT-based sequences. Hence, the performance of M-sequences, when used for autonomous RB assignments in relay-assisted MTC systems, will be upper bounded by that of the cyclic and the CRT-based sequences. However, the performance gap between the cyclic and the M-sequences will depend on the cardinality of the set of M-sequences and their structure. This performance gap is evaluated in Example 17 of Section 7.4 for M-sequences of length, $N = 2^5 - 1 = 31$.

7.3.2 Generation of binary M-sequences using the ACG structure

From Lemma 5, it can be concluded that there exists at least one one-to-one and onto function that maps the elements of a subset of the cyclically-generated sequences

by the ACG structure to those of the M-sequences. In addition, we note that the cyclically-generated sequences by a primitive element and its conjugates yield the same binary M-sequence. Hence, it follows that the set of binary M-sequences can be easily generated by using the ACG structure in three steps: 1) generate the set of cyclic sequences using the ACG structure; 2) use a group isomorphism function to map the resulting set of sequences; and 3) convert the mapped sequences into binary and discard the duplicate binary M-sequences that are generated by the conjugates of the primitive elements.

7.3.3 Generation of cyclic sequences using shift registers

One of the favorable properties of M-sequences is that they can be easily generated by using shift registers. This feature renders M-sequences appealing in many systems that exploit devices with limited processing capabilities; e.g., MTC systems. Hence, in this section, we will introduce a technique that invokes the graphical representation of cyclic groups to extend the ease of generation property to the cyclically-generated sequences of length, $N = 2^m - 1$, $m \in \mathbb{Z}$. In particular, this technique involves two main steps: 1) the generation of a basis cyclic sequence, i.e., an M-sequence, using shift registers; and 2) using the graphical representation of cyclic groups and their properties to obtain the set of cyclically-generated sequences from the basis sequence. The details of these two steps are discussed next.

7.3.3.1 Generation of the basis cyclic sequence

In this section, we focus attention on generating a basis cyclic sequence that will be used to obtain the corresponding set of cyclically-generated sequences. We note that this sequence is not unique. In other words, any cyclic sequence of length N can

be used to obtain the corresponding set of N -length cyclically-generated sequences; i.e., the basis cyclic sequence can be randomly selected. This is because, for any sequence length, N , the set of cyclically-generated sequences is unique, cf. Theorem 2. Now, since each M-sequence is cyclic by definition, i.e., can be cyclically-generated from a single seed, then any M-sequence can be used as the basis cyclic sequence. Hence, to obtain the basis cyclic sequence of length $N = 2^m - 1, m \geq 1$, we begin by randomly selecting a primitive polynomial of order m . Using this polynomial, a set of binary shift registers are connected; cf. Figure 7.1.

Figure 7.1: Multiplicative shift register-based generation of the M-sequence corresponding to the primitive polynomial, $x^4 + x + 1$.

We note that at each clock cycle, the equivalent decimal value of the states of the binary shift registers, i.e., $\{a_0, a_1, \dots, a_m\}$, represents an element of the generated cyclic sequence. Finally, since the registers are connected based on a primitive polynomial, the resulting cyclically-generated sequence will be of length $N = 2^m - 1$. This sequence is then considered as the basis cyclic sequence.

Next, we will show how the set of cyclically-generated sequences can be obtained from the basis sequence.

7.3.3.2 Generation of the cyclic sequences set

In this step, we use two properties of the cyclically-generated sequences, namely the inverse and the graphical representation of cyclic groups, to generate the set of

cyclically-generated sequences from a basis sequence; cf. Chapter 5. We note that this step does not involve any complex computations and can be summarized as follows:

1. Obtain the inverse of the basis cyclic sequence; i.e., the reversed version of the basis cyclic sequence.
2. Use the basis sequence and its inverse to construct the first pattern in the graphical representation of the cyclic group of order N ; i.e., the outer circle.
3. Construct the remaining patterns of the graphical representation of the cyclic group of order N ; cf. Figure 7.2.
4. From each pattern, obtain the corresponding cyclically-generated sequence.
5. Reverse each of the obtained cyclic sequences to get its inverse.

The resulting sequences constitute the set of N -length cyclically-generated sequences. Next, we will provide numerical examples to compare the performance and the properties of M-sequences to those of cyclic and CRT-based ones.

7.4 Simulation results

In this section, we compare the performance of the sequences obtained by the proposed CRT-based scheme of Chapter 6 and the cyclically-generated sequences by the MCG structure proposed in [61] with that of the PN-based M-sequences and the ZC sequences that are commonly used in practical systems; e.g., the primary and secondary synchronization signals in LTE [63, 100–102]. We will begin by discussing the simulation environment.

Figure 7.2: A graphical representation of a cyclic group of order 7. This cyclic group has 3 group generator pairs, each yielding a unique pattern (the two inner patterns and the outer circle).

7.4.1 Simulation environment

In this work, two simulation environments are considered. In the first environment, we numerically evaluate the performance of CRT-based, cyclic, and M-sequences in terms of hit occurrences; cf. Example 17. In the second environment, we evaluate the correlation properties of CRT-based sequences and compare them to those of M-sequences and ZC ones; cf. Examples 18, 19, 20, and 21.

For ease of exposition, in Example 17, we will restrict attention to the case of

$M = 3$ RTs and apply an exhaustive search based technique for the selection of cyclic, CRT-based, and M-sequences. In this example, the interference incurred by the RTs is measured by the average number of hits over all possible load combinations for different relative loads $\frac{L}{N}$, where L is the number of assigned RBs; i.e., over all ordered triples $(\ell_1, \ell_2, \ell_3) \in \mathcal{L} \triangleq \{(\ell_1, \ell_2, \ell_3) | \sum_{i=1}^3 \ell_i = L, \ell_i \geq 0\}$, where ℓ_1, ℓ_2 , and ℓ_3 are the load levels of RTs 1, 2, and 3, respectively. Note that when each MTC device is assigned an RB, the load ℓ_i represents the number of MTC devices served by RT i and L becomes the total number of MTC devices.

In Examples 18 and 20, we evaluate the autocorrelation and the cross-correlation properties of the CRT-based sequences and compare them to those of M-sequences and ZC ones. In these examples, we demonstrate the tradeoff between the cross-correlation and the autocorrelation properties of CRT-based, ZC, and M-sequences. In particular, we show the cross-correlation gain achieved by the CRT-based sequences over M-sequences and ZC ones at the expense of an autocorrelation penalty.

In Examples 19 and 21, we extend our evaluation of the autocorrelation properties of the CRT-based sequences in the previous two examples by considering a more practical scenario with carrier frequency offsets. In particular, we evaluate the ambiguity function for CRT-based, ZC, and M-sequences; i.e., the autocorrelation of these sequences under different carrier frequency offsets. This metric is essential for determining the applicability of a set of sequences in many practical applications; e.g., the applicability of a set of sequences for synchronization signalling in cellular systems such as the LTE and the 3GPP 5G NR. This is because practical systems suffer from carrier frequency drifts due to using imperfect oscillators. Hence, for these systems to operate properly, it is essential that the synchronization sequences maintain their favorable autocorrelation properties even in the presence of carrier frequency offsets.

7.4.2 Numerical examples

Example 17. In this example, we compare the performance of the sequences obtained by the proposed CRT-based sequence generation scheme of Chapter 6, the MCG scheme proposed in [61], and the M-sequences. The lengths of the CRT-based, the cyclic, and the M-sequences are set to 30, 30, and 31, respectively. This is because the length of the binary M-sequences must be of the form $2^k - 1, k \geq 1$. The RTs apply the HIA technique developed in [61]. For comparison, the performance of the randomly-generated sequences for $N = 30$ is also shown.

From Figure 7.3, it can be seen that all structured sequences, i.e., the CRT-based sequences, the cyclically-generated sequences, and the M-sequences, outperform the randomly-generated ones. For example, at a relative load, $\frac{L}{N} = 90\%$, the average number of hits resulting from using the CRT-based sequences, the cyclic sequences, the M-sequences, and the random ones are 0.78, 2.1, 3.48, and 6.3, respectively. Figure 7.3 also shows that both the CRT-based sequences and the cyclic ones outperform the M-sequences at all relative loads. This performance gain also increases as the relative load increases. For example, at a relative load, $\frac{L}{N} = 100\%$, the average number of hits resulting from using the the CRT-based sequences, the cyclic sequences, and the M-sequences are 1.89, 2.9, and 4.67, respectively; i.e. the cyclic and the CRT-based sequences offer a gain of at least 38% over the M-sequences. This is mainly due to the fact that the M-sequences constitutes a subset of the cyclic sequences of [61] and the CRT-based sequences of Chapter 6 and thus their performance is inferior to those sequences; cf. Lemma 5. □

Example 18. In this example, we focus attention on the correlation properties of CRT-based and M-sequences. In particular, we set $N = 15$ and evaluate the

Figure 7.3: A performance comparison between M-sequences, Chinese remainder theorem-based sequences, cyclically-generated sequences, and random assignments.

normalized autocorrelation and cross-correlation of the two sets of sequences in the absence of carrier frequency offsets; i.e., $f_o = 0$. Note that, for this example, the autocorrelation and the cross-correlation are normalized by N to eliminate the effect of the sequence length on the performance. The CRT-based sequences are selected based on their cross-correlation properties. In particular, the selected CRT-based sequences are the ones that yield minimal normalized cross-correlation peaks.

(a) Normalized autocorrelation of the Chinese remainder theorem-based and the M-sequences for $N = 15$.

(b) Normalized cross-correlation of the Chinese remainder theorem-based and the M-sequences for $N = 15$.

Figure 7.4: A comparison between the autocorrelation and cross-correlation properties of the Chinese remainder theorem-based sequences and the M-sequences, when $f_o = 0$.

From Figure 7.4a, it can be seen that the normalized autocorrelation of M-sequences outperforms that of CRT-based ones when $f_o = 0$. In particular, the normalized autocorrelation of M-sequences is equal to 1 at 0 time offset and $\frac{-1}{N} = -0.067$ otherwise, whereas that of CRT-based sequences has multiple side peaks as high as 0.2. This inferior performance of the CRT-based sequences is mainly due to the fact that the CRT-based sequences are selected based on their cross-correlation properties.

From Figure 7.4b, it can be seen that the normalized cross-correlation of CRT-based sequences outperforms that of M-sequences. In particular, the normalized cross-correlation of CRT-based sequences has only two levels with a peak equal to 0.2, whereas that of M-sequences has multiple levels with peaks as high as 0.467. This performance gain is mainly due to the richness of the set of CRT-based sequences when compared to the set of M-sequences; cf. Lemma 5. Hence, these sequences can be favorable in practical applications such as synchronization that can benefit from the low cross-correlation peaks.

Example 19. In this example, we evaluate the effect of carrier frequency offsets due to imperfect oscillators on the autocorrelation properties of M-sequences and CRT-based ones. To do this, we evaluate the ambiguity function for the two sets of sequences, when $N = 15$. Similar to the previous example, the CRT-based sequences are selected based on their cross-correlation properties. The sampling frequency, f_s , is set to 30.72 MHz similar to that of LTE [83] and the ambiguity function is evaluated for carrier frequency offsets, f_o , up to 1.23 MHz.

From Figures 7.5a and 7.5b, it can be seen that both the CRT-based and the M-sequences yield a high peak-to-side-lobe ratio even in the presence of high carrier frequency offsets. In other words, both sequence sets offer low ambiguity side lobes

(a) The ambiguity function of an M-sequence.

(b) The ambiguity function of a Chinese remainder theorem-based sequence.

Figure 7.5: A comparison between the ambiguity functions of the Chinese remainder theorem-based sequences of Chapter 6 and the M-sequences for $N = 15$.

for frequency offsets up to 1.23 MHz. For instance, the observed peak-to-side-lobe ratios for the M-sequences and the CRT-based ones are 3.06 and 2.87, respectively. These high peak-to-side-lobe ratios indicate that the two sets are favorable for synchronization applications as they offer a large noise margin; i.e., the ability to achieve synchronization in a noisy channel. Figures 7.5a and 7.5b also show that the performance of the selected CRT-based sequences in this example is slightly inferior to that of M-sequences. This is because, the CRT-based sequences are selected based on their cross-correlation properties. In particular, the CRT-based sequences that yield the lowest cross-correlation peaks are selected since it is desirable for the synchronization sequences to possess ‘good’ cross-correlation and autocorrelation properties. Finally, we note that the cross-correlation of the selected CRT-based sequences significantly outperform that of M-sequences at an expense of a slight increase in their peak-to-side-lobe ratio; cf. Example 18. $\square$

Unlike Examples 18 and 19, in the following two examples, we compare the performance of the CRT-based sequences with that of the ZC ones, which are commonly used in practice; e.g., the primary synchronization signal in LTE [100]. In particular, in Example 20, we focus attention on the autocorrelation and cross-correlation properties of the two sets of sequences. This focus is then extended in Example 21 by considering the effect of practical system impairments, i.e., carrier frequency offsets due to imperfect oscillators, on the autocorrelation properties of ZC and CRT-based sequences.

Example 20. In this example, we evaluate the correlation properties of CRT-based and ZC sequences. In particular, we set $N = 15$ and evaluate the normalized autocorrelation and cross-correlation of the two sets of sequences in the absence of carrier frequency offsets; i.e., $f_o = 0$. Similar to Example 18, the CRT-based sequences are

selected based on their cross-correlation properties and the autocorrelation and the cross-correlation are normalized by N to eliminate the effect of the sequence length on the performance.

From Figure 7.6a, it can be seen that the normalized autocorrelation of ZC sequences outperforms that of CRT-based ones, when $f_o = 0$. In particular, the normalized autocorrelation of ZC sequences is equal to 1 at 0 time offset and 0 otherwise, whereas that of CRT-based sequences has multiple side peaks as high as 0.2. We note that this inferior performance of the CRT-based sequences is mainly due to the fact that the CRT-based sequences are selected based on their cross-correlation properties.

From Figure 7.6b, it can be seen that the normalized cross-correlation of CRT-based sequences outperforms that of ZC sequences. In particular, the normalized cross-correlation peak of CRT-based sequences is equal to 0.2, whereas that of ZC ones is as high as 0.258. This performance gain is mainly due to the richness of the set of CRT-based sequences. Hence, it can be appealing to use the CRT-based sequences in practical applications that require low cross-correlation; e.g., synchronization.

Example 21. In this example, we evaluate the effect of carrier frequency offsets due to imperfect oscillators on the autocorrelation properties of ZC sequences and CRT-based ones. Similar to Example 19, we set $N = 15$ and evaluate the ambiguity function for the two sets of sequences. The CRT-based sequences are selected based on their cross-correlation properties and the sampling frequency, f_s , is set to 30.72 MHz [83]. The ambiguity function is evaluated for carrier frequency offsets, f_o , up to 1.84 MHz.

From Figures 7.7a and 7.7b, it can be seen that CRT-based sequences yield a significantly higher peak-to-side-lobe ratio than the one yielded by ZC sequences. For

(a) Normalized autocorrelation of the Chinese remainder theorem-based and the Zadoff-Chu sequences for $N = 15$.

(b) Normalized cross-correlation of the Chinese remainder theorem-based and the Zadoff-Chu sequences for $N = 15$.

Figure 7.6: A comparison between the autocorrelation and the cross-correlation properties of the Chinese remainder theorem-based sequences and the Zadoff-Chu ones, when $f_o = 0$.

(a) The ambiguity function of a Zadoff-Chu sequence.

(b) The ambiguity function of a Chinese remainder theorem-based sequence.

Figure 7.7: A comparison between the ambiguity functions of the Chinese remainder theorem-based sequences of Chapter 6 and the Zadoff-Chu sequences for $N = 15$.

instance, for $N = 15$, the observed peak-to-side-lobe ratios for CRT-based sequences and ZC ones are 2.87 and 1, respectively. Hence, the CRT-based sequences can be more favorable for synchronization applications than their ZC counterparts since the

latter ones have high ambiguity side lobes, which impedes their performance in noisy channels. Finally, we note that although ZC sequences suffer from a low peak-to-side-lobe ratio, they were still selected for the primary synchronization signalling of LTE because of two favorable features: 1) at the absence of frequency offsets, the value of the autocorrelation of ZC sequences is either N or 0, cf. Example 20; and 2) the locations of the side lobes are known to the receiver and thus it can correctly synchronize with the transmitter in noisy channels. $\square$

Chapter 8

Summary and Future Work

In this thesis, we addressed the problem of resource block assignment for machine-type communication systems with self-organizing relaying terminals. In particular, we proposed sequence-based RB assignment schemes by which the RTs can autonomously assign the available RBs to their assisted MTC devices in the absence of CSI. Since the RBs are autonomously assigned by the RTs, the proposed schemes dispense with the need for coordination between the RTs and its associated overhead. In the proposed schemes, the RB assignments are independent of CSI and thus they are easily applicable in the highly dynamic terminal relaying topology with its fast channel variations. In addition, the structured sequences underlying the proposed schemes were shown to outperform their PN-based counterparts. We conclude this thesis by highlighting the key points in our proposed work and introducing the open research directions for future work.

8.1 Thesis summary

The key points of the proposed work in this thesis can be presented by the following items:

- In Chapter 4, we use group isomorphism to eliminate the constraint on the number of RBs to be a prime number minus one and to show that the optimal

assignment sequences generated by a specific cyclic group are globally optimal over the set of all cyclically-generated sequences.

- In Chapter 5, we develop a greedy algorithm with polynomial complexity for the sequential selection of RB assignment sequences in systems with large numbers of RTs and arbitrary device distributions. This algorithm is further simplified by invoking the graphical representation of cyclic groups. The resulting graphical greedy algorithm is more efficient and thus suitable for generating assignment sequences for relay-assisted MTC systems with a massive number of devices.
- In Chapter 6, we extend the RB assignment scheme by using multi-generator sequences. In particular, we use the CRT to combine the cyclic sequences generated by simple cyclic group structures into longer ones. The combining process introduces additional degrees of freedom in sequence generation, thereby enriching the set of available sequences for RB assignments. We also develop an efficient greedy-based algorithm for CRT-based sequences selection. The computational complexity of this algorithm is significantly lower than exhaustive search, thereby enabling the application of the CRT-based RB assignment sequences in systems with large numbers of RTs and MTC devices.
- In Chapter 7, we use group isomorphism to show that the set of PN-based M-sequences constitutes a subset of that of the set of cyclically-generated sequences by the ACG structure. Subsequently, we conclude that the performance of M-sequences, when used for autonomous RB assignments, will be upper bounded by that of the cyclic and the CRT-based sequences. In addition, we develop

a technique that uses the properties of cyclic groups and the isomorphism of M-sequences to extend the simple shift register-based generation property of M-sequences to their corresponding cyclically-generated supersets. Finally, we compare the correlation properties of CRT-based, ZC, and M-sequences and accordingly discuss the applicability of the CRT-based sequences for synchronization signalling in practical systems that suffer from high carrier frequency offsets.

8.2 Proposed future work

In this section, we provide directions for future research that can potentially improve the performance of autonomous RB assignments in relay-assisted MTC systems. These directions can be summarized in the following bullets:

- In the current research, we proposed an autonomous RB assignment scheme for relay-assisted MTC systems that depends on cyclically-generated sequences, i.e., sequences obtained from a single generator; cf. Chapter 5. This work was further extended in Chapter 6 by proposing CRT-based sequences that are constructed from multiple seed cyclic sequences. To further reduce the number of assignment conflicts between neighboring RTs, one might explore different tools from number theory and their properties for potential sequence generation; e.g., circular permutations. Subsequently, this can yield a set of structured RB assignment sequences with performance superior to that obtained by the proposed CRT-based sequences; cf. Figure 8.1.
- In the current research, RTs are assumed not to communicate with one another and not to have any channel state information. However, in some scenarios,

Figure 8.1: Relationship between the set of cyclically-generated sequences, the set of CRT-based sequences and the set of all available sequences of a given length.

RTs can temporarily obtain long term average channel parameters, thereby creating an opportunity for performance enhancement; e.g., in scenarios, wherein the RTs and MTC devices are static for a relatively long duration. In particular, we would like to investigate the design updates necessary to improve the overall system performance if RTs can acquire the long term average channel parameters of their assisted MTC devices.

- We assume that RTs are not capable of sensing the spectrum before assigning

RBs to their assisted MTC devices. While this assumption has its advantages in reducing the consumed power and eliminating the delay imposed by the sensing process, we would like to investigate the effect of employing occasional and partial spectrum sensing on the performance of the proposed HIA algorithm. In particular, if the RTs are able to occasionally sense a portion of the available spectrum, they will be able to identify some RBs that are occupied by other RTs. To utilize this information, the RTs can possibly apply a modified version of the HIA algorithm to identify the load levels of neighboring relays and avoid future hits accordingly. However, the sensing strategy and the number of reserved resources for sensing should be carefully designed so that it does not infringe on the resources available for communications.

- In the current research, we assumed perfect backhaul links between the RTs and the BS, thereby eliminating the impact of the BS-RT links on the performance of RB assignments by RTs. However, in practical scenarios, RTs may be required to negotiate the available resources with the BS based on their load, thereby creating an opportunity for performance enhancement. This is because, in such scenarios, the BS can inform each RT of the number of its neighboring RTs as well as their current load levels. Using this information, the RT can adjust its RB assignment sequence to reduce the number of potential hits with its neighboring RTs, thereby enhancing the performance of the MTC system. Despite its advantages, too frequent negotiations with the BS can impact the system performance due to the signalling overhead. Hence, the signalling strategy between the BS and the RTs must be carefully designed to realize its benefits without significantly reducing the communication resources.

Appendix A

Proof of Theorem 2

A.1 Proof of Theorem 2

Let $\mathbb{Z}_N$ be an ACG and $\mathbb{G}_N$ be an arbitrary cyclic group, both of order N . From Lemma 3, there is at least one isomorphism function that maps the elements of $\mathbb{G}_N$ to those of $\mathbb{Z}_N$. Furthermore, Theorem 3.26 in [70], asserts that such an isomorphism maps every group generator in $\mathbb{G}_N$ to a distinct group generator in $\mathbb{Z}_N$. Using these facts, we will develop a constructive proof to show that the sequences obtained by the generators of $\mathbb{G}_N$ are isomorphic to those obtained by the generators of $\mathbb{Z}_N$.

Let g_1 and g_2 be two distinct generators of $\mathbb{G}_N$ and let F be the isomorphism that maps these generators to two group generators of $\mathbb{Z}_N$, say z_1 and z_2 , i.e., $g_1 \xrightarrow{F} z_1$ and $g_2 \xrightarrow{F} z_2$. To establish the desired isomorphism, we will generate the sequences in $\mathbb{Z}_N$ that are isomorphic to the ones generated by g_1 and g_2 in two different ways. In the first way, the sequences in $\mathbb{Z}_N$ are generated by applying F to every element of the sequences generated by g_1 and g_2 . However, at this point, there is no guarantee that the sequences obtained in this way can be cyclically-generated by the group generators of $\mathbb{Z}_N$. To prove that this is indeed the case, in the second way, we will make use of Theorem 3.26 in [70] to determine z_1 and z_2 , and to obtain their corresponding sequences. The isomorphism between the cyclic sequences obtained by $\mathbb{G}_N$ and $\mathbb{Z}_N$ is established by showing that the sequences of the two ways are identical.

For the first way, we note that the cyclic sequences of g_1 and g_2 can be expressed as

$$S_1 = (g_1^1(\bmod N+1), \dots, g_1^N(\bmod N+1)), \text{ and} \quad (\text{A.1})$$

$$S_2 = (g_2^1(\bmod N+1), \dots, g_2^N(\bmod N+1)). \quad (\text{A.2})$$

Since the sequence generated by any of the group generators of $\mathbb{G}_N$ spans the complete sequence of integers from 1 to N , then there exists an integer $1 \leq q \leq N$ such that $g_2 = g_1^q \pmod{N+1}$. Subsequently, S_2 can be written as

$$S_2 = (g_1^q(\bmod N+1), \dots, g_1^{Nq}(\bmod N+1)), \quad (\text{A.3})$$

where in writing (A.3), we have used the fact that $(g_1^q \pmod{N+1})^m \pmod{N+1} = g_1^{mq} \pmod{N+1} \forall m \in \{1, \dots, N\}$.

Next, we define an isomorphism function, F that maps the element $g_1^k \pmod{N+1} \in \mathbb{G}_N$ to the element $k \pmod{N} \in \mathbb{Z}_N$, $k = 0, \dots, N-1$. Applying this isomorphism on the sequences S_1 and S_2 yields

$$S'_1 = (1, 2, \dots, N-1, 0), \text{ and} \quad (\text{A.4})$$

$$S'_2 = (q(\bmod N), 2q(\bmod N), \dots, Nq(\bmod N)). \quad (\text{A.5})$$

We will now provide another way for generating these sequences. Applying F on g_1 and g_2 yields z_1 and z_2 , which are group generators of $\mathbb{Z}_N$, cf. [70, Theorem 3.26], where

$$z_1 = F(g_1(\bmod N+1)) = 1, \quad \text{and} \quad (\text{A.6})$$

$$\begin{aligned} z_2 &= F(g_2(\bmod N+1)) \\ &= F(g_1^q(\bmod N+1)) = q(\bmod N). \end{aligned} \quad (\text{A.7})$$

Since z_1 and z_2 are group generators of $\mathbb{Z}_N$, applying (2.4) on them yields the cyclic sequences

$$\tilde{S}_1 = (z_1(\bmod N), 2z_1(\bmod N), \dots, Nz_1(\bmod N)), \quad (\text{A.8})$$

$$\tilde{S}_2 = (z_2(\bmod N), 2z_2(\bmod N), \dots, Nz_2(\bmod N)). \quad (\text{A.9})$$

Substituting (A.6) and (A.7) in (A.8) and (A.9) yields

$$\tilde{S}_1 = (1, 2, \dots, 0) = S'_1, \quad \text{and} \quad (\text{A.10})$$

$$\tilde{S}_2 = (q(\bmod N), 2q(\bmod N), \dots, Nq(\bmod N)) = S'_2. \quad (\text{A.11})$$

The equalities in (A.10) and (A.11) yield that the set of sequences generated by an arbitrary cyclic group $\mathbb{G}_N$ are isomorphic to the set of sequences generated by the ACG $\mathbb{Z}_N$. □

Appendix B

Proof and Discussion on The Upper Bound on $q(N)$ in Lemma 4

We begin by proving the upper bound on $q(N)$ in Lemma 4. This is then followed by a discussion on the tightness of this bound for asymptotically large values of N .

B.1 Proof of Lemma 4: upper bound on $q(N)$

Let $\{p_1, p_2, \dots, p_n\}$ be a set of distinct prime numbers such that $p_1^{c_1} p_2^{c_2} \dots p_n^{c_n} \leq N$, where c_i is the repetitions of the i -th prime factor p_i . It follows that the cardinality of this set, i.e., n , is maximized when the prime numbers $p_1, p_2, \dots, p_n$ are selected to be the smallest distinct primes, i.e., $p_1 = 2, p_2 = 3, p_3 = 5, \dots$, and the constants $c_1, \dots, c_n$ are set to, $c_1 = \dots = c_n = 1$. We will refer to the maximum value of n as $\bar{n}$. Now, $q(N)$ is defined as the number of unique prime factors of N , cf. Section 6.1.4, and thereby is equal to the cardinality of the set of distinct primes $\{p_1, p_2, \dots, p_{q(N)}\}$ such that $p_1^{t_1} p_2^{t_2} \dots p_{q(N)}^{t_{q(N)}} = N$, where t_i is the repetitions of the i -th prime factor, p_i . Note that, for any N , $\{p_1, p_2, \dots, p_{q(N)}\} \subseteq \{p_1, p_2, \dots, p_{\bar{n}}\}$; i.e., $q(N) \leq \bar{n}$. Hence, to obtain an upper bound on $q(N)$, it suffices to derive an upper bound on $\bar{n}$.

We begin by noting that for any $N > 1$ to have $\bar{n}$ distinct primes, it is necessary that $N \geq p_1 p_2 \dots p_{\bar{n}}$. Now, apart from the first four primes (i.e., 2, 3, 5, and 7), all

prime numbers are larger than 10. Hence, any increase in $\bar{n}$ beyond four, will require increasing the value of N by a factor that is larger than 10. Hence, one possible bound on $\bar{n} > 4$, is given by $\lceil \log_{10}(N) \rceil + 1$ since it increases by at least 1 as N increases by a factor of 10 or more. However, we need to show that this bound holds for $\bar{n} \leq 4$. For N to have 1, 2, 3, and 4 distinct primes, it follows that N must be larger than or equal $p_1 = 2$, $p_1 p_2 = 6$, $p_1 p_2 p_3 = 30$, and $p_1 p_2 p_3 p_4 = 210$, respectively. For each of these four values of N , the corresponding values of $\lceil \log_{10}(N) \rceil + 1$ are 2, 2, 3, and 4, respectively; i.e., $\lceil \log_{10}(N) \rceil + 1 \geq \bar{n}$ for $\bar{n} \leq 4$. Hence, for any value of N , the value of $\bar{n}$, and subsequently the value of $q(N)$, can be both upper bounded by $\lceil \log_{10}(N) \rceil + 1$.

B.2 Discussion on the tightness of the bound on $q(N)$

As shown in the previous section, the bound in Lemma 4 is valid for any value of N . However, it becomes less tight as the value of N increases. This can be shown by considering the rate of increase of the upper bound with N and comparing it with that of the actual number of distinct prime numbers.

Let N be an asymptotically large number, where $N = p_1 \dots p_{\bar{n}}$. For this value of N , the number of distinct primes $q(N) = \bar{n}$ and is upper bounded by $\lceil \log_{10}(N) \rceil + 1$. Let $p_{\bar{n}+1}$ be the smallest prime following $p_{\bar{n}}$ and let $\hat{N} = p_1 \dots p_{\bar{n}+1}$. Note that $p_{\bar{n}+1} \gg 10$ for large values of N . Now, the contribution of this prime, i.e., $p_{\bar{n}+1}$, to $q(\hat{N})$ is one, i.e., $q(\hat{N}) = q(N) + 1$, whereas its contribution to the upper bound is $\log_{10}(p_{\bar{n}+1}) \gg 1$; i.e., $\lceil \log_{10}(\hat{N}) \rceil = \lceil \log_{10}(N) + \log_{10}(p_{\bar{n}+1}) \rceil$. Hence, the upper bound in Lemma 4 increases at a rate higher than that of $q(N)$ for asymptotically large values of N , thereby yielding a loose bound.

References

- [1] J. Kim, J. Lee, J. Kim, and J. Yun, “M2M service platforms: Survey, issues, and enabling technologies,” *IEEE Commun. Surveys & Tutorials*, vol. 16, no. 1, pp. 61–76, April 2014.
- [2] S. Lien and K. Chen, “Massive access management for QoS guarantees in 3GPP machine-to-machine communications,” *IEEE Commun. Letters*, vol. 15, no. 3, pp. 311–313, March 2011.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A survey on enabling technologies, protocols, and applications,” *IEEE Commun. Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, June 2015.
- [4] K. Zheng, S. Ou, J. Alonso-Zarate, M. Dohler, F. Liu, and H. Zhu, “Challenges of massive access in highly dense LTE-Advanced networks with machine-to-machine communications,” *IEEE Wireless Communications*, vol. 21, no. 3, pp. 12–18, June 2014.
- [5] T. C. Hsieh and Y. C. Chen, “Traffic aggregation for overload control in LTE-based machine type communications,” in *Proc. IEEE International Symposium on Wireless Personal Multimedia Communications (WPMC)*, China, Nov. 2016, pp. 199–204.
- [6] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, “Toward massive machine type cellular communications,” *IEEE Wireless Communications*, vol. 24, no. 1, pp. 120–128, Feb. 2017.
- [7] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jntti, and A. Ghosh, “Machine-type communications: current status and future perspectives toward 5G systems,” *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 10–17, Sept. 2015.
- [8] M. Hasan, E. Hossain, and D. Niyato, “Random access for machine-to-machine communication in LTE-Advanced networks: Issues and approaches,” *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, June 2013.

- [9] S.-Y. Lien, K.-C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3GPP machine-to-machine communications," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 66–74, April 2011.
- [10] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. Johnson, "M2M: From mobile to embedded internet," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 36–43, April 2011.
- [11] S. Wang, H. Su, H. Hsieh, S. Yeh, and M. Ho, "Random access design for clustered wireless machine to machine networks," in *Proc. IEEE Int. Black Sea Conf. on Commun. and Networking*, Georgia, July 2013, pp. 107–111.
- [12] S. E. Wei, H. Y. Hsieh, and H. J. Su, "Joint optimization of cluster formation and power control for interference-limited machine-to-machine communications," in *Proc. IEEE Glob. Commun. Conf.*, California, Dec. 2012, pp. 5512–5518.
- [13] A. Asadi and V. Mancuso, "Network-assisted outband D2D-clustering in 5G cellular networks: Theory and practice," *IEEE Trans. Mob. Computing*, vol. 16, no. 8, pp. 2246–2259, Aug. 2017.
- [14] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 65–75, Nov. 2014.
- [15] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-Advanced networks," *IEEE Wireless Communications*, vol. 19, no. 3, pp. 96–104, June 2012.
- [16] 3GPP TR 36.843 version 12.0.1, *Study on LTE Device to Device Proximity Services, Radio Aspects (Release 12)*, March 2014, available at: http://www.3gpp.org/ftp/specs/archive/36_series/36.843.
- [17] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3GPP device-to-device proximity services," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 40–48, April 2014.
- [18] 3GPP TR 38.913 version 14.0.0, *Study on Scenarios and Requirements for Next Generation Access Technologies (Release 14)*, Oct. 2016, available at: http://www.3gpp.org/ftp/specs/archive/38_series/38.913.
- [19] J. Schlien and A. Roessler, "Device to device communication in LTE," *Rohde & Schwarz Inc.*, Sept. 2015.

- [20] Q. Li, R. Q. Hu, Y. Qian, and G. Wu, "Cooperative communications for wireless networks: techniques and applications in LTE-Advanced systems," *IEEE Wireless Communications*, vol. 19, no. 2, pp. 22–29, April 2012.
- [21] J. Lianghai, L. Man, and H. D. Schotten, "Context-aware cluster based device-to-device communication to serve machine type communications," in *Proc. IEEE Int. Conf. Commun. wkshp. (ICC Workshop)*, Paris, May 2017, pp. 1011–1017.
- [22] W. Dang, M. Tao, H. Mu, and J. Huang, "Subcarrier-pair based resource allocation for cooperative multi-relay OFDM systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1640–1649, May 2010.
- [23] Q. Song, Z. Ning, Y. Huang, and L. Guo, "A tabu-based optimal sub-carrier allocation algorithm in multi-hop OFDMA relaying network," in *Proc. IEEE Int. Conf. Info. and Commun. Technologies*, India, April 2013, pp. 381–386.
- [24] Q. Song, Y. Huang, Z. Ning, and F. Wang, "Subcarrier allocation in multi-hop orthogonal frequency division multiple access wireless networks," *Elsevier Journal of Computers and Electrical Engineering*, vol. 40, no. 2, pp. 599–611, Feb. 2014.
- [25] Y. Cao, T. Jiang, and C. Wang, "Cooperative device-to-device communications in cellular networks," *IEEE Wireless Commun. Mag.*, vol. 22, no. 3, pp. 124–129, June 2015.
- [26] P. Phunchongharn, E. Hossain, and D. I. Kim, "Resource allocation for device-to-device communications underlaying LTE-Advanced networks," *IEEE Wireless Commun. Mag.*, vol. 20, no. 4, pp. 91–100, Aug. 2013.
- [27] P. N. Alevizos, E. A. Vlachos, and A. Bletsas, "Inference-based distributed channel allocation in wireless sensor networks," March 2017, available at <https://arxiv.org/abs/1703.06652>.
- [28] Q. Ye, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "Distributed resource allocation in device-to-device enhanced cellular networks," *IEEE Trans. Commun.*, vol. 63, no. 2, pp. 441–454, June 2015.
- [29] R. Yin, G. Yu, C. Zhong, and Z. Zhang, "Distributed resource allocation for D2D communication underlaying cellular networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, June 2013, pp. 138–143.
- [30] T. Wang, G. B. Giannakis, and R. Wang, "Smart regenerative relays for link-adaptive cooperative communications," *IEEE Trans. Commun.*, vol. 56, no. 11, pp. 1950–1960, Nov. 2008.

- [31] W. Rhee and J. Cioffi, "Increase in capacity of multiuser OFDM system using dynamic subchannel allocation," in *Proc. IEEE Vehic. Tech. Conf. (VTC)*, Tokyo, May 2000, pp. 1085–1089.
- [32] A. Behnad, X. Gao, and X. Wang, "Distributed resource allocation for multi-hop decode-and-forward relay systems," *IEEE Trans. Vehicular Tech.*, vol. 64, no. 10, pp. 4821–4826, Oct. 2015.
- [33] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.
- [34] J. Goseling, C. Stefanovic, and P. Popovski, "Sign-compute-resolve for random access," in *Proc. 52nd Annual Allerton Conference on Comm., Control, and Computing (Allerton)*, Illinois, Sept. 2014, pp. 675–682.
- [35] E. Paolini, G. Liva, and A. G. i Amat, "A structured irregular repetition slotted ALOHA scheme with low error floors," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, May 2017, pp. 1–6.
- [36] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.
- [37] F. Clazzer, E. Paolini, I. Mambelli, and . Stefanovi, "Irregular repetition slotted ALOHA over the rayleigh block fading channel with capture," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, May 2017, pp. 1–6.
- [38] M. Hasan and E. Hossain, "Distributed resource allocation for relay-aided device-to-device communication under channel uncertainties: A stable matching approach," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3882–3897, Oct. 2015.
- [39] Z. Chang, T. Ristaniemi, and Z. Niu, "Radio resource allocation for collaborative OFDMA relay networks with imperfect channel state information," *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2824–2835, May 2014.
- [40] S. Mallick, R. Devarajan, M. M. Rashid, and V. K. Bhargava, "Resource allocation for selective relaying based cellular wireless system with imperfect CSI," *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1822–1834, May 2013.
- [41] H. Wu, X. Gao, X. Wang, and X. You, "Sum-rate-optimal precoding for multi-cell large-scale MIMO uplink based on statistical CSI," *IEEE Trans. Commun.*, vol. 63, no. 8, pp. 2924–2935, Aug. 2015.

- [42] N. Abuzainab, W. Saad, and H. V. Poor, "Cognitive hierarchy theory for heterogeneous uplink multiple access in the Internet of Things," in *Proc. IEEE Int. Symp. Inf. Theory*, Barcelona, July 2016, pp. 1252–1256.
- [43] N. Abuzainab, W. Saad, C. S. Hong, and H. V. Poor, "Cognitive hierarchy theory for distributed resource allocation in the internet of things," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 7687–7702, Dec. 2017.
- [44] C. Koutsimanis and G. Fodor, "A dynamic resource allocation scheme for guaranteed bit rate services in OFDMA networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Beijing, May 2008, pp. 2524–2530.
- [45] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. United Kingdom: John Wiley, 2007.
- [46] C. H. Wei, R. G. Cheng, and S. L. Tsao, "Modeling and estimation of one-shot random access for finite-user multichannel slotted ALOHA systems," *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1196–1199, Aug. 2012.
- [47] J. B. Seo and V. C. M. Leung, "Design and analysis of cross-layer contention resolution algorithms for multi-packet reception slotted ALOHA systems," *IEEE Trans. Wireless Commun.*, vol. 10, no. 3, pp. 825–833, March 2011.
- [48] A. Laya, C. Kalalas, F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Goodbye, ALOHA!" *IEEE Access*, vol. 4, pp. 2029–2044, April 2016.
- [49] G. Margelis, R. Piechocki, D. Kaleshi, and P. Thomas, "Low throughput networks for the IoT: Lessons learned from industrial implementations," in *Proc. IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec. 2015, pp. 181–186.
- [50] C. Goursaud and Y. Mo, "Random unslotted time-frequency ALOHA: Theory and application to IoT UNB networks," in *Proc. IEEE 23rd International Conference on Telecommunications (ICT)*, May 2016, pp. 1–5.
- [51] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 60–67, Oct. 2016.
- [52] G. Alfano, M. Garetto, and E. Leonardi, "New directions into the stochastic geometry analysis of dense CSMA networks," *IEEE Trans. Mobile Comp.*, vol. 13, no. 2, pp. 324–336, Feb. 2014.

- [53] N. Achir, Y. Bouchaala, P. Muhlethaler, and O. Shagdar, "Comparison of spatial ALOHA and CSMA using simple stochastic geometry models for 1D and 2D networks," in *Proc. IEEE 23rd International Conference on Telecommunications (ICT)*, May 2016, pp. 1–7.
- [54] B. Shrestha, E. Hossain, and K. W. Choi, "Distributed and centralized hybrid CSMA/CA-TDMA schemes for single-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 4050–4065, July 2014.
- [55] T. Nguyen, F. Baccelli, Z. Kai, S. Subramanian, and W. Xinzhou, "A performance analysis of CSMA based broadcast protocol in VANETs," in *Proc. IEEE Int. Conf. Comp. Commun. (INFOCOM)*, April 2013, pp. 2805–2813.
- [56] A. M. Cailean, B. Cagneau, L. Chassagne, V. Popa, and M. Dimian, "A survey on the usage of DSRC and VLC in communication-based vehicle safety applications," in *Proc. IEEE 21st Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov. 2014, pp. 69–74.
- [57] Q. Ye and W. Zhuang, "Distributed and adaptive medium access control for Internet-of-Things-enabled mobile networks," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 446–460, April 2017.
- [58] J. Chen, Q. Yu, B. Chai, Y. Sun, Y. Fan, and X. Shen, "Dynamic channel assignment for wireless sensor networks: A regret matching based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 95–106, Jan. 2015.
- [59] B. Karaoglu and W. Heinzelman, "Cooperative load balancing and dynamic channel allocation for cluster-based mobile ad hoc networks," *IEEE Trans. Mobile Computing*, vol. 14, no. 5, pp. 951–963, May 2015.
- [60] Y. Sun and J. sheng Qian, "Distributed channel fast allocation in cognitive wireless sensor networks," *Springer Wireless Personal Communications Journal*, vol. 96, no. 4, pp. 5931–5945, Oct. 2017.
- [61] Y. M. M. Fouad, R. H. Gohary, and H. Yanikomeroglu, "An autonomous resource block assignment scheme for OFDMA-based relay-assisted cellular networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 637–647, Feb. 2012.
- [62] C. F. Shih, A. E. Khafa, and J. Zhou, "Practical frequency hopping sequence design for interference avoidance in 802.15.4e TSCH networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, June 2015, pp. 6494–6499.

- [63] I. Kim, Y. Han, and H. K. Chung, "An efficient synchronization signal structure for OFDM-based cellular systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 99–105, Jan. 2010.
- [64] Intel Corporation, "NR PSS and SSS design," in *3GPP TSG RAN WG1 Meeting RAN1 Number 88*, April 2017, Document Number: R1-1706156, Available at: http://www.3gpp.org/ftp/TSG_RAN/WG1_RL1/TSGR1-88b/Docs/R1-1706156.zip.
- [65] W. W. Adams and L. J. Goldstein, *Introduction to Number Theory*. New Jersey: Prentice-Hall Inc., 1976.
- [66] H. M. Stark, *An Introduction to Number Theory*. United States: First MIT Press, 1987.
- [67] J. A. Gallian, *Contemporary Abstract Algebra*. Boston: Houghton Mifflin, 1998.
- [68] G. H. Hardy and E. M. Wright, *An Introduction to the Theory of Numbers*. United Kingdom: Clarendon Press, 1979.
- [69] C. Ding, D. Pei, and A. Salomaa, *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography*. Singapore: World Scientific Publishing Company, 1996.
- [70] L. Gilbert and J. Gilbert, *Elements of Modern Algebra*. United States: Brooks/Cole, 2005.
- [71] A. Celik, R. M. Radaydeh, F. S. Al-Qahtani, and M. S. Alouini, "Resource allocation and interference management for D2D-enabled DL/UL decoupled Het-Nets," *IEEE Access*, vol. 5, pp. 22 735–22 749, Oct. 2017.
- [72] H. B. Chang and I. Rubin, "Optimal downlink and uplink fractional frequency reuse in cellular wireless networks," *IEEE Trans. Vehicular Tech.*, vol. 65, no. 4, pp. 2295–2308, April 2016.
- [73] W. S. Jeon, J. Kim, and D. G. Jeong, "Downlink radio resource partitioning with fractional frequency reuse in femtocell networks," *IEEE Trans. Vehicular Tech.*, vol. 63, no. 1, pp. 308–321, Jan. 2014.
- [74] A. Mahmud and K. A. Hamdi, "A unified framework for the analysis of fractional frequency reuse techniques," *IEEE Trans. Commun.*, vol. 62, no. 10, pp. 3692–3705, Oct. 2014.

- [75] Z. Kostic and N. Sollenberger, “Performance and implementation of dynamic frequency hopping in limited-bandwidth cellular systems,” *IEEE Trans. Wireless Commun.*, vol. 1, no. 1, pp. 28–36, Jan. 2002.
- [76] P. Popovski, H. Yomo, S. Aprili, and R. Prasad, “Frequency rolling: A cooperative frequency hopping for mutually interfering WPANS,” in *Proc. ACM Int. Symp. Mobile Ad Hoc Ntwk. Computing (MOBIHOC)*, Tokyo, May 2004, pp. 199–209.
- [77] A. A. Shaar and L. Fukshansky, “A new family of one-coincidence sets of sequences with dispersed elements for frequency hopping CDMA systems,” Jan. 2017, available at <http://arxiv.org/abs/1701.05209>.
- [78] D. Dummit and R. Foote, *Abstract Algebra*. United Kingdom: John Wiley, 2003.
- [79] S. Roman, *Fundamentals of Group Theory: An Advanced Approach*. Boston: Springer, 2011.
- [80] G. E. Andrews, *The Theory of Partitions*. Cambridge: Cambridge University Press, 1976.
- [81] G. Hahn and G. Sabidussi, *Graph Symmetry: Algebraic Methods and Applications*. Netherlands: K. A. Publishers, 1997.
- [82] 3GPP TR 36.814 version 9.0.0, *Further advancements for E-UTRA physical layer aspects (Release 9)*, March 2010, available at: http://www.3gpp.org/ftp/specs/archive/36_series/36.814.
- [83] C. Johnson, *Long Term Evolution in bullets*. United Kingdom: Johnson, 2012.
- [84] C. Tseng, H. Wang, F. Kuo, K. Ting, H. Chen, and G. Chen, “Delay and power consumption in LTE/LTE-A DRX mechanism with mixed short and long cycles,” *IEEE Trans. Vehicular Tech.*, vol. 65, no. 3, pp. 1721–1734, March 2016.
- [85] C. A. Charalambides, *Enumerative Combinatorics*. Cambridge: Cambridge University Press, 1997.
- [86] G. Bi and Y. Zeng, *Transforms and Fast Algorithms for Signal Analysis and Representations*. New York: Springer, 2004.
- [87] B. C. Rennie and A. J. Dobson, “On stirling numbers of the second kind,” *J. Combinatorial Theory*, vol. 7, no. 2, pp. 116–121, Sept. 1969.

- [88] P. Ribenboim, *The Little Book of Bigger Primes*. New York: Springer, 2004.
- [89] J. M. Muller, *Elementary Functions: Algorithms and Implementation*. New York: Springer, 1997.
- [90] S. Zoppi, H. M. Grsu, M. Vilgelm, and W. Kellerer, “Reliable hopping sequence design for highly interfered wireless sensor networks,” in *Proc. IEEE Int. Symp. Local Metropolitan Area Networks (LANMAN)*, Japan, June 2017, pp. 1–7.
- [91] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Tholeyre, “Is local blacklisting relevant in slow channel hopping low-power wireless networks?” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, May 2017, pp. 1–6.
- [92] S. Linder, K. Wiklundh, and L. Junholm, “Performance of frequency hopping systems with adjacent channel interference,” in *Proc. IEEE Int. Symp. on Electromagnetic Compatibility - EMC EUROPE*, Poland, Sept. 2016, pp. 376–381.
- [93] N. Saquib, E. Hossain, and D. I. Kim, “Fractional frequency reuse for interference management in LTE-Advanced hetnets,” *IEEE Wireless Communications*, vol. 20, no. 2, pp. 113–122, April 2013.
- [94] C. D. Lee, Y. P. Huang, Y. Chang, and H. H. Chang, “Perfect Gaussian integer sequences of odd period $2^m - 1$,” *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 881–885, July 2015.
- [95] J. Sachs, R. Herrmann, M. Kmec, M. Helbig, and K. Schilling, “Recent advances and applications of m-sequence based ultra-wideband sensors,” in *IEEE Int. Conf. on Ultra-Wideband*, Singapore, Sept. 2007, pp. 50–55.
- [96] Y. Zhou, K. Panetta, and S. Agaian, “An image scrambling algorithm using parameter bases m-sequences,” in *IEEE Int. Conf. on Machine Learning and Cybernetics*, vol. 7, China, July 2008, pp. 3695–3698.
- [97] F. J. MacWilliams and N. J. A. Sloane, “Pseudo-random sequences and arrays,” *IEEE Proceedings*, vol. 64, no. 12, pp. 1715–1729, Dec. 1976.
- [98] G. L. Mullen and D. Panario, *Handbook of Finite Fields*. CRC Press, 2013.
- [99] N. Xiang, “Using m-sequences for determining the impulse responses of LTI systems,” *Elsevier Signal Processing*, vol. 28, no. 2, pp. 139–152, Aug. 1992.
- [100] N. M. Balasubramanya, L. Lampe, G. Vos, and S. Bennett, “On timing reacquisition and enhanced primary synchronization signal (ePSS) design for energy efficient 3GPP LTE MTC,” *IEEE Trans. on Mobile Computing*, vol. 16, no. 8, pp. 2292–2305, Aug. 2017.

- [101] M. Mansour, “Optimized architecture for computing Zadoff-Chu sequences with application to LTE,” in *Proc. IEEE Glob. Commun. Conf.*, Hawaii, Nov. 2009, pp. 1–6.
- [102] 3GPP TS 36.211 version 12.3.0, *Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (Release 12)*. ETSI TS, Oct. 2014, available at: <http://www.itu.int/pub/R-REP-M.2135-1-2009>.