# ABSTRACT

## CLOUD-AIDED WIRELESS SYSTEMS: COMMUNICATIONS AND RADAR APPLICATIONS

**by**
**Shahrouz Khalili**

This dissertation focuses on cloud-assisted radio technologies for communication, including mobile cloud computing and Cloud Radio Access Network (C-RAN), and for radar systems.

This dissertation first concentrates on cloud-aided communications. Mobile cloud computing, which allows mobile users to run computationally heavy applications on battery limited devices, such as cell phones, is considered initially. Mobile cloud computing enables the offloading of computation-intensive applications from a mobile device to a cloud processor via a wireless interface. The interplay between offloading decisions at the application layer and physical-layer parameters, which determine the energy and latency associated with the mobile-cloud communication, motivates the inter-layer optimization of fine-grained task offloading across both layers. This problem is modeled by using application call graphs, and the joint optimization of application-layer and physical-layer parameters is carried out via a message passing algorithm by minimizing the total energy expenditure of the mobile user.

The concept of cloud radio is also being considered for the development of two cellular architectures known as Distributed RAN (D-RAN) and C-RAN, whereby the baseband processing of base stations is carried out in a remote Baseband Processing Unit (BBU). These architectures can reduce the capital and operating expenses of dense deployments at the cost of increasing the communication latency. The effect of this latency, which is due to the fronthaul transmission between the Remote Radio Head (RRH) and the BBU, is then studied for implementation of Hybrid Automatic

Repeat Request (HARQ) protocols. Specifically, two novel solutions are proposed, which are based on the control-data separation architecture. The trade-offs involving resources such as the number of transmitting and receiving antennas, transmission power and the blocklength of the transmitted codeword, and the performance of the proposed solutions is investigated in analysis and numerical results.

The detection of a target in radar systems requires processing of the signal that is received by the sensors. Similar to cloud radio access networks in communications, this processing of the signals can be carried out in a remote Fusion Center (FC) that is connected to all sensors via limited-capacity fronthaul links. The last part of this dissertation is dedicated to exploring the application of cloud radio to radar systems. In particular, the problem of maximizing the detection performance at the FC jointly over the code vector used by the transmitting antenna and over the statistics of the noise introduced by quantization at the sensors for fronthaul transmission is investigated by adopting the information-theoretic criterion of the Bhattacharyya distance and information-theoretic bounds on the quantization rate.

# CLOUD-AIDED WIRELESS SYSTEMS: COMMUNICATIONS AND RADAR APPLICATIONS

by
Shahrouz Khalili

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

May 2016

## APPROVAL PAGE

## CLOUD-AIDED WIRELESS SYSTEMS: COMMUNICATIONS AND RADAR APPLICATIONS

**Shahrouz Khalili**

Dr. Osvaldo Simeone, Dissertation Advisor Date
Associate Professor of Electrical and Computer Engineering, NJIT

Prof. Alexander Haimovich, Committee Member Date
Distinguished Professor of Electrical and Computer Engineering, NJIT

Dr. Joerg Kliewer, Committee Member Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Ali Abdi, Committee Member Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Onur Sahin, Committee Member Date
InterDigital Inc.

## BIOGRAPHICAL SKETCH

**Author:** Shahrouz Khalili

**Degree:** Doctor of Philosophy

**Date:** May 2016

**Undergraduate and Graduate Education:**

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2016.
- Master of Science in Electrical Engineering,
  Shiraz University, Shiraz, Iran, 2012
- Bachelor of Science in Electrical Engineering,
  Shiraz University, Shiraz, Iran, 2009

**Major:** Electrical Engineering

**Presentations and Publications:**

S. Khalili, O. Simeone, A.M Haimovich, "Cloud Radio-Multistatic Radar: Joint Optimization of Code Vector and Backhaul Quantization," *IEEE Signal Processing Letters*, vol. 22, no. 4, pages 494-498, April 2015.

S. Khalili, O. Simeone, "Inter-Layer Per-Mobile Optimization of Cloud Mobile Computing: A Message-Passing Approach," to be published in *Transactions on Emerging Telecommunications Technologies (ETT)*, January 2016.

S. Jeong, S. Khalili, O. Simeone, A.M. Haimovich and J. Kang, "Multistatic Cloud Radar Systems: Joint Waveform and Backhaul Optimization," to be published in *Transactions on Emerging Telecommunications Technologies (ETT)*, January 2016.

S. Khalili, O. Simeone, "Control-Data Separation in Cloud RAN: The Case of Uplink HARQ," in Proc. *IEEE Information Theory and Applications (ITA)*, San Diego, CA, January, 2016.

S. Khalili, O. Simeone, "Uplink HARQ for Distributed and Cloud RAN via Separation of Control and Data Planes," tsubmitted to *IEEE Transactions on Vehicular Technology*, 2015.

S. Khalili, J. Feng, O. Simeone, J. Tang, Z. Wen, A.M. Haimovich, and M. Zhou, “Code-Aided Channel Tracking and Decoding over Sparse Fast-Fading Multipath Channels with Application to Train Backbone Networks,” submitted to *IEEE Transactions on Intelligent Transportation Systems*, 2015.

*Dedicated to my parents whom I owe all of my success and achievements. Without their help and support, I could have not gone this far and none of this would be possible without their kind and useful guidance.*

# ACKNOWLEDGMENT

Hereby, I express my gratitude and deepest appreciation to my adviser, Dr. Osvaldo Simeone for his wisdom, commitment and guidance during my PhD program. I also appreciate the time he dedicated for my research and being available during his vacation even though he is heavily involved with so many activities, projects.

Specifically, I want to thank Prof. Alexander Haimovich for his guidelines, countless hours of revisions, advice on my research, and I appreciate his willingness to work on a tight schedule.

I would also like to thank my committee members and my sincere gratitude to Dr. Joerg Kliewer, Dr. Ali Abdi, and Dr. Onur Sahin; and I appreciate their time as well as their encouraging and constructive comments, feedbacks and guides on my dissertation.

Ms. Kathleen Bosco and Ms. Angela Retino deserve a very special acknowledgment from all of students at CWCSPR. They were always ready to help us and they have made everything easy.

Further thanks go to Ms. Clarisa Gonzalez-Lenahan, the staff of the Graduate Studies office of NJIT and the staff of the Office of Global Initiatives and faculty for their advice, help and support with administrative matters during my PhD studies.

At the end, I want to thank my family, and specifically my parents, Shahryar Khalili and Nahid Nemati, whom I owe all of my achievement to. Without them, none of these would be possible and hereby, I would like to express my deepest gratitude and appreciation for all they have done for me.

## TABLE OF CONTENTS

**Chapter** **Page**

## TABLE OF CONTENTS
## (Continued)

**Chapter** **Page**

## TABLE OF CONTENTS (Continued)

**Chapter** **Page**

# LIST OF FIGURES

## LIST OF FIGURES
## (Continued)

**Figure** **Page**

## LIST OF FIGURES
## (Continued)

**Figure** **Page**

# LIST OF TABLES

# CHAPTER 1

# MOTIVATION AND OVERVIEW

The current trend in wireless communication traffic suggests that the data traffic volume will be 1000 larger in the following 10 years [25]. Moreover, beside handling the additional traffic, the next wireless standard, i.e., 5G, should be capable of supporting low-latency communication and massive number of devices. The current consensus is that this can be achieved by means of an architectural transformation of wireless network that includes ultra dense deployments, massive MIMO, mm-wave transmission, and cloud-aided solutions such as mobile cloud computing and Distributed Radio Access Network (D-RAN) and Cloud RAN (C-RAN) [11, 64, 81]. This dissertation focuses on the cloud-aided radio techniques with application to both communication and radar systems.

Cloud computing refers to a network of remote servers, tipically hosted on the Internet, which can store, manage and process data. The idea of "cloud computing" was conceived in 1960s by Joseph Licklider in the Advanced Research Projects Agency Network (ARPANET) project, which was an early packet switching network. In the 1990s, telecommunication companies that offered Virtual Private Network (VPN) services used the term "cloud" to fix the boundary between what the provider was responsible for and what users were responsible for. More recently, cloud computing has extended this boundary to include in the "cloud" servers as well as the network infrastructure. In 2008, NASA's OpenNebula was the first software that used clouds [70] and later, in 2011 to 2012, IBM and Oracle announced their own cloud framework [2].

With the current widespread use of smart phones, there is an increasing demand on the users' part for applications that require heavy computations to be

**Figure 1.1** An example of a cloud server that is connected to difference devices to provide them a cloud storage or cloud computing service[2].

run on battery-powered mobile devices, such as video processing, gaming, automatic translation, object recognition and medical monitoring. Offloading energy-consuming tasks from a mobile device to a cloud server – known in the literature as cyber foraging, computation offloading [46] and, more commonly, cloud mobile computing [26] – provides a viable solution to this problem, as attested to by systems such as Google Voice Search, Apple Siri and Shazam and by implementations such as MAUI [23] and ThinkAir [45].

Cloud mobile computing combines the idea of cloud computing, mobile computing and wireless networks to provide mobile users with strong computational resources. This approach is based on sending and receiving information to and from the cloud using uplink and downlink transmissions. These transmissions entail energy consumption and latency which may neutralize the potential gains of offloading. To overcome this problem, in Chapter 2, an inter-layer optimization approach is advocated that encompasses the physical layer, via power allocation, and the application layer, via code partitioning. The joint optimization of physical

[2] Source: http://www.gadgetreview.com/cloud-storage-vs-cloud-computing-which-are-you-using (accessed on March 2016).

layer and the application layer parameters is obtained for the first time for serial and parallel implementations by means of a low-complexity message passing algorithm. Furthermore, the advantages of parallel implementations which allows for the pipelining of communication and computation is also investigated. The material in this chapter has been reported in the document:

- S. Khalili, O. Simeone, “Inter-Layer Per-Mobile Optimization of Cloud Mobile Computing: A Message-Passing Approach,” to be published in *Transactions on Emerging Telecommunications Technologies (ETT)*, January 2016.

As discussed the increase in the traffic load can be accommodated by means of ultra dense networks [5], which require the deployment of more Base Stations (BTS). The cost to build and operate such as infrastructure, as well as severe interference potentially created by concurrent transmissions make a traditional cellular architecture inefficient in this context. In Chapter 3 of this dissertation, D-RAN and C-RAN architectures are studied which attempt to overcome the aforementioned limitations. C-RAN was first introduced by China Mobile Research Institute in 2010 [56] with the aim of reducing the cost of a BTS by centralizing the BBU in a remote location. D-RAN is a variation of C-RAN in which the centralized BBU of each base station is separate.

In D-RAN and C-RAN, the BBU is virtualized at a “cloud” processor. This virtualization yields the separation between the remote radio head (RRH) that implements the radio functionalities of the base station and a centralized BBU that is charged with higher-layer tasks, including the physical layer. The centralization of control and data processing generally increases the latency due to the fronthaul transmission between the BBU and RRH. This latency may significantly affect the operation of HARQ scheme. HARQ is a crucial part of wireless systems that is responsible for securing reliable transmission over fading channels.

In Chapter 3, two novel ideas are proposed to tackle the extra latency introduced due to centralization of control and data for D-RAN and C-RAN, which are based on the separation of control and data planes. The key idea is based on the *separation of control and data* planes, in which retransmission control decisions are made at the edge of the network, that is, by the RRHs or User Equipments (UEs), while data decoding is carried out remotely at the BBUs. This architecture enables *low-latency local retransmission decisions* to be made at the RRHs or UEs, which are not subject to the fronthaul latency constraints, while at the same time leveraging the decoding capability of the BBUs. Moreover, the effect of different parameters on the performance of wireless network is investigated and closed form equations are derived to evaluate the performance of the considered system under different HARQ schemes such as Chase Combining HARQ and Incremental redundancy.

The material in this chapter has been reported in the document:

- S. Khalili, O. Simeone, "Uplink HARQ for Distributed and Cloud RAN via Separation of Control and Data Planes," submitted to *IEEE Transactions on Vehicular Technology*, 2015.

and

- S. Khalili, O. Simeone, "Control-Data Separation in Cloud RAN: The Case of Uplink HARQ," in *Proc. of Information Theory and Applications Workshop (ITA)*, San Diego, USA 2016.

In Chapter 4, the application of the cloud in radar systems is studied. A multistatic radar set-up with distributed receive sensors (also known as receive antennas) that are connected to a Fusion Center (FC) via limited-capacity backhaul links resembles a cloud radio access network in communication systems. Receive sensors measure signals sent by a transmit element and reflected from a target, and possibly clutter, in the presence of interference and noise. The receive sensors communicate over non-ideal backhaul links with the fusion center, or cloud processor, where the presence or absence of the target is determined.

Waveform design has been a topic of great interest to radar designers, [13], [69], [48]. For the problem of signal detection, the shape of the transmitted waveform may greatly affect detection performance when the radar operates in a clutter environment in which detection is subject to signal-dependent interference. The optimal waveform in the Neyman-Pearson (NP) sense has been studied for monostatic radars [21] [38]. Existing waveform design techniques such as those discussed in [39] [59], assume infinite-capacity links between a set of distributed radar elements and a FC that performs target detection. In scenarios in which the receive antennas are distributed over a large geographical area to capture a target's spatial diversity [31] and no wired backhaul infrastructure is in place, this assumption should be revised.

In order to cope with the capacity limitations of the backhaul links, inspired by the cloud radio access architecture in cellular communication systems [56], in Chapter 4, it is assumed that the receive sensors quantize the received baseband signal prior to the transmission to the FC. Hence, the FC operates on the quantized received baseband signals. This system is known as Cloud Radio-Multistatic Radar (CR-MR). The problem of jointly optimizing over the code vector and over the operation of the quantizers at the receive antennas is formulated and tackled by adopting information-theoretic criteria. The Bhattacharyya distance is used to evaluate the detection performance at the FC. The proposed joint optimization is addressed via a Block Coordinate Descent (BCD) method coupled with Majorization-Minimization (MM). Numerical results demonstrate the advantages of the proposed joint optimization approach over more conventional solutions that perform separate optimization.

The material in this chapter has been reported in the document:

- S. Jeong, S. Khalili, O. Simeone, A.M. Haimovich and J. Kang, "Multistatic Cloud Radar Systems: Joint Waveform and Backhaul Optimization," *Transactions on Emerging Telecommunications Technologies (ETT)*, January 2016.

and

- S. Khalili, O. Simeone, A.M Haimovich, "Cloud Radio-Multistatic Radar: Joint Optimization of Code Vector and Backhaul Quantization," *IEEE Signal Processing Letters*, vol. 22, no. 4, pages 494-498, April 2015.

# CHAPTER 2

# INTER-LAYER PER-MOBILE OPTIMIZATION OF CLOUD MOBILE COMPUTING: A MESSAGE-PASSING APPROACH

Cloud mobile computing enables the offloading of computation-intensive applications from a mobile device to a cloud processor via a wireless interface. In light of the strong interplay between offloading decisions at the application layer and physical-layer parameters, which determine the energy and latency associated with the mobile-cloud communication, this chapter investigates the inter-layer optimization of fine-grained task offloading across both layers. Algorithmic solutions are proposed that leverage the structure of the call graphs of typical applications by means of message passing on the call graph, under both serial and parallel implementations of processing and communication. For call trees, the proposed solutions have a linear complexity in the number of tasks, and efficient extensions are presented for more general call graphs that include ”map” and ”reduce”-type tasks. Moreover, the proposed schemes are optimal for the serial implementation, and provide principled heuristics for the parallel implementation. Extensive numerical results yield insights into the impact of inter-layer optimization and on the comparison of the two implementations.

## 2.1 Introduction

With the current widespread use of smart phones, there is an increasing demand on the users' part for applications that require heavy computations to be run on battery-powered mobile devices, such as video processing, gaming, automatic translation, object recognition and medical monitoring. Offloading energy-consuming tasks from a mobile device to a cloud server – known in the literature as cyber foraging,

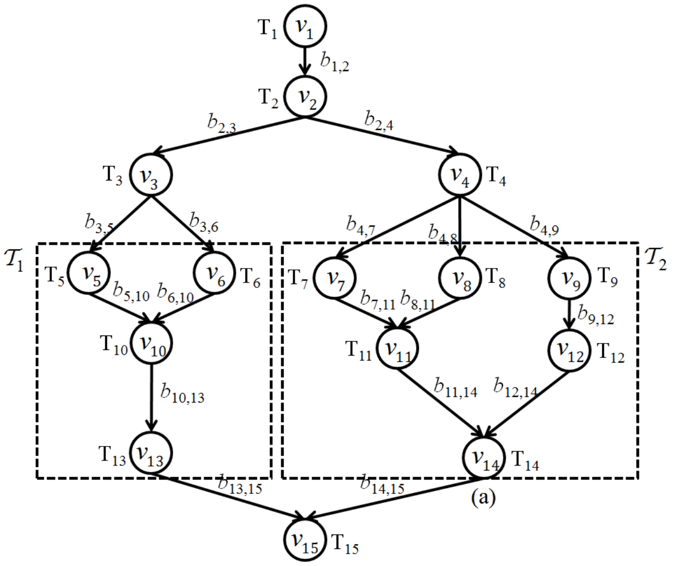


**Figure 2.1** An example of a call graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that is adopted from [37].

computation offloading [46] and, more commonly, cloud mobile computing [26] – provides a viable solution to this problem, as attested to by systems such as Google Voice Search, Apple Siri and Shazam and by implementations such as MAUI [23] and ThinkAir [45]. Moreover [87] proposes a blind scheduling in mobile media cloud to achieve fairness, simplicity and asymptotic optimality.

A mobile application can be partitioned into its component tasks via profiling, producing a *call graph* for the program [72]. The call graph describes the functional dependence between the different tasks (see Figure 2.1 for an example). Offloading can either take place at the coarser granularity of entire applications, as in, e.g., [75], or at the finer scale of individual tasks, see [23]. In the latter case, each task may be either offloaded to the cloud or performed locally. Moreover, processing and communication processes can either be implemented one after another in a serial fashion, as assumed in most prior art, or may be parallelized in the case of non-conflicting tasks as in [36] [37].

**State of the Art**: The large majority of prior works on the subject of optimal fine-grained offloading tackles the problem on a per-mobile basis, and assumes a *fixed physical layer*, which provides given information rate and latency. Examples of this approach for the serial implementation include [83], which uses a graph partitioning formulation; [66], which presents a heuristic on-line approach to task partitioning to improve latency; and [33] and [86], which assume a time-varying channel and propose adaptive solutions based on Lyapunov optimization and a constrained shortest path problem, respectively. Instead, for the parallel implementation, references [36] [37] propose a dynamic programming solution, again with a fixed physical layer.

While the assumption of a fixed physical layer made in all reviewed works simplifies the problem formulation, there is an evident interplay between decisions at the physical layer and offloading decisions at the application layer. Most fundamentally, the choice of the physical layer mode, e.g., of the transmission power and information rate, determines the mobile energy consumption, as well as the corresponding latency, for mobile-cloud communication. Therefore, a proper adaptation of the physical layer is instrumental in making cloud mobile computing viable.

Recognizing this critical interplay, more recent work has tackled the *inter-layer optimization of the physical and of the application layers.* Specifically, references [73] [74] studied this problem for a general network of interfering mobile devices by assuming *coarse-grained offloading.* Fine-grained offloading is instead studied in [49], where the authors focus on a per-mobile formulation under a serial implementation. To reduce the complexity of the resulting mixed integer program in [49], a method is proposed that limits the exponential number of alternative offloading decisions based on feasibility arguments. Furthermore, for fixed offloading decisions, the problem is shown to have useful convexity properties. A similar problem formulation is also studied in [52].

**Main Contributions**: In this chapter, the per-mobile inter-layer fine-grained optimization of offloading decisions at the application layer and of the transmission powers at the physical layer is investigated, with the aim of minimizing energy and latency for *both* serial and parallel implementations. As discussed, prior works, including [49] [52], formulate the problem as a mixed integer program, whose complexity is generally exponential in the size of the call graph for an arbitrary graph. However, it can be observed that most call graphs have specific structures that can be leveraged to reduce the computational complexity. For instance, Figure 2.1 shows a typical example of an application that is composed of "map" tasks, which perform operations such as filtering, features extraction or sorting, and allow the successive tasks to be decomposed into independent operations (see tasks $\mathrm{T}_2$, $\mathrm{T}_3$, $\mathrm{T}_4$); along with "reduce" tasks, which perform summary operations such as classification or regression (see tasks $\mathrm{T}_{10}$, $\mathrm{T}_{11}$ and $\mathrm{T}_{14}$) [47]. This chapter shows that, for structured graphs, solutions based on message passing can be developed for the both standard *serial* implementation, (see Section 2.4), as well as the *parallel* implementation (see Section 2.5).

In particular, for applications with a tree structure, such as the subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ in Figure 2.1, optimal efficient message passing algorithm for the serial implementation is developed, whose complexity is of the order $O(|\mathcal{V}|d_{in})$, where $|\mathcal{V}|$ is the number of nodes of the call graph and $d_{in}$ is the maximum in-degree. For the more challenging parallel implementation, the proposed method yields a principled suboptimal scheme whose complexity is of the same order as for the serial case. The performance of this scheme is evaluated by means of a dynamic model also introduced here.

For more general call graphs, such as the one in Figure 2.1, the proposed solutions can be generalized to yield a complexity of the order $O(2^{|\mathcal{V}_s|}|\mathcal{V}|d_{in})$, where $|\mathcal{V}_s|$ is the number of nodes that, if removed, decompose the graph into subtrees (such as $\mathrm{T}_2$, $\mathrm{T}_3$ and $\mathrm{T}_4$ in Figure 2.1, so that $|\mathcal{V}_s| = 3$ for this call graph). With reference to prior work, it should be noted that the proposed approach for parallel case generalizes the schemes in [36] and [37] by encompassing also the optimization of the physical layer.

Extensive simulation results, presented in Section 2.6, bring insight into the impact of inter-layer optimization and of the call graph structure on the performance of the cloud mobile computing.

*Notation*: Throughout this chapter, the graph terminology of, e.g., [44] is used. Accordingly, for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a node $a$ with an incoming edge from another node $b$ is referred to as a *child* of the *parent* node $b$. $\mathcal{P}(n)$ and $\mathcal{C}(n)$ are the sets containing parents and children, respectively, of a node $n \in \mathcal{V}$. Given a set $\mathcal{A} \subseteq \mathbb{N}$, where $\mathbb{N}$ is the set of integers and variables $X_i$ with $i \in \mathbb{N}$, $X_{\mathcal{A}}$ is the set defined as $X_{\mathcal{A}} = \{X_i | i \in \mathcal{A}\}$; similarly, for variables $X_{i,j}$ with $j \in \mathbb{N}$, $X_{\mathcal{A},j}$ is the set defined as $X_{\mathcal{A},j} = \{X_{i,j}, i \in \mathcal{A}\}$.

## 2.2 System Model

In this chapter, a per-mobile problem formulation is considered in which a mobile aims at running a given application with minimal energy expenditure and latency. For this purpose, the mobile may offload some of the computing tasks to a cloud processor, also referred to as server. A configuration with a single processor both at mobile and cloud is considered. This section starts by introducing the key quantities at the *application layer* and then at the *physical layer*.

### 2.2.1 Application Layer

A computer application can be described by its call graph [72]. A call graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a *directed acyclic graph* which is used to represent the casual relation among the tasks in which a program can be partitioned. An example is shown in Figure 2.1.

Each vertex, or node, in $\mathcal{V}$ represents a particular task to be carried out within the application, e.g., data preparation, edge recognition or transform coding. The task nodes are denoted as $\mathcal{V} = \{\mathrm{T}_1, ..., \mathrm{T}_{|\mathcal{V}|}\}$. However, the shortcut notation $n \in \mathcal{V}$ is also used in lieu of $\mathrm{T}_n \in \mathcal{V}$, where no confusion can arise. In the call graph $\mathcal{G}$, a directed edge $(\mathrm{T}_m, \mathrm{T}_n) \in \mathcal{E}$ with $\mathrm{T}_m \in \mathcal{V}$ and $\mathrm{T}_n \in \mathcal{V}$ denotes the invocation of a "child" task $\mathrm{T}_n$ by a "parent" task $\mathrm{T}_m$.

Each task node $\mathrm{T}_n$ is characterized by a parameter $v_n$, which is the number of CPU cycles required for task $\mathrm{T}_n$ to be completed. Let us define as $f^l$ and $f^r$ the number of CPU cycles/sec that can be run at the mobile (i.e., locally) and the cloud (i.e., remotely), respectively. The latency $L_n^l = v_n/f^l$ is then the time required to compute task $\mathrm{T}_n$ locally and $L_n^r = v_n/f^r$ is the latency to run that task remotely in the case the respective processors are devoted only to the completion of task $\mathrm{T}_n$. Each edge $(\mathrm{T}_m, \mathrm{T}_n) \in \mathcal{E}$ is instead labeled by the number of bits $b_{m,n}$ that must be transferred by the parent task $\mathrm{T}_m$ in order to allow the computation of the child task $\mathrm{T}_n$.

To complete the description of the quantities of interest at the application layer, we introduce the *offloading decision variables*. Specifically, we define $I_n \in \{0, 1\}$ as the indicator variable that determines whether task $\mathrm{T}_n$ should be executed locally or remotely, where $I_n = 0$ indicates the local execution of the task and $I_n = 1$ represents the offloading of the task to the remote server. Not all the tasks may be eligible for offloading. In particular, a mobile application typically operates on input data, e.g., images or videos, that reside in the mobile device. This can be accounted for by identifying a subset $\mathcal{V}_\mathrm{D} \subseteq \mathcal{V}$ of task nodes that represent input data preparation processes, such that for every task $\mathrm{T}_m \in \mathcal{V}_\mathrm{D}$ we have $I_m = 0$, i.e., local processing. These nodes are assumed to have no parents and have the role of initializing the application (see, e.g., [36] [37]). For instance, in Figure 2.1, we may have $\mathcal{V}_\mathrm{D} = \{\mathrm{T}_1\}$. Moreover, for any graph, we assume, without loss of generality, that there is a final task to be carried out at the mobile that has no children and completes the application by, e.g., showing the results on the mobile screen. An example is task $\mathrm{T}_{15}$ in Figure 2.1 for which we then have $I_{15} = 0$.

### 2.2.2 Physical Layer

We now describe the parameters and the optimization variables relative to the *physical layer*. The parameter $P^l$ represents the local processing power of the mobile and $P^{rf}$ is the power required to keep the mobile's RF circuits active during both transmission and reception, while $P^{rx}$ is the power needed to process the received baseband signal for decoding at the mobile. All powers are measured in Watts. The parameter $C^{dl}$ (bits/s) is the downlink capacity available to transfer the information bits from the server to the mobile. Uplink and downlink are assumed to be operated over orthogonal spectral resources.

The optimization variable $P^{ul}_{m,n}$ is the uplink power used by the mobile to transfer the necessary $b_{m,n}$ bits in case a parent task $\mathrm{T}_m$ is run locally ($I_m = 0$) and a child task $\mathrm{T}_n$ is performed remotely ($I_n = 1$) for all $(\mathrm{T}_m, \mathrm{T}_n) \in \mathcal{E}$. Note that we allow the uplink transmit powers $P^{ul}_{m,n}$ to be different for every edge in $\mathcal{E}$, hence enabling a more flexible joint optimization of application and physical layers as in [49]. Given an uplink power $P$, we denote as

$$C^{ul}(P) = B \log_2 \left(1 + \frac{\gamma P}{N_0 B}\right) \tag{2.1}$$

the uplink rate (bits/s) between the mobile and the server, where $\gamma$ accounts for the channel gain between mobile and the server, $B$ is the available bandwidth and $N_0$ (Watts/Hz) is noise power spectral density.

## 2.3 Problem Formulation

Here, we aim at optimizing the application layer variables $\mathbf{I} = \{I_n\}_{n=1}^{|\mathcal{V}|}$, with $I_n = 0$ for $n \in \mathcal{V}_{\mathrm{D}}$ and for the root node, and the physical layer variables $\mathbf{P} = \{P_{m,n}^{ul}\}_{(m,n)\in\mathcal{E}}$. We consider separately serial and parallel implementations.

### 2.3.1 Serial Implementation

In this section, as in most prior work, it is assumed that at any time, only one operation, either computation or communication, may take place, either at the mobile or at the server. Therefore, the operations needed to run a given application are performed in a serial fashion one after another. Note that the order in which these operations are scheduled is arbitrary as long as it is consistent with the procedures encoded in the call graph. For instance, for the tree $\mathcal{T}_1$ in Figure 2.1 if $I_5 = I_6 = I_{13} = 0$ and $I_{10} = 1$, tasks $\mathrm{T}_5$ and $\mathrm{T}_6$ can be first carried out in any order at the mobile; then, $b_{5,10}$ and $b_{6,10}$ bits are transferred in the uplink in any order; then, node $\mathrm{T}_{10}$ is processed at the cloud; and finally $b_{10,13}$ bits are downloaded by the mobile, which performers task $\mathrm{T}_{13}$.

Under a serial implementation, the overall latency is the sum of all the latencies required to communicate and compute across all task nodes, which can be written as (see also [49])

$$L(\mathbf{I},\mathbf{P}) = \sum_{n=1}^{|\mathcal{V}|} L_n^c(I_n) + \sum_{n=1}^{|\mathcal{V}|}\sum_{m\in\mathcal{P}(n)} L_{m,n}^{ul}(I_{\{m,n\}}, P_{m,n}^{ul}) + \sum_{n=1}^{|\mathcal{V}|}\sum_{m\in\mathcal{P}(n)} L_{m,n}^{dl}(I_{\{m,n\}}), \quad (2.2)$$

where $L_n^c(I_n) = (1 - I_n)L_n^l + I_n L_n^r$ denotes the delay required to perform the computations associated with task $\mathrm{T}_n$ either locally or remotely; $L_{m,n}^{ul}(I_{\{m,n\}}, P_{m,n}^{ul}) = I_n(1 - I_m)b_{m,n}/C^{ul}(P_{m,n}^{ul})$ accounts for the delay caused by the transfer of $b_{m,n}$

bits to the server if task $\mathrm{T}_n$ is offloaded ($I_n = 1$) but $\mathrm{T}_m$ is not ($I_m = 0$); $L^{dl}_{m,n}(I_{\{m,n\}}) = (1 - I_n)I_m b_{m,n}/C^{dl}$ represents the latency caused by the transfer of $b_{m,n}$ bits at the mobile if $\mathrm{T}_m$ is offloaded ($I_m = 1$) and $\mathrm{T}_n$ is run locally ($I_n = 0$).

The energy spent by the mobile for given variables is similarly given as the sum (see also [49])

$$E(\mathbf{I}, \mathbf{P}) = \sum_{n=1}^{|\mathcal{V}|} E^c_n(I_n) + \sum_{n=1}^{|\mathcal{V}|} \sum_{m \in \mathcal{P}(n)} E^{ul}_{m,n}(I_{\{m,n\}}, P^{ul}_{m,n}) + \sum_{n=1}^{|\mathcal{V}|} \sum_{m \in \mathcal{P}(n)} E^{dl}_{m,n}(I_{\{m,n\}}), \quad (2.3)$$

where the term $E^c_n(I_n) = (1 - I_n)P^l L^l_n$ measures the energy consumed by the mobile to perform each task $\mathrm{T}_n$ locally if $I_n = 0$; the term $E^{ul}_{m,n}(I_{\{m,n\}}, P^{ul}_{m,n}) = (P^{ul}_{m,n} + P^{rf})L^{ul}_{m,n}(I_{\{m,n\}}, P^{ul}_{m,n})$ is the energy required, for a task $\mathrm{T}_n$ with $I_n = 1$, to transfer information from all the parent tasks $m \in \mathcal{P}(n)$ that are performed locally, namely with $I_m = 0$; and finally $E^{dl}_{m,n}(I_{\{m,n\}}) = (P^{rf} + P^{rx})L^{dl}_{m,n}(I_{\{m,n\}})$ is the energy consumed, for a task $\mathrm{T}_n$ with $I_n = 0$, to transfer and decode the information in the downlink from parent tasks $m \in \mathcal{P}(n)$ with $I_m = 1$.

### 2.3.2 Parallel Operation

As an alternative to the serial operation discussed in Section 2.3.1, we now consider an implementation that allows to potentially reduce the latency by parallelizing computing and communication. This implementation was implicitly assumed in [36] [37] but without consideration for the optimization of the physical layer. According to this implementation, tasks are processed as soon as they receive the necessary information from their parents. It is then possible for uplink transmissions, downlink transmissions, local and remote computations to occur at the same time.

As an example, consider the call tree $\mathcal{T}_2$ in Figure 2.1 with $I_7 = I_8 = I_9 = I_{14} = 0$ and $I_{11} = I_{12} = 1$. An illustrative timeline is shown in Figure 2.2, where $\mathrm{CP}^{\mathrm{l}}$

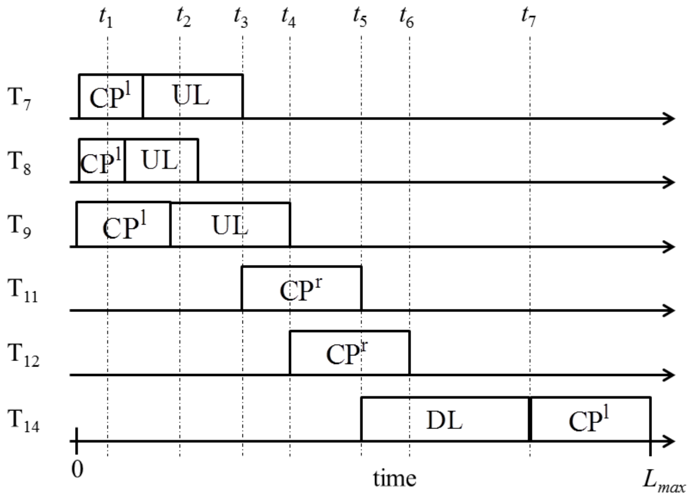


**Figure 2.2** An example of a timeline for the parallel implementation of the call tree $\mathcal{T}_2$ in Figure 2.1 with $I_7 = I_8 = I_9 = I_{14} = 0$ and $I_{11} = I_{12} = 1$.

denotes local computing and CP$^\mathrm{r}$ denotes remote computing; UL indicates that the task is uploading information bits in the uplink; and DL means that the task is receiving information from one or more of its parent task nodes in the downlink. It can be seen that, for instance, task T$_{11}$ can be processed remotely as soon as the information from tasks T$_7$ and T$_8$ has been received by the server at time $t_3$, while uplink transmission for task T$_9$ may be still ongoing. Observe that, whenever multiple concurrent uplink/downlink transfers take place at the same time, the uplink/downlink spectral resources have to be properly divided (e.g., for tasks T$_7$, T$_8$ and T$_9$ at time $t_1$). This requires an adequate allocation of the spectral resources, such as time-frequency resource blocks in LTE. An analogous discussion applies to the computational resources.

Assuming the feasibility of allocating communication and computation resources as discussed above, the Appendix B details a dynamic model that enables the evaluation of the energy and latency of the parallel implementation for given physical- and application-layer variables $\mathbf{P}$ and $\mathbf{I}$. This framework will be used in Section 3.7 to evaluate the performance of the parallel implementation using numerical results. However, the framework in the Appendix B does not lend itself to the development of efficient optimization algorithms due to the complexity of accounting for the mentioned reallocation of the communication and computation resources. In Section 2.5, useful heuristics are developed for this purpose.

### 2.3.3 Problem Formulation

In order to optimize physician and application layer variables, two different standard approaches are considered (see, e.g, [15]). In the first problem formulation, a weighted sum of energy and latency is minimized via the problem

$$[\text{P.1}] \quad \underset{\mathbf{I},\mathbf{P}}{\text{minimize}}\ E(\mathbf{I},\mathbf{P}) + \lambda L(\mathbf{I},\mathbf{P}), \tag{2.4}$$

where $\lambda$ is a non-negative constant that determines the trade-off between energy and latency. By varying $\lambda$, one can explore the trade-off between latency and energy [15]. An alternative problem formulation is to minimize the energy (2.3) with a latency constraint as

$$\begin{aligned} [\text{P.2}] \qquad & \underset{\mathbf{I},\mathbf{P}}{\text{minimize}}\ E(\mathbf{I},\mathbf{P}) \\ & \text{subject to } L(\mathbf{I},\mathbf{P}) \leq L_{max}, \end{aligned} \tag{2.5}$$

where $L_{max}$ is the maximum allowed delay. Note that, in (2.4) and (2.5), the domains of variables $\mathbf{I}$ and $\mathbf{P}$ are implicit. As it will be illustrated in the next sections, it is analytically convenient to tackle problem [P.1] for the serial implementation and problem [P.2] for the parallel implementation.

*Remark* 2.1. References [36] [37] tackled problem [P.2] for the parallel implementation under the assumption that the call graph is a tree or a parallel/serial combination of trees, and assuming that the physical-layer parameters $\mathbf{P}$ are not subject to optimization. Moreover, the papers [36] [37] implicitly assume that parallel communication and computation do not entail a division of the available resources, hence bypassing the issue discussed above. Under these assumptions, it is shown that the problem can be efficiently, albeit approximately, solved via dynamic programming by quantizing the set of possible delays. Reference [49] studied instead problem [P.2] for the serial implementation. The solution given in [49] prescribes a properly pruned exhaustive search over the variables $\mathbf{I}$, and leverages the fact that, for a fixed $\mathbf{I}$, the problem of optimization over $\mathbf{P}$, upon a proper change of variables, is convex.

## 2.4 Optimal Task Offloading for Serial Processing

In this section, the problem [P.1] is tackled for serial processing. The key idea of the proposed approach is to leverage the factorization of the objective function in [P.1] in order to apply the min-sum message passing algorithm. We first detail the mentioned factorization in Section 2.4.1. Then, in Section 2.4.2, the proposed efficient optimal method is discussed based on min-sum message passing [44] for the special case of a call tree. Then, in Section 2.4.3, the proposed algorithm is extended to call graphs with more general structure.

### 2.4.1 Factorization of the Cost Function

The objective function for problem [P.1] can be factorized over the task nodes as follows:

$$\sum_{n\in\mathcal{V}} \Phi_n \left( I_{\{n\}\cup\mathcal{P}(n)}, P^{ul}_{\mathcal{P}(n),n} \right), \tag{2.6}$$

where the factor $\Phi_n(I_{\{n\}\cup\mathcal{P}(n)}, P^{ul}_{\mathcal{P}(n),n})$ accounts for the weighted sum of energy and latency associated with the local or the remote computation of node $\mathrm{T}_n$ and with the transmissions in uplink and/or downlink related to the edges connecting the parents of node $\mathrm{T}_n$ to node $\mathrm{T}_n$. This function is given, from (2.2) and (2.3), as

$$\begin{aligned}\Phi_n\left(I_{\{n\}\cup\mathcal{P}(n)}, P^{ul}_{\mathcal{P}(n),n}\right) &= (1-I_n)P^l L^l_n + \lambda L^c_n(I_n) + \sum_{m\in\mathcal{P}(n)} (P^{ul}_{m,n} + P^{rf} + \lambda) L^{ul}_{m,n}(I_{\{m,n\}}, P^{ul}_{m,n}) \\ &+ \sum_{m\in\mathcal{P}(n)} (P^{rf} + P^{rx} + \lambda) L^{dl}_{m,n}(I_{\{m,n\}}).\end{aligned} \tag{2.7}$$

We now show that the optimization in [P.1] over the transmission powers $\mathbf{P}$ can be carried out analytically, yielding new factors that are independent of the powers. In fact, given that each power $P^{ul}_{m,n}$ appears separately in the factors of (2.6), the optimization of all powers can be carried out independently. In particular, the optimum power $\bar{P}^{ul}_{m,n}$ for all edges $(m,n)\in\mathcal{E}$ is given by the solution of the problem

$$\bar{P}^{ul}_{m,n} = \arg\min_{P^{ul}_{m,n}\geq 0} \frac{P^{ul}_{m,n} + P^{rf} + \lambda}{C^{ul}(P^{ul}_{m,n})}. \tag{2.8}$$

As discussed in [49], the optimization problem in (2.8) becomes strictly convex with the change of variables $y_{m,n} = C^{ul}(P^{ul}_{m,n})$ and hence its unique solution can be easily found[1]. Note that the optimum values $\bar{P}^{ul}_{m,n}$ for all $(m,n)\in\mathcal{E}$ are equal.

[1] This follows from the convexity of the function $(2^x + a)/x$ for $x > 0$ and any constant $a \geq 0$.

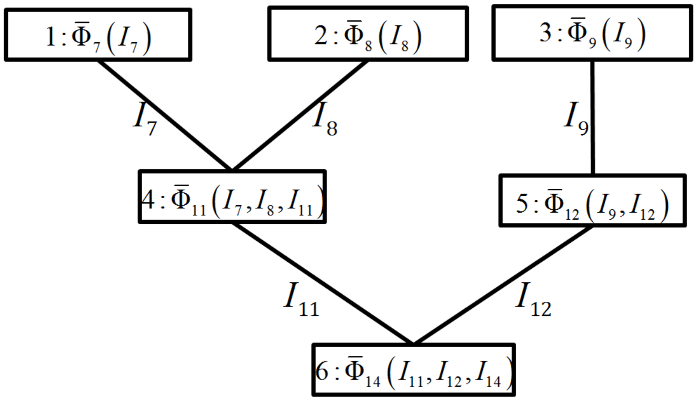

$1:\bar{\Phi}_7(I_7)$
$2:\bar{\Phi}_8(I_8)$
$3:\bar{\Phi}_9(I_9)$
$I_7$
$I_8$
$I_9$
$4:\bar{\Phi}_{11}(I_7,I_8,I_{11})$
$5:\bar{\Phi}_{12}(I_9,I_{12})$
$I_{11}$
$I_{12}$
$6:\bar{\Phi}_{14}(I_{11},I_{12},I_{14})$

**Figure 2.3** The clique tree $\mathcal{T}_c$ corresponding to the call tree $\mathcal{T}_2$ in Figure 2.1.

Substituting the optimum powers from (2.8) into (2.6), the problem [P.1] can be rewritten as

$$\text{[P.1]} \quad \underset{\mathbf{I}}{\text{minimize}} \sum_{n \in \mathcal{V}} \bar{\Phi}_n \left( I_{\{n\} \cup \mathcal{P}(n)} \right), \tag{2.9}$$

where the factors are defined as

$$\bar{\Phi}_n \left( I_{\{n\} \cup \mathcal{P}(n)} \right) = \Phi_n \left( I_{\{n\} \cup \mathcal{P}(n)}, \bar{P}^{ul}_{\mathcal{P}(n),n} \right). \tag{2.10}$$

### 2.4.2 Message Passing for a Call Tree

For a given call tree $\mathcal{T}$, as for $\mathcal{T}_1$ and $\mathcal{T}_2$ in Figure 2.1, the problem [P.1] in (2.9) can be solved *exactly* via the *min-sum message passing* algorithm with a complexity of the order $O(|\mathcal{V}|d_{in})$, where $d_{in}$ is the maximum in-degree in the call graph. We refer to [44] for an introduction to message passing algorithms.

The algorithm operates on a clique tree $\mathcal{T}_c$ that is associated with the call tree $\mathcal{T}$. The clique tree $\mathcal{T}_c$ can be constructed from $\mathcal{T}$ as follows: $(i)$ replace the directed edges in $\mathcal{T}$ with undirected ones; and $(ii)$ substitute each task node $\mathrm{T}_n$ in $\mathcal{T}$ with a node of $\mathcal{T}_c$, which is labeled as the $n$th cluster node. Each cluster node $n$ is assigned the factors $\bar{\Phi}_n\left(I_{\{n\}\cup\mathcal{P}(n)}\right)$ in (2.10). Each edge that connects clusters $n$ and $m$ is labeled with the variable $I_m$ that appears in both clusters $n$ and $m$. An example of a call tree and its corresponding clique tree is illustrated in Figure 2.3.

Once the clique tree is constructed, the min-sum message passing algorithm can be directly obtained following the standard rules as detailed in [44, Ch. 10] (see also Appendix A). To elaborate, we define $\{E^l(n), E^r(n)\}$ as the message sent by the $n$th cluster node on the edge labeled by $I_n$, to its child cluster, where $E^l(n)$ is the value of the message corresponding to $I_n = 0$ (local processing) and $E^r(n)$ is the value of the message for $I_n = 1$ (remote processing). Note that the definition of the parents and children nodes follows that used for the call tree $\mathcal{T}$. The messages of the clusters that are not leaves can be calculated recursively as

$$E^l(n) = \sum_{m\in\mathcal{P}(n)} \min\left\{E^l(m) + \bar{\Phi}_n\left(I_n = 0, I_m = 0\right),\ E^r(m) + \bar{\Phi}_n\left(I_n = 0, I_m = 1\right)\right\}, \tag{2.11}$$

and

$$E^r(n) = \sum_{m\in\mathcal{P}(n)} \min\Big\{E^l(m) + \bar{\Phi}_n\left(I_n = 1, I_m = 0\right), E^r(m) + \bar{\Phi}_n\left(I_n = 1, I_m = 1\right)\Big\}. \tag{2.12}$$

In order to keep track of the optimal decision $\mathbf{I}$, for each cluster $n$ and parent cluster $m$, we also define the functions $I^l_m(n)$ and $I^r_m(n)$, where $I^l_m(n) = 0$ if the first argument in the min operation in (2.11) is smaller and $I^l_m(n) = 1$ otherwise; and $I^r_m(n)$ is defined analogously with respect to (2.12).

**Table 2.1** Message Passing Algorithm for the Serial Implementation

```
1: Calculate the powers $P^{ul}_{m,n}$ for all
   $(m, n) \in \mathcal{E}$ using (2.8).
2: Build the corresponding clique tree as explained in
   Section 2.4.2 (see Figure 2.3).
3: for $n = 1{:}|\mathcal{V}|$ do
     if $n$ is a leaf cluster
       $E^l(n) = 0$
       $E^r(n) = \infty$
     else
       Update $E^l(n)$ and $E^r(n)$ by using (2.11) and (2.12)
       and calculate $I^l_m(n)$ and $I^r_m(n)$ for all $m \in \mathcal{P}(n)$
       as explained in Section 2.4.2.
4: Trace back the optimum decisions.
```

As detailed in Table 2.1, the messages are first sent by the leaf clusters, and then each cluster transmits its message $\{E^l(n), E^r(n)\}$ to its child cluster as soon as it has received the message from all its parents. The message passing algorithm is detailed in Table 2.1. The optimum decisions are finally obtained via *backtracking*, starting from the root node $\mathcal{V}$ so that for any node $n$ and every parent $m \in \mathcal{P}(n)$, we set $I_m = I^l_m(n)$ if $I_n = 0$ and $I_m = I^r_m(n)$ otherwise.

*Complexity and optimality*: The presented scheme is optimal due to the well known properties of min-sum message passing [44]. Furthermore, while the optimization of the powers is performed in (2.8), the final selected powers depend on the optimal offloading decisions identified during backtracking step. Finally, from (2.11) and (2.12), the complexity of serial implementation is of order $O(|\mathcal{V}|d_{in})$, since every node needs to sum at most $d_{in}$ metrics, each of which only requires two sums and a binary comparison.

### 2.4.3 Message Passing for a General Graph

In the case of a more general call graph $\mathcal{G}$, it is not possible to directly convert the call graph to a clique tree as done above for a call tree.

Two solutions to this problem is outlined here. First, assume that the call graph is such that by removing a small number subset $\mathcal{V}_{\mathrm{S}}$ of nodes, one can partition the graph into subtrees. This is the case for typical graphs, such as that in Figure 2.1, with a small number of "map" and "reduce" nodes (see Section 3.1). For such graphs, similar to the observation in [37], one can apply message passing scheme introduced above on each subtree for all possible instantiations of the offloading decisions for the mentioned fixed nodes. Then, the minimum value of the function in (2.9) is calculated over all such instantiations. The complexity of this approach is of the order $O(2^{|\mathcal{V}_s|}|\mathcal{V}|d_{in})$.

For graphs with an even more general structure, the junction tree algorithm can be applied to obtain a clique tree [44, Ch. 10]. Once the clique tree is obtained, message passing can be implemented by extending the approach described in the previous subsection. The complexity of this scheme depends on the treewidth of the graph [44]. In general, unless $|\mathcal{V}_{\mathrm{S}}|$ is prohibitively large, the previous approach is to be preferred due to the possibility to reuse efficient algorithm in Table 2.1.

## 2.5 Optimization of Task Offloading for Parallel Processing

In this section, the problem [P.2] is tackled in the presence of parallel processing. As for the serial case, we concentrate on call trees in Section 2.5.1, and in Section 2.5.2 the extensions to more general call graphs is discussed.

As explained in Section 2.3, in order to evaluate energy and latency of a parallel implementation, one needs to keep track of the number of concurrent processes that use the local and remote CPUs as well as the uplink and downlink bandwidth. While the dynamic model presented in the Appendix B is able to do so, its use for optimization appears to be challenging. Hence, in this section, in order to develop a useful optimization *heuristic*, the allocation of computation and communication resources is fixed among a given number of concurrent uploads, downloads, local computations and remote computations. Under this simplifying constraint, an algorithm is proposed that solves problem [P.2] to any arbitrary precision with linear complexity via message passing, and, specifically, via dynamic programming. The performance of the obtained heuristic solution is then evaluated by means of the dynamic model described in the Appendix B.

To elaborate, we fix the number of concurrent upload and download transmissions to $N^{ul}$ and $N^{dl}$, respectively, and, the number of concurrently computed tasks locally or remotely as $N^{l}$ and $N^{r}$, respectively. This is done in order to constrain the available uplink and downlink capacities as

$$C^{ul}_{par}(P^{ul}_{m,n}) = \frac{C^{ul}(N^{ul}P^{ul}_{m,n})}{N^{ul}} \tag{2.13a}$$

$$\text{and } C^{dl}_{par} = \frac{\log_2\left(1+(2^{C^{dl}}-1)N^{dl}\right)}{N^{dl}}, \tag{2.13b}$$

which correspond to the rates achievable when the spectral resources, either in the time or in the frequency, are equally divided into $N^{ul}$ and $N^{dl}$ parts, respectively. Similarly, the frequency of the local and the remote processors can be obtained by

$$f^{l}_{par} = \frac{f^{l}}{N^{l}} \text{ and } f^{r}_{par} = \frac{f^{r}}{N^{r}}. \tag{2.14}$$

The fixed values of $N^{ul}$, $N^{dl}$, $N^{l}$ and $N^{r}$ define parameters that can be set by the designer, yielding different optimization solutions that can be evaluated via the dynamic model in the Appendix B. More discussion on the selection of these parameters can be found in Section 3.7.

Following [36], it can be observed that, for each task $\mathrm{T}_n$, the delay required to complete the tasks of the subtree in $\mathcal{G}$ rooted at any task node $\mathrm{T}_n$ can be calculated recursively, given that the completion of task $\mathrm{T}_n$ requires completion of all the parent tasks. Specifically the time $L_{par}^{(n)}(\mathbf{I}, \mathbf{P})$ by which the subtree rooted at $\mathrm{T}_n$ is completed, given the decisions $(\mathbf{I}, \mathbf{P})$, can be written in terms of the same quantities for its parents as

$$L_{par}^{(n)}(\mathbf{I}, \mathbf{P}) = \max_{m \in \mathcal{P}(n)} \left\{ L_{par}^{(m)}(\mathbf{I}, \mathbf{P}) + L_{m,n}^{ul}(I_{\{m,n\}}, P_{m,n}^{ul}) + L_{m,n}^{dl}(I_{\{m,n\}}) \right\} + L_n^c(I_n), \quad (2.15)$$

where the $L_{par}^{(m)}(\mathbf{I}, \mathbf{P})$ is the latency of the subtree rooted at the parent node $\mathrm{T}_m$ and the latency terms are defined as in (2.2). Note that since $I_n = 0$ for the leaf nodes in $\mathcal{V} - \mathrm{D}$, we have $L_{par}^{(n)}(\mathbf{I}, \mathbf{P}) = 0$ for $n \in \mathcal{V}_{\mathrm{D}}$. The expression (2.15) can be then calculated recursively starting from the leaf nodes, and the final delay is given by $L_{par}(\mathbf{I}, \mathbf{P}) = L_{par}^{(|\mathcal{V}|)}(\mathbf{I}, \mathbf{P})$.

### 2.5.1 Message Passing for a Call Tree

We aim at developing an approximate solution to problem [P.2] under the constraints that the communication and computation resources are allocated as in (2.13)-(2.14). To this end, as in [36], the set of possible delays is partitioned into $K$ intervals by means of the quantization function

$$q(t) = t_k \quad \text{if } t \in (t_{k-1}, t_k], \quad (2.16)$$

where $0 \leq t_1 \leq t_2 \leq ... \leq t_K = L_{max}$ are given predefined latency values. For simplicity $t_k$ is set as $t_k = (k-1)\epsilon$ for a given quantization step $\epsilon > 0$. The algorithm presented below provides an approximation of the optimal solution of the constrained program at hand, which, following the same arguments as in [36] [37], become increasingly accurate as $\epsilon$ becomes smaller.

$\mathcal{T}_n$ is defined as the subtree $\mathcal{G}$ that is rooted at the task $\mathrm{T}_n$. Moreover, let $E^l(n,k)$ denote the minimum energy needed to run the the tasks in $\mathcal{T}_n$ if node $\mathrm{T}_n$ is executed locally and under the constraint that the latency is less than $t_k$. Note that the energy $E^l(n,k)$ is minimized with respect to the offloading variables in vector $\mathbf{I}$ corresponding to the task nodes in the mentioned subtree except $\mathrm{T}_n$, as well as over the uplink powers in vector $\mathbf{P}$ corresponding to all the edges within the subtree. Similarly, $E^r(n,k)$ is defined as the minimum energy cost for $\mathcal{T}_n$ if $\mathrm{T}_n$ is performed remotely and under the delay constraint $t_k$. We also correspondingly define the set $\mathcal{I}^l(n,k) = \{I^l_m(n,k)\}_{m\in\mathcal{P}(n)}$ that contains the optimum offloading decisions for the parent nodes $\mathrm{T}_m$ of node $\mathrm{T}_n$ if the latter is performed locally under the latency $t_k$ for the subtree rooted at $\mathrm{T}_n$. Similarly, $\mathcal{I}^r(n,k) = \{I^r_m(n,k)\}_{m\in\mathcal{P}(n)}$ is defined as the set containing the optimum decisions for the parent nodes $\mathrm{T}_m$ of node $\mathrm{T}_n$, if the latter is performed remotely with the latency constraint $t_k$.

The proposed dynamic programming algorithm computes the cost functions $E^l(n,k)$ and $E^r(n,k)$ and the sets $\mathcal{I}^l(n,k)$ and $\mathcal{I}^r(n,k)$ recursively from the energy cost functions $E^l(m,j)$ and $E^r(m,j)$ of all the parent nodes $m \in \mathcal{P}(n)$ under all the delay constraints $t_j$ with $j = 1,...,k-1$. Specifically, we set $E^l(n,k) = \infty$ and $E^r(n,k) = \infty$ for $k \leq 0$. The recursive relationship can be obtained as

$$\begin{aligned} E^l(n,k) = P^l L^l_n + \sum_{m\in\mathcal{P}(n)} \min\Big\{ E^l\Big(m, k - Q(L^l_n)\Big), \\ E^r\left(m, k - Q\left(L^l_n + \frac{b_{m,n}}{C^{dl}_{par}}\right)\right) + (P^{rf} + P^{rx})\frac{b_{m,n}}{C^{dl}_{par}} \Big\}, \end{aligned} \tag{2.17}$$

where the function $Q$ is defined as $Q(t) = k$ if $t \in [t_{k-1}, t_k)$ for all $k \in \{1,...,K\}$.

Equation (2.17) accounts for the fact that the minimum energy cost required to run the task in the subtree $\mathcal{T}_n$ within a latency $t_k$ if $\mathrm{T}_n$ is run locally is given by the sum of the local processing energy $P^l L_n^l$ (see $E_n^c(I_n)$ in (2.3)) and of the energies required to run all the subtrees $\mathcal{T}_m$ with $m \in \mathcal{P}(n)$. For the latter, each parent node $\mathrm{T}_m$ can be run either locally, requiring energy $E^l(m, k - Q(L_n^l))$, or remotely, with an energy $E^r(m, k - Q(L_n^l + \frac{b_{m,n}}{C_{par}^{dl}}))$. It is observed that, if node $\mathrm{T}_m$ is performed locally, the latency allowed for the subtree $\mathcal{T}_m$ is $t_k - q(L_n^l)$ and hence the corresponding minimum energy is $E^l(m, k - Q(L_n^l))$, and similarly for the case in which $\mathcal{T}_m$ is carried out remotely the energy can be calculated as in (2.17). In (2.17), the $\min\{\cdot,\cdot\}$ operation accounts for the choice of whether node $\mathrm{T}_n$ should be performed locally or remotely. Accordingly, the set $\mathcal{I}^l(n,k) = \{I_m^l(n,k)\}_{m\in\mathcal{P}(n)}$ can be evaluated during calculation of $E^l(n,k)$ in (2.17) by observing which term in the function $\min\{\cdot,\cdot\}$ is smaller. Specifically, we can write $I_m^l(n,k) = 0$ if the first term is smaller and $I_m^l(n,k) = 1$ otherwise.

Similar to (2.17), we can also write

$$\begin{aligned} E^r(n,k) = \sum_{m\in\mathcal{P}(n)} \min\Bigg\{ \Bigg( (\bar{P}^{ul}_{m,n,k} + P^{rf}) \frac{b_{m,n}}{C^{ul}_{par}(\bar{P}^{ul}_{m,n,k})} \\ + E^l\left(m, k - Q\left(L_n^r + \frac{b_{m,n}}{C^{ul}_{par}(\bar{P}^{ul}_{m,n,k})}\right)\right)\Bigg), E^r\Big(m, k - Q(L_n^r)\Big)\Bigg\}, \end{aligned} \tag{2.18}$$

where uplink $\bar{P}^{ul}_{m,n,k}$ is selected as detailed below. The two arguments of the $\min\{\cdot,\cdot\}$ operator measures the energy cost of the subtree $\mathcal{T}_m$ in the case that the parent node $\mathrm{T}_m$ is performed locally or remotely, respectively, and are explained in an analogous fashion as for (2.17). Furthermore, the set $\mathcal{I}^r(n,k) = \{I_m^r(n,k)\}_{m\in\mathcal{P}(n)}$ can be evaluated during calculation of $E^r(n,k)$ in analogous fashion as $I_m^l(n,k)$.

Once equations (2.17)-(2.18) are evaluated starting from the leaf nodes of $\mathcal{G}$ to the root, the optimum powers $\mathbf{P}$ and offloading decisions $\mathbf{I}$ are obtained via

**Table 2.2** Dynamic Programming Solution for Parallel Implementation

```
for $n = 1{:}|\mathcal{V}|$ do
    if $\mathrm{T}_n \in \mathcal{V}_{\mathrm{D}}$
      $E^l(n,k) = \; 0$ for all $k$
      $E^r(n,k) = \; \infty$ for all $k$
    else
      for $k = 1,\, K$ do
        Calculate the powers $\bar{P}^{ul}_{m,n,k}$ for all $(m,n) \in \mathcal{E}$
        using (2.19).
        Update $E^l(n,k)$, $E^r(n,k)$, $\mathcal{I}^l(n,k)$ and $\mathcal{I}^r(n,k)$
        by using (2.17)-(2.18).
Trace back the optimum decisions from $E^l(|\mathcal{V}|,k)$
  using the algorithm in Table 2.3.
```

backtracking from the root to the leaves of $\mathcal{G}$. Specifically, since the root node must be performed locally within the delay constraint $L_{max}$, the optimum solution ($\mathbf{I}$,$\mathbf{P}$) can be found starting from the optimal decisions associated with $E^l(|\mathcal{V}|, L_{max})$ by keeping track of the maximum allowed delay $t_n$ for each subtree $\mathcal{T}_n$. The complete dynamic complete programming algorithm is presented in Table 2.2 and the backtracking method is explained in Table 2.3.

Optimization of the powers is carried out by observing that, thanks to the decomposition made possible by dynamic programming, the powers $P^{ul}_{m,n,k}$ appear in separate terms in (2.18). Therefore, without loss of optimality, the powers $P^{ul}_{m,n,k}$ can be optimized separately from each term in (2.18). This optimization is complicated by the presence of the non-differentiable term $Q(L^r_n + \frac{b_{m,n}}{C^{ul}_{par}(\bar{P}^{ul}_{m,n,k})})$. To address this issue, for each $(m,n) \in \mathcal{E}$ and each $k \in \{1, ..., K\}$ we calculate

$$\bar{P}^{ul}_{m,n,k} = \arg \min_{P^{ul}_{m,n} \geq 0} E^r(n,k,P^{ul}_{m,n}), \tag{2.19}$$

where

$$E^r(n, k, P^{ul}_{m,n}) \triangleq (P^{ul}_{m,n} + P^{rf}) \frac{b_{m,n}}{C^{ul}_{par}(P^{ul}_{m,n})} + E^l \left( m, k - Q \left( L^r_n + \frac{b_{m,n}}{C^{ul}_{par}(P^{ul}_{m,n})} \right) \right). \tag{2.20}$$

by solving $k - Q(L^r_n) + 1$ convex subproblems. Note that the equality $Q(L^r_n + b_{m,n}/C^{ul}_{par}(P^{ul}_{m,n})) = j$ holds as long as the inclusion $P^{ul}_{m,n} \in \mathcal{R}_{m,n,j}$ is satisfied with

$$\mathcal{R}_{m,n,j} = \left( \left( 2^{\frac{b_{m,n}}{B(t_j - L^r_n)}} - 1 \right) / \gamma', \left( 2^{\frac{b_{m,n}}{B(t_{j-1} - L^r_n)}} - 1 \right) / \gamma' \right], \tag{2.21}$$

where $\gamma'$ is defined as $\gamma' = \frac{\gamma N^{ul}}{B N_0}$. Then, $\bar{P}^{ul}_{m,n,k}$ in (2.19) can be calculated by first solving the problems

$$P^{ul}_{m,n,j} = \arg \min_{P^{ul}_{m,n} \in \mathcal{R}_{m,n,j}} (P^{ul}_{m,n} + P^{rf}) \frac{b_{m,n}}{C^{ul}_{par}(P^{ul}_{m,n})}, \tag{2.22}$$

for all $j \in \{Q(L^r_n), ..., k\}$ and then set

$$\begin{aligned} \bar{P}^{ul}_{m,n,k} = \arg \min_{j \in \{Q(L^r_n), ..., k\}} & (P^{ul}_{m,n,j} + P^{rf}) \frac{b_{m,n}}{C^{ul}_{par}(P^{ul}_{m,n,j})} \\ & + E^l \left( m, k - Q \left( L^r_n + \frac{b_{m,n}}{C^{ul}_{par}(P^{ul}_{m,n,j})} \right) \right). \end{aligned} \tag{2.23}$$

Each problem (2.22) becomes convex by means of the change of variable $y_{m,n} = C^{ul}_{par}(P^{ul}_{m,n})$ [49].

*Complexity*: Since the maximum number of convex optimizations that need to be solved at each time instant for each node can be upper bounded by $d_{in}K$, and $K$

**Table 2.3** Backtracking Algorithm for Table 2.2

```
Set $L_{|\mathcal{V}|} = L_{max}$ and $I_{|\mathcal{V}|} = 0$.
for $n = |\mathcal{V}| : 1$ do
    for all $m \in \mathcal{P}(n)$ do
        if $I_n = 0$
            if $I^l_m(n, Q(L_n)) = 0$
                Set $I_m = 0$ and $L_m = L_n - L^l_n$.
            else
                Set $I_m = 1$ and $L_m = L_n - \left(L^l_n + \frac{b_{m,n}}{C^{dl}_{par}}\right)$.
        else
            if $I^r_m(n, Q(L_n)) = 0$
                Set $I_m = 0$, $\bar{P}^{ul}_{m,n} = \bar{P}^{ul}_{m,n,Q(L_n)}$
                and $L_m = L_n - \left(L^r_n + \frac{b_{m,n}}{C^{ul}_{par}(\bar{P}^{ul}_{m,n})}\right)$ .
            else
                Set $I_m = 1$ and $L_m = L_n - L^r_n$.
```

is proportional to $1/\epsilon$, the complexity of the proposed algorithm in Table 2.2 is given by $O(|\mathcal{V}|d_{in}/\epsilon^2)$.

### 2.5.2 Message Passing for a General Call Graph

Similar to Section 2.4.3, for a graph with the structure discussed in Section 3.1, the problem [P.2] can be solved, for fixed parameters $N^l$, $N^r$, $N^{ul}$ and $N^{dl}$. This is done by identifying a subset $\mathcal{V}_{\mathrm{S}}$ of nodes such that, when removed, the graph is decomposed into disjoint trees. Then, for fixed offloading decisions of this set of nodes, the algorithm in Tables 2.2 and 2.3 are applied to each subtree. Finally, the optimum solution is found by comparing the energy obtained from different offloading decisions of the nodes in the set $\mathcal{V}_{\mathrm{S}}$. Following the discussion in Section 2.4.3, the resulting solution has a complexity of order $O(2^{|\mathcal{V}_s|}|\mathcal{V}|d_{in}/\epsilon^2)$, since there are $2^{\mathcal{V}_{\mathrm{S}}}$ possible offloading decisions for the nodes in $\mathcal{V}_{\mathrm{S}}$.

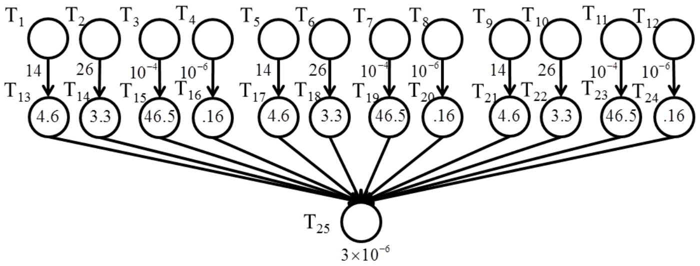


**Figure 2.4** The call tree graph used for the examples in Figure 2.5-2.7. The numbers shown next to the edges that are connected to the input task nodes represent the sizes of input bits $b_{m,n}$ in Mbits and the numbers in the task nodes (circles) represent the number of CPU cycles $v_n$ normalized by $10^9$ CPU cycles (empty circles with $v_1 = ... = v_{12} = 0$). The remaining values for case (a) are: $b_{13,25} = 7.3 \times 10^9$, $b_{14,25} = 1.4 \times 10^3$, $b_{15,25} = 1.4 \times 10^3$, $b_{16,25} = 1.4 \times 10^7$ bits, $b_{17,13} = b_{21,25} = b_{13,25}$, $b_{18,25} = b_{22,25} = b_{14,25}$, $b_{19,25} = b_{23,25} = b_{15,13}$ and $b_{20,25} = b_{24,25} = b_{16,25}$. In case (b), all the parameters are the same as case (a) except for $b_{3,15} = b_{4,16} = b_{7,19} = b_{8,20} = b_{11,23} = b_{12,24} = 11.4$ Mbits, $b_{14,25} = b_{15,25} = b_{16,25} = b_{18,25} = b_{19,25} = b_{20,25} = b_{22,25} = b_{23,25} = b_{24,25} = 14.6 \times 10^7$ bits, $b_{13,25} = b_{17,25} = b_{21,25} = 7.3 \times 10^7$ bits and $v_{15} = v_{19} = v_{23} = 4.6 \times 10^9$, $v_{16} = v_{20} = v_{24} = 3.6 \times 10^9$ and $v_{25} = 3.42 \times 10^9$ CPU cycles.

## 2.6 Simulation Results

In this section, some numerical example are provided based on the analysis developed in the previous sections. We start by considering the call tree in Figure 2.4 in order to simplify the interpretation of the results and gain an insight into the performance of the considered techniques. In this example, $\mathrm{T}_{13}, ..., \mathrm{T}_{24}$ process input data present at the mobile device, represented by nodes $\mathcal{V}_{\mathrm{D}} = \{\mathrm{T}_1, ..., \mathrm{T}_{12}\}$, e.g., to extract some features, and then root node $\mathrm{T}_{25}$ performs a “reduce” operation, such as classification, on the extracted features at the mobile ($I_{25} = 0$). We set $P^l = 0.4$ Watts, which is a common for smart phones [3, 4, 36]; $f^l = 10^9$ CPU cycles/s (e.g., Apple iPhone 6 processor has maximum clock rate of 1.4 Ghz); $f^r = 10^{10}$ CPU cycles/s (e.g., AMD FX-9590 has a clock rate of 5 Ghz [1]); $\gamma/(BN_0) = 27$ dB, $P^{rf} = 0$ W, $P^{rx} = 0$ W, $B = 1$ MHz, $C^{dl} = 200$ Mbits/s unless stated otherwise. For both the serial implementation (solid lines) and the parallel implementation (dashed lines), optimization is performed according to the algorithms described in Section 2.4 and

Section 2.5, respectively, and, for the parallel implementation, the performance is evaluated using the dynamic model presented in the Appendix B with step size $\epsilon_d = 0.1$. For parallel optimization, we set $N^{ul} = N^{dl} = N^l = N^r$ in (2.13) and (2.14) to an optimized value in the range $[1, 4]$ and we have $\epsilon = 0.1$. Note that the performance of the optimization was found not to be significantly improved with smaller values of $\epsilon$ and not to be increased by choosing larger values for $N^{ul} = N^{dl} = N^l = N^r$.

In Figure 2.5, the mobile energy cost for the serial and the parallel implementations are plotted versus the latency, along with their communication and computation components for the graph in Figure 2.4 with the selection of parameters marked as case (a) in the caption of Figure 2.4. The parameters of the graph are chosen to yield the same range of latencies and energy consumptions as in [23] and [37]. With the selected parameters, performing the application locally requires an energy equal to 65.6 J and has a latency of 164 s (outside the range of Figure 2.5). Figure 2.5 shows that significantly smaller latencies and energy expenditures can be obtained by properly optimizing the offloading decisions and the communication strategy. For instance, with an energy expenditure of 6.5 J, an optimized parallel implementation yields a latency of around 20 s, while an optimized serial implementation requires a latency of around 45 s.

The parallel implementation is shown here to have the potential to strictly outperform the serial implementation and to enable the operation at latencies that are unattainable with the serial implementation. For example, latencies in the range $[16, 38]$ s can be attained with an energy smaller than 10 J via the parallel implementation, but they cannot be achieved via the serial implementation. Moreover, as the latency increases, the energy can be seen to decrease mostly due to the fact that the communication powers can be reduced. An exception to this trend is observed for the serial implementation around the latency $L = 42$ s, due to the fact

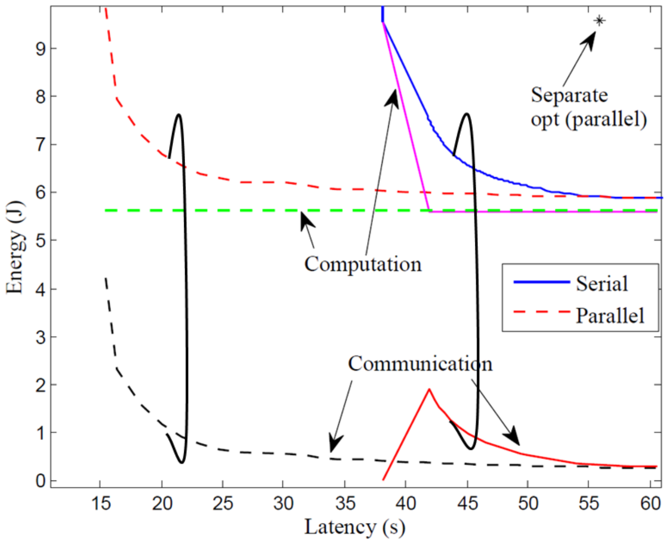


**Figure 2.5** Energy and latency trade-off for the call graph $\mathcal{G}$ in Figure 2.4 (case (a)). The program can be completely performed locally with $E = 65.6$ J and $L = 164$ s. Moreover, separate optimization for serial implementation yields $E = 9.7$ J and $L = 178$ s.

that the optimum application layer decisions prescribe more tasks to be offloaded for $L \geq 42$ s.

In order to provide a further reference performance for inter-layer optimization, we consider a conventional *separate design* strategy, whereby: $(i)$ the uplink transmission power for each task is obtained by imposing the constraint that transmitting in the uplink require a time no larger than that necessary to perform that task locally (see [49, Section 3] for a similar approach); $(ii)$ the optimization of the offloading decisions is carried out by following the proposed algorithms with fixed uploading powers, which amount to the schemes in [36] [37] for the parallel implementations. For the serial implementation, this separate approach yields a latency of 178 s and an energy expenditure of 9.7 J, which is outside the range of Figure 2.5, while for parallel processing the observed energy-latency power is illustrated in this figure. Note that separate optimization does not attempt to adapt the physical layer to the application layer requirements, and hence, it yields a single energy-latency point in the considered latency range.

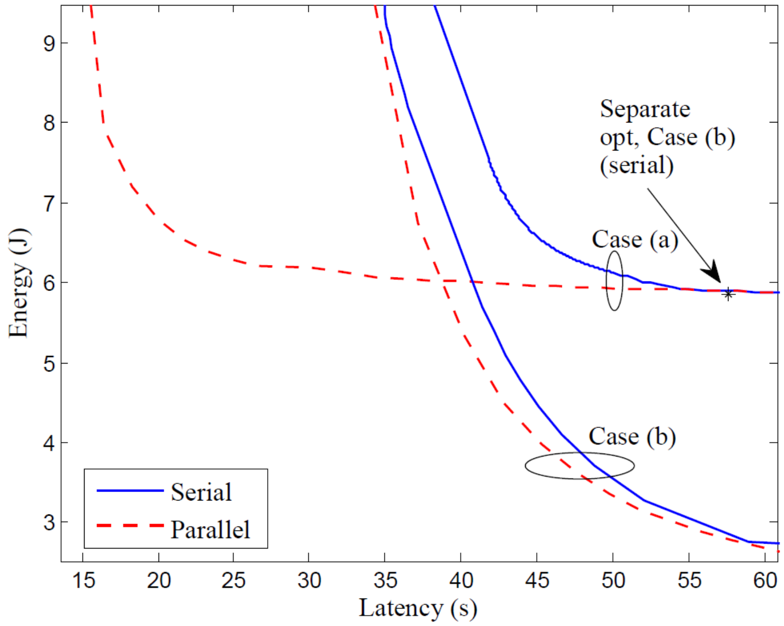


**Figure 2.6** Energy and latency trade-off for the call graph $\mathcal{G}$ in Figure 2.4 for case (a) and case (b). Separate optimization for the parallel implementation yields $E = 22.5$ J and $L = 38.5$ s for case (b) (not shown).

Figure 2.6 shows the energy-latency trade-off for the call graph in Figure 2.4 for both case (a) and case (b) as detailed in the caption of Figure 2.4. Note that the separate optimization for case (b) with the parallel implementation yields $E = 22.5$ J for $L = 38.5$, which is out of the range of Figure 2.6. The results in Figure 2.6 suggest that the gains offered by the parallel implementation over the serial implementation depend strongly on the chosen call graph. For instance, in case (b), the maximum observed energy gain of the parallel implementation is 1 J, whereas, for case (a), latencies in the interval $[16, 38]$ s require energies above 10 J for the serial implementation, while this is not the case for the parallel implementation. As another specific operating point, the parallel implementation provides 4 J gain for $L = 38$ s over the serial implementation.

To gain more insight into this point, Figure 2.7 illustrates the timeline corresponding to the parallel implementation for case (a) and case (b) for $L = 20$ s. Here, the same definition for $\{\mathrm{ID}, \mathrm{CP}^{\mathrm{l}}, \mathrm{CP}^{\mathrm{r}}, \mathrm{UL}, \mathrm{DL}\}$ is used as in Figure 2.7. It can be

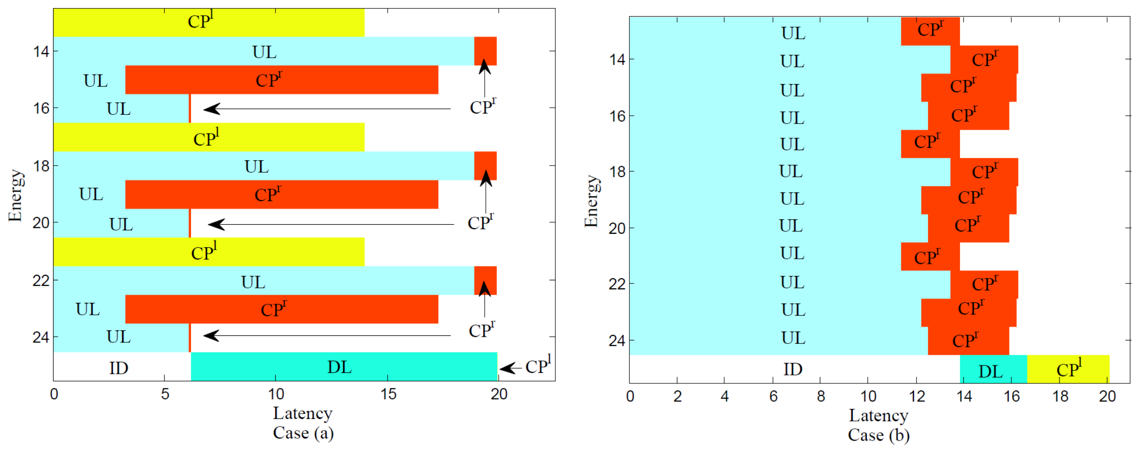


**Figure 2.7** Timeline for the parallel implementation corresponding to the optimum solution for $L = 20$ s for the call graph in Figure 2.4 (see Figure 2.6).

seen that in case (a), several communication and computation operations take place in parallel for a significant fraction of the time, and hence the parallel implementation is advantageous as compared to the serial implementation. Instead, for case (b) most of the time is spent for uplink transmissions and hence the opportunities for parallel processing are much reduced.

In order to complement the insight obtained from the study of the call graph in Figure 2.5, here we elaborate on the impact of the structure of the call graph by considering the graph in Figure 2.1. The performance of the serial and parallel implementations for the call graph $\mathcal{G}$ is plotted as well as for the subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ in Figure 2.8. The relative values of the parameters in the call graph $\mathcal{G}$ is obtained from [37], and their exact values are defined in the caption of this figure. As expected, the energy required to run the application for a given latency increases as one considers a larger call graph. More importantly, the opportunities for concurrent computations and communications are enhanced on larger subgraphs, and, as a result, for $\mathcal{T}_2$ and $\mathcal{G}$, parallel processing provides more substantial gain over the serial implementation than in $\mathcal{T}_1$.

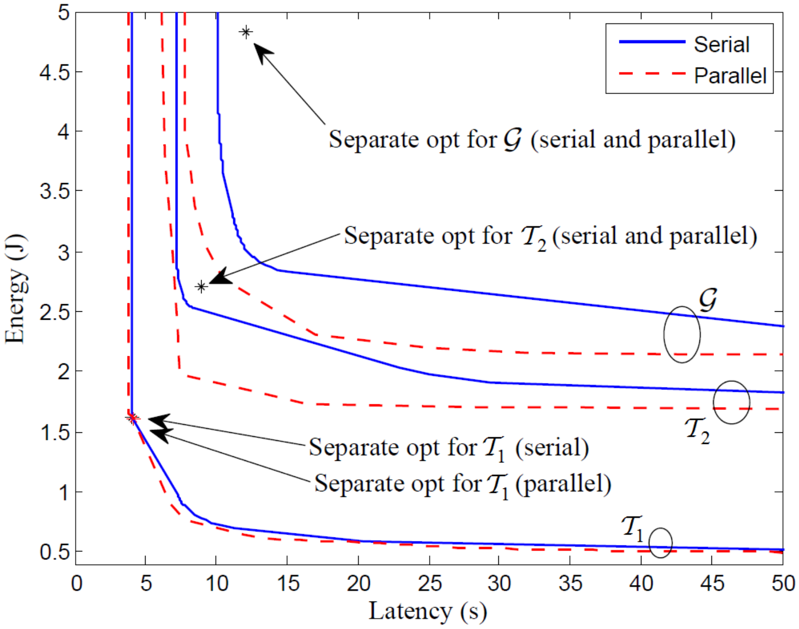


**Figure 2.8** Energy and latency trade-off for call graph $\mathcal{G}$ in Figure 2.1 and the subtrees $\mathcal{T}_1$ and $\mathcal{T}_2$ with $v_1 = 0$, $v_2 = v_4 = v_{12} = 0.6 \times 10^9$, $v_3 = 0.24 \times 10^9$, $v_5 = 0.4 \times 10^9$, $v_6 = v_9 = v_{14} = 2 \times 10^9$, $v_7 = v_8 = 1.1 \times 10^9$, $v_{10} = 0.66 \times 10^9$, $v_{11} = v_{13} = 1 \times 10^9$, $v_{15} = 0.2 \times 10^9$ CPU cycles, $b_{1,2} = b_{3,5} = b_{3,6} = b_{5,10} = b_{9,12} = b_{11,14} = b_{12,14} = 5 \times 10^6$, $b_{2,3} = 15 \times 10^6$, $b_{2,4} = 9.7 \times 10^6$, $b_{4,7} = b_{4,8} = 8.5 \times 10^6$, $b_{4,9} = 3 \times 10^6$, $b_{6,10} = 8 \times 10^6$, $b_{7,11} = b_{8,11} = 1.2 \times 10^6$, $b_{10,13} = b_{13,15} = 10 \times 10^6$ and $b_{14,15} = 15.5 \times 10^6$ bits.

## 2.7 Concluding Remarks

In this chapter, the inter-layer optimization of cloud mobile computing systems over the power allocation at the physical layer and offloading decisions at the application layer is studied with the aim of exploring the achievable trade-offs between the mobile energy expenditure and latency. Unlike prior work in which the problem is formulated as a mixed integer program, here a message-passing framework is proposed that leverage the typical structure of call graphs to drastically reduce complexity. In particular, we focused on call graphs that can be decomposed into combination of a small number of subtrees when fixing the decisions of a subset of nodes, obtaining a complexity that grows exponentially only in the size of such set of nodes rather the size of the call graph. Moreover, unlike prior art, the framework is applied to both the conventional serial implementation and a parallel implementation that enables the concurrent schedule of communication and computation. Via simulation results, we demonstrated the impact of the call graph structure on the relative performance of the parallel and serial implementations, and shed light on the impact of inter-layer optimization.

# CHAPTER 3

# UPLINK HARQ FOR DISTRIBUTED AND CLOUD RAN VIA SEPARATION OF CONTROL AND DATA PLANES

Distributed-Radio Access Network (D-RAN) and Cloud-RAN (C-RAN) are new cellular architectures that are candidate for next generation wireless technology 5G. However, the implementation of uplink HARQ in D-RAN or C-RAN architecture is constrained by the two-way latency on the fronthaul links connecting the Remote Radio Heads (RRHs) with the Baseband Units (BBUs) that perform decoding. To overcome this limitation, in this chapter an architecture based on the separation of control and data planes is considered, in which retransmission decisions are made at the edge of the network, that is, by the RRHs or User Equipments (UEs), while data decoding is carried out remotely at the BBUs. This architecture enables *low-latency local retransmission decisions* to be made at the RRHs or UEs, which are not subject to the fronthaul latency constraints, while at the same time leveraging the decoding capability of the BBUs. A D-RAN system is first considered in which *low-latency local feedback* from the RRH assigned to a given UE is used to drive the UE's HARQ process. Throughput and probability of error of this solution are analyzed for the three standard HARQ modes of Type-I, Chase Combining and Incremental Redundancy over a general fading MIMO link. Then, novel *user-centric low-latency feedback* strategies are proposed and analyzed for the C-RAN architecture based on limited "hard" or "soft" local feedback from the RRHs to the UE and on retransmission decisions taken at the UE. The presented analysis allows the optimization of the considered schemes, as well as the investigation of the impact of system parameters such as HARQ protocol type, blocklength and number of antennas on the performance of low-latency local HARQ decisions in D-RAN and C-RAN architectures.

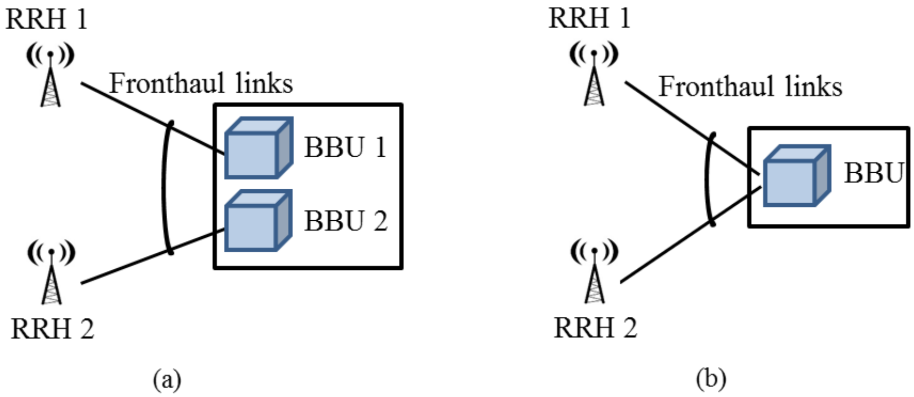


**Figure 3.1** Illustration of the (a) D-RAN and (b) C-RAN architecture ($L = 2$ RRHs).

## 3.1 Introduction

Distributed and Cloud Radio Access Network, abbreviated as D-RAN and C-RAN, respectively, are candidate cellular architectures for 5G systems, in which the baseband processing unit (BBU) of each base station is virtualized at a "cloud" processor. This virtualization yields the separation between the remote radio head (RRH) that implements the radio functionalities of the base station and a centralized BBU that is charged with higher-layer tasks, including the physical layer.

In a D-RAN, as seen in Figure 3.1-(a), the BBU of each base station is hosted at a remote site, which is accessed by the RRH via connections known as *fronthaul* links. In a D-RAN, the BBUs of different RRHs are hence distinct. The D-RAN architecture lowers the expenditure needed to deploy and operate dense cellular networks, by simplifying the base stations hardware and by enabling flexible upgrading and easier maintenance (see, e.g., [17, 56, 60]). D-RAN also allows limited forms of cooperation to be implemented among base stations, particularly in the downlink, by leveraging an X2 interface that may connect the BBUs with one another within the same "cloud" [60]. Nevertheless, joint baseband decoding in the uplink is generally not feasible in a D-RAN, since it requires the exchange of baseband signals among BBUs, rather

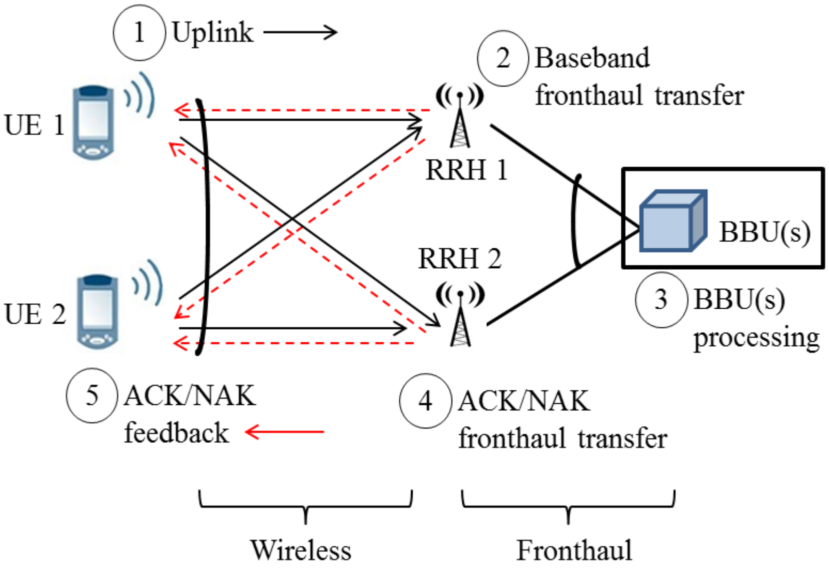


**Figure 3.2** Conventional HARQ in D-RAN or C-RAN. The numbers indicate the sequence of events associated with a transmission. Fronthaul latency is associated with the fronthaul transmissions at steps 2 and 4 and with the part of BBU processing at step 3 needed to encode and decode transmissions on the fronthaul links. The cross-links in the uplink carry interference in a D-RAN and useful signals in a C-RAN. The dashed cross-links in the ACK/NAK feedback path are used only in the C-RAN architecture.

than user-plane data as allowed by an X2 interface (see e.g., [17, 56, 60]). Therefore, a D-RAN operates as a conventional cellular system, in which each user equipment (UE) is assigned to one RRH.

In a C-RAN architecture, instead, a unique BBU is shared among multiple RRHs, as depicted in Figure 3.1-(b). Therefore, C-RAN enables joint baseband processing across all the RRHs connected to the same BBU. In addition to the gains achieved by D-RAN, C-RAN can hence also benefit from the statistical multiplexing and interference management capabilities that are made possible by joint baseband processing across multiple RRHs. Furthermore, no UE-RRH assignment is ncecessary (see, e.g., [17, 56]).

**Main Problem:** The implementation of the D-RAN and C-RAN architectures needs to contend with the potentially significant latencies needed for the transfer and processing of the baseband signals on the fronthaul links to and from the BBU(s) [61].

The communication protocols that are most directly affected by fronthaul delays are the Automatic Repeat Request (ARQ) and Hybrid ARQ (HARQ)[1] protocols at layer 2 of the protocol stack. In fact, in a conventional cellular network, upon receiving a codeword from an UE, the local base station performs decoding, and, depending on the decoding outcome, feeds back an Acknowledgment (ACK) or a Negative Acknowledgment (NAK) to the UE. In contrast, in a D-RAN or C-RAN, as illustrated in Figure 3.2, the outcome of decoding at the BBU may only become available at the RRHs after the time required for the transfer of the baseband signals from the RRHs to the BBU(s) on the fronthaul links, for processing at the BBU(s), and for the transmission of the decoding outcome from the BBU(s) to the RRHs on the fronthaul links.

The fronthaul latency may significantly affect the performance of retransmission protocols. For instance, in LTE with frequency division multiplexing, the feedback latency should be less than 3 ms in order not to disrupt the operation of the system [22][2]. We also refer to [32] for a discussion on the effect of the latency on ARQ protocols in C-RANs.

**A Solution Based on the Separation of Control and Data Planes:** Fronthaul latency is unavoidable in conventional D-RAN and C-RAN architectures in which the RRHs only retain radio functionalities. Nevertheless, alternative functional splits are currently being investigated whereby the RRH may implement some additional functions [17, 20, 22, 82]. In this chapter, we consider a functional split that enables the *separation of control and data planes associated with the HARQ protocol*, with the aim of alleviating the problem of fronthaul latency. We note that

[1] In ARQ protocols, different transmissions of a packet are performed independently, whereas, in HARQ schemes, decoding and/or coding can be performed across multiple retransmissions [19].

[2] By interleaving multiple HARQ processes, as discussed in [22], the tolerated latency can be increased to $3 + n8$ ms, where $n$ is a positive integer, albeit at the cost of possibly reducing the throughput. Of the mentioned 3 ms latency, it has been recently specified that the one-way transport delay on the fronthaul should be no larger than around 400 $\mu$s [61].

the approach studied here can be seen as an instance of the more general principle of control and data separation, for which an overview of the literature can be found in [57].

In particular, we investigate an architecture in which retransmission decisions are made at the edge of the network, that is, by the RRHs or UEs, while data decoding is carried out remotely at the BBUs as in a conventional D-RAN or C-RAN. This architecture enables *low-latency local retransmission decisions* to be made at the RRHs or UEs, which are not subject to the fronthaul latency constraints, while at the same time leveraging the decoding capability of the BBUs.

Low-latency local control of the retransmission process is made possible by an RRH-BBU functional split whereby each RRH can perform synchronization and resource demapping, so as to distinguish the different fields of a frame [22][3]. In fact, this functional split allows the RRHs to gather information about the modulation and coding scheme (MCS) used in the uplink packet, as well as on the channel state information (CSI) about the local uplink channels. As proposed in [22] and in [71], for a D-RAN system, based on the available MCS and CSI, the RRH assigned to a UE can make local decisions about whether successful or unsuccessful decoding is expected to occur at the BBU, feeding back an ACK/NAK message to the UE accordingly. The RRH associated to a UE makes this *local low-latency feedback decision* without waiting to be notified about the actual decoding outcome at the BBU and without running the channel decoder, which is implemented only at the BBU. Figure 3.4 presents an illustration of the outlined low-latency approach.

The local feedback approach under discussion introduces possible errors due to the mismatch between the local decision at the RRH and the actual decoding outcome at the BBU. Indeed, the RRH may request an additional retransmissions for a packet

[3] In an OFDM system, such as LTE, this requires also the implementation of an FFT block.

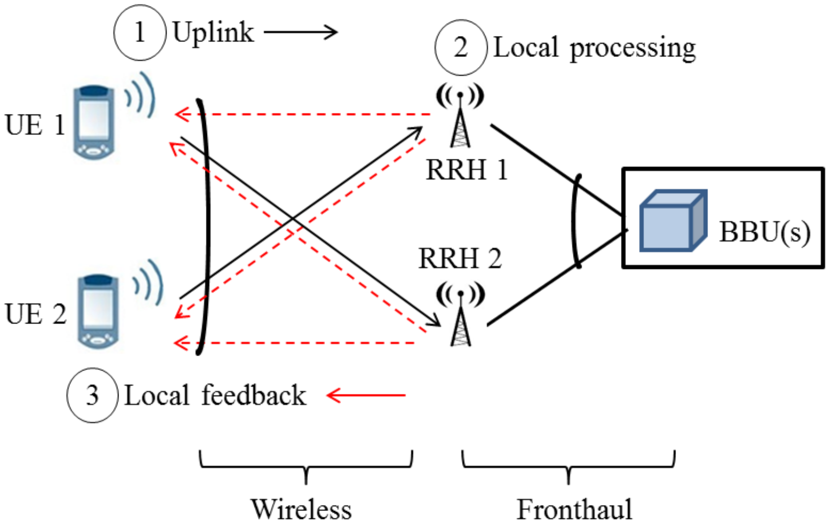


**Figure 3.3** HARQ in D-RAN and C-RAN systems via low-latency local feedback based on separation of control and data planes. HARQ control is carried out at the network edge based on local low-latency feedback from the RRHs, while data decoding is carried out at the BBUs. The cross-links in the uplink carry interference in a D-RAN and useful signals in a C-RAN. The cross-links in the feedback path are used only in the C-RAN architecture.

that the BBU is able to decode, or acknowledge correct reception of a packet for which decoding eventually fails at the BBU, hence causing a throughput degradation.

In a C-RAN, which is characterized by joint baseband processing across multiple RRHs, the outlined approach based on local feedback is complicated by the fact that the channel state information between the UE and each RRH is not known to other RRHs. Therefore, it is not possible for the RRHs to directly agree on HARQ control decisions, making the local feedback mechanism proposed in [22] and [71] not applicable.

**Main contributions:** The main contributions of this chapter are summarized as follows.

- For D-RAN, we analyze throughput and probability of error of low-latency local feedback for the three standard HARQ modes of Type-I (TI), Chase Combining (CC) and Incremental Redundancy (IR) over a multi-antenna, or MIMO, link with coding blocks (packets) of arbitrary finite length. This is done by leveraging recently derived finite-blocklength tight capacity bounds [63]. As a result, unlike the existing literature [22] and [71], the analysis allows the investigation of the impact of system parameters such as HARQ protocol type, blocklength and number of antennas. We note that the analysis in [71] focuses on the throughput of single-antenna links in a D-RAN with HARQ-IR and is based on an error exponent framework, which is known to be provide an inaccurate evaluation of the probability of error in the practical finite-blocklength regime [63, Eq. (54)] [62, Section 1.2 and Section 1.3].

- We propose and analyze *user-centric low-latency feedback* schemes for C-RAN systems. According to these proposed techniques, limited-feedback information is sent from each RRH to an UE in order to allow the latter to make a low-latency local control decision about the need for a retransmission. A "hard feedback" approach is first proposed that directly generalizes the D-RAN scheme described above and requires a one-bit feedback message from each RRH. Then, a "soft feedback" strategy is proposed in which the UE decision is based on multi-bit feedback from the RRHs, consisting of quantized local CSI.

The rest of this chapter is organized as follows. In Section 3.2, the system model for D-RAN and C-RAN systems is introduced. Section 3.3 details the principles underlying the proposed low-latency local feedback solutions for D-RAN and C-RAN. The metrics used to evaluate the performance of the proposed schemes and some preliminaries are discussed in Section 3.4. In Sections 3.5 and 3.6, the analysis of D-RAN and C-RAN strategies is presented. In Section 3.7, the numerical results are provided, and Section 4.6 concludes this chapter.

*Notation*: Bold letters denote matrices and superscript $^H$ denote Hermitian conjugation. $\mathcal{CN}(\mu, \sigma^2)$ denotes a complex normal distribution with mean $\mu$ and variance $\sigma^2$; and $\mathcal{X}_k^2$ a Chi-Squared distribution with $k$ degrees of freedom. $f_{\mathcal{A}}(x)$ and $F_{\mathcal{A}}(x)$ represent the probability density function and the cumulative distribution function of a distribution $\mathcal{A}$ evaluated at $x$, respectively. $\mathbf{A} = \text{diag}([\mathbf{A}_1, ..., \mathbf{A}_n])$ is a block diagonal matrix with block diagonal given by the matrices $[\mathbf{A}_1, ..., \mathbf{A}_n]$. The indicator function $\mathbf{1}(x)$ equals 1 if $x = \text{true}$ and 0 if $x = \text{false}$.

## 3.2 System Model

We study the uplink of both D-RAN and C-RAN systems as illustrated in Figure 3.1. In this section, the system model and performance metrics are detailed.

### 3.2.1 System Model

As seen in Figure 3.1, each RRH is connected by means of orthogonal fronthaul links to a dedicated BBU for D-RAN and to a single BBU for C-RAN systems. The BBUs perform decoding, while the RRHs have limited baseband processing functionalities that allow resource demapping and the inference of CSI and MCS information as discussed in Section 3.1 and further detailed below. Different UEs are served in distinct time-frequency resources, as done for instance in LTE, and hence we focus here on the performance of a given UE.

Each packet transmitted by the UE contains $k$ encoded complex symbols and is transmitted within a coherence time/frequency interval of the channel, which is referred to as *slot*. The transmission rate of the first transmission of an information message is defined as $r$ bits per symbol, so that $kr$ is the number of information bits in the information message.

Each transmitted packet is acknowledged via the transmission of a feedback message by the RRHs. We assume that these feedback messages are correctly decoded by the UE. We will first assume that messages are limited to binary positive or negative acknowledgments, i.e., ACK or NAK messages, in Section 3.3.1, and we will consider the more general case in which feedback messages may consist of $b \geq 1$ bits in Section 3.3.2. The same information message may be transmitted for up to $n_{max}$ successive slots using standard HARQ protocols such as TI, CC and IR, to be recalled in Section 3.3.

The UE is equipped with $m_t$ transmitting antennas, while $m_{r,l}$ receiving antennas are available at the $l$th RRH. The received signal for any $n$th slot at the $l$th RRH can be expressed as

$$\mathbf{y}_{l,n} = \sqrt{\frac{s}{m_t}}\mathbf{H}_{l,n}\mathbf{x}_n + \mathbf{w}_{l,n}, \tag{3.1}$$

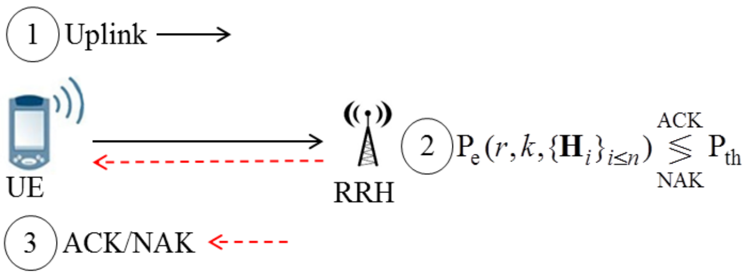
where $s$ measures the average SNR per receive antenna; $\mathbf{x}_n \in \mathbb{C}^{m_t \times 1}$ represents the symbols sent by the transmit antennas at a given channel use, whose average power is normalized as $\mathrm{E}[||\mathbf{x}_n||^2] = 1$; $\mathbf{H}_{l,n} \in \mathbb{C}^{m_{r,l} \times m_t}$ is the channel matrix, which is assumed to have independent identically distributed (i.i.d.) $\mathcal{CN}(0,1)$ entries (Rayleigh fading); and $\mathbf{w}_{l,n} \in \mathbb{C}^{m_{r,l} \times 1}$ is an i.i.d. Gaussian noise vector with $\mathcal{CN}(0,1)$ entries. The channel matrix $\mathbf{H}_{l,n}$ are independent for different RRHs $l \in \{1, ..., L\}$ and also change independently in each slot $n$. Moreover, they are assumed to be known to the $l$th RRH and to the BBU. We assume the use of Gaussian codebooks with an equal power allocation across the transmit antennas, although the analysis could be extended to arbitrary power allocation and antenna selection schemes.

### 3.2.2 Performance Metrics

The main performance metrics of interest are as follows.

- Throughput $T$: The throughput measures the average rate, in bits per symbol, at which information can be successfully delivered from the UE to the BBU;
- Probability $\mathrm{P_s}$ of success: The metric $\mathrm{P_s}$ measures the probability of a successful transmission within a given HARQ session, which is the event that, in one of the $n_{max}$ allowed transmission attempts, the information message is decoded successfully at the BBU.

Note that errors in the HARQ sessions can be dealt with by higher layers, as done by the RLC layer in LTE [22], albeit at the cost of large delays. For this reason,



1 Uplink
UE
RRH
2 $\mathrm{P_e}(r,k,\{\mathbf{H}_i\}_{i\leq n}) \underset{\mathrm{NAK}}{\overset{\mathrm{ACK}}{\lessgtr}} \mathrm{P_{th}}$
3 ACK/NAK

**Figure 3.4** Low-latency local feedback scheme for D-RAN systems: ACK/NAK messages are sent by the assigned RRH to a given UE according to the local decision rule (3.2).

in Section 3.7, we will pay special attention to the throughput that can be obtained under a given constraint on the probability of success $\mathrm{P_s}$. Typical values for $\mathrm{P_s}$, on which one can base the design of higher layers, are in the order of 0.99 – 0.999 [60] [24]. We elaborate on the evaluation of these metrics in Section 3.4.

## 3.3 Low-Latency Local Feedback

In this section, the key working principles underlying low-latency local feedback solutions for D-RAN and C-RAN is introduced.

### 3.3.1 RRH-Based Low-Latency Local Feedback for D-RAN

In a D-RAN architecture, each pair of RRH and corresponding BBU operates as a base station in a conventional cellular system [56] [17]. Therefore, an UE is assigned to a specific RRH-BBU pair by following standard user association rules. For D-RAN, as in [71], we can then focus on a single RRH, i.e., $L = 1$, with the understanding that the noise term in (3.1) may account also for the interference from UEs associated to other RRH-BBU pairs. When studying D-RAN systems, we hence drop the subscript $l$ indicating the RRH index.

The low-latency local feedback scheme for D-RAN, first proposed in [22], is illustrated in Figure 3.4. At each transmission attempt $n$, the RRH performs resource demapping and obtains CSI about the channel $\mathbf{H}_n$ and the MCS used for data transmission. The MCS amounts here to the rate $r$ and packet length $k$. Based on this information, the RRH can compute the probability of error $\mathrm{P_e}(r, k, \{\mathbf{H}_i\}_{i\leq n})$ for decoding at the BBU, where we emphasized the possible dependence of the probability of error $\mathrm{P_e}$ on all channel matrices $[\mathbf{H}_1, \cdots, \mathbf{H}_n]$ corresponding to prior and current transmission attempts. We note that the probability $\mathrm{P_e}$ may be read on a look-up table or obtained from some analytical approximations as discussed in the next section. As proposed in [71], if the decoding error probability $\mathrm{P_e}(r, k, \{\mathbf{H}_i\}_{i\leq n})$ is smaller than a given threshold $\mathrm{P_{th}}$, the RRH sends an ACK message to the UE, predicting a positive decoding event at the BBU; while, otherwise, a NAK message is transmitted, that is,

$$\mathrm{P_e}(r, k, \{\mathbf{H}_i\}_{i\leq n}) \underset{\mathrm{NAK}}{\overset{\mathrm{ACK}}{\lesseqgtr}} \mathrm{P_{th}}. \tag{3.2}$$

As we will discuss in Section 3.7, the optimization of the threshold $\mathrm{P_{th}}$ needs to strike a balance between the probability of success $\mathrm{P_s}$, which would call for a smaller $\mathrm{P_{th}}$ and hence more retransmissions, and the throughput $T$, which may be generally improved by a larger $\mathrm{P_{th}}$, resulting in the transmission of new information.

### 3.3.2 User-Centric Low-Latency Local Feedback for C-RAN

In C-RAN, unlike D-RAN systems, a BBU jointly processes the signals received by several connected RRHs (Figure 3.1-(b)). Therefore, a UE-RRH assignment step is not needed as the BBU performs decoding based on the signals received from all connected RRHs. The development of a local feedback solution for C-RAN is hence complicated by the fact that the BBU decoding error probability $\mathrm{P_e}(r, k, \{\mathbf{H}_i\}_{i\leq n})$

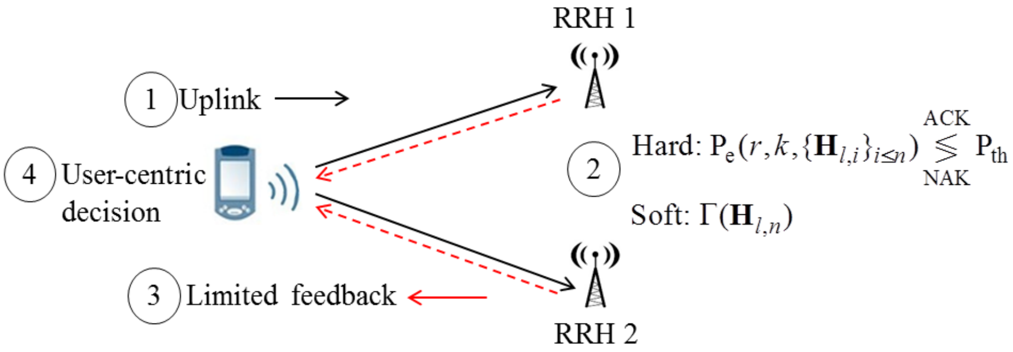

RRH 1
1 Uplink
4 User-centric decision
2 Hard: $\mathrm{P_e}(r,k,\{\mathbf{H}_{l,i}\}_{i\le n}) \underset{\text{NAK}}{\overset{\text{ACK}}{\lessgtr}} \mathrm{P_{th}}$
Soft: $\Gamma(\mathbf{H}_{l,n})$
3 Limited feedback
RRH 2

**Figure 3.5** Low-latency local feedback scheme for C-RAN systems: The UE collects limited-feedback messages from the RRHs to make a local decision on whether another transmission attempt is necessary.

depends on the CSI $\{\mathbf{H}_i\}_{i\le n}$ between the UE and all RRHs, while each RRH $l$ is only aware of the CSI $\{\mathbf{H}_{l,i}\}_{i\le n}$ between the UE and itself. Therefore, the decoding error probability $\mathrm{P_e}(r,k,\{\mathbf{H}_i\}_{i\le n})$ cannot be calculated at any RRH as instead done for D-RAN.

To overcome this problem, in this chapter, a user-centric low-latency local HARQ mechanism is proposed, whereby the UE collects limited-feedback messages from the RRHs, based on which it makes a local decision about whether a further retransmission attempt is needed or not, illustrated in Figure 3.5. We allow for multi-bit feedback messages from the RRHs to the UE, and study methods based on *hard feedback*, and *soft feedback*, as explained next.

**Hard Feedback** The hard feedback scheme is a direct extension of the local feedback solution explained in Section 3.3.1 for D-RAN. Since at the $n$th transmission attempt, the $l$th RRH is only aware of the CSI $\{\mathbf{H}_{l,i}\}_{i\le n}$ between itself and the UE, it can only calculate the decoding error probability $\mathrm{P_e}(r,k,\{\mathbf{H}_{l,n}\}_{i\le n})$, which corresponds to a scenario in which the BBU decodes solely based on the signal received by the $l$th RRH. Then, each RRH $l$ uses a 1-bit quantizer, which maps the probability $\mathrm{P_e}(r,k,\{\mathbf{H}_{l,n}\}_{i\le n})$ to an ACK/NAK message according to the same rule used in D-RAN system, i.e.,

$$\mathrm{P_e}(r, k, \{\mathbf{H}_{l,n}\}_{i \leq n}) \underset{\mathrm{NAK}}{\overset{\mathrm{ACK}}{\lesseqgtr}} \mathrm{P_{th}}. \tag{3.3}$$

The UE decides to retransmit the packet if all RRHs return a NAK message and to stop retransmissions if at least one ACK is received.

**Soft Feedback** The soft feedback schemes aims at leveraging multi-bit feedback messages, composed of $b \geq 1$ bits, from each RRH to the UE. The key idea here is that the UE can estimate the decoding error probability $\mathrm{P_e}(r, k, \{\mathbf{H}_i\}_{i \leq n})$ of the BBU upon receiving information from each RRH $l$ about the local CSI $\mathbf{H}_{l,n}$. To this end, in the soft feedback scheme, each RRH quantizes its own CSI $\mathbf{H}_{l,n}$ by using vector quantization [50] with $b$ bits and sends the quantized CSI $\Gamma(\mathbf{H}_{l,n}) = \hat{\mathbf{H}}_{l,n}$ to the UE via a $b$-bit feedback message. Then, the UE performs a retransmission if the estimated decoding error probability $\mathrm{P_e}(r, k, \{\hat{\mathbf{H}}_i\}_{i \leq n})$, with $\hat{\mathbf{H}}_i$ collecting all the quantized matrices $\hat{\mathbf{H}}_{l,n}$ for $l \in \{1, ..., L\}$, is larger than a threshold $\mathrm{P_{th}}$ and stop retransmission otherwise, as in

$$\mathrm{P_e}(r, k, \{\hat{\mathbf{H}}_i\}_{i \leq n}) \underset{\mathrm{NAK}}{\overset{\mathrm{ACK}}{\lesseqgtr}} \mathrm{P_{th}}. \tag{3.4}$$

## 3.4 Performance Criteria and Preliminaries

In this section, we discuss the general approach that will be followed to evaluate throughput and probability of success for the considered schemes in D-RAN and C-RAN systems.

### 3.4.1 Throughput and Probability of Success

To start, let us denote as $\mathrm{RTX}_n$ the event that a retransmission decision is made for all the first $n$ transmission attempts of an information message. In a similar manner, $\mathrm{STOP}_n$ is defined as the event that a decision is made to stop the retransmission of a packet at the $n$th attempt, and hence $n-1$ retransmission attempts have been performed before. As discussed in Section 3.3, these decisions are made at the RRH for the low-latency local feedback scheme in D-RAN and at the UE in the proposed user-centric low-latency strategies for C-RAN. By definition, the probabilities of these events satisfy the equality

$$\mathrm{P}(\mathrm{STOP}_n) = \mathrm{P}(\mathrm{RTX}_{n-1}) - \mathrm{P}(\mathrm{RTX}_n). \tag{3.5}$$

**Remark**: In case of ideal feedback from the BBU, a STOP/RTX event reflects correct/incorrect decoding at the BBU, whereas this is not the case for the local feedback schemes due to the possible mismatch between the RRHs' or users' decisions and the decoding outcome at the BBU. In particular, there are two types of error as summarized in Table 3.1. In the first type of error, the transmitted packet is not decodable at the BBU, but a STOP decision is made by the local feedback scheme. This type of mismatch needs to be dealt with by higher layers, introducing significant delays. In the second type of error, the received packet is decodable at the BBU, but an RTX decision is made. In this case, the UE performs an unnecessary retransmission. It is observed that the first type of error is more deleterious to the performance as it affects directly the probability of success. ■

We now elaborate on the calculation of the throughput $T$ and probability of success $\mathrm{P_s}$ for both the local feedback schemes and reference ideal case of zero-delay feedback from the BBU. For all schemes, based on standard renewal theory arguments,

**Table 3.1** Error Types Due to Low-latency Local Feedback

| **BBU decoding outcome** | **Local feedback decision** | **Consequence** |
|---|---|---|
| Undecodable | STOP | Delays due to higher-layer protocols |
| Decodable | RTX | HARQ retransmission |

the throughput can be calculated as [16]

$$T = \frac{r\mathrm{P_s}}{\mathrm{E}[N]}, \tag{3.6}$$

where we recall that $r$ is the transmission rate, and the random variable $N$ denotes the number of transmission attempts for a given information message. The average number of transmissions can be computed directly as

$$\mathrm{E}[N] = \sum_{n=1}^{n_{max}-1} n\mathrm{P}(\mathrm{STOP}_n) + n_{max}\mathrm{P}(\mathrm{RTX}_{n_{max}-1}). \tag{3.7}$$

Moreover, the probability of a successful transmission for the case of zero-delay feedback from the BBU is given as

$$\mathrm{P_s} = 1 - \mathrm{P}(\mathrm{RTX}_{n_{max}}). \tag{3.8}$$

Instead, with local feedback, a transmission is considered as successful if a decision is made to stop the retransmission of a packet within one of the $n_{max}$ allowed transmissions attempts *and* if the BBU can correctly decode. Hence, by the law of total probability, the probability of success $\mathrm{P_s}$ can be written as

$$\mathrm{P_s} = \sum_{n=1}^{n_{max}} \mathrm{P}(\mathrm{D}_n|\mathrm{STOP}_n)\mathrm{P}(\mathrm{STOP}_n), \tag{3.9}$$

where $\mathrm{D}_n$ is the event that the BBU can correctly decode at the $n$th transmission.

In summary, in order to evaluate the throughput, (3.5)-(3.7) is used for both ideal and local feedback; while, for the probability of success $\mathrm{P_s}$, (3.8) is used for the case of ideal feedback and (3.9) for local feedback. Therefore, to compute both metrics, we only need to calculate the probabilities $\mathrm{P}(\mathrm{RTX}_n)$, for both ideal and local feedback, and the probabilities $\mathrm{P}(\mathrm{D}_n|\mathrm{STOP}_n)$ for local feedback, with $n = 1, ..., n_{max}$. This approach is used in the next two sections for D-RAN and C-RAN systems.

### 3.4.2 Gaussian Approximation

Throughout this chapter, the Gaussian approximation proposed in [84] is adopted, based on the work in [63], to evaluate the probability $\mathrm{P_e}(r, k, \mathbf{H})$ of decoding error for a transmission at rate $r$ in a slot of $k$ channel uses when the channel matrix is $\mathbf{H}$. This amounts to

$$\mathrm{P_e}(r, k, \mathbf{H}) = Q\left(\frac{C(\mathbf{H}) - r}{\sqrt{\frac{V(\mathbf{H})}{k}}}\right), \tag{3.10}$$

where we have defined

$$C(\mathbf{H}) = \sum_{j=1}^{m_{rt}} \log_2\left(1 + \frac{s\lambda_j}{m_t}\right) \text{ and } V(\mathbf{H}) = \left(m_{rt} - \sum_{j=1}^{m_{rt}} \frac{1}{\left(1 + \frac{s\lambda_j}{m_t}\right)^2}\right) \log_2^2 e, \tag{3.11}$$

with $m_{rt} = \min(m_r, m_t)$; $\{\lambda_j\}_{j=1,\dots,m_{rt}}$ being the eigenvalues of the matrix $\mathbf{H}^H\mathbf{H}$; and $Q(\cdot)$ being the Gaussian complementary cumulative distribution function. Expressions obtained by means of the Gaussian approximation (3.10) will be marked for simplicity of notation as equalities in the following.

For future reference, we note that we have the limit

$$\lim_{k \longrightarrow \infty} \mathrm{P_e}(r, k, \mathbf{H}) = \begin{cases} 1 & \text{if } C(\mathbf{H}) < r \\ 0 & \text{if } C(\mathbf{H}) > r \end{cases} \tag{3.12}$$

in the asymptotic regime of large blocklengths.

## 3.5 Analysis of RRH-Based Low-Latency Local Feedback for D-RAN

In this section, we analyze the performance in terms of throughput and probability of success of the low-latency local feedback scheme for D-RAN introduced in Section 3.3.1. We focus separately on the three standard modes of HARQ-TI, CC and IR, in order of complexity [27]. We recall that, in the considered low-latency scheme, a decision to stop retransmissions is made by the RRH by sending an ACK message, while a retransmission is decided by the transmission of a NAK message. $\mathrm{ACK}_n$ is defined as the event that an ACK message is sent at the $n$th transmission attempt and as $\mathrm{NAK}_n$ the event that a NAK message is sent for all the first $n$ transmissions. Therefore, in applying the analytical expression introduced in the previous section, we can focus on the evaluation of the probabilities $\mathrm{P}(\mathrm{RTX}_n) = \mathrm{P}(\mathrm{NAK}_n)$ and $\mathrm{P}(\mathrm{D}_n|\mathrm{STOP}_n) = \mathrm{P}(\mathrm{D}_n|\mathrm{ACK}_n)$ in order to calculate throughput and probability of success. Throughout, we use the Gaussian approximation for the probability of error discussed in Section 3.4.2.

### 3.5.1 HARQ-TI

With HARQ-TI, the same packet is retransmitted by the UE upon reception of a NAK message until the maximum number $n_{max}$ of retransmissions is reached or until an ACK message is received. Moreover, decoding at the BBU is based on the last received packet only. HARQ-TI is hence a standard ARQ strategy [19].

**Ideal Feedback** For reference, we first study the ideal case in which zero-delay feedback is available directly from BBU. Using the approximation (3.10) and averaging over the channel distribution, the approximate probability of an erroneous decoding at the BBU at the $n$th retransmission is given by $\mathrm{E}\left[\mathrm{P}_e(r,k,\mathbf{H}_n)\right]$. Accordingly, since with HARQ-TI the BBU performs decoding independently for each slot, we obtain

$$\mathrm{P}(\mathrm{NAK}_n) = \left(\mathrm{E}\left[\mathrm{P}_e(r,k,\mathbf{H})\right]\right)^n . \tag{3.13}$$

As discussed, throughput and the probability of success now can be calculated as (2)-(4) and (5), where the throughput can be simplified as

$$T = r\left(1 - \mathrm{E}\left[\mathrm{P}_e(r,k,\mathbf{H})\right]\right). \tag{3.14}$$

The average in (3.14) can be computed numerically based on the known distribution of the eigenvalues the Wishart-distributed matrix $\mathbf{H}^H\mathbf{H}$, see [78, Theorem 2.17]. As an important special case, for a SISO link ($m_t = m_r = 1$), we have $|H|^2 \sim \mathcal{X}_2^2$ and hence

$$\mathrm{E}\left[\mathrm{P}_e(r,k,H)\right] = \int_0^\infty \mathrm{P}_e(r,k,\sqrt{x}) f_{\mathcal{X}_2^2}(x)\mathrm{d}x. \tag{3.15}$$

**Local Feedback** With local feedback, as discussed, at each transmission attempt $n$, the RRH estimates the current channel realization $\mathbf{H}_n$ and decides whether it expects the BBU to decode correctly or not by comparing the probability of error by using the following rule (3.2), which reduces to

$$\mathrm{P_e}(r,k,\mathbf{H}_n) \underset{\mathrm{NAK}}{\overset{\mathrm{ACK}}{\lesseqgtr}} \mathrm{P_{th}}, \tag{3.16}$$

since decoding is done only based on the last received packet. We observe that, in the case of a single antenna at the transmitter and/or the receiver, the rule (3.16) only requires the RRH to estimate the SNR $s||\mathbf{H}_n||^2/m_t$.

The quantities that are needed to calculate the performance metrics under study can be then directly obtained from their definitions as

$$\mathrm{P}(\mathrm{D}_n|\mathrm{ACK}_n) = 1 - \mathrm{E}\left[\mathrm{P_e}(r,k,\mathbf{H})|\mathrm{P_e}(r,k,\mathbf{H}) \leq \mathrm{P_{th}}\right] \tag{3.17}$$

$$\text{and } \mathrm{P}(\mathrm{NAK}_n) = \left(\mathrm{P}\left(\mathrm{P_e}(r,k,\mathbf{H}) > \mathrm{P_{th}}\right)\right)^n. \tag{3.18}$$

As discussed, (3.17) and (3.18) can be obtained by averaging over the distribution of the eigenvalues of $\mathbf{H}^H\mathbf{H}$. As an example, for a SISO link, we obtain

$$\mathrm{P}(\mathrm{D}_n|\mathrm{ACK}_n) = 1 - \frac{1}{1-F_{\mathcal{X}_2^2}(\gamma(\mathrm{P_{th}}))}\int_{\gamma(\mathrm{P_{th}})}^{\infty} \mathrm{P_e}(r,k,\sqrt{x}) f_{\mathcal{X}_2^2}(x)\mathrm{d}x \tag{3.19}$$

$$\text{and } \mathrm{P}(\mathrm{NAK}_n) = \left(F_{\mathcal{X}_2^2}(\gamma(\mathrm{P_{th}}))\right)^n, \tag{3.20}$$

where $\gamma(\mathrm{P_{th}})$ is calculated by solving the non-linear equation

$$\mathrm{P_e}\left(r,k,\sqrt{\gamma(\mathrm{P_{th}})}\right) = \mathrm{P_{th}}, \tag{3.21}$$

e.g., by means of bisection.

### 3.5.2 HARQ-CC

With HARQ-CC, every retransmission of the UE consists of the same encoded packet as for TI. However, at the $n$th transmission attempt, the BBU uses maximum ratio combining (MRC) of all the $n$ received packets in order to improve the decoding performance. For HARQ-CC, we only consider here a SISO link. This is because MRC requires to compute the weighted sum of the received signals across multiple transmission attempts, where the weight is given by the corresponding scalar channel for a SISO link. Note that SIMO and MISO links could also be tackled in a similar way by considering weights obtained from the effective scalar channels. Due to MRC, at the $n$th retransmission, the received signal can be written as

$$\bar{y}_n = \frac{\sum_{i=1}^{n} H_i^* y_i}{\bar{S}_n}, \tag{3.22}$$

or equivalently as

$$\bar{y}_n = \bar{S}_n x + \bar{w}_n, \tag{3.23}$$

where $y_n$ is the $n$th received packet, the noise $\bar{w}_n$ is distributed as $\mathcal{CN}(0,1)$ and the effective channel gain of the combined signal is given by $\bar{S}_n = \sqrt{\sum_{i=1}^{n} |H_i|^2}$.

**Ideal Feedback** The probability that the BBU does not decode correctly when the effective SNR is $\bar{S}_n^2$ is given as $\mathrm{P_e}(r, k, \bar{S}_n)$. Let $\bar{\mathrm{D}}_n$ denote the event that the $n$th transmission is not decoded correctly at the BBU. The probability of the event $\mathrm{NAK}_n$ is then given as $\mathrm{P}(\mathrm{NAK}_n) = \mathrm{P}(\bigcap_{j=1}^{n} \bar{\mathrm{D}}_j)$, which can be upper bounded, using the chain rule of probability, as

$$\mathrm{P}(\mathrm{NAK}_n) = \mathrm{P}\left(\bar{\mathrm{D}}_n\right)\mathrm{P}\left(\bar{\mathrm{D}}_{n-1}|\bar{\mathrm{D}}_n\right)\cdots\mathrm{P}\left(\bar{\mathrm{D}}_1|\bigcap_{j=2}^{n}\bar{\mathrm{D}}_j\right) \leq \mathrm{P}\left(\bar{\mathrm{D}}_n\right) = \mathrm{E}\left[\mathrm{P}_\mathrm{e}(r,k,\bar{S}_n)\right]. \tag{3.24}$$

The usefulness of the bound (3.24) for small values of k will be validated in Section 3.7 by means of a comparison with Monte Carlo simulations. We also refer to [65] where the same bound is proposed as an accurate approximation of the probability of error for HARQ-CC. We note that the inequality (3.24) is asymptotically tight in the limit of a large blocklength, since the limit $\mathrm{P}(\bar{\mathrm{D}}_m|\bigcap_{j=m+1}^{n}\bar{\mathrm{D}}_j) \to 1$ as $k \to \infty$ holds for a fixed $r$ due to (3.12) and to the inequality $\bar{S}_n \geq \bar{S}_m$ for $n \geq m$. The usefulness of the bound (3.24) for small values of $k$ will be validated in Section 3.7 by means of a comparison with Monte Carlo simulations. Since the effective SNR is distributed as $\bar{S}_n^2 = \sum_{i=1}^{n}|H_i|^2 \sim \mathcal{X}_{2n}^2$, the bound (3.24) can be calculated as

$$\mathrm{P}(\mathrm{NAK}_n) \leq \int_0^\infty \mathrm{P}_\mathrm{e}(r,k,\sqrt{x})f_{\mathcal{X}_{2n}^2}(x)\mathrm{d}x. \tag{3.25}$$

**Local Feedback** With local feedback, the RRH decision is made according to the rule $\mathrm{P}_\mathrm{e}(r,k,\bar{S}_n) \lessgtr_{\mathrm{NAK}}^{\mathrm{ACK}} \mathrm{P}_\mathrm{th}$, for a threshold $\mathrm{P}_\mathrm{th}$ to be optimized. Similar to (3.17) and (3.18), we can compute the probabilities

$$\mathrm{P}(\mathrm{D}_n|\mathrm{ACK}_n) = 1 - \mathrm{E}\Big[\mathrm{P}_\mathrm{e}(r,k,\bar{S}_n)|\{\mathrm{P}_\mathrm{e}(r,k,\bar{S}_{n-1}) > \mathrm{P}_\mathrm{th}\}\bigcap\{\mathrm{P}_\mathrm{e}(r,k,\bar{S}_n) \leq \mathrm{P}_\mathrm{th}\}\Big] \tag{3.26}$$

$$\text{and } \mathrm{P}(\mathrm{NAK}_n) = \mathrm{P}[\mathrm{P}_\mathrm{e}(r,k,\bar{S}_n) > \mathrm{P}_\mathrm{th}]. \tag{3.27}$$

Note that in (3.26)-(3.27) we used the fact that, if the condition $\mathrm{P_e}(r,k,\bar{S}_n) > \mathrm{P_{th}}$ holds, then we also have the inequality $\mathrm{P_e}(r,k,\bar{S}_i) > \mathrm{P_{th}}$ for all the indices $i < n$ due to the monotonicity of the probability $\mathrm{P_e}(r,k,\bar{S})$ as a function of $\bar{S}$. Furthermore, noting that we can write $\bar{S}_n^2 = \bar{S}_{n-1}^2 + |H_n|^2$, where $\bar{S}_{n-1}^2 \sim \mathcal{X}_{2n-2}^2$ and $|H_n|^2 \sim \mathcal{X}_2^2$ are independent, from (3.26) and (3.27), we have

$$\begin{aligned}\mathrm{P}(\mathrm{D}_n|\mathrm{ACK}_n) &= 1 - \mathrm{E}\Big[\mathrm{P_e}(r,k,\bar{S}_n)|\left\{\bar{S}_{n-1}^2 < \gamma(\mathrm{P_{th}})\right\}\bigcap\left\{\bar{S}_{n-1}^2 + |H_n|^2 \geq \gamma(\mathrm{P_{th}})\right\}\Big]\\ &= 1 - \frac{1}{\Delta(\gamma(\mathrm{P_{th}}))}\int_0^{\gamma(\mathrm{P_{th}})}\int_{\gamma(\mathrm{P_{th}})-y}^{\infty}\mathrm{P_e}(r,k,\sqrt{x+y})f_{\mathcal{X}_2^2}(x)f_{\mathcal{X}_{2n-2}^2}(y)\mathrm{d}x\mathrm{d}y\\ \text{and } \mathrm{P}(\mathrm{NAK}_n) &= F_{\mathcal{X}_{2n}^2}(\gamma(\mathrm{P_{th}})),\end{aligned} \tag{3.28}$$

where $\Delta(\gamma(\mathrm{P_{th}}))$ is defined as

$$\Delta(\gamma(\mathrm{P_{th}})) = \int_0^{\gamma(\mathrm{P_{th}})}\int_{\gamma(\mathrm{P_{th}})-y}^{\infty} f_{\mathcal{X}_2^2}(x)f_{\mathcal{X}_{2n-2}^2}(y)\mathrm{d}x\mathrm{d}y. \tag{3.29}$$

### 3.5.3 HARQ-IR

With HARQ-IR, the UE transmits new parity bits at each transmission attempt and the BBU performs decoding based on all the received packets.

**Ideal Feedback** With HARQ-IR, a set of $n$ transmission attempt for a given information messages can be treated as the transmission over $n$ parallel channels (see, e.g., [16]), and hence the error probability at the $n$th transmission can be computed as $\mathrm{P_e}(r,k,\mathcal{H}_n)$ where $\mathcal{H}_n = \mathrm{diag}([\mathbf{H}_1,...,\mathbf{H}_n])$ [84]. Moreover, following the same argument as (3.24), the decoding error at the $n$th transmission can be upper bounded as

$$\mathrm{P}(\mathrm{NAK}_n) \leq \mathrm{P}(\bar{\mathrm{D}}_n) = \mathrm{E}[\mathrm{P_e}(r, k, \mathcal{H}_n)], \tag{3.30}$$

which is tight for large values of $k$ due to (3.12). This can be computed using the known distribution of the eigenvalues of the matrices $\mathbf{H}_i^H \mathbf{H}_i$ and the independence of the matrices $\mathbf{H}_i$ for $i = 1, ..., n$. For instance in the SISO case, we get

$$\mathrm{P}(\mathrm{NAK}_n) \leq \int_0^\infty \cdots \int_0^\infty \mathrm{P_e}(r, k, \mathrm{diag}([\sqrt{x_1}, ..., \sqrt{x_n}])) \prod_{i=1}^{n} f_{\mathcal{X}_2^2}(x_i) \mathrm{d}x_1 \cdots \mathrm{d}x_n. \tag{3.31}$$

**Local Feedback** With local feedback, at the $n$th retransmission, the RRH sends feedback to the UE according to the rule $\mathrm{P_e}(r, k, \mathcal{H}_n) \lessgtr_{\mathrm{NAK}}^{\mathrm{ACK}} \mathrm{P_{th}}$. Due to the monotonicity of the probability $\mathrm{P_e}(r, k, \mathcal{H}_n)$ as a function of each eigenvalue, we have that the probability $\mathrm{P_e}(r, k, \mathcal{H}_n)$ is no larger than $\mathrm{P_e}(r, k, \mathcal{H}_{n-1})$. Therefore, similar to CC, we can calculate

$$\mathrm{P}(\mathrm{D}_n | \mathrm{ACK}_n) = 1 - \mathrm{E}[\mathrm{P_e}(r, k, \mathcal{H}_n) | \mathcal{A}(\mathrm{P_{th}})] \tag{3.32}$$

$$\text{and } \mathrm{P}(\mathrm{NAK}_n) = \mathrm{P}(\mathrm{P_e}(r, k, \mathcal{H}_n) > \mathrm{P_{th}}), \tag{3.33}$$

where we have defined the event $\mathcal{A}(\mathrm{P_{th}}) = \{\{\mathrm{P_e}(r,k,\mathcal{H}_{n-1}) > \mathrm{P_{th}}\} \bigcap \{\mathrm{P_e}(r,k,\mathcal{H}_n) \leq \mathrm{P_{th}}\}\}$. For the SISO case, we can calculate these quantities as

$$\mathrm{P}(\mathrm{D}_n|\mathrm{ACK}_n) = 1 - \frac{1}{\Delta(\mathrm{P_{th}})} \int_0^\infty \cdots \int_0^\infty \mathrm{P_e}(r,k,\mathrm{diag}([\sqrt{x_1},...,\sqrt{x_n}])) \\ \mathbf{1}\left(\mathcal{A}(\mathrm{P_{th}})\right) \prod_{i=1}^{n} f_{\mathcal{X}_2^2}(x_i)\mathrm{d}x_1 \cdots \mathrm{d}x_n$$

$$\text{and } \mathrm{P}(\mathrm{NAK}_n) = \int_0^\infty \cdots \int_0^\infty \mathbf{1}\left(\mathrm{P_e}(r,k,\mathrm{diag}([\sqrt{x_1},...,\sqrt{x_n}])) > \mathrm{P_{th}}\right) \prod_{i=1}^{n} f_{\mathcal{X}_2^2}(x_i)\mathrm{d}x_1 \cdots \mathrm{d}x_n, \tag{3.34}$$

where

$$\Delta(\mathrm{P_{th}}) = \int_0^\infty \cdots \int_0^\infty \mathbf{1}\left(\mathcal{A}(\mathrm{P_{th}})\right) \prod_{i=1}^{n} f_{\mathcal{X}_2^2}(x_i)\mathrm{d}x_1 \cdots \mathrm{d}x_n. \tag{3.35}$$

## 3.6 Analysis of User-Centric Low-Latency Local Feedback for C-RAN

In this section, we turn to the analysis of the user centric low-latency local feedback schemes introduced in Section 3.3.2 for C-RAN. Throughout, we focus on HARQ-IR for its practical relevance, see, e.g., [24]. Furthermore, we consider the case where each RRH has only one receiving antenna, i.e., $m_{r,l} = 1$ for $l = 1,..,L$. Extensions to other HARQ protocols and to scenarios with large number of antennas at the RRHs are possible by following similar arguments as in the previous sections and will not be further discussed here. We recall that in a C-RAN with local feedback, the retransmission decisions are made at the UE based on feedback from the RRHs. We treat separately the case of ideal zero-delay feedback from the BBU, and the hard and soft feedback schemes in the following.

### 3.6.1 Ideal Feedback

We first consider for reference the case of zero-delay ideal feedback from the BBU. Since the BBU jointly processes all the received signals for decoding, at the $n$th retransmission, the signal available at the BBU can be written, using (3.1), as $\mathbf{y}^n = [\mathbf{y}_1^T, ..., \mathbf{y}_n^T]^T$, where

$$\mathbf{y}_n = \sqrt{\frac{s}{m_t}} \mathbf{H}_n \mathbf{x}_n + \mathbf{w}_n, \tag{3.36}$$

with $\mathbf{H}_n = [\mathbf{h}_{1,n}^T \ \mathbf{h}_{2,n}^T \cdots \mathbf{h}_{L,n}^T]^T$ and $\mathbf{w}_n = [w_{1,n} \ w_{2,n} \cdots w_{L,n}]^T$. We emphasize that we denoted here as $\mathbf{h}_{l,n}$ instead of $\mathbf{H}_{l,n}$ the vector containing the channel coefficients between the UE and $l$th RRH in the $n$th retransmission, so as to stress the focus on single-antenna RRHs. The effective received signal is hence given by

$$\mathbf{y}^n = \sqrt{\frac{s}{m_t}} \mathcal{H}_n [\mathbf{x}_1^T \cdots \mathbf{x}_n^T]^T + [\mathbf{w}_1^T \cdots \mathbf{w}_n^T]^T, \tag{3.37}$$

with $\mathcal{H}_n = \text{diag}([\mathbf{H}_1, ..., \mathbf{H}_n])$. Therefore, the decoding error probability at the $n$th transmission is given by $\mathrm{P_e}(r, k, \mathcal{H}_n)$.

The C-RAN performance in terms of throughput and the probability of success under ideal feedback can be obtained following the discussion in Section 3.4 by computing the probability $\mathrm{P}(\mathrm{RTX}_n)$ that a retransmission is required at the $n$th transmission attempt. This can be bounded similar to (3.30) as $\mathrm{P}(\mathrm{RTX}_n) \leq \mathrm{P}(\bar{\mathrm{D}}_n) = \mathrm{E}[\mathrm{P_e}(r, k, \mathcal{H}_n)]$.

### 3.6.2 Hard Feedback Scheme

With the hard feedback low-latency scheme described in Section 3.3.2, each RRH calculates its own decoding error probability $\mathrm{P_e}(r, k, \mathcal{H}_{l,n})$ with

$\mathcal{H}_{l,n} = \mathrm{diag}(\mathbf{h}_{l,1}\ \mathbf{h}_{l,2} \cdots \mathbf{h}_{l,n})$ and uses the rule (3.3), which reduces to

$$\mathrm{P_e}(r, k, \mathcal{H}_{l,n}) \underset{\mathrm{NAK}}{\overset{\mathrm{ACK}}{\lessgtr}} \mathrm{P_{th}}. \tag{3.38}$$

Each RRH sends a single bit indicating the ACK/NAK feedback to the UE. The UE decides that a retransmission is necessary as long as all the RRHs return a NAK message, and it stops retransmission otherwise.

Throughput and probability of success can be computed as detailed in Section 3.4 by using the following probabilities

$$\mathrm{P}(\mathrm{D}_n|\mathrm{STOP}_n) = 1 - \mathrm{E}\left[\mathrm{P_e}(r, k, \mathcal{H}_n)\middle|\prod_{l=1}^{L}\mathbf{1}\left(\mathrm{P_e}(r, k, \mathcal{H}_{l,n}) > \mathrm{P_{th}}\right) = 0\right] \tag{3.39}$$

$$\text{and } \mathrm{P}\left(\mathrm{RTX}_n\right) = \mathrm{P}\left(\prod_{l=1}^{L}\mathbf{1}\left(\mathrm{P_e}(r, k, \mathcal{H}_{l,n}) > \mathrm{P_{th}}\right) = 1\right). \tag{3.40}$$

The above probabilities can be calculated similar to the equations derived in Section 3.5 by averaging over the distribution of the eigenvalues of the involved channel matrices.

### 3.6.3 Soft Feedback Scheme

With the soft feedback introduced in Section 3.3.2, each RRH quantizes the local CSI $\mathbf{h}_{l,n}$ with $b$ bits. From the $b$ feedback bits received from each RRH, the UE obtains the quantized channel vectors $\hat{\mathbf{h}}_{l,n}$ for $l \in \{1, ..., L\}$. Based of these, the decision (3.4) is adopted, which reduces to

$$\mathrm{P_e}(r, k, \hat{\mathcal{H}}_n) \underset{\mathrm{RTX}}{\overset{\mathrm{STOP}}{\lessgtr}} \mathrm{P_{th}}, \tag{3.41}$$

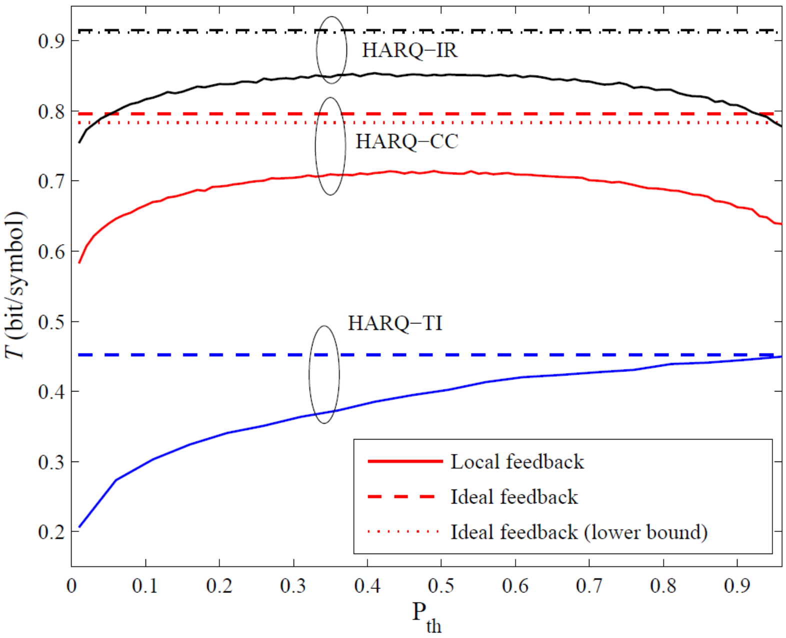


**Figure 3.6** Throughput versus threshold $\mathrm{P_{th}}$ for ideal feedback and local feedback in a D-RAN system ($s = 3$ dB, $n_{max} = 5$, $r = 2$ bit/symbol, $k = 50$, $m_t = 1$ and $m_r = 1$).

where $\hat{\mathcal{H}}_n = \mathrm{diag}(\hat{\mathbf{H}}_1, ..., \hat{\mathbf{H}}_n)$ and $\hat{\mathbf{H}}_n = [\hat{\mathbf{h}}_{1,n}^T \cdots \hat{\mathbf{h}}_{2,n}^T \cdots \hat{\mathbf{h}}_{L,n}^T]^T$ collect the quantized CSI. Accordingly, we can compute the desired probabilities as

$$\mathrm{P}(\mathrm{D}_n|\mathrm{STOP}_n) = 1 - \mathrm{E}[\mathrm{P_e}(r, k, \mathcal{H}_n)|\mathrm{P_e}(r, k, \hat{\mathcal{H}}_n) \leq \mathrm{P_{th}}] \tag{3.42}$$

$$\text{and } \mathrm{P}(\mathrm{RTX}_n) = \mathrm{P}(\mathrm{P_e}(r, k, \hat{\mathcal{H}}_n) > \mathrm{P_{th}}). \tag{3.43}$$

The above probabilities can be computed analytically or via Monte Carlo simulations by averaging over the distribution of the eigenvalues similar to Section 3.5.

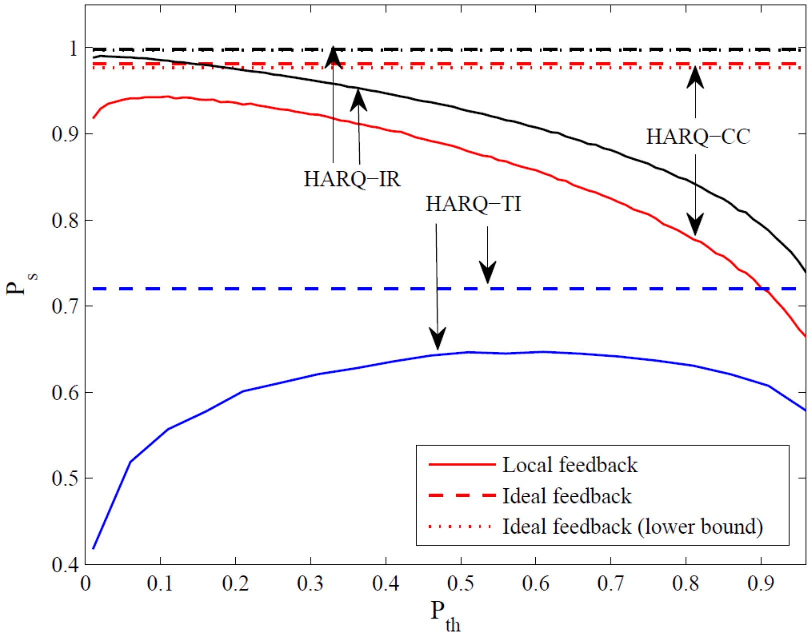


**Figure 3.7** Probability of success versus threshold $\mathrm{P_{th}}$ in a D-RAN system ($s = 3$ dB, $n_{max} = 5$, $r = 2$ bit/symbol, $k = 50$, $m_t = 1$ and $m_r = 1$).

## 3.7 Numerical Results and Discussion

In this section, we validate the analysis presented in the previous sections and provide insights on the performance comparison of ideal and local feedback schemes for D-RAN and C-RAN systems via numerical examples.

### 3.7.1 D-RAN

We first study the optimization of the threshold $\mathrm{P_{th}}$ used in the local feedback schemes. As an exemplifying case study, we consider the D-RAN strategy described in Section 3.5. In Figures 3.6 and 3.7, respectively, the throughput $T$ and the probability of success $\mathrm{P_s}$ are shown versus $\mathrm{P_{th}}$ for $s = 3$ dB, $n_{max} = 5$ retransmissions, $r = 2$ bit/symbol and blocklength $k = 50$ for a SISO link, i.e., for $m_t = 1$ and $m_r = 1$. The curves have been computed using both the equations derived in Section 3.5 and Monte Carlo simulations. The latter refer to the simulation of the HARQ process in which the probability of error at the BBU is modeled by means of the Gaussian approximation. The analytical results are confirmed to match with the Monte Carlo simulations, except for the ideal feedback performance of HARQ-CC and HARQ-IR, for which, as discussed in Section 3.5, the expressions (3.25) and (3.31) yield lower bounds on throughput and probability of success. As seen in the figures, the bounds are very accurate for $k$ as small as 50.

From Figures 3.6 and 3.7, it is also concluded that throughput and probability of success are maximized for different values of threshold $\mathrm{P_{th}}$, with the throughput metric requiring a larger threshold. In fact, a larger value of $\mathrm{P_{th}}$, while possibly causing the acknowledgement of packets that will be incorrectly decoded at the BBU, may enhance the throughput by allowing for the transmission of fresh information in a new HARQ session. This is particularly evident for HARQ-TI, for which setting $\mathrm{P_{th}} = 1$ guarantees a throughput equal to the case of ideal feedback, but at the cost of a loss in the probability of success. It is also observed that more powerful HARQ schemes such as CC and IR are more robust to a suboptimal choice of $\mathrm{P_{th}}$ in terms of throughput, although lower values of $\mathrm{P_{th}}$ are necessary in order to enhance the probability of success by avoiding a premature transmission of an ACK message.

We now illustrate in Figure 3.8 the throughput loss of local feedback as compared to the ideal feedback case, as a function of the blocklength $k$, for two rates $r = 1$ bit/symbol and $r = 3$ bit/symbol for HARQ-CC and HARQ-IR in a D-RAN system. Henceforth, to avoid clutter in the figures, we only show Monte Carlo results, given the match with analysis discussed above. The simulation are performed by setting $s = 4$ dB, $n_{max} = 10$ and we focus on a SISO link. For every value of $k$, the threshold $\mathrm{P_{th}}$ is optimized to maximize the throughput $T$ under the constraint that the probability of success satisfies the requirement $\mathrm{P_s} > 0.99$ (see, e.g., [24] and [51]). It can be seen that, as the blocklength increases, the performance loss of local feedback decreases significantly. This reflects a fundamental insight: The performance loss of

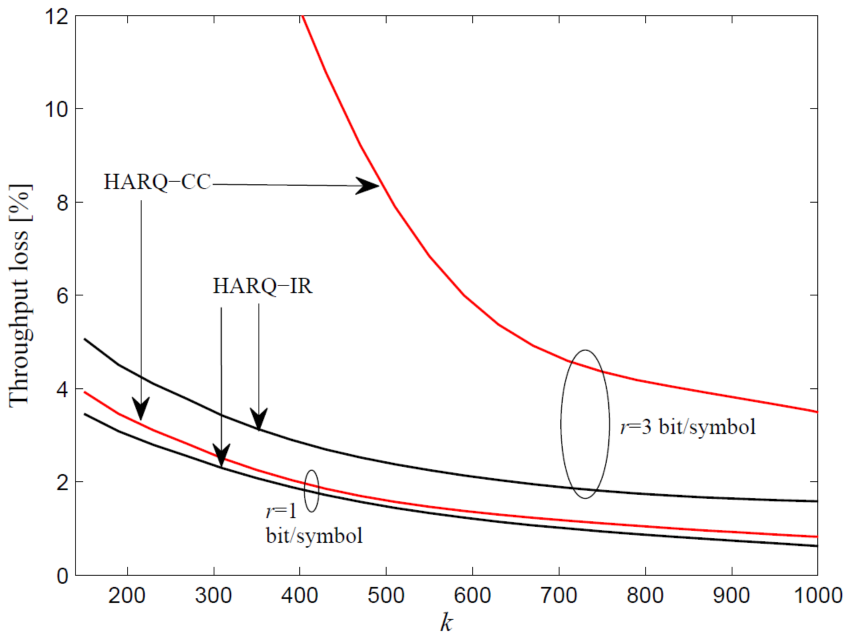


**Figure 3.8** Throughput loss versus blocklength $k$ for HARQ-CC and HARQ-IR in a D-RAN system ($s = 4$ dB, $n_{max} = 10$ $m_t = 1$, $m_r = 1$, $\mathrm{P_s} > 0.99$ for $r = 1$ bit/symbol and $r = 3$ bit/symbol).

local feedback is due to the fact that the local decisions are taken by the RRH based only on channel state information, without reference to the specific channel noise realization that affects the received packet. Therefore, as the blocklength $k$ increases, and hence as the errors due to atypical channel noise realizations become less likely, the local decisions tend to be consistent with the actual decoding outcomes at the BBU. In other words, as the blocklength $k$ grows larger, it becomes easier for the RRH to predict the decoding outcome at the BBU: In the Shannon regime of infinite $k$, successful or unsuccessful decoding depends deterministically on wether the rate $r$ is above or below capacity.

A related conclusion can be reached from Figure 3.9, where we investigate the throughput for MIMO ($m_t = m_r = m$), MISO ($m_t = m$ and $m_r = 1$) and SIMO ($m_t = 1$ and $m_r = m$) links versus the number of antennas $m$ for HARQ-IR, with $s = 1$ dB, $n_{max} = 10$, $r = 5$ bit/symbol, $k = 100$. As in Figure 3.8, the threshold $\mathrm{P_{th}}$

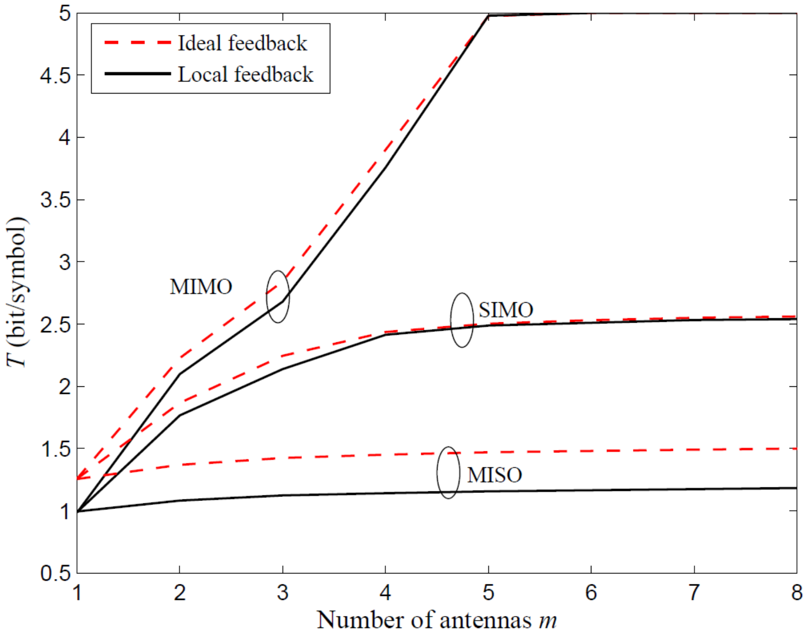


**Figure 3.9** Throughput versus the number of antennas for MISO, SIMO and MIMO with HARQ-IR in a D-RAN system ($s = 1$ dB, $n_{max} = 10$, $r = 5$ bit/symbol, $k = 100$ and $\mathrm{P_s} > 0.99$).

is optimized here, and henceforth, to maximize the throughput under the constraint $\mathrm{P_s} > 0.99$. As $m$ grows large, it is seen that the throughput of SIMO and MIMO increases significantly, while, at the same time, the throughput loss of the local feedback decreases. This is due to the fact that increasing the number of receive antennas effectively boosts the received SNR and hence reduces the impact of the noise on the decoding outcome. This is unlike the case with MISO, since an increase in the number of transmit antennas only enhances the diversity order but does not improve the average received SNR.

### 3.7.2 C-RAN

We now turn our attention to the performance of low-latency local feedback for HARQ over C-RAN systems with $L > 1$ single-antenna RRHs and $m_t = 4$ antennas at the UE. Throughout, we consider the throughput of local feedback based on hard or soft feedback, under the constraint $\mathrm{P_s} > 0.99$ on the probability of success. As a reference, we also consider the performance of a D-RAN system, i.e., with $L = 1$, under both ideal and local feedback (we mark the latter as "hard feedback" following the discussion in Section 3.6.2).

For soft feedback, we set different values for the number of feedback bits $b$, including $b = \infty$, with the latter being equivalent to a D-RAN system with three co-located antennas at the RRH (i.e., $m_{r,1} = 3$ and $L = 1$). We use a vector quantizer for each RRH $l$, in which $b' \leqslant b$ bits are used to quantize the channel direction $\mathbf{h}_{l,n}/||\mathbf{h}_{l,n}||$ and $b - b'$ bits for the amplitude $||\mathbf{h}_{l,n}||$. For vector quantization, we generate randomly quantization codebooks with normalized columns (see, e.g., [50]) until finding one for which the constraint on the probability of success is met. The amplitude $||\mathbf{h}_{l,n}||$ of each channel vector is quantized with the remaining $b' - b$ using a quantizer with numerically optimized thresholds. For $b = 3$, $b = 6$, $b = 9$ and $b = 16$, the number of bits used for the quantization of the direction of each channel vector are $b' = 1$, $b' = 4$, $b' = 5$ and $b' = 12$.

In Figure 3.10, the throughput of the schemes outlined above is shown versus the SNR parameter $s$. We first observe that hard feedback, which only require 1 bit of feedback per RRH, is able to improve over the performance of D-RAN, but the throughput is limited by the errors due to the user-centric local decisions based on partial feedback from the RRHs. This limitation is partly overcome by implementing the soft feedback scheme, whose throughput increases for a growing feedback rate. Note that, even with an infinite feedback rate, the performance of local feedback still exhibits a gap as compared to ideal feedback for the same reasons discussed above for D-RAN systems. Also, the flattening of the throughput of less performing schemes around $T = 2.5$ for intermediate SNR levels is due to the need to carry out at least two retransmissions unless the SNR is sufficiently large (see, e.g., [76]).

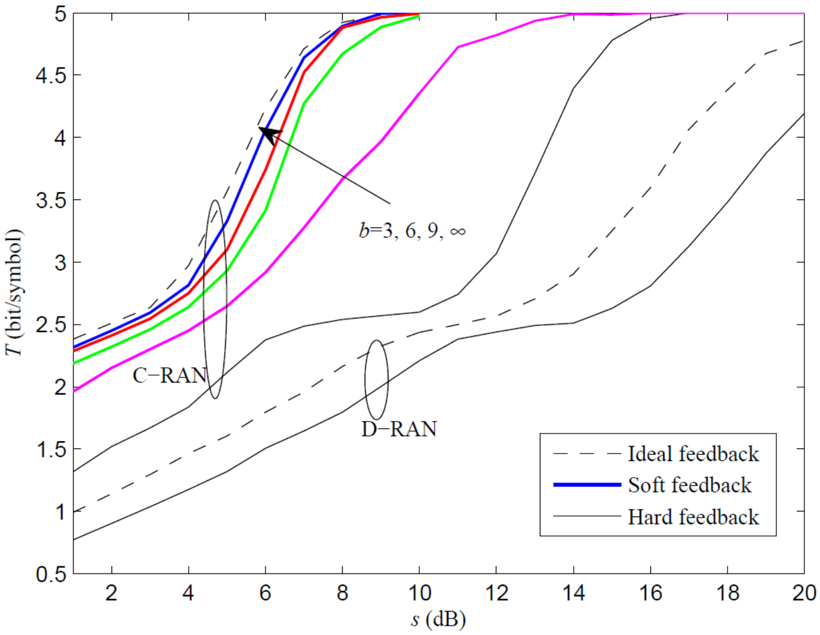


**Figure 3.10** Throughput versus SNR $s$ for D-RAN ($L = 1$) and C-RAN ($L = 3$) systems ($n_{max} = 10$, $r = 5$ bit/symbol, $k = 100$, $\mathrm{P_s} > 0.99$, $m_t = 4$, $m_{r,l} = 1$).

We finally show in Figure 3.11 the throughput of ideal and soft feedback schemes versus the blocklength $k$ for a C-RAN system with $L = 2$ and $L = 3$. We observe that, in a C-RAN system with a sufficiently small feedback rate such as $b = 3$ and $b = 6$, an increase in the blocklength $k$ does not significantly increase the throughput, which is limited by the CSI quantization error. However, with a larger $b$, such as $b = 16$, the throughput can be more significantly improved towards the performance of ideal feedback, especially for a smaller number of RRHs.

## 3.8 Concluding Remarks

The performance of D-RAN and C-RAN systems is currently under close scrutiny as limitations due to constraints imposed by fronthaul capacity and latency are increasingly brought to light (see, e.g., [56]). An important enabling technology to bridge the gap between the desired lower cost and higher spectral efficiency of

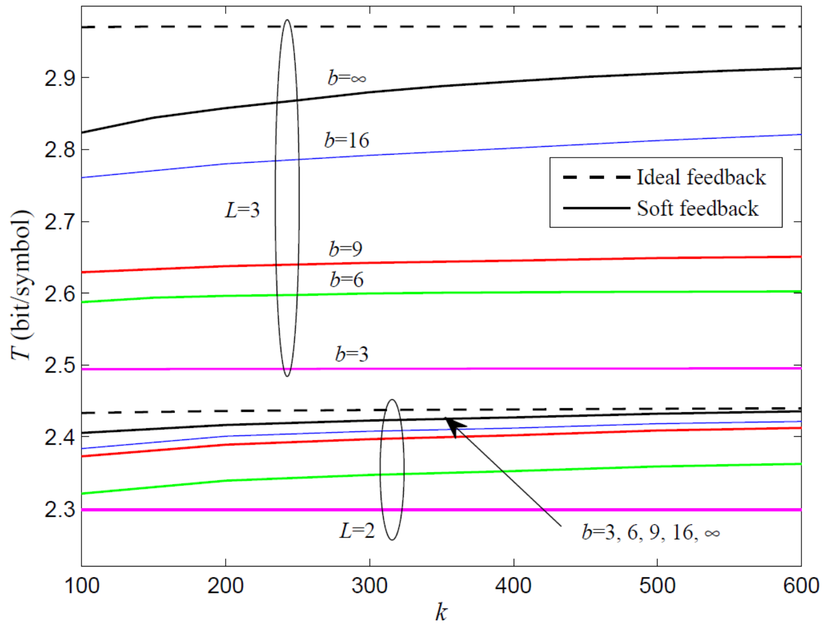


**Figure 3.11** Throughput versus blocklength $k$ for hard and soft feedback schemes in C-RAN with $L = 3$ and $L = 2$. The throughput of the hard feedback scheme for $L = 2$ and $L = 3$ (not shown) are $T = 1.7$ and $T = 1.77$, respectively ($s = 4$ dB, $n_{max} = 10$, $r = 5$ bit/symbol, $\mathrm{P_s} > 0.99$, $m_t = 4$, $m_{r,l} = 1$).

D-RAN and C-RAN and its potentially poor performance in terms of throughput at higher layers is the recently proposed control and data separation architecture [57]. In this context, this chapter has considered D-RAN and C-RAN systems in which retransmission decisions are made at the edge of the network, that is, by the RRHs or UEs, while data decoding is carried out in a centralized fashion at the BBUs.

As shown, for D-RAN, this class of solutions has the potential to yield throughput values close to those achievable with ideal zero-delay feedback from the BBUs, particularly when the packet length is sufficiently long or the number of received antennas is large enough. For C-RAN, it was argued that multi-bit feedback messages from the RRHs are called for in order to reduce the throughput loss and a specific scheme based on vector quantization was proposed to this end.

Interesting future work include the analysis of control and data separation architectures for C-RAN systems for the purpose of user detection activity in random access in scenarios with a massive number of devices.

# CHAPTER 4

# CLOUD RADIO-MULTISTATIC RADAR: JOINT OPTIMIZATION OF CODE VECTOR AND BACKHAUL QUANTIZATION

This chapter aims at extending the idea of cloud radio networks in radar systems in order to improve the performance of detection. In a multistatic cloud radar system, receive sensors measure signals sent by a transmit element and reflected from a target and possibly clutter, in the presence of interference and noise. The receive sensors communicate over non-ideal backhaul links with a fusion center, or cloud processor, where the presence or absence of the target is determined. The backhaul architecture can be characterized either by an orthogonal-access channel or by a non-orthogonal multiple-access channel. To this end, two backhaul transmission strategies are considered iin this chapter, namely compress-and-forward (CF), which is well suited for the orthogonal-access backhaul, and amplify-and-forward (AF), which leverages the superposition property of the non-orthogonal multiple-access channel. The *joint* optimization of the sensing and backhaul communication functions of the cloud radar system is also studied. Specifically, the transmitted waveform is jointly optimized with backhaul quantization in the case of CF backhaul transmission and with the amplifying gains of the sensors for the AF backhaul strategy. In both cases, the information-theoretic criterion of the Bhattacharyya distance is adopted as a metric for the detection performance. Algorithmic solutions based on successive convex approximation are developed under different assumptions on the available channel state information (CSI).

## 4.1 Introduction

This chapter addresses a distributed radar system that involves sensing and communication: a transmit element illuminates an area of interest, in which a target may be present, and the signals returned from the target are observed by sensors. The sensors have a minimal processing capabilities, but communicate over a backhaul network with a processing center, referred to henceforth as a *fusion center*, where target detection takes place (see Figure 4.1). Such architecture is different from a classical multistatic radar system in which each constituent radar performs the full array of radar functions, including target detection and tracking. The considered architecture is motivated by the proliferation of low-cost, mobile or fixed sensors in the "Internet of Things," which are supported by global synchronization services such as the global positioning system (GPS), and are capable of communicating with a fusion center in the "*cloud*" through a backhaul wireless or wired network. For example, the receive sensors could be mounted on light poles, trucks or unmanned aerial vehicles (UAV's) and could be connected to a wireless access point via Wi-Fi or dedicated mmWave links. As this architecture can be implemented by means of cloud computing technology, we refer to it as "*cloud radar*."

The main purpose of this chapter is to study the interaction between the sensing and backhaul communication functions in a cloud radar architecture, and to develop an understanding of the performance gain to be expected by means of a joint optimization of these two functions, namely of waveform design for sensing and of backhaul transmission.

### 4.1.1 Background

The separate design of radar waveforms, under the assumption of an ideal backhaul, has long been a problem of great interest [7, 54]. For monostatic radar systems, i.e.,

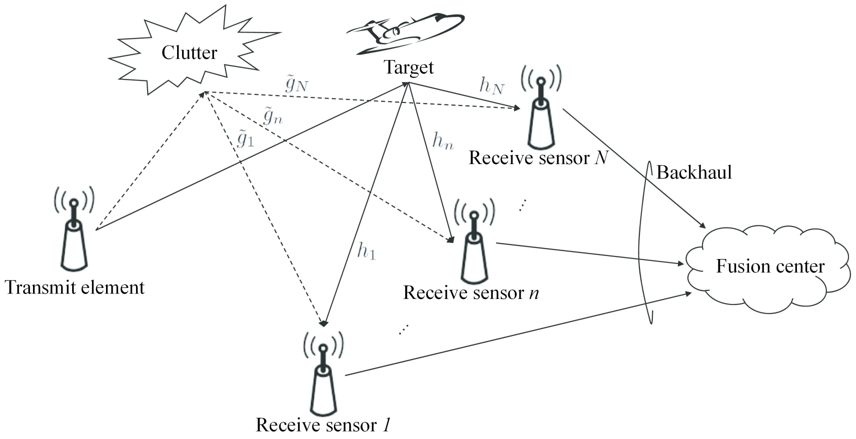


**Figure 4.1** Illustration of a multistatic cloud radar system, which consists of a transmit element, $N$ receive sensors, and a fusion center. All the nodes are configured with a single antenna. The receive sensors are connected to the fusion center via orthogonal-access or non-orthogonal multiple-access backhaul links.

radars with single transmit and receive elements, optimal waveforms for detection in the Neyman-Pearson sense were studied in [41]. In a multistatic radar system, where the signals received by a set of distributed sensors are processed jointly, the performance of the Neyman-Pearson optimal detector is in general too complex to be suitable as a design metric. As a result, various information-theoretic criteria such as the Bhattacharyya distance, the Kullback Leibler divergence, the J-divergence and the mutual information, which can be shown to provide various bounds to the probability of error (missed detection, false alarm and Bayesian risk), have been considered as alternative design metrics [35, 42, 58].

Instead, the separate design of backhaul communication functions, for fixed radar waveforms was studied in [12, 14, 18, 80] under a compress-and-forward (CF) strategy, for which backhaul quantization was optimized, and in [9, 10] under an amplify-and-forward (AF) scheme, for which the power allocation at the sensors was investigated using the minimum mean square error (MMSE) as the performance criterion.

### 4.1.2 Main Contributions

*Unlike prior work, this chapter tackles the problem of jointly designing the waveform, or code vector, and the transmission of the receive sensors over the backhaul.* This approach is motivated by the strong interplay between waveform and backhaul transmission designs. For instance, waveform design may allocate more power at frequencies that are less affected on average by clutter and interference, while the backhaul transmission strategies are adapted accordingly to devote most backhaul resources, namely capacity or power, to the transmission of such frequencies to the fusion center.

Two basic types of backhaul links between the radar receive sensors and the fusion center are considered, namely orthogonal and non-orthogonal access backhaul. In the former, no interference exists between the sensors, as in a wired backhaul, while in the latter, the backhaul forms a multiple-access channel, where channels are subject to mutual interference, as in a wireless backhaul. Furthermore, two standard backhaul transmission schemes are investigated, namely CF and AF. As in the Cloud Radio Access Network (C-RAN) architecture in communication [56], CF is particularly well suited to an orthogonal backhaul architecture: each sensor satisfies the backhaul capacity constraint quantizing the received baseband signals prior to transmission to the fusion center. AF, instead, is better matched to a non-orthogonal multiple-access backhaul: each receive sensor amplifies and forwards the received signal to the fusion center so that the signals transmitted by the receive sensors are superimposed at the fusion center (see, e.g., [9, 10]).

Our specific contributions are as follows:

- CF: The joint optimization of the waveform and the quantization strategy is investigated for CF, with a focus on orthogonal-access backhaul. To reflect practical constraints, only stochastic channel state information (CSI) is assumed on the channel gains between target or clutter and the receive sensors. For an optimization objective, we adopt the information-theoretic criterion of the Bhattacharyya distance in order to account for the detection performance [35, 42, 58].

• AF: The joint optimization of the waveform and the amplifying gains of the receive sensors is studied for AF, by concentrating on non-orthogonal multiple-access backhaul. We adopt the performance criterion and main assumptions of CF. Furthermore, we consider both instantaneous and stochastic CSI on the receive sensors-to-fusion center channels.

Throughout, we assume tractable and well accepted models in order to gain insight into the problem at hand. With this insight gained, subsequent work may explore more detailed configurations.

The rest of this chapter is organized as follows. In Section 4.2, we present the signal model and cover the two types of backhaul links, namely orthogonal-access and non-orthogonal multiple-access backhaul. In Section 4.3, after describing the CF backhaul transmission strategies and reviewing the optimal detectors, we present the optimization of the multistatic cloud radar system with CF. In Section 4.4, we focus on the AF backhaul transmission, and optimize the system with both instantaneous and stochastic CSI under AF. Numerical results are provided in Section 4.5, and, finally, conclusions are drawn in Section 4.6.

## 4.2 System Model

Consider a multistatic cloud radar system consisting of a transmit element, $N$ receive sensors, and a fusion center, or cloud processor, as illustrated in Figure 4.1. The receive sensors communicate with the fusion center over an orthogonal-access backhaul or a non-orthogonal multiple-access backhaul. All the nodes are equipped with a single antenna, and the set of receive sensors is denoted $\mathcal{N} = \{1, \ldots, N\}$.

The system aims to detect the presence of a single stationary target in a clutter field. To this end, each sensor receives a noisy version of the signal transmitted by the transmit element and reflected from the surveillance area, which is conveyed to the fusion center on the backhaul channels after either quantizing or amplifying the received signals as discussed below. It is assumed that perfect timing information is available at the fusion center, such that samples of the received signal may be associated with specific locations in some coordinate system. For such a location, and based on all the signals forwarded from the different receive sensors, the fusion center makes a decision about the presence of the target (see, e.g. [9, 10, 12, 14, 18, 58, 80]). Note that, as argued in [42], the assumption of stationary target and scatterers can be regarded as a worst-case scenario for more general set-ups with non-zero Doppler.

We consider a pulse compression radar in which the transmitted signal given in baseband form is

$$s(t) = \sum_{k=1}^{K} x_k \phi(t - (k-1)T_c), \tag{4.1}$$

where $\phi(t)$ is, for example, a square root Nyquist with chip rate $1/T_c$, so that $\{\phi(t - (k-1)T_c)\}_{k=1}^{K}$ are orthonormal; and $\{x_k\}_{k=1}^{K}$ is a sequence of (deterministic) complex coefficients that modulate the waveform. The vector $\boldsymbol{x} = [x_1 \ \cdots \ x_K]^T$ is referred to as *waveform* or *code vector*, on which we impose the transmit power constraint $\boldsymbol{x}^H\boldsymbol{x} \leq P_T$. The design of the waveform $\boldsymbol{x}$ determines both target and clutter response, and thus has a key role in the performance of the radar system.

The baseband signal received at the sensor $n \in \mathcal{N}$, which is backscattered by a stationary target, can be expressed as

$$r_n(t) = h_n s(t - \tau_n) + c_n(t) + w_n(t), \tag{4.2}$$

where $h_n$ is the random complex amplitude of the target return, which includes the effects of the channel and follows a Swerling I target-type model having a Rayleigh envelope, i.e., $h_n \sim \mathcal{CN}(0, \sigma_{t,n}^2)$; $c_n(t)$ represents the clutter component; $w_n(t)$ is a Gaussian random process representing the signal-independent interference, which aggregates the contributions of thermal noise, interference and jamming and is assumed to be correlated over time, as detailed below; and $\tau_n$ is the propagation delay for the path from the transmit element to the target and thereafter to the sensor $n$, which is assumed to satisfy the condition $\tau_n \geq KT_c$ in order for the target to be detectable. The clutter component $c_n(t)$ consists of signal echoes generated by stationary point scatterers, whose echoes have independent return amplitudes and arrival times. Accordingly, the clutter component $c_n(t)$ is expressed as

$$c_n(t) = \sum_{v=1}^{N_c} g_{n,v} s(t - \tau_{n,v}), \tag{4.3}$$

where $N_c$ is the number of point scatters; $g_{n,v}$ is the amplitude of the return from scatterer $v$; and $\tau_{n,v}$ is the propagation delay for the path from the transmit element to the scatterer $v$ and to the sensor $n$, which satisfies the condition $\tau_{n,v} \leq KT_c$.

After matched filtering of the received signal (4.2) with the impulse response $\phi^*(-t)$, and after range-gating by sampling the output of the matched filter at the chip rate, the discrete-time signal at receive sensor $n$ for $n \in \mathcal{N}$ can be written as

$$r_{n,k} = h_n x_k + \tilde{g}_n x_k + w_{n,k}, \tag{4.4}$$

where $r_{n,k}$ is the output of the matched filter at the receive sensor $n$ sampled at time $t = (k-1)T_c + \tau_n$; the term $\tilde{g}_n = \sum_{v=1}^{N_c} g_{n,v} \Psi(\tau_n - \tau_{n,v})$ with $\Psi(t) \triangleq \int_{-\infty}^{\infty} \phi(\tau - t)\phi^*(\tau) d\tau$ being the auto-correlation function of $\phi(t)$, represents the contribution of clutter scatterers, which can be modeled, invoking the central limit theorem, as a zero mean Gaussian random variable with a given variance $\sigma_{c,n}^2$ (see [58, Appendix A]); and $w_{n,k}$ is the $k$th sample of $w_n(t)$ after matched filtering at the sensor $n$.

In vector notation, we can write (4.4) as

$$\boldsymbol{r}_n = \boldsymbol{s}_n + \boldsymbol{c}_n + \boldsymbol{w}_n, \tag{4.5}$$

where we defined $\boldsymbol{r}_n \triangleq [r_{n,1} \ \cdots \ r_{n,K}]^T$, $\boldsymbol{s}_n \triangleq h_n \boldsymbol{x}$ and $\boldsymbol{c}_n \triangleq \tilde{g}_n \boldsymbol{x}$; and the noise vector $\boldsymbol{w}_n \triangleq [w_{n,1} \ \cdots \ w_{n,K}]^T$ follows a zero-mean Gaussian distribution with temporal correlation $\boldsymbol{\Omega}_{w,n}$, i.e., $\boldsymbol{w}_n \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{\Omega}_{w,n})$. The variables $h_n$, $\tilde{g}_n$ and $\boldsymbol{w}_n$ for all $n \in \mathcal{N}$, are assumed to be independent for different values of $n$ under the assumption that the receive sensors are sufficiently separated [42]. Moreover, their second-order statistics $\sigma_{t,n}^2$, $\sigma_{c,n}^2$ and $\boldsymbol{\Omega}_{w,n}$ are assumed to be known to the fusion center, for all $n \in \mathcal{N}$, e.g., from prior measurements or prior information [29, 30].

To summarize, the signal received at sensor $n$ can be written as

$$\mathcal{H}_0 : \boldsymbol{r}_n = \boldsymbol{c}_n + \boldsymbol{w}_n, \tag{4.6a}$$

$$\mathcal{H}_1 : \boldsymbol{r}_n = \boldsymbol{s}_n + \boldsymbol{c}_n + \boldsymbol{w}_n, \ \ n \in \mathcal{N}, \tag{4.6b}$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ represent the hypotheses under which the target is absent or present, respectively.

In the rest of this section, we detail the assumed model for both orthogonal-access and non-orthogonal multiple-access backhaul.

**Orthogonal-access Backhaul:** For the orthogonal-access backhaul case, each receive sensor $n$ is connected to the fusion center via an orthogonal link of limited capacity $C_n$ bits per received sample. The capacity $C_n$ is assumed to be known to the fusion center for all $n \in \mathcal{N}$ and to change sufficiently slowly so as to enable the adaptation of the waveform and of the transmission strategy of the sensors to the values of the capacities $C_n$ for all $n \in \mathcal{N}$.

**Non-orthogonal Multiple-access Backhaul:** For the non-orthogonal multiple-access backhaul, the signal received at the fusion center is the superposition of the signals sent by all receive sensors, where channels are subject to mutual interference. Accordingly, the received signal at the fusion center $\tilde{\boldsymbol{r}} = [\tilde{r}_1 \ \cdots \ \tilde{r}_K]^T$ is given by

$$\tilde{\boldsymbol{r}} = \sum_{n=1}^{N} f_n \boldsymbol{t}_n + \boldsymbol{z}, \tag{4.7}$$

where $\boldsymbol{t}_n = [t_{n,1} \ \cdots \ t_{n,K}]^T$ is the signal sent by the receive sensor $n$ on the backhaul to the fusion center; $f_n$ is the complex-valued channel gain between the receive sensor $n$ and the fusion center; and $\boldsymbol{z} = [z_1 \ \cdots \ z_K]^T$ is the noise vector having a zero-mean Gaussian distribution with correlation matrix $\boldsymbol{\Omega}_z$, i.e., $\boldsymbol{z} \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{\Omega}_z)$. Based on prior information or measurements, the second-order statistics of the channel gains between the target and the receive sensors, and of the noise terms, namely $\sigma^2_{t,n}$, $\sigma^2_{c,n}$, $\boldsymbol{\Omega}_{w,n}$ and $\boldsymbol{\Omega}_z$, are assumed to be known to the fusion center for all $n \in \mathcal{N}$. The channel between receive sensors and fusion center $\boldsymbol{f} = [f_1 \ \cdots \ f_N]^T$ are also assumed to be known at the fusion center, via training and channel estimation.

## 4.3 CF Backhaul Transmission

In this section, we consider orthogonal-access backhaul and CF transmission. With CF, each receive sensor quantizes the received vector $\boldsymbol{r}_n$ in (4.6), and sends a quantized version of $\boldsymbol{r}_n$ to the fusion center. Note that, since the receive sensor does not know whether the target is present or not, the quantizer cannot depend on the correct hypothesis $\mathcal{H}_0$ or $\mathcal{H}_1$. In order to facilitate analysis and design, we follow the standard random coding approach of rate-distortion theory of modeling the effect of quantization by means of an additive quantization noise (see, e.g., [6, 77]) as in

$$\tilde{\boldsymbol{r}}_n = \boldsymbol{r}_n + \boldsymbol{q}_n, \tag{4.8}$$

where $\tilde{\boldsymbol{r}}_n = [\tilde{r}_{n,1} \ \cdots \ \tilde{r}_{n,K}]^T$ is the quantized signal vector of $\boldsymbol{r}_n$; and $\boldsymbol{q}_n \sim \mathcal{CN}(\mathbf{0}, \boldsymbol{\Omega}_{q,n})$ is the quantization error vector, which is characterized by a covariance matrix $\boldsymbol{\Omega}_{q,n}$. Based on random coding arguments, while (4.8) holds on an average over randomly generated quantization codebooks, the results derived in this chapter can be obtained by means of some (deterministic) high-dimensional vector quantizer (see, e.g., [28]). For instance, as discussed in [85], a Gaussian quantization noise $\boldsymbol{q}_n$ with any covariance $\boldsymbol{\Omega}_{q,n}$ can be realized in practice via a linear transform, obtained from the eigenvectors of $\boldsymbol{\Omega}_{q,n}$, followed by a multi-dimensional dithered lattice quantizer such as Trellis Coded Quantization (TCQ) [55].

Based on (4.8), the signal received at the fusion center from receive sensor $n$ is given as

$$\begin{aligned} \mathcal{H}_0 : \ \tilde{\boldsymbol{r}}_n &= \boldsymbol{c}_n + \boldsymbol{w}_n + \boldsymbol{q}_n, \\ \mathcal{H}_1 : \ \tilde{\boldsymbol{r}}_n &= \boldsymbol{s}_n + \boldsymbol{c}_n + \boldsymbol{w}_n + \boldsymbol{q}_n. \end{aligned} \tag{4.9}$$

As further elaborated in the following, the covariance matrix $\boldsymbol{\Omega}_{q,n}$ determines the bit rate required for backhaul communication between the receive sensor $n$ and the fusion center [6, 77] and is subject to design.

To set the model (4.9) in a more convenient form, the signal received at the fusion center is whitened with respect to the overall additive noise $\boldsymbol{c}_n + \boldsymbol{w}_n + \boldsymbol{q}_n$, and the returns from all sensors are collected, leading to the model

$$
\begin{aligned}
&\mathcal{H}_0: \ \boldsymbol{y} \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I}), \\
&\mathcal{H}_1: \ \boldsymbol{y} \sim \mathcal{CN}(\boldsymbol{0}, \boldsymbol{DSD} + \boldsymbol{I}),
\end{aligned}
\tag{4.10}
$$

where $\boldsymbol{y} = [\boldsymbol{y}_1^T \ \cdots \ \boldsymbol{y}_N^T]^T$, $\boldsymbol{y}_n = \boldsymbol{D}_n \tilde{\boldsymbol{r}}_n$, $\boldsymbol{D}_n$ is the whitening matrix associated with the receive sensor $n$ and is given by $\boldsymbol{D}_n = (\sigma_{c,n}^2 \boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n})^{-1/2}$, $\boldsymbol{D}$ is the block diagonal matrix $\boldsymbol{D} = \text{diag}\{\boldsymbol{D}_1, ..., \boldsymbol{D}_N\}$, and $\boldsymbol{S}$ is the block diagonal matrix $\boldsymbol{S} = \text{diag}\{\sigma_{t,1}^2 \boldsymbol{x}\boldsymbol{x}^H, ..., \sigma_{t,N}^2 \boldsymbol{x}\boldsymbol{x}^H\}$. The detection problem formulated in (4.10) has the standard Neyman-Pearson solution given by the test

$$
\boldsymbol{y}^H \boldsymbol{T} \boldsymbol{y} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \nu,
\tag{4.11}
$$

where we have defined $\boldsymbol{T} = \boldsymbol{DSD}(\boldsymbol{DSD} + \boldsymbol{I})^{-1}$, and the threshold $\nu$ is set based on the tolerated false alarm probability [40].

In the rest of this section, we aim to find the optimum code vector $\boldsymbol{x}$ and quantization error covariance matrices $\boldsymbol{\Omega}_{q,n}$ in (4.9), for given backhaul capacity constraints $C_n$, for all $n \in \mathcal{N}$. Before we proceed, for reference, we first discuss the standard distributed detection approach that combines hard local decisions at the receive sensors and a majority-rule detection at the fusion center (see, e.g., [8, 79]).

### 4.3.1 Distributed Detection

Here, we describe the standard distributed detection approach applied to multistatic radar system (see, e.g., [8, 79]). With this approach, each receive sensor $n$ makes its own decision based on the likelihood test given by $\boldsymbol{y}_n^H \boldsymbol{T}_n \boldsymbol{y}_n \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma_n$, where $\gamma_n$ is the threshold for receive sensor $n$, which is calculated based on the tolerated false alarm probability [40], and we have defined $\boldsymbol{T}_n = \boldsymbol{D}_n\boldsymbol{S}_n\boldsymbol{D}_n(\boldsymbol{D}_n\boldsymbol{S}_n\boldsymbol{D}_n+\boldsymbol{I})^{-1}$ with $\boldsymbol{y}_n = \boldsymbol{D}_n\boldsymbol{r}_n$, $\boldsymbol{D}_n = (\sigma_{c,n}^2\boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n})^{-1/2}$ and $\boldsymbol{S}_n = \sigma_{t,n}^2\boldsymbol{x}\boldsymbol{x}^H$, for all $n \in \mathcal{N}$. The receive sensors transmit the obtained one-bit hard decision to the fusion center. Note that this scheme is feasible as long as the backhaul capacity available for each receive sensor-to-fusion center channel is larger than or equal to $1/K$ bits/sample, i.e., $C_n \geq 1/K$, for $n \in \mathcal{N}$. The fusion center decides on the target's presence based on the majority rule: if the number of receive sensors $k$ that decide for $\mathcal{H}_0$ satisfies $k \geq N/2$, the fusion center chooses $\mathcal{H}_0$, and vice versa if $k \leq N/2$.

### 4.3.2 Performance Metrics and Constraints

To start the analysis of the cloud radar system, we discuss the criterion that is adopted to account for the detection performance, namely the Bhattacharyya distance and the approach used to model the effect of the quantizers at the receive sensors.

**Bhattacharyya Distance:** For two zero-mean Gaussian distributions with covariance matrix of $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$, the Bhattacharyya distance $\mathcal{B}$ is given by [35]

$$\mathcal{B} = \log\left(\frac{|0.5(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)|}{\sqrt{|\boldsymbol{\Sigma}_1||\boldsymbol{\Sigma}_2|}}\right). \tag{4.12}$$

Therefore, for the signal model (4.9), the Bhattacharyya distance between the distributions under the two hypotheses can be calculated as

$$\begin{aligned}\mathcal{B}(\boldsymbol{x}, \boldsymbol{\Omega}_q) &= \log\left(\frac{|\boldsymbol{I} + 0.5\boldsymbol{D}\boldsymbol{S}\boldsymbol{D}|}{\sqrt{|\boldsymbol{I} + \boldsymbol{D}\boldsymbol{S}\boldsymbol{D}|}}\right)\\ &= \sum_{n=1}^{N} \mathcal{B}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n})\\ &= \sum_{n=1}^{N} \log\left(\frac{1 + 0.5\lambda_n}{\sqrt{1+\lambda_n}}\right),\end{aligned} \tag{4.13}$$

where we have made explicit the dependence on $\boldsymbol{x}$ and $\boldsymbol{\Omega}_{q,n}$; $\boldsymbol{\Omega}_q$ collects all the covariance matrices of quantization noise and is given as $\boldsymbol{\Omega}_q = \{\boldsymbol{\Omega}_{q,n}\}_{n\in\mathcal{N}}$; and we have defined

$$\lambda_n = \sigma_{t,n}^2 \boldsymbol{x}^H \left(\sigma_{c,n}^2 \boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n}\right)^{-1} \boldsymbol{x}. \tag{4.14}$$

We observe that (4.13) is valid under the assumption that the effect of the quantizers can be well approximated by additive Gaussian noise as per (4.9). This is discussed next.

**Quantization:** From rate-distortion theory, a vector quantizer exists that is able to realize the additive quantization noise model (4.8), when operating over a sufficiently large number of measurement vectors (4.6), as long as the capacity $C_n$ is no smaller than the mutual information $I(\boldsymbol{r}_n;\tilde{\boldsymbol{r}}_n)/K$ [28]. For example, a dithered lattice vector quantizer achieves this result [85]. These considerations motivate the selection of the mutual information $I(\boldsymbol{r}_n;\tilde{\boldsymbol{r}}_n)$ as a measure of the backhaul rate required for the transmission to the fusion center.

While the mutual information $I(\boldsymbol{r}_n;\tilde{\boldsymbol{r}}_n)$ depends on the actual hypothesis $\mathcal{H}_0$ or $\mathcal{H}_1$, it is easy to see that $I(\boldsymbol{r}_n;\tilde{\boldsymbol{r}}_n)$ is larger under hypothesis $\mathcal{H}_1$. Based on this, the mutual information $I(\boldsymbol{r}_n;\tilde{\boldsymbol{r}}_n)$ evaluated under $\mathcal{H}_1$ is adopted here as the measure of the bit rate required between receive sensor $n$ and the fusion center. This can be easily calculated as $I(\boldsymbol{r}_n;\tilde{\boldsymbol{r}}_n) = \mathcal{I}_n(\boldsymbol{x},\boldsymbol{\Omega}_{q,n})$ by using the expression of the mutual information for multivariate Gaussian distribution (see, e.g., [77]) with

$$\begin{aligned} \mathcal{I}_n(\boldsymbol{x},\boldsymbol{\Omega}_{q,n}) &= \log\left|\boldsymbol{I} + (\boldsymbol{\Omega}_{q,n})^{-1}\boldsymbol{\Omega}_{w,n}\right| \\ &+ \log\left(1 + (\sigma_{t,n}^2 + \sigma_{c,n}^2)\boldsymbol{x}^H(\boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n})^{-1}\boldsymbol{x}\right), \end{aligned} \tag{4.15}$$

where again we have made explicit the dependence of mutual information on $\boldsymbol{x}$ and $\boldsymbol{\Omega}_{q,n}$.

In the following, we formulate and solve the problem of jointly optimizing the Bhattacharyya distance criterion over the waveform $\boldsymbol{x}$ at the transmit element and over the covariance matrices $\boldsymbol{\Omega}_q$ of the quantizers at the receive sensors in Section 4.3.3 and in Section 4.3.4, respectively.

### 4.3.3 Problem Formulation

The problem of maximizing the Bhattacharyya distance in (4.13) over the waveform $\boldsymbol{x}$ and the covariance matrices $\boldsymbol{\Omega}_q$ under the backhaul capacity constraints is stated as

$$\underset{\boldsymbol{x},\boldsymbol{\Omega}_q}{\text{minimize}}\ \bar{\mathcal{B}}(\boldsymbol{x},\boldsymbol{\Omega}_q)=\sum_{n=1}^{N}\bar{\mathcal{B}}_n(\boldsymbol{x},\boldsymbol{\Omega}_{q_n}) \tag{4.16a}$$

$$\text{s.t.}\quad \mathcal{I}_n(\boldsymbol{x},\boldsymbol{\Omega}_{q,n})\leq KC_n=\bar{C}_n,\ \ n\in\mathcal{N}, \tag{4.16b}$$

$$\boldsymbol{x}^H\boldsymbol{x}\leq P_T, \tag{4.16c}$$

$$\boldsymbol{\Omega}_{q,n}\succeq 0,\ \ n\in\mathcal{N}, \tag{4.16d}$$

where we have formulated the problem as the minimization of the negative distance $\bar{\mathcal{B}}(\boldsymbol{x},\boldsymbol{\Omega}_q) = \sum_{n=1}^{N} \bar{\mathcal{B}}_n(\boldsymbol{x},\boldsymbol{\Omega}_{q,n})$, with $\bar{\mathcal{B}}(\boldsymbol{x},\boldsymbol{\Omega}_q) = -\mathcal{B}(\boldsymbol{x},\boldsymbol{\Omega}_q)$ and $\bar{\mathcal{B}}_n(\boldsymbol{x},\boldsymbol{\Omega}_{q,n}) = -\mathcal{B}_n(\boldsymbol{x},\boldsymbol{\Omega}_{q,n})$, following the standard convention in [15]. The power of the waveform $\boldsymbol{x}$ is constrained not to exceed a prescribed value of transmit power $P_T$. We observe that the constraint (4.16b) ensures that the transmission rate with $K$ chips between each receive sensor and the fusion center is smaller than $\bar{C}_n$, according to the adopted information-theoretic metric. Note also that the problem (4.16) is not a convex program, since the objective function (4.16a) and the constraints (4.16b) are not convex.

### 4.3.4 Proposed Algorithm

Since both functions $\bar{\mathcal{B}}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q_n})$ and $\mathcal{I}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q_n})$ in (4.16) are non-convex in $\boldsymbol{x}$ and $\boldsymbol{\Omega}_{q,n}$, the optimization problem (4.16) is not convex, and hence it is difficult to solve. To obtain a locally optimal solution, we approach the joint optimization of $\boldsymbol{x}$ and $\boldsymbol{\Omega}_q$ in (4.16) via successive convex approximations. Specifically, in an outer loop, Block Coordinate Descent (BCD) is applied to update $\boldsymbol{x}$ and $\boldsymbol{\Omega}_q$ one at a time, while an inner loop implemented via Majorization-Minimization (MM) solves the optimization of $\boldsymbol{x}$ and $\boldsymbol{\Omega}_q$ separately. This approach was first introduced in [43] for a sum-capacity backhaul constraint. By the properties of MM (see, e.g., [34, 67]), the algorithm provides a sequence of feasible solutions with non-increasing cost function, which guarantees convergence of the cost function. Note that, due to the non-convexity of the problem, no claim of convergence to a local or global optimum is made here.

At the $i$th iteration of the outer loop, the optimum waveform $\boldsymbol{x}^{(i)}$ is obtained by solving (4.16) for matrices $\boldsymbol{\Omega}_q = \boldsymbol{\Omega}_q^{(i-1)}$ obtained at the previous iteration; subsequently, the matrices $\boldsymbol{\Omega}_q^{(i)}$ are calculated by solving (4.16) with $\boldsymbol{x} = \boldsymbol{x}^{(i)}$. These two separate optimizations are carried out by the MM method, which, as described in Appendix A, requires the solution of a quadratically constrained quadratic programs (QCQP). The proposed algorithm coupling BCD and MM to solve problem (4.16), is summarized in Table Algorithm 4.1. In Algorithm 4.1, we use the superscript $i$ to identify the iterations of the outer loop, and the superscript $j$ as the index of the inner iteration of the MM method (e.g., $\boldsymbol{x}^{(i,j)}$ indicates the waveform optimized at the $j$th iteration of the inner loop of the MM method and the $i$th iteration of the outer loop). In Appendix A, we present the MM steps and the overall proposed algorithm in detail.

The complexity of Algorithm 4.1 by using standard convex optimization tools is polynomial in $K$ and $N$ since, at each outer iteration, MM requires to solve the problems (C.3) and (C.6), whose sizes of the optimization domains are $K$ and $NK^2$, and numbers of constraints are $N+1$ and $2N$, respectively [15, 53].

## 4.4 AF Backhaul Transmission

In this section, we consider AF transmission on a non-orthogonal multiple-access backhaul. With AF, sensor $n \in \mathcal{N}$ amplifies the received signal $\boldsymbol{r}_n$ in (4.6) and then forwards the amplified signal $\boldsymbol{t}_n = \alpha_n \boldsymbol{r}_n$ to the fusion center, where $\alpha_n$ is is the amplification coefficient at the receive sensor $n$. From (4.7), the fusion center is faced with the following detection hypothesis problem

$$\begin{aligned} \mathcal{H}_0 :\ \tilde{\boldsymbol{r}} &= \sum_{n=1}^{N} f_n \boldsymbol{t}_n + \boldsymbol{z} = \sum_{n=1}^{N} f_n \alpha_n \left(\boldsymbol{c}_n + \boldsymbol{w}_n\right) + \boldsymbol{z}, \\ \mathcal{H}_1 :\ \tilde{\boldsymbol{r}} &= \sum_{n=1}^{N} f_n \boldsymbol{t}_n + \boldsymbol{z} \sum_{n=1}^{N} f_n \alpha_n \left(\boldsymbol{s}_n + \boldsymbol{c}_n + \boldsymbol{w}_n\right) + \boldsymbol{z}. \end{aligned} \tag{4.17}$$

The variables $h_n$, $\tilde{g}_n$, $\boldsymbol{w}_n$, $f_n$ and $\boldsymbol{z}$, for all $n \in \mathcal{N}$, are assumed to be mutually independent. Since only the second-order statistics of the channel gains $h_n$, $n \in \mathcal{N}$, are known to the receive sensors and the fusion center, no coherent gains may be achieved by optimizing the amplifying gains, and hence one can focus, without loss of optimality, only on the receive sensors' power gains $\boldsymbol{p} = [p_1 \cdots p_N]^T$, with $p_n = |\alpha_n|^2$, for $n \in \mathcal{N}$.

As in the CF backhaul transmission in Section 4.3, we can write the hypotheses (4.17) in a standard form by whitening the signal received at the fusion center, and consequently the detection problem can be expressed as (4.10), where we have redefined $\boldsymbol{y} = \boldsymbol{D}\tilde{\boldsymbol{r}}$; $\boldsymbol{D} = (\sum_{n=1}^{N}(|f_n|^2 p_n \sigma_{c,n}^2 \boldsymbol{x}\boldsymbol{x}^H + |f_n|^2 p_n \boldsymbol{\Omega}_{w,n}) + \boldsymbol{\Omega}_z)^{-1/2}$ is the

**Algorithm 4.1** Joint optimization of waveform and quantization noise covariances (4.16)

```
**Initialization (outer loop)**: Initialize $\boldsymbol{x}^{(0)} \in C^{K\times 1}$, $\boldsymbol{\Omega}_q^{(0)} \succeq 0$ and set $i = 0$.
**Repeat (BCD method)**
    $i \leftarrow i + 1$
    **Initialization (inner loop)**: Initialize $\boldsymbol{x}^{(i,0)} =$
    $\boldsymbol{x}^{(i-1)}$ and set $j = 0$.
    **Repeat (MM method for $\boldsymbol{x}^{(i)}$)**
        $j \leftarrow j + 1$
        Find $\boldsymbol{x}^{(i,j)}$ by solving the problem (C.3) with
        $\boldsymbol{\Omega}_q = \boldsymbol{\Omega}_q^{(i-1)}$.
    **Until** a convergence criterion is satisfied.
    **Update** $\boldsymbol{x}^{(i)} \leftarrow \boldsymbol{x}^{(i,j)}$
    **Initialization (inner loop)**: Initialize $\boldsymbol{\Omega}_q^{(i,0)} =$
    $\boldsymbol{\Omega}_q^{(i-1)}$ and set $j = 0$.
    **Repeat (MM method for $\boldsymbol{\Omega}_q^{(i)}$)**
        $j \leftarrow j + 1$
        Find $\boldsymbol{\Omega}_q^{(i,j)}$ by solving the problem (C.6) with
        $\boldsymbol{x} = \boldsymbol{x}^{(i)}$.
    **Until** a convergence criterion is satisfied.
    **Update** $\boldsymbol{\Omega}_q^{(i)} \leftarrow \boldsymbol{\Omega}_q^{(i,j)}$
**Until** a convergence criterion is satisfied.
**Solution**: $\boldsymbol{x} \leftarrow \boldsymbol{x}^{(i)}$ and $\boldsymbol{\Omega}_q \leftarrow \boldsymbol{\Omega}_q^{(i)}$
```

whitening filter with respect to the overall additive noise $\sum_{n=1}^{N} f_n \alpha_n (\boldsymbol{c}_n + \boldsymbol{w}_n) + \boldsymbol{z}$; and $\boldsymbol{S} = \sum_{n=1}^{N} |f_n|^2 p_n \sigma_{t,n}^2 \boldsymbol{x}\boldsymbol{x}^H$ is the correlation matrix of the desired signal part. Accordingly, the detection problem has the standard estimator-correlator solution given by the test in (4.11). In the rest of this section, we seek to optimize the detection performance with respect to the waveform $\boldsymbol{x}$ and the power gains $\boldsymbol{p}$, under power constraints on the transmit element and receive sensors. As done above, we adopt the Bhattacharyya distance as the performance metric. As per (4.12), the Bhattacharyya distance between the distributions (4.17) of the signals received at the fusion center under the two hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ can be calculated as

$$\begin{aligned}\mathcal{B}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f}) &= \log\left(\frac{|\boldsymbol{I} + 0.5\boldsymbol{DSD}|}{\sqrt{|\boldsymbol{I} + \boldsymbol{DSD}|}}\right) \\ &= \log\left(\frac{1 + 0.5\lambda}{\sqrt{1+\lambda}}\right), \qquad (4.18)\end{aligned}$$

where $\lambda = \boldsymbol{f}^H\boldsymbol{P}\boldsymbol{\Sigma}_t\boldsymbol{f}\boldsymbol{x}^H(\boldsymbol{f}^H\boldsymbol{P}\boldsymbol{\Sigma}_c\boldsymbol{f}\boldsymbol{x}\boldsymbol{x}^H + (\boldsymbol{f}\otimes\boldsymbol{I}_K)^H\ (\boldsymbol{P}\otimes\boldsymbol{I}_K)\boldsymbol{\Omega}_w(\boldsymbol{f}\otimes\boldsymbol{I}_K) + \boldsymbol{\Omega}_z)^{-1}\boldsymbol{x}$; $\boldsymbol{\Sigma}_t = \text{diag}\{\sigma_{t,1}^2, \dots\ , \sigma_{t,N}^2\}$ and $\boldsymbol{\Sigma}_c = \text{diag}\{\sigma_{c,1}^2, \dots, \sigma_{c,N}^2\}$ are the diagonal matrices whose components are the second-order statistics of channel amplitudes of target return and clutter, respectively; $\boldsymbol{\Omega}_w = \text{diag}\{\boldsymbol{\Omega}_{w,1}, \dots, \boldsymbol{\Omega}_{w,N}\} \in R^{NK\times NK}$ is a block diagonal matrix containing all the noise covariance matrices at the receive sensors; and $\boldsymbol{P} = \text{diag}\{\boldsymbol{p}\} \in R^{N\times N}$ is the diagonal matrix that contains the receive sensors' power gains. Note that we have made explicit the dependence of the Bhattacharyya distance $\mathcal{B}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f})$ on the channels $\boldsymbol{f}$ at the fusion center, as well as on the waveform $\boldsymbol{x}$ and the receive sensors' power gains $\boldsymbol{p}$.

### 4.4.1 Short-Term Adaptive Design

We first consider the case in which design of the waveform $\boldsymbol{x}$ and of the receive sensors' gains $\boldsymbol{p}$ depends on the instantaneous gain of the CSI of the receive sensors-to-fusion center channels $\boldsymbol{f}$. Note that this design requires to modify the solution vector $(\boldsymbol{x},\boldsymbol{p})$ at the time scale at which the channel vector $\boldsymbol{f}$ varies, hence entailing a potentially large feedback overhead from the fusion center to the receive sensors and the transmit element. The problem of maximizing the Bhattacharyya distance (4.18) over the waveform $\boldsymbol{x}$ and the power gains $\boldsymbol{p}$ under the power constraints for transmit element and receive sensors, is stated as

$$\underset{\boldsymbol{x},\boldsymbol{p}}{\text{minimize}} \quad \bar{\mathcal{B}}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f}) \tag{4.19a}$$

$$\text{s.t.} \quad \boldsymbol{x}^H\boldsymbol{x} \le P_T, \tag{4.19b}$$

$$\mathbf{1}^T\boldsymbol{p} \le P_R, \tag{4.19c}$$

$$p_n \ge 0,\ \ n \in \mathcal{N}, \tag{4.19d}$$

where we have defined $\bar{\mathcal{B}}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f}) = -\mathcal{B}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f})$ to formulate the problem as the minimization of the negative Bhattacharyya distance $\bar{\mathcal{B}}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f})$. We observe that the problem (4.19) may be easily modified to include individual power constraints at the receive sensors, but this is not further explored here. Moreover, the problem (4.19) is not a convex program, since the objective function (4.19a) is not convex.

We propose an algorithm to solve the optimization problem (4.19). As in Section 4.3.4, due to the difficulty of obtaining a global optimal solution, we develop a descent algorithm, and adopt the BCD method coupled with MM. The proposed algorithm is summarized in Table Algorithm 4.2 and further detailed in Appendix B. The complexity of the Algorithm 4.2 by using standard convex optimization tool is polynomial in $K$ and $N$ since, at each outer iteration, MM requires to solve the problems (D.2) and (D.4), whose sizes of the optimization domains are $K$ and $N$, and numbers of constraints are 1 and $N + 1$, respectively [15, 53].

**Algorithm 4.2** Short-term adaptive design of waveform and amplifier gain (4.19)

```
**Initialization (outer loop)**: Initialize $\boldsymbol{x}^{(0)} \in C^{K\times 1}$, $\boldsymbol{p}^{(0)} \succeq 0$ and set $i = 0$.
**Repeat (BCD method)**
    $i \leftarrow i + 1$
    **Initialization (inner loop)**: Initialize $\boldsymbol{x}^{(i,0)} =$
    $\boldsymbol{x}^{(i-1)}$ and set $j = 0$.
    **Repeat (MM method for $\boldsymbol{x}^{(i)}$)**
        $j \leftarrow j + 1$
        Find $\boldsymbol{x}^{(i,j)}$ by solving the problem (D.2) with
        $\boldsymbol{p} = \boldsymbol{p}^{(i-1)}$.
    **Until** a convergence criterion is satisfied.
    **Update** $\boldsymbol{x}^{(i)} \leftarrow \boldsymbol{x}^{(i,j)}$
    **Initialization (inner loop)**: Initialize $\boldsymbol{p}^{(i,0)} =$
    $\boldsymbol{p}^{(i-1)}$ and set $j = 0$.
    **Repeat (MM method for $\boldsymbol{p}^{(i)}$)**
        $j \leftarrow j + 1$
        Find $\boldsymbol{p}^{(i,j)}$ by solving the problem (D.4) with
        $\boldsymbol{x} = \boldsymbol{x}^{(i)}$.
    **Until** a convergence criterion is satisfied.
    **Update** $\boldsymbol{p}^{(i)} \leftarrow \boldsymbol{p}^{(i,j)}$
**Until** a convergence criterion is satisfied.
**Solution**: $\boldsymbol{x} \leftarrow \boldsymbol{x}^{(i)}$ and $\boldsymbol{p} \leftarrow \boldsymbol{p}^{(i)}$
```

### 4.4.2 Long-Term Adaptive Design

Here, in order to avoid the possibly excessive feedback overhead between fusion center and the transmit element and receive sensors of the short-term adaptive solution,

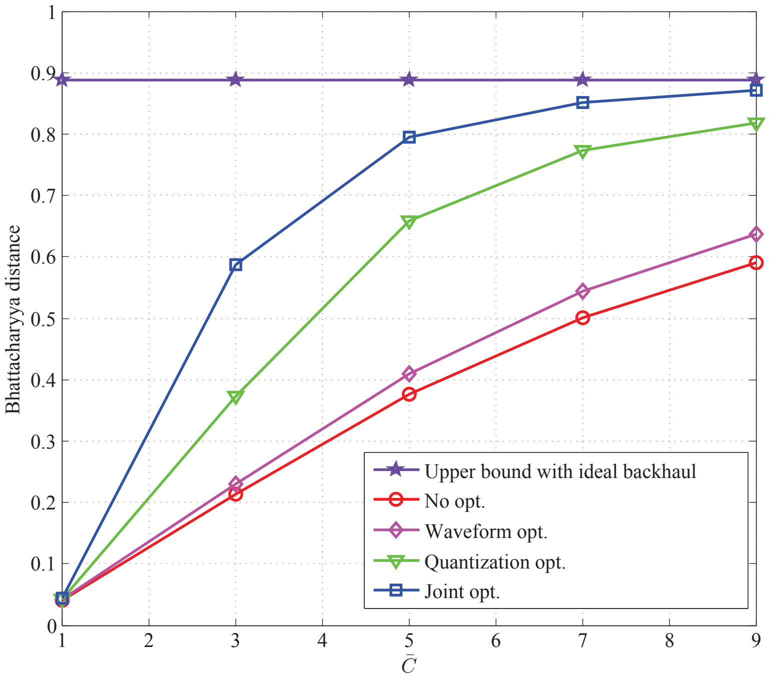


**Figure 4.2** Bhattacharyya distance versus the backhaul capacity $\bar{C}_n = \bar{C}$, $n \in \mathcal{N}$ for CF backhaul transmission, with $P_T = 10$ dB, $K = 13$, $N = 3$, $\sigma^2_{t,n} = 1$, $\sigma^2_{c,1} = 0.125$, $\sigma^2_{c,2} = 0.25$, $\sigma^2_{c,3} = 0.5$ and $[\mathbf{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$ for $n \in \mathcal{N}$.

we adopt the average Bhattacharyya distance, as the performance criterion, where the average is taken with respect to the distribution of the receive sensors-to-fusion center channels $\boldsymbol{f}$. In this way, the waveform $\boldsymbol{x}$ and receive sensors' gains $\boldsymbol{p}$ have to be updated only at the time scale at which the statistics of channels and noise terms vary. Then, the problem for the long-term adaptive design is formulated from problem (4.19) by substituting the objective function $\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f})$ with $E_{\boldsymbol{f}}[\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f})]$, yielding

$$\underset{\boldsymbol{x},\boldsymbol{p}}{\text{minimize}} \quad E_{\boldsymbol{f}}\left[\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f})\right] \tag{4.20a}$$

$$\text{s.t.} \quad (4.19\text{b}) - (4.19\text{d}). \tag{4.20b}$$

Note that the problem (4.20) is a stochastic program with a non-convex objective function (4.20a).

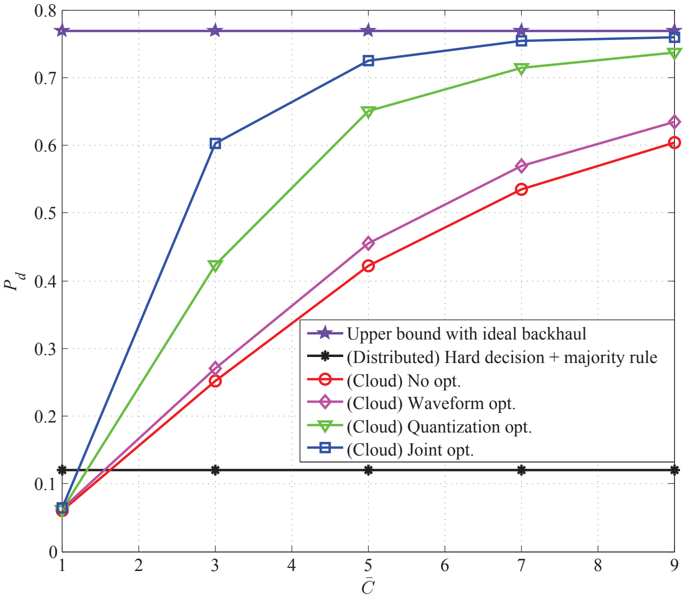


**Figure 4.3** Probability of detection $P_d$ versus the backhaul capacity $\bar{C}_n = \bar{C}$ for CF backhaul transmission, $n \in \mathcal{N}$, with $P_T = 10$ dB, $K = 13$, $N = 3$, $\sigma^2_{t,n} = 1$, $\sigma^2_{c,1} = 0.125$, $\sigma^2_{c,2} = 0.25$, $\sigma^2_{c,3} = 0.5$, $[\mathbf{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$ for $n \in \mathcal{N}$ and $P_{fa} = 0.01$.

Since the stochastic program (4.20) has a non-convex objective function, we apply the stochastic successive upper-bound minimization method (SSUM) [68], which minimizes at each step an approximate ensemble average of a locally tight upper bound of the cost function. Specifically, we develop a BCD scheme similar to the one detailed in Table Algorithm 4.2 that uses SSUM in lieu of the MM scheme. Details are provided in Appendix C. The final algorithm for long-term adaptive design can be summarized as in Table Algorithm 4.2 by substituting (D.2) and (D.4) with (E.1) and (E.2), respectively. Convergence of the SSUM algorithm is proved in [68] and the algorithm guarantees feasible iterates. The complexity of the proposed algorithm by using standard convex optimization tool is polynomial in $K$ and $N$ since, at each outer iteration, SSUM requires to solve the problems (E.1) and (E.2), whose sizes of the optimization domains are $K$ and $N$, and numbers of constraints are 1 and $N + 1$, respectively [15, 53].

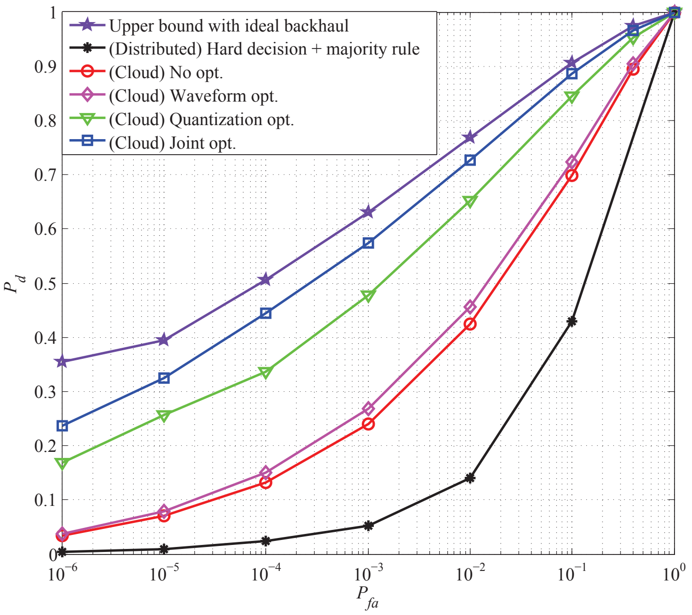


**Figure 4.4** ROC curves for CF backhaul transmission with $P_T = 10$ dB, $K = 13$, $N = 3$, $\sigma^2_{t,n} = 1$, $\sigma^2_{c,1} = 0.125$, $\sigma^2_{c,2} = 0.25$, $\sigma^2_{c,3} = 0.5$, $[\mathbf{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$ and $\bar{C}_n = \bar{C} = 5$ for $n \in \mathcal{N}$.

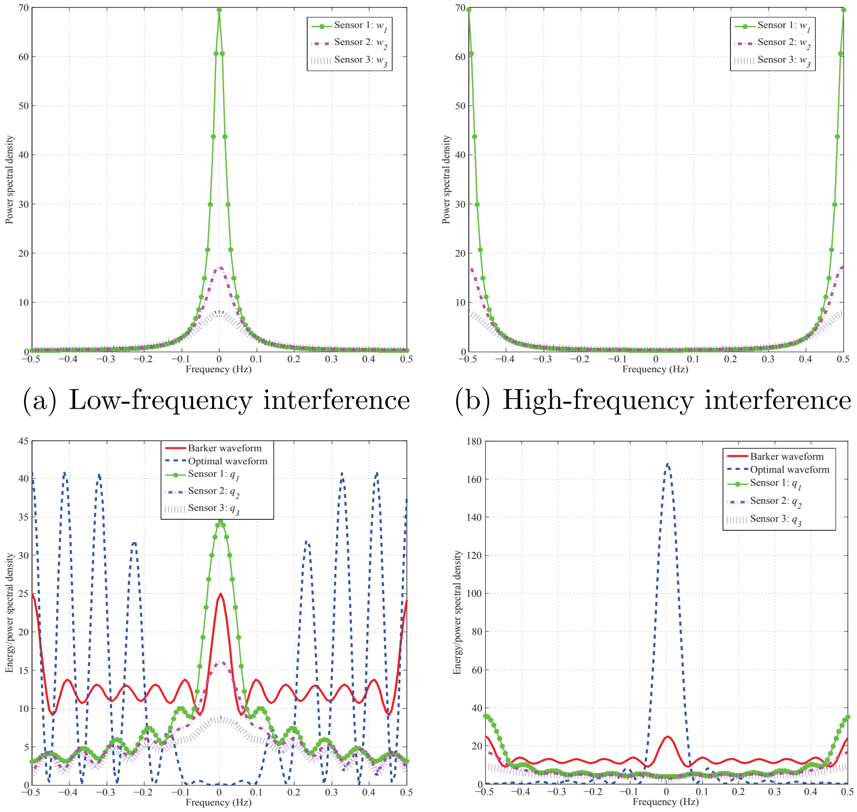
## 4.5 Numerical Results

In the following, the performance of the proposed algorithms that perform joint optimization of the waveform $\boldsymbol{x}$ and of the quantization noise covariance matrices $\mathbf{\Omega}_q$ for the CF, and of the waveform $\boldsymbol{x}$ and of the power gains $\boldsymbol{p}$ for AF, are investigated via numerical results in Section 4.5.1 and in Section 4.5.2, respectively. Throughout, we set the length of the waveform to $K = 13$ and the variances of the target amplitudes as $\sigma^2_{t,n} = 1$ for $n \in \mathcal{N}$. For reference, we consider a baseline waveform with Barker code of length 13, i.e., $\boldsymbol{b}_{13} = [1\ \ 1\ \ 1\ \ 1\ \ 1\ -1\ \ -1\ \ 1\ \ 1\ \ -1\ \ 1\ -1\ \ 1]^T$. Moreover, unless stated otherwise, we model the noise with covariance matrices $[\mathbf{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$ and $[\mathbf{\Omega}_z]_{i,j} = (1 - 0.6)^{|i-j|}$ as in [58], hence accounting for temporally correlated interference. The channel coefficients $f_n$ have unit variance, i.e., $\sigma^2_{f_n} = 1$.

### 4.5.1 CF Backhaul Transmission

In this section, the performance of the proposed joint optimization of the waveform $\boldsymbol{x}$ and of the quantization noise covariance matrices $\boldsymbol{\Omega}_q$ in Section 4.3 is verified via numerical results. Note that some limited results for a sum-backhaul constraint were presented in [43]. For reference, we consider the performance of the upper bound obtained with infinite capacity backhaul links, distributed detection using the Barker waveform (see, Section 4.3.1), and the following strategies: $(i)$ *No optimization (No opt.)*: Set $\boldsymbol{x} = \sqrt{P_T/K}\boldsymbol{b}_{13}$ and $\boldsymbol{\Omega}_{q,n} = \epsilon \boldsymbol{I}$, for $n \in \mathcal{N}$, where $\epsilon$ is a constant that is found by satisfying the constraint (4.16b) with equality; $(ii)$ *Waveform optimization (Waveform opt.)* : Optimize the waveform $\boldsymbol{x}$ by using the algorithm in [58], which is given in Algorithm 4.1 by setting $\boldsymbol{\Omega}_{q,n} = 0$ for $n \in \mathcal{N}$, and set $\boldsymbol{\Omega}_{q,n} = \epsilon \boldsymbol{I}$, for $n \in \mathcal{N}$, as explained above; $(iii)$ *Quantization noise optimization (Quantization opt.)*: Optimize the covariance matrices $\boldsymbol{\Omega}_q$ as per Algorithm 4.1 with $\boldsymbol{x} = \sqrt{P_T/K}\boldsymbol{b}_{13}$. In the following, we set the number of receive sensors, the transmit power and the variance of the clutter amplitudes as $N = 3$, $P_T = 10$ dB, $\sigma_{c,1}^2 = 0.125$, $\sigma_{c,2}^2 = 0.25$ and $\sigma_{c,3}^2 = 0.5$, respectively. Also, the backhaul rate constraints $\bar{C}_n$ are assumed to be equal, i.e., $\bar{C}_n = \bar{C}$ for all $n \in \mathcal{N}$.

In Figure 4.2 the Bhattacharyya distance is plotted versus the available backhaul capacity $\bar{C}$. For intermediate and large values of $\bar{C}$, the proposed joint optimization of waveform and quantization noise is seen to be significantly beneficial over all separate optimization strategies. In order to study the actual detection performance and validate the results in Figure 4.2, Figure 4.3 shows the detection probability $P_d$ as a function of the available backhaul capacity $\bar{C}$ when the false alarm probability is $P_{fa} = 0.01$. The curve was evaluated via Monte Carlo simulations by implementing the optimum test detector (4.11). We also implemented the distributed detection scheme described in Section 4.3.1 by setting the threshold $\gamma_n$ to be equal for $n \in \mathcal{N}$ for simplicity. It can be noted that the relative gains predicted by the Bhattacharyya



(a) Low-frequency interference (b) High-frequency interference

(c) Optimal waveform and quantization noise with low-frequency interference

(d) Optimal waveform and quantization noise with high-frequency interference

**Figure 4.5** Comparison of the energy/power spectral densities of the waveforms obtained with a Barker code (Barker waveform) and with an optimal code $\boldsymbol{x}$ (Optimal waveform), and of optimal quantization noise $\{\boldsymbol{q}_n\}_{n=1}^{N}$ obtained by Algorithm 4.1 when $P_T = 10$ dB, $K = 13$, $N = 3$, $\sigma_{t,n}^2 = 1$, $\sigma_{c,1}^2 = 0.125$, $\sigma_{c,2}^2 = 0.25$, $\sigma_{c,3}^2 = 0.5$, and $\bar{C}_n = \bar{C} = 5$ for $n \in \mathcal{N}$: (a) and (c) consider receive sensors with low-frequency interference having temporal correlation $[\boldsymbol{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$, (b) and (d) consider receive sensors with high-frequency interference having temporal correlation $[\boldsymbol{\Omega}_{w,n}]_{i,j} = (-1 + 0.12n)^{|i-j|}$.

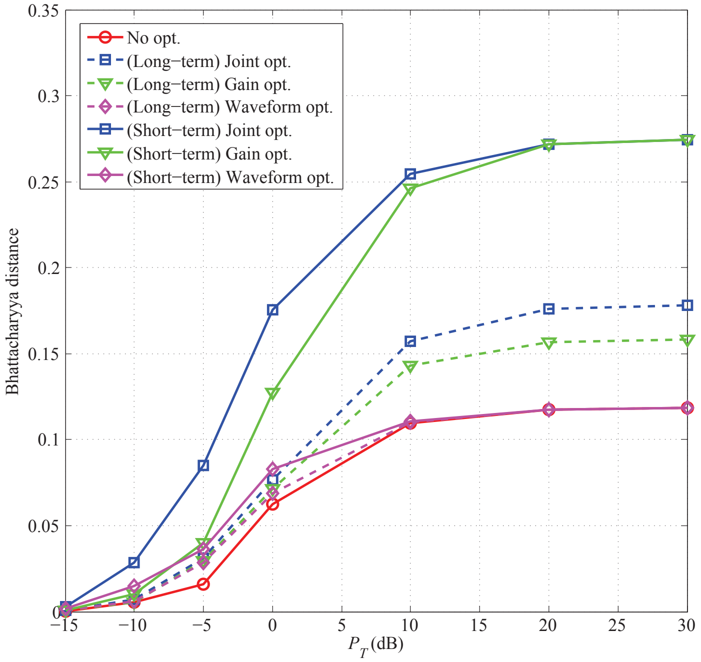


**Figure 4.6** Bhattacharyya distance versus the transmit element's power $P_T$ for AF backhaul transmission with $P_R = 10$ dB, $K = 13$, $N = 3$, $\sigma^2_{f_n} = 1$, $\sigma^2_{t,n} = 1$, $\sigma^2_{c,1} = 0.25$, $\sigma^2_{c,2} = 0.5$, $\sigma^2_{c,3} = 1$, $[\mathbf{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$ and $[\mathbf{\Omega}_z]_{i,j} = (1 - 0.6)^{|i-j|}$ for $n \in \mathcal{N}$.

distance criterion in Figure 4.2 are consistent with the performance shown in Figure 4.3. Moreover, for small values of $\bar{C}$, distributed detection outperforms cloud detection due to the performance degradation caused by the large quantization noise on the cloud-based schemes. However, as the available backhaul capacity $\bar{C}$ increases, the cloud detection approach considerably outperforms distributed detection.

Figure 4.4 plots the Receiving Operating Characteristic (ROC), i.e., the detection probability $P_d$ versus false alarm probability $P_{fa}$, for $\bar{C} = 5$. It is confirmed that the proposed joint optimization method provides remarkable gains over all separate optimization schemes as well as over the distributed detection approach. For instance, for $P_{fa} = 0.01$, joint optimization yields $P_d = 0.7251$, while waveform optimization only yields $P_d = 0.4556$.

Figure 4.5 shows the energy/power spectral density functions of the waveform with Barker code *(Barker waveform)* and with optimal code $\boldsymbol{x}$ *(Optimal waveform)*,

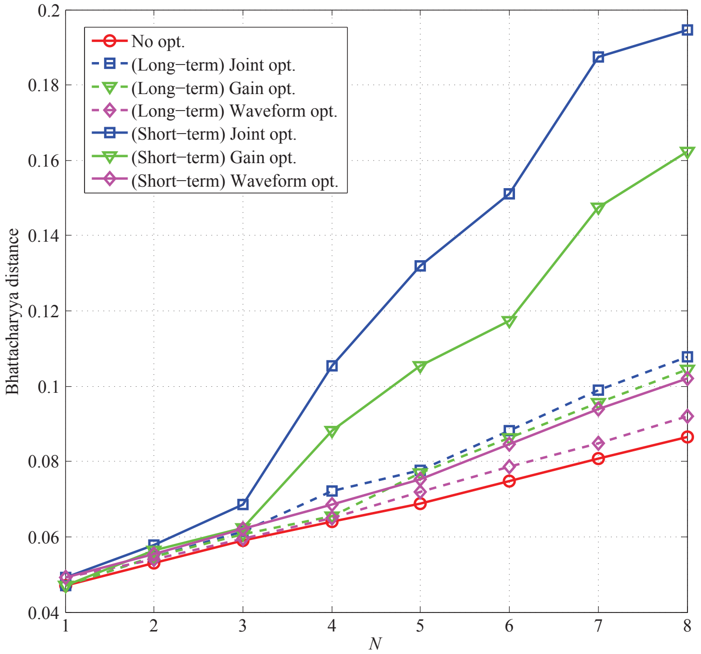


**Figure 4.7** Bhattacharyya distance versus the number receive sensors $N$ for AF backhaul transmission with $P_T = 5$ dB, $P_R = 10$ dB, $K = 13$, $\sigma_{f_n}^2 = 1$, $\sigma_{t,n}^2 = 1$, $\sigma_{c,1}^2 = 1$, $\sigma_{c,2}^2 = 0.9$, $\sigma_{c,3}^2 = 0.75$, $\sigma_{c,4}^2 = 0.5$, $\sigma_{c,5}^2 = 0.35$, $\sigma_{c,6}^2 = 0.25$ and $\sigma_{c,7}^2 = 0.125$, $\sigma_{c,8}^2 = 0.05$, $[\boldsymbol{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$ and $[\boldsymbol{\Omega}_z]_{i,j} = (1 - 0.6)^{|i-j|}$ for $n \in \mathcal{N}$.

and of optimal quantization noise $\{\boldsymbol{q}_n\}_{n=1}^N$ obtained by Algorithm 4.1 when a square root Nyquist chip waveform $\phi(t)$ with duration $T_c$ is adopted, and $\bar{C}_n = \bar{C} = 5$ for $n \in \mathcal{N}$. We consider two types of interference at the receive sensors, namely (a) low-frequency interference with temporal correlation $[\boldsymbol{\Omega}_{w,n}]_{i,j} = (1-0.12n)^{|i-j|}$ which has a single spectral peak at zero frequency; and (b) high-frequency interference with temporal correlation $[\boldsymbol{\Omega}_{w,n}]_{i,j} = (-1 + 0.12n)^{|i-j|}$, having a minimum at zero frequency. It is observed in Figure 4.5(c) and Figure 4.5(d) that the spectrum of the optimal waveform concentrates the transmitted energy at frequencies for which the interference power is less pronounced, while the spectrum of the quantization noise concentrates at frequencies and sensors for which the interference power is more pronounced.

### 4.5.2 AF Backhaul Transmission

In this section, we evaluate the performance of the proposed algorithms that perform the joint optimization of the waveform $\boldsymbol{x}$ and of the amplifying power gains $\boldsymbol{p}$ for the short-term (Section 4.4.1) and long-term (Section 4.4.2) adaptive designs. For reference, we consider the following schemes; *(i) No opt.*: Set $\boldsymbol{x} = \sqrt{P_T/K}\boldsymbol{b}_{13}$ and $\boldsymbol{p} = P_R/N\mathbf{1}_N$; *(ii) Waveform opt.*: Optimize the waveform $\boldsymbol{x}$ as per Algorithm 4.2 (with (E.1) in lieu of (D.2) for the long-term adaptive design) with $\boldsymbol{p} = P_R/N\mathbf{1}_N$; and *(iii) Gain optimization (Gain opt.)*: Optimize the gains $\boldsymbol{p}$ as per Algorithm 4.2 (with (E.2) in lieu of (D.4) for the long-term adaptive design) with $\boldsymbol{x} = \sqrt{P_T/K}\boldsymbol{b}_{13}$. We set the total receive sensors' power as $P_R = 10$ dB. Note that the upper bound with ideal backhaul is far from the performance achieved with AF over a non-orthogonal backhaul even for large sensors' power $P_R$, and it is hence not shown here. The gap between the AF performance and the upper bound is due to the fact that, in order to obtain an ideal backhaul, one needs to code across long block lengths whereas AF operates on block length of size equal to the waveform $K$ (here $K = 13$).

Figure 4.6 shows the Bhattacharyya distance as a function of the transmit element's power $P_T$, with $N = 3$, $\sigma^2_{c,1} = 0.25$, $\sigma^2_{c,2} = 0.5$ and $\sigma^2_{c,3} = 1$. For small values of $P_T$, optimizing the waveform is more advantageous than optimizing the amplifying gains, due to the fact that performance is limited by the transmit element-to-receive sensors connection. In contrast, for intermediate and large values of $P_T$, the optimization of the receive sensors' gains is to be preferred, since the performance becomes limited by the channels between the receive sensors and the fusion center. Joint optimization significantly outperforms all other schemes, except in the very low- and large-power regimes, in which, as discussed, the performance is limited by either the transmit element-to-receive sensors or the receive sensors-to-fusion center channels. In addition, we observe that the long-term adaptive scheme loses about 30% in terms of the Bhattacharyya distance with respect to the short-term adaptive design

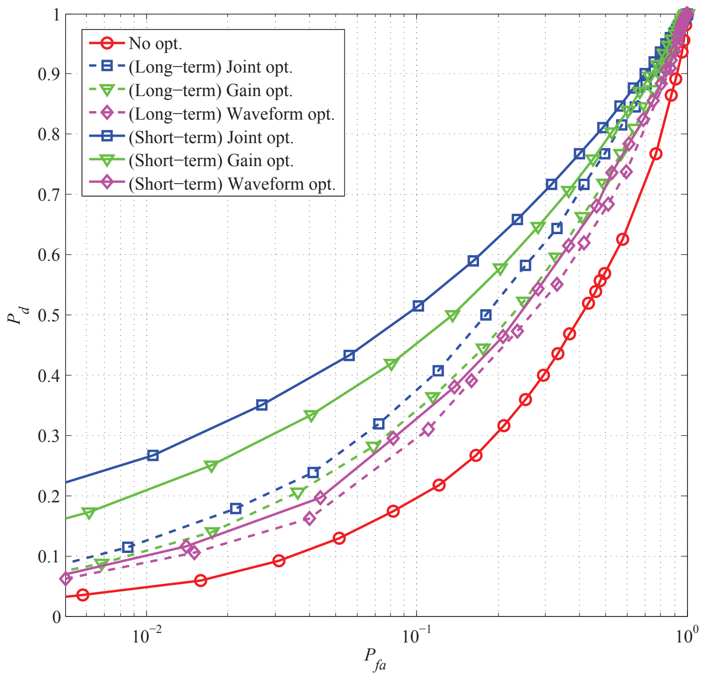


**Figure 4.8** ROC curves for AF backhaul transmission with $P_T = 5$ dB, $P_R = 10$ dB, $K = 13$, $N = 3$, $\sigma^2_{f_n} = 1$, $\sigma^2_{t,n} = 1$, $\sigma^2_{c,1} = 0.25$, $\sigma^2_{c,2} = 0.5$, $\sigma^2_{c,3} = 1$, $[\boldsymbol{\Omega}_{w,n}]_{i,j} = (1 - 0.12n)^{|i-j|}$ and $[\boldsymbol{\Omega}_z]_{i,j} = (1 - 0.6)^{|i-j|}$ for $n \in \mathcal{N}$.

in the high SNR regime. The results in Figure 4.6 can be interpreted by noting that the joint optimization seeks to design the transmitted signal $\boldsymbol{x}$ such that it reduces the power transmitted at the frequencies in which the receive sensors observe the largest interference, while, at the same time, allocating more power to receive sensors suffering from less interference and, with the short-term adaptive design, having better channels to the fusion center.

In Figure 4.7, the Bhattacharyya distance is plotted versus the number receive sensors $N$ with $P_T = 5$ dB, $\sigma^2_{c,1} = 1$, $\sigma^2_{c,2} = 0.9$, $\sigma^2_{c,3} = 0.75$, $\sigma^2_{c,4} = 0.5$, $\sigma^2_{c,5} = 0.35$, $\sigma^2_{c,6} = 0.25$, $\sigma^2_{c,7} = 0.125$ and $\sigma^2_{c,8} = 0.05$. Optimizing the receive sensors' power gains is seen to be especially beneficial at large $N$, due to the ability to allocate more power to the receive sensors in better condition in terms of interference and channels to the fusion center. For instance, even with the long-term adaptive design, optimizing the receive sensors' power gains outperforms waveform optimization with short-term adaptive design for sufficiently large $N$.

Figure 4.8 plots the ROC curves with $P_T = 5$ dB, $N = 3$, $\sigma_{c,1}^2 = 0.25$, $\sigma_{c,2}^2 = 0.5$ and $\sigma_{c,3}^2 = 1$. The curve was evaluated via Monte Carlo simulations by implementing the optimum test detector (4.11) as discussed in Section 4.4. It can be observed that the gains observed in the previous figures directly translate into a better ROC performance of joint optimization. Note also that power gain optimization is seen to be advantageous due to sufficient value of $P_T$ as predicted based on Figure 4.6.

## 4.6 Concluding Remarks

We have studied a multistatic cloud radar system, where the receive sensors and fusion center are connected via an orthogonal-access backhaul or a non-orthogonal multiple-access backhaul channel. In the former case, each receive sensor quantizes and forwards the signal sent by transmit element to a fusion center following a compress-and-forward protocol, while amplify-and-forward of the received signal is carried out over the multiple-access backhaul. The fusion center collects the signals from all the receive sensors and determines the target's presence or absence. We have investigated the joint optimization of waveform and backhaul transmission so as to maximize the detection performance. As the performance metric, we adopted the Bhattacharyya distance and the proposed algorithmic solutions were based on successive convex approximations. Overall, joint optimization was seen to have remarkable gains over the standard separate optimization of waveform and backhaul transmission. Moreover, cloud processing is found to outperform the standard distributed detection approach as long as the backhaul capacity is large enough.

# APPENDIX A

# MIN-SUM MESSAGE PASSING ALGORITHM

In this appendix, we briefly detail the basics of min-sum message passing algorithm for a clique tree [44]. Let $\mathcal{T}_c$ be a clique tree with cluster nodes $C_1, \cdots, C_n$, where each cluster $C_i$ is associated with a factor $\Phi_i$. Moreover, we define $\mathcal{I}_{C_i,C_j}$ the set of variables that appear as argument of the factors $\Phi_i$ and $\Phi_j$, associated with two cluster nodes $C_i$ and $C_j$ that are connected via an (undirected) edge. The min-sum message passing algorithm works as follows: $(i)$ Starting from the leaves of the clique tree and moving toward the root cluster node, each cluster node sends a message to its child. The message $\delta_{i\to j}$ that is sent from the cluster parent $C_i$ to the cluster child node $C_j$ is given by

$$\delta_{i\to j} = \min_{\mathcal{I}\setminus\mathcal{I}_{C_i,C_j}} \left\{ \Phi_i + \sum_{k\in\mathcal{P}(i)} \delta_{k\to i} \right\}, \tag{A.1}$$

where $\mathcal{I}$ is the set of all variables and we recall that $\mathcal{P}(i)$ is the set of parent clusters of cluster node $C_i$. The above equation indicates that the cluster node $C_i$ sums all incoming messages from its parents with its factor $\Phi_i$ and then minimizes the sum over all the variables except those that are common between $C_i$ and $C_j$; $(ii)$ This process is repeated until the root node receives all the messages from its parents.

# APPENDIX B
# EVALUATING ENERGY AND LATENCY FOR THE PARALLEL IMPLEMENTATION

In Section 2.5, we proposed an analytically convenient approximation for the energy and latency of the parallel implementation. Here, we develop a dynamic model that enables the evaluation of upper bounds on the energy and latency of the parallel implementation for a fixed set of variables ($\mathbf{I}$,$\mathbf{P}$) by tracking the state of each task over time. To this end, we quantize the time axis similar to (2.16) with a generally different time step $\epsilon_d$. By construction, the upper bounds calculated here become increasingly tighter as the quantization step $\epsilon_d$ decreases.

Define as $X_n(k)$ the state of task node $\mathrm{T}_n$ at time instant $t_k = (k-1)\epsilon_d$. The state of each node remains constant in the time range $(t_k, t_{k+1}]$ and may take any value in the set $\{\mathrm{ID}, \mathrm{CM}, \mathrm{CP}^{\mathrm{l}}, \mathrm{CP}^{\mathrm{r}}, \mathrm{UL}, \mathrm{DL}\}$, where ID indicates that a task is idle in the sense that it has not started processing yet. Instead, CM indicates that a task is completed in terms of processing and uplink/downlink communication and other state are defined in Section 2.3.2. For all $n \in \mathcal{V}_{\mathrm{D}}$, we initialize the state as $X_n(1) = \mathrm{CP}^{\mathrm{l}}$.

To keep track of the state of the uplink and downlink transmissions, we define the following variables. The variable $b_n^{ul}(k)$ indicates the remaining information bits that task $\mathrm{T}_n$ still needs to send in the uplink at time $t_k$. For $k = 1$, we have $b_n^{ul}(1) = b_{n,\mathcal{C}(n)}$ for all tasks $\mathrm{T}_n$ that are not directly connected to a leaf node with $I_n = 0$ and

$I_{\mathcal{C}(n)} = 1$; instead, if $I_n = 1$ and $\mathcal{P}(n) \in \mathcal{V}_{\mathrm{D}}$, we set $b_n^{ul}(k) = b_{\mathcal{P}(n),n}$; and we have $b_n^{ul}(k) = 0$ otherwise. Similarly, the variable $b_{m,n}^{dl}(k)$ for $m \in \mathcal{P}(n)$ represents the remaining output bits of task $\mathrm{T}_m$ that task $\mathrm{T}_n$ needs to receive in the downlink at time $t_k$. For $k = 1$, we have $b_{m,n}^{dl}(1) = b_{m,n}$ for all pairs $(m, n)$ such that $I_n = 0$ and $I_m = 1$, and $b_{m,n}^{dl}(1) = 0$ otherwise.

In order to track the state of the tasks in terms of computations, we define as $c_n^l(k)$ the number of CPU cycles that are left at time $t_k$ to finish a task $\mathrm{T}_n$ with $I_n = 0$, while $c_n^r(k)$ denotes the corresponding number of remaining CPU cycles for a task $\mathrm{T}_n$ with $I_n = 1$. Thus, we have $c_n^l(1) = v_n$ if $I_n = 0$ and $c_n^r(1) = v_n$ if $I_n = 1$, while we set $c_n^l(1) = c_n^r(1) = 0$ otherwise.

Let us define $N^l(k)$ as the number of tasks that are running locally and $N^r(k)$ as the number of tasks that are running remotely at time $t_k$. Similarly, we define $N^{ul}(k)$ and $N^{dl}(k)$ as the number of concurrent uplink and downlink transmissions at time $t_k$, respectively. In the proposed approach, as described below, we update the state $X_n(k)$ of each task node by making the assumption that the quantities $N^l(k)$, $N^r(k)$, $N^{ul}(k)$ and $N^{dl}(k)$ remain constant through the time interval $(t_k, t_{k+1}]$. As argued below, this lead to the desired upper bounds on energy and latency. In the following, we treat separately the state update of each task $\mathrm{T}_n$ in any interval $(t_k, t_{k+1}]$ depending on the state $X_n(k)$ at time $t_k$.

If $X_n(k) = \mathrm{UL}$, the amount of information that can be transmitted to the server in the time slot $(t_k, t_{k+1}]$ should be calculated in order to update the variable

$b_n^{ul}(k)$. If $I_n = 1$ we have $b_n^{ul}(k+1) = [b_n^{ul}(k) - (C^{ul}(N^{ul}(k)\bar{P}_{\mathcal{P}(n),n})/N^{ul}(k))\epsilon]^+$ due to the uploading of information from the connected leaf node, where $[x]^+$ is equal to $x$ if $x > 0$ and $x$ is equal to 0 otherwise. Instead, if $I_n = 0$, we have $b_n^{ul}(k+1) = [b_n^{ul}(k) - (C^{ul}(N^{ul}(k)\bar{P}_{n,\mathcal{C}(n)})/N^{ul}(k))\epsilon]^+$, due to the uploading of information to the child task $\mathrm{T}_{\mathcal{C}(n)}$. As a result, the state of the node changes as

$$X_n(k+1) = \begin{cases} \text{UL} & \textbf{if } b_n^{ul}(k+1) > 0 \\ \text{CM} & \textbf{if } I_n = 0 \text{ and } b_n^{ul}(k+1) = 0 \\ \text{CP}^{\text{r}} & \textbf{if } I_n = 1 \text{ and } b_n^{ul}(k+1) = 0 \end{cases}, \tag{B.1}$$

since when $I_n = 0$, the task is completed, and when $I_n = 1$, the task $\mathrm{T}_n$ needs to be computed remotely.

Following similar consideration, if $X_n(k) = \text{DL}$, the state of the task node $\mathrm{T}_n$ can be updated as

$$X_n(k+1) = \begin{cases} \text{DL} & \textbf{if } b_{m,n}^{dl}(k+1) > 0 \text{ for any } m \in \mathcal{P}(n) \\ \text{CP}^{\text{l}} & \textbf{if } b_{m,n}^{dl}(k+1) = 0 \text{ and } X_m(k) = \text{CM} \\ & \text{for all } m \in \mathcal{P}(n) \end{cases}. \tag{B.2}$$

Moreover, if $X_n(k) = \mathrm{CP^l}$, we have

$$X_n(k+1) = \begin{cases} \mathrm{CP^l} & \textbf{if } c_n^l(k+1) > 0 \\ \mathrm{UL} & \textbf{if } I_{\mathcal{C}(n)} = 1 \text{ and } c_n^l(k+1) = 0 \text{ and } n \in \mathcal{V} \backslash \mathcal{V}_\mathrm{D} \\ \mathrm{CM} & \text{otherwise} \end{cases}, \tag{B.3}$$

and, if $X_n(k) = \mathrm{CP^r}$, we can write

$$X_n(k+1) = \begin{cases} \mathrm{CP^r} & \textbf{if } c_n^r(k+1) > 0 \\ \mathrm{CM} & \textbf{if } c_n^r(k+1) = 0 \end{cases}, \tag{B.4}$$

where $c_n^r(k+1)$ is calculated as $c_n^r(k+1) = [c_n^r(k) - (f^r/N^r(k))\epsilon]^+$. If $X_n(k) = \mathrm{CM}$, we always have $X_n(k+1) = \mathrm{CM}$ and, if $X_n(k) = \mathrm{ID}$, we have

$$X_n(k+1) = \begin{cases} \mathrm{DL} & \textbf{if } I_n = 0 \text{ and } I_m = 1 \text{ for some } m \in \mathcal{P}(n) \text{ with } X_m(k) = \mathrm{CM} \\ \mathrm{UL} & \textbf{if } I_n = 1 \text{ and } X_m(k) = \mathrm{CM} \text{ for all } m \in \mathcal{P}(n) \text{ and } m \in \mathcal{V}_\mathrm{D} \\ \mathrm{CP^l} & \textbf{if } I_n = 0 \text{ and } I_m = 0 \text{ for all } m \in \mathcal{P}(n) \text{ with } X_m(k) = \mathrm{CM} \\ \mathrm{CP^r} & \textbf{if } I_n = 1 \text{ and } X_m(k) = \mathrm{CM} \text{ for all } m \in \mathcal{P}(n) \text{ and } m \in \mathcal{V} \backslash \mathcal{V}_\mathrm{D} \\ \mathrm{ID} & \text{otherwise} \end{cases}. \tag{B.5}$$

Based on the discussion above, the values $N^l(k)$, $N^r(k)$, $N^{ul}(k)$ and $N^{dl}(k)$ are calculated at each time $t_k$ according to the states of nodes as $N^{ul}(k) = \sum_{n=1}^{|\mathcal{V}|} 1(X_n(k) = \mathrm{UL})$, $N^l(k) = \sum_{n=1}^{|\mathcal{V}|} 1(X_n(k) = \mathrm{CP^l})$, $N^r(k) = \sum_{n=1}^{|\mathcal{V}|} 1(X_n(k) =$

CPr) and $N^{dl}(k) = \sum_{n=1}^{|\mathcal{V}|} \sum_{m \in \mathcal{P}(n)} 1(X_n(k) = \text{DL and } b^{dl}_{m,n}(k) > 0 \text{ and } X_m(k) = \text{CM}$, where $1(\cdot)$ is the indicator function.

Finally, at the end of each time interval $(t_k, t_{k+1}]$ the energy consumed by the mobile is updated as

$$
\begin{aligned}
E(k+1) = E(k) + &\sum_{n \in \mathcal{V}} \sum_{m \in \mathcal{P}(n)} 1\left(X_n(k) = \text{DL and } b^{dl}_{m,n}(k) > 0 \text{ and } X_m(k) = \text{CM}\right) \\
&(P^{rx} + P^{rf})\epsilon + \sum_{n \in \mathcal{V}} 1\left(X_n(k) = \text{UL}\right)(\bar{P}_{n,\mathcal{C}(n)} + P^{rf})\epsilon \\
&+ \sum_{n \in \mathcal{V}} 1\left(X_n(k) = \text{CP}^{\text{l}}\right) \frac{P^l}{N^l(k)} \epsilon.
\end{aligned}
\tag{B.6}
$$

The latency is instead given by the smallest value $t_k$ such that $X_{|\mathcal{V}|}(k) = \text{CM}$ for the root node $\text{T}_{|\mathcal{V}|}$. We observe that (B.6) assumes that transmissions and computations last for the period of duration $\epsilon_d$ even if the task completed at some time within the interval. This implies that (B.6) and the corresponding latency are upper bounds on the actual energy and latency that become increasingly tight as $\epsilon_d$ become smaller.

# APPENDIX C
# DETAILS OF CF OPTIMIZATION

## C.1 Review of MM Method

We start by reviewing the MM method. For a non-convex function $f(\boldsymbol{t})$ of a generic variable $\boldsymbol{t}$, which may appear either in the cost function or among the constraints, the MM method substitutes at the $l$th iteration, a convex approximation $f(\boldsymbol{t}|\boldsymbol{t}^{(l-1)})$ of $f(\boldsymbol{t})$, such that the global upper bound property $f(\boldsymbol{t}|\boldsymbol{t}^{(l-1)}) \geq f(\boldsymbol{t})$ is satisfied for all $\boldsymbol{t}$ in the domain, along with the local tightness condition $f(\boldsymbol{t}^{(l-1)}|\boldsymbol{t}^{(l-1)}) = f(\boldsymbol{t}^{(l-1)})$. These properties guarantee the feasibility of all iterates and the descent property that the object function does not increase along the iterations.

## C.2 Details of the Proposed Algorithm 4.1

In the following, we discuss the application of the MM method to perform optimizations over $\boldsymbol{x}$ and $\boldsymbol{\Omega}_q$ in Algorithm 4.1, respectively.

**Optimization over $\boldsymbol{x}$:** Here, the goal is to obtain the optimal value of $\boldsymbol{x}^{(i)}$ for problem (4.16) given $\boldsymbol{\Omega}_q = \boldsymbol{\Omega}_q^{(i-1)}$. To this end, we apply the MM method. Specifically, at the $j$th iteration of the MM method and the $i$th iteration of the outer loop, the MM method solves a QCQP and obtains a solution $\boldsymbol{x}^{(i,j)}$ by substituting the non-convex objective function $\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{\Omega}_q)$ with a tight upper bound $\mathcal{U}^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_q|\boldsymbol{x}^{(i,j-1)})$ around the current iterate $\boldsymbol{x}^{(i,j-1)}$. This bound is obtained by linearizing the difference-of-convex functions in $\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{\Omega}_q)$ via the first-order Taylor approximation [34], which follows the same steps as in [58, eq. (34) and (50) in Section IV], and is given by

$$\begin{aligned}\mathcal{U}^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_q | \boldsymbol{x}^{(i,j-1)}) &= \sum_{n=1}^{N} \mathcal{U}_n^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n} | \boldsymbol{x}^{(i,j-1)}) \\ = \sum_{n=1}^{N} \phi_n^{(i,j-1)} \boldsymbol{x}^H \left(\boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n}\right)^{-1} \boldsymbol{x} \\ & -\mathfrak{Re}\left(\left(\boldsymbol{d}_n^{(i,j-1)}\right)^H \boldsymbol{x}\right), \end{aligned} \tag{C.1}$$

where

$$\begin{aligned} \phi_n^{(i,j-1)} &= \frac{\beta_n}{1 + \beta_n y_n^{(i,j-1)}} + \beta_n(1 + 0.5\gamma_n) \\ &\quad + \frac{0.5\gamma_n}{1 + \lambda_n^{(i,j-1)}} \frac{\beta_n}{\left(1 + \beta_n y_n^{(i,j-1)}\right)^2}; \\ \boldsymbol{d}_n^{(i,j-1)} &= \left(\frac{2\beta\left(1 + 0.5\gamma_n\right)}{1 + \beta_n y_n^{(i,j-1)}\left(1 + 0.5\gamma_n\right)}\right. \\ &\quad \left. + 2\beta_n\left(1 + 0.5\gamma_n\right)\right)\left(\boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n}\right)^{-1} \boldsymbol{x}^{(i,j-1)}; \\ y_n^{(i,j-1)} &= \left(\boldsymbol{x}^{(i,j-1)}\right)^H \left(\boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n}\right)^{-1} \boldsymbol{x}^{(i,j-1)}; \\ \lambda_n^{(i,j-1)} &= \gamma_n - \frac{\gamma_n}{1 + \beta_n y_n^{(i,j-1)}}. \end{aligned}$$

with $\beta_n = \sigma_{c,n}^2$ and $\gamma_n = \sigma_{t,n}^2 / \beta_n$. A bound with the desired property can also be easily derived for $\mathcal{I}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n})$ by using the inequality $\log(1+t) \leq \log(1+t^{(l)}) + 1/(1+t^{(l)})(t - t^{(l)})$, for $t = (\sigma_{t,n}^2 + \sigma_{c,n}^2)\boldsymbol{x}^H(\boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n})^{-1}\boldsymbol{x}$, leading to

$$\mathcal{U}_n^{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n}|\boldsymbol{x}^{(i,j-1)}) = \log\left|\boldsymbol{I} + (\boldsymbol{\Omega}_{q,n})^{-1}\boldsymbol{\Omega}_{w,n}\right|$$

$$+\log(1+t^{(i,j-1)}) + \frac{1}{1+t^{(i,j-1)}}\left((\sigma_{c,n}^2 + \sigma_{t,n}^2)\right.$$

$$\left.\boldsymbol{x}^H(\boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n})^{-1}\boldsymbol{x} - t^{(i,j-1)}\right). \tag{C.2}$$

At the $j$th iteration of the MM method and the $i$th outer loop, we evaluate the new iterate $\boldsymbol{x}^{(i,j)}$ by solving the following QCQP problem

$$\boldsymbol{x}^{(i,j)} \leftarrow \underset{\boldsymbol{x}}{\operatorname{argmin}}\ \mathcal{U}^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_q|\boldsymbol{x}^{(i,j-1)}) \tag{C.3a}$$

$$\text{s.t.}\quad \mathcal{U}_n^{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n}|\boldsymbol{x}^{(i,j-1)}) \leq \bar{C}_n,\ \ n \in \mathcal{N}, \tag{C.3b}$$

$$\boldsymbol{x}^H\boldsymbol{x} \leq P_T. \tag{C.3c}$$

The MM method obtains the solution $\boldsymbol{x}^{(i)}$ for the $i$th iteration of the outer loop by solving the problem (C.3) iteratively over $j$ until a convergence criterion is satisfied.

**Optimization over $\boldsymbol{\Omega}_q$:** In this part, we consider the optimization of matrices $\boldsymbol{\Omega}_q^{(i)}$ for a given $\boldsymbol{x} = \boldsymbol{x}^{(i)}$. Similar to the optimization over $\boldsymbol{x}^{(i)}$, we use upper bounds of $\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{\Omega}_q)$ and $\mathcal{I}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n})$ for optimization. First, by rewriting $\mathcal{I}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n})$ as $\mathcal{I}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n}) = \log|\boldsymbol{\Omega}_{q,n} + (\sigma_{t,n}^2 + \sigma_{c,n}^2)\boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n}| - \log|\boldsymbol{\Omega}_{q,n}|$, we obtain difference-of-convex functions with respect to $\boldsymbol{\Omega}_{q,n}$. Then, by linearizing negative convex component via its first-order Taylor approximation, upper bounds $\mathcal{U}_n^{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n}|\boldsymbol{\Omega}_{q,n}^{(i,j-1)})$ and $\mathcal{U}^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_q|\boldsymbol{\Omega}_q^{(i,j-1)})$ with the desired properties of MM method are derived for functions $\mathcal{I}_n(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n})$ and $\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{\Omega}_q)$, respectively, as follows:

$$
\begin{aligned}
&\mathcal{U}_n^{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n} | \boldsymbol{\Omega}_{q,n}^{(i,j-1)}) \\
&= \log |\boldsymbol{\Omega}_{q,n}^{(i,j-1)} + (\sigma_{t,n}^2 + \sigma_{c,n}^2)\boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n}| \\
&\quad - \log |\boldsymbol{\Omega}_{q,n}| + \mathrm{tr}\left\{ \left(\boldsymbol{\Omega}_{q,n}^{(i,j-1)} + (\sigma_{t,n}^2 + \sigma_{c,n}^2)\boldsymbol{x}\boldsymbol{x}^H \right.\right. \\
&\qquad\quad \left.\left. + \boldsymbol{\Omega}_{w,n}\right)^{-1} \left(\boldsymbol{\Omega}_{q,n} - \boldsymbol{\Omega}_{q,n}^{(i,j-1)}\right) \right\}
\end{aligned}
\tag{C.4}
$$

and

$$
\begin{aligned}
&\mathcal{U}^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_q | \boldsymbol{\Omega}_q^{(i,j-1)}) = \sum_{n=1}^{N} \mathcal{U}_n^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n} | \boldsymbol{\Omega}_{q,n}^{(i,j-1)}) \\
&= \sum_{n=1}^{N} - \log |(0.5\sigma_{t,n}^2 + \sigma_{c,n}^2)\boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n}| \\
&+ 0.5\mathrm{tr}\left\{ \left((\sigma_{t,n}^2 + \sigma_{c,n}^2)\boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n}^{(i,j-1)}\right)^{-1} \right. \\
&\left. \times \boldsymbol{\Omega}_{q,n} \right\} + 0.5\mathrm{tr}\left\{ \left(\sigma_{c,n}^2 \boldsymbol{x}\boldsymbol{x}^H + \boldsymbol{\Omega}_{w,n} + \boldsymbol{\Omega}_{q,n}^{(i,j-1)}\right)^{-1} \right. \\
&\qquad\qquad \left. \times \boldsymbol{\Omega}_{q,n} \right\}.
\end{aligned}
\tag{C.5}
$$

The $j$th iteration of the MM method then evaluates the matrices $\boldsymbol{\Omega}_q^{(i,j)} = \{\boldsymbol{\Omega}_{q,n}^{(i,j)}\}_{n \in \mathcal{N}}$ by solving the following convex optimization problem

$$
\boldsymbol{\Omega}_q^{(i,j)} \leftarrow \underset{\boldsymbol{\Omega}_q}{\mathrm{argmin}}\, \mathcal{U}^{\bar{\mathcal{B}}}(\boldsymbol{x}, \boldsymbol{\Omega}_q | \boldsymbol{\Omega}_q^{(i,j-1)}) \tag{C.6a}
$$

$$
\text{s.t.} \quad \mathcal{U}_n^{\mathcal{I}}(\boldsymbol{x}, \boldsymbol{\Omega}_{q,n} | \boldsymbol{\Omega}_{q,n}^{(i,j-1)}) \le \bar{C}_n, \;\; n \in \mathcal{N}, \tag{C.6b}
$$

$$
\boldsymbol{\Omega}_{q,n} \succeq 0, \;\; n \in \mathcal{N}. \tag{C.6c}
$$

By repeating the procedure (C.6) over $j$ until the convergence is attained, the solution $\mathbf{\Omega}_q^{(i)}$ is obtained for the $i$th outer loop.

# APPENDIX D

# DETAILS OF AF SHORT-TERM ADAPTIVE DESIGN

## D.1 Optimization over $\boldsymbol{x}$

Here, the goal is to optimize the objective function (4.19) over the waveform $\boldsymbol{x}^{(i)}$ given the gains $\boldsymbol{p} = \boldsymbol{p}^{(i-1)}$. For this purpose, we apply the MM method. Specifically, at the $j$th iteration of the MM method and the $i$th iteration of the outer loop, the MM method solves a convex QCQP and obtains a solution $\boldsymbol{x}^{(i,j)}$ by substituting the non-convex objective function $\bar{\mathcal{B}}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f})$ with a tight upper bound $\mathcal{U}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f}|\boldsymbol{x}^{(i,j-1)})$ around the current iterate $\boldsymbol{x}^{(i,j-1)}$. This bound is obtained by following the same steps as in Appendix C.2 and is given by

$$\begin{aligned}\mathcal{U}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f}|\boldsymbol{x}^{(i,j-1)}) = \phi^{(i,j-1)}\boldsymbol{x}^H \Big(&(\boldsymbol{f}\otimes\boldsymbol{I}_K)^H(\boldsymbol{P}\otimes\boldsymbol{I}_K)\boldsymbol{\Omega}_w(\boldsymbol{f}\otimes\boldsymbol{I}_K)\\ &+\boldsymbol{\Omega}_z)^{-1}\boldsymbol{x} - \mathfrak{Re}\left\{\left(\boldsymbol{d}^{(i,j-1)}\right)^H\boldsymbol{x}\right\},\end{aligned}\tag{D.1}$$

where

$$\begin{aligned}
\phi^{(i,j-1)} &= \frac{\beta}{1+\beta y^{(i,j-1)}} + \beta(1+0.5\gamma) + \frac{0.5\gamma}{1+\lambda^{(i,j-1)}}\frac{\beta}{\left(1+\beta y^{(i,j-1)}\right)^2};\\
\boldsymbol{d}^{(i,j-1)} &= \Bigg(\frac{2\beta\,(1+0.5\gamma)}{1+\beta y^{(i,j-1)}\,(1+0.5\gamma)}\ +2\beta\,(1+0.5\gamma))\Big((\boldsymbol{f}\otimes\boldsymbol{I}_K)^H(\boldsymbol{P}\otimes\boldsymbol{I}_K)\boldsymbol{\Omega}_w\\
&\qquad\qquad(\boldsymbol{f}\otimes\boldsymbol{I}_K)+\boldsymbol{\Omega}_z)^{-1}\boldsymbol{x}^{(i,j-1)};
\end{aligned}$$

$$\begin{aligned}
\beta &= \boldsymbol{f}^H\boldsymbol{P}\boldsymbol{\Sigma}_c\boldsymbol{f};\\
\gamma &= \frac{\boldsymbol{f}^H\boldsymbol{P}\boldsymbol{\Sigma}_t\boldsymbol{f}}{\beta};\\
y^{(i,j-1)} &= \left(\boldsymbol{x}^{(i,j-1)}\right)^H\Big((\boldsymbol{f}\otimes\boldsymbol{I}_K)^H(\boldsymbol{P}\otimes\boldsymbol{I}_K)\boldsymbol{\Omega}_w\ (\boldsymbol{f}\otimes\boldsymbol{I}_K)+\boldsymbol{\Omega}_z)^{-1}\boldsymbol{x}^{(i,j-1)};\\
\lambda^{(i,j-1)} &= \gamma - \frac{\gamma}{1+\beta y^{(i,j-1)}}.
\end{aligned}$$

At the $j$th iteration of the MM method and the $i$th outer loop, we evaluate the new iterate $\boldsymbol{x}^{(i,j)}$ by solving the following QCQP problem

$$\boldsymbol{x}^{(i,j)} \leftarrow \underset{\boldsymbol{x}}{\operatorname{argmin}} \quad \mathcal{U}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f}|\boldsymbol{x}^{(i,j-1)}) \tag{D.2a}$$

$$\text{s.t.} \qquad \boldsymbol{x}^H\boldsymbol{x} \leq P_T. \tag{D.2b}$$

The MM method obtains the solution $\boldsymbol{x}^{(i)}$ for the $i$th iteration of the outer loop by solving the problem (D.2) iteratively over $j$ until a convergence criterion is satisfied.

$$
\begin{aligned}
&\mathcal{U}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f}|\boldsymbol{p}^{(i,j-1)})\\
&= -\ln\left|\boldsymbol{f}^H\boldsymbol{P}\left(0.5\boldsymbol{\Sigma}_t+\boldsymbol{\Sigma}_c\right)\boldsymbol{f}\boldsymbol{x}\boldsymbol{x}^H+\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)^H\left(\boldsymbol{P}\otimes\boldsymbol{I}_K\right)\boldsymbol{\Omega}_w\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)+\boldsymbol{\Omega}_z\right|\\
&+0.5\mathrm{tr}\left\{\left(\boldsymbol{f}^H\boldsymbol{P}^{(i,j-1)}\left(\boldsymbol{\Sigma}_t+\boldsymbol{\Sigma}_c\right)\boldsymbol{f}\boldsymbol{x}\boldsymbol{x}^H+\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)^H\left(\boldsymbol{P}^{(i,j-1)}\otimes\boldsymbol{I}_K\right)\boldsymbol{\Omega}_w\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)+\boldsymbol{\Omega}_z\right)^{-1}\right.\\
&\qquad\qquad\left.\times\left(\boldsymbol{f}^H\boldsymbol{P}\left(\boldsymbol{\Sigma}_t+\boldsymbol{\Sigma}_c\right)\boldsymbol{f}\boldsymbol{x}\boldsymbol{x}^H+\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)^H\left(\boldsymbol{P}\otimes\boldsymbol{I}_K\right)\boldsymbol{\Omega}_w\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)\right)\right\}\\
&\qquad\qquad+0.5\mathrm{tr}\left\{\left(\boldsymbol{f}^H\boldsymbol{P}^{(i,j-1)}\Sigma_c\boldsymbol{f}\boldsymbol{x}\boldsymbol{x}^H+\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)^H\left(\boldsymbol{P}^{(i,j-1)}\otimes\boldsymbol{I}_K\right)\boldsymbol{\Omega}_w\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)+\boldsymbol{\Omega}_z\right)^{-1}\right.\\
&\qquad\qquad\left.\times\left(\boldsymbol{f}^H\boldsymbol{P}\Sigma_c\boldsymbol{f}\boldsymbol{x}\boldsymbol{x}^H+\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)^H\left(\boldsymbol{P}\otimes\boldsymbol{I}_K\right)\boldsymbol{\Omega}_w\left(\boldsymbol{f}\otimes\boldsymbol{I}_K\right)\right)\right\}. \qquad \text{(D.3)}
\end{aligned}
$$

## D.2 Optimization over $\boldsymbol{p}$

We consider now the optimization of the gains $\boldsymbol{p}^{(i)}$, when the waveform $\boldsymbol{x} = \boldsymbol{x}^{(i)}$ is given. Similar to the optimization over $\boldsymbol{x}^{(i)}$ in the previous section, we also use the MM method for the optimization over $\boldsymbol{p}$. Towards this goal, we obtain the upper bound $\mathcal{U}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f}|\boldsymbol{p}^{(i,j-1)})$ of the objective function $\bar{\mathcal{B}}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f})$ around the current iterate $\boldsymbol{p}^{(i,j-1)}$. This bound is derived by linearizing the difference-of-convex functions via the first-order Taylor approximation [34]. The bound can then be obtained in (D.3) at the top of the next page. Then, the new iterate $\boldsymbol{p}^{(i,j)}$ at the $j$th iteration of the MM method and the $i$th iteration of the outer loop can be obtained by solving the following optimization problem:

$$\boldsymbol{p}^{(i,j)} \leftarrow \underset{\boldsymbol{p}}{\mathrm{argmin}} \quad \mathcal{U}(\boldsymbol{x},\boldsymbol{p};\boldsymbol{f}|\boldsymbol{p}^{(i,j-1)}) \tag{D.4a}$$

$$\text{s.t.} \qquad \mathbf{1}^T\boldsymbol{p} \leq P_R, \tag{D.4b}$$

$$p_n \geq 0, \;\; n \in \mathcal{N}. \tag{D.4c}$$

By repeating the procedure (D.4) over $j$ until a convergence criterion is satisfied, the solution $\boldsymbol{p}^{(i)}$ is determined for the $i$th outer loop.

## D.3 Summary of the Proposed Algorithm 4.2

In summary, in order to solve problem (4.19), we propose an algorithm (described in Table Algorithm 4.2) that alternates between the optimization over $\boldsymbol{x}$, described in Appendix D.1 and the optimization over $\boldsymbol{p}$, discussed in Appendix D.2. In particular, at the $i$th iteration of the outer loop, the iterate $\boldsymbol{x}^{(i)}$ is obtained by solving a sequence of convex problems (Appendix D.1) via the MM method for a fixed $\boldsymbol{p} = \boldsymbol{p}^{(i-1)}$. Then, the iterate $\boldsymbol{p}^{(i)}$ is found by solving a sequence of convex problems (Appendix D.2) via the MM method with $\boldsymbol{x} = \boldsymbol{x}^{(i)}$ attained in the previous step. According to the the properties of the MM method [34, 67], the proposed scheme yields feasible iterates and a non-increasing objective function along the outer and inner iterations, hence ensuring convergence of the cost function.

# APPENDIX E

# DETAILS OF AF LONG-TERM ADAPTIVE DESIGN

## E.1 Optimization over $\boldsymbol{x}$

Following the SSUM scheme, at the $j$th inner iteration and the $i$th outer iteration, we optimize the waveform $\boldsymbol{x}^{(i,j)}$ given $\boldsymbol{p} = \boldsymbol{p}^{(i-1)}$ by solving the following convex problem

$$\boldsymbol{x}^{(i,j)} \leftarrow \underset{\boldsymbol{x}}{\text{argmin}} \quad \frac{1}{j} \sum_{l=1}^{j} \mathcal{U}^{(l)}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f}^{(l)} | \boldsymbol{x}^{(i,l-1)}) \tag{E.1a}$$

$$\text{s.t.} \qquad \boldsymbol{x}^{H}\boldsymbol{x} \leq P_T, \tag{E.1b}$$

where $\boldsymbol{f}^{(l)}$ denotes a channel vector $\boldsymbol{f}$ for the fusion center that is randomly and independently generated at the $l$th iteration according to the known distribution of $\boldsymbol{f}$, and $\mathcal{U}^{(l)}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f}^{(l)} | \boldsymbol{x}^{(i,l-1)})$ is the locally tight convex upper bound (D.1) on the negative Bhattacharyya distance around the point $\boldsymbol{x}^{(i,l-1)}$. Note that the cost function (E.1a) depends on all the realizations of the channel vectors $\boldsymbol{f}^{(l)}$ for $l = 1, \ldots, j$. The solution $\boldsymbol{x}^{(i)}$ for the $i$th iteration of the outer loop is obtained by solving the problem (E.1) iteratively over $j$, until a convergence criterion is satisfied.

## E.2 Optimization over $\boldsymbol{p}$

With the optimized waveform $\boldsymbol{x} = \boldsymbol{x}^{(i)}$, SSUM calculates the iterates $\boldsymbol{p}^{(i,j)}$ by solving iteratively the following problems

$$\boldsymbol{p}^{(i,j)} \leftarrow \underset{\boldsymbol{p}}{\operatorname{argmin}} \frac{1}{j} \sum_{l=1}^{j} \mathcal{U}^{(l)}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f}^{(l)} | \boldsymbol{p}^{(i,l-1)}) \tag{E.2a}$$

$$\text{s.t. } \mathbf{1}^T \boldsymbol{p} \leq P_R, \tag{E.2b}$$

$$p_n \geq 0, \;\; n \in \mathcal{N}, \tag{E.2c}$$

where $\mathcal{U}^{(l)}(\boldsymbol{x}, \boldsymbol{p}; \boldsymbol{f}^{(l)} | \boldsymbol{p}^{(i,l-1)})$ is the convex upper bound (D.3) on the negative Bhattacharyya distance around the point $\boldsymbol{p}^{(i,l-1)}$. The iterate $\boldsymbol{p}^{(i)}$ is obtained by solving the problem (E.2) iteratively over $j$ until convergence of the cost function.

# BIBLIOGRAPHY

[1] Amd unleashes first-ever 5 ghz processor. http://www.amd.com/en-us/press-releases/Pages/amd-unleashes-2013jun11.aspx. accessed 1/12/2015.

[2] Cloud computing. https://en.wikipedia.org/wiki/Cloud_computing. (accessed 1/12/2015).

[3] Samsung exynos 4 quad (exynos 4412). http://www.samsung.com/global/business/semiconductor/file/product/Exynos_4_Quad_User_Manaul_Public_REV1.00-0.pdf. (accessed 1/12/2015).

[4] Samsung exynos 4412 quad. http://www.notebookcheck.net/Samsung-Exynos-4412-Quad-ARM-SoC.86876.0.html. (accessed 1/12/2015).

[5] A. Alexiou A. G. Gotsis, S. Stefanatos. Ultra dense networks: The new wireless frontier for enabling 5G access. arXiv, Oct. 2015.

[6] A. Gersho and R. Grey. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, Boston, 1992.

[7] A. W. Rihaczek. *Principles of high-resolution radar.* New York: Wiley, 1969.

[8] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication*, 4(1):40–62, Mar. 2011.

[9] G. Alirezaei, M. Reyer, and R. Mathar. Optimum power allocation in sensor networks for passive radar applications. *IEEE Trans. Wireless Comm.*, 13(6):3222–3231, Jun. 2014.

[10] G. Alirezaei, O. Taghizadeh, and R. Mathar. Optimum power allocation with sensitivity analysis for passive radar applications. *IEEE Sensors Journal*, 14(11):3800–3809, Jun. 2014.

[11] J.G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J.C. Zhang. What will 5G be? *IEEE J. Sel. Areas Commun.*, 32(6):1065–1082, Jun. 2014.

[12] M. Barkat and P. K. Varshney. Decentralized CFAR signal detection. *IEEE Trans. on Aerosp. Electron. Syst.*, 25(2):141–149, Mar. 1989.

[13] M. Bernfeld. *Radar signals: An introduction to theory and application.* Elsevier, 2012.

[14] R. S. Blum, S. A. Kassam, and H. V. Poor. Distributed detection with multiple sensors: Part II – Advanced topics. *Proc. of the IEEE*, 85(1):64–79, Jan. 1997.

[15] S. P. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[16] G. Caire and D. Tuninetti. The throughput of hybrid-ARQ protocols for the Gaussian collision channel. *IEEE Trans. Inf. Theory*, 47(5):1971–1988, Jul. 2001.

[17] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. Cloud RAN for mobile networks-a technology overview. *IEEE Commun. Surveys Tuts.*, 17(1):405–426, First quarter 2015.

[18] V. S. Chemyak. *Fundamentals of multisite radar systems: multistatic radars and multistatic radar systems.* Gordon and Breach Science Publishers, 1998.

[19] A. M. Cipriano, P. Gagneur, G. Vivier, and S. Sezginer. Overview of ARQ and HARQ in beyond 3G systems. In *Proc. IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications*, pages 424–429, Istanbul, Turkey, Sep. 2010.

[20] A. De La Oliva, X. Costa Perez, A. Azcorra, A. Di Giglio, F. Cavaliere, D. Tiegelbekkers, J. Lessmann, T. Haustein, A. Mourad, and P. Iovanna. Xhaul: toward an integrated fronthaul/backhaul architecture in 5G networks. *IEEE Wireless Commun.*, 22(5):32–40, Oct. 2015.

[21] D. DeLong and E.M. Hofstetter. On the design of optimum radar waveforms for clutter rejection. *IEEE Trans. Inf. Theory*, 13(3):454–463, Jul. 1967.

[22] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier. Quantitative analysis of split base station processing and determination of advantageous architectures for LTE. *Bell Labs Technical Journal*, 18(1):105–128, Jun. 2013.

[23] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, P. Bahl. Maui: Making smartphones last longer with code offload. In *Proc. of 8th ACM MobiSys*, pages 49–62, 2010.

[24] E. Dahlman, S. Parkvall, J. Skold, P. Bemin. *3G Evolution: HSPA and LTE for Mobile Broadband.* Academic Press, second edition, 2008.

[25] Sony ERICSSON. 5G systems. ERICSSON White Paper, Jan. 2015.

[26] Niroshinie F., Seng W. L., and W. Rahayu. Mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(1):84–106, Jan. 2013.

[27] P. Frenger, S. Parkvall, and E. Dahlman. Performance comparison of HARQ with chase combining and incremental redundancy for HSDPA. In *Proc. IEEE Vehicular Technology Conf.*, volume 3, pages 1829–1833, Atlantic City, New Jersey, USA, Oct. 2001.

[28] A. El Gamal and Y.-H. Kim. *Network Information Theory.* Cambridge University Press, 2011.

[29] F. Gini and M. Rangaswamy. *Knowledge based radar detection, tracking and classification*, volume 52. John Wiley & Sons, 2008.

[30] J. R. Guerci. Cognitive radar: A knowledge-aided fully adaptive approach. In *Proc. IEEE Radar Conf.*, pages 1365–1370, Washington, DC, May 2010.

[31] A.M. Haimovich, R.S. Blum, and L.J. Cimini. MIMO radar with widely separated antennas. *IEEE Signal Process. Mag.*, 25(1):116–129, Jan. 2008.

[32] Q. Han, Ch. Wang, M. Levorato, and O. Simeone. On the effect of fronthaul latency on ARQ in C-RAN systems. *CoRR*, abs/1510.07176, 2015.

[33] D. Huang, P. Wang, and D. Niyato. A dynamic offloading algorithm for mobile computing. *IEEE Trans. Wireless Commun.*, 11(6):1991–1995, Jun. 2012.

[34] David R Hunter and Kenneth Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, Feb. 2004.

[35] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Trans. Comm.*, 15(1):52–60, Feb. 1967.

[36] Y.-H. Kao and B. Krishnamachari. Optimizing mobile computational offloading with delay constraints. In *Proc. of Global Communication Conf.*, pages 8–12, Dec. 2014.

[37] Yi-Hsuan Kao, B. Krishnamachari, Moo-Ryong Ra, and Fan Bai. Hermes: Latency optimal task assignment for resource-constrained mobile computing. In *Proc. IEEE INFOCOM*, Apr. 2015.

[38] S. Kay. Optimal signal design for detection of gaussian point targets in stationary gaussian clutter/reverberation. *IEEE J. Sel. Topics Signal Process.*, 1(1):31–41, Jun. 2007.

[39] S. Kay. Waveform design for multistatic radar detection. *IEEE Trans. Aerosp. Electron. Syst.*, 45(3):1153–1166, Jul. 2009.

[40] S. M. Kay. *Fundamentals of Signal Processing-Estimation Theory.* Prentice Hall, Englandwood Cliffs, NJ, 1993.

[41] S. M. Kay. Optimal signal design for detection of Gaussian point targets in stationary Gaussian clutter/reverberation. *IEEE Jour. Select. Topics in Sig. Proc.*, 1(1):31–41, Jun. 2007.

[42] S. M. Kay. Waveform design for multistatic radar detection. *IEEE Trans. Aerosp. Electron. Syst.*, 45(3):1153–1166, Jul. 2009.

[43] S. Khalili, O. Simeone, and A. M. Haimovich. Cloud Radio-Multistatic Radar: Joint optimization of code vector and backhaul quantization. *IEEE Sig. Proc. Lett.*, 22(4):494–498, Oct. 2014.

[44] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques.* The MIT Press, 2009.

[45] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proc. of INFOCOM*, pages 945–953, Mar. 2012.

[46] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava. A survey of computation offloading for mobile systems. *Mobile Networks and Applications*, 18(1):129–140, Feb. 2013.

[47] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon. Parallel data processing with mapreduce: A survey. *SIGMOD Rec.*, 40(4):11–20, Jan. 2012.

[48] N. Levanon and E. Mozeson. *Radar signals.* John Wiley Sons, 2004.

[49] P. D. Lorenzo, S. Barbarossa, and S. Sardellitti. Joint optimization of radio resources and code partitioning in mobile cloud computing. *Submitted to IEEE Transactions Mobile Comput.*, Jul. 2016.

[50] D. J. Love, R. W. Heath, and T. Strohmer. Grassmannian beamforming for multiple-input multiple-output wireless systems. *IEEE Trans. Inf. Theory*, 49(10):2735–2747, Oct. 2003.

[51] A. Lozano and N. Jindal. Are yesterday's information-theoretic fading models and performance metrics adequate for the analysis of today's wireless systems? *IEEE Commun. Mag.*, 50(11):210–217, Nov. 2012.

[52] C. Luo, L.T. Yang, P. Li, X. Xie, and H.-C. Chao. A holistic energy optimization framework for cloud-assisted mobile computing. *IEEE Trans. Wireless Commun.*, 22(3):118–123, Jun. 2015.

[53] Z. Q. Luo, W. K. Ma, A. M. C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Proc. Magazine*, 27(3):20–34, May 2010.

[54] M. Bernfeld. *Radar signals: An introduction to theory and application.* Elsevier, 2012.

[55] M. W. Marcellin and T. R. Fischer. Trellis coded quantization of memoryless and Gauss-Markov sources. *IEEE Trans. Comm.*, 38(1):82–93, Jan. 1996.

[56] China Mobile. C-RAN: The road towards green RAN. White Paper, ver. 2.5, China Mobile Research Institute, Oct. 2011.

[57] A. Mohamed, O. Onireti, M. Imran, A. Imran, and R. Tafazolli. Control-data separation architecture for cellular radio access networks: A survey and outlook. *To appear in IEEE Commun. Surveys Tuts.*, 2015.

[58] M. Naghsh, M. Modarres-Hashemi, S. Shahbazpanahi, M. Soltanalian, and P. Stoica. Unified optimization framework of multi-static radar code design using information-theoretic criteria. *IEEE Trans. Sig. Proc.*, 61(21):5401–5416, Nov. 2013.

[59] M. M. Naghsh, M. Modarres-Hashemi, S. Shahbazpanahi, M. Soltanalian, and P. Stoica. Unified optimization framework for multi-static radar code design using information-theoretic criteria. *IEEE Trans. Signal Process.*, 61(21):5401–5416, Nov. 2013.

[60] M. Nahas, A. Saadani, J. Charles, and Z. El-Bazzal. Base stations evolution: Toward 4G technology. In *Proc. Int. Conf. on Telecommunications (ICT)*, 2012.

[61] NGMN Alliance. Further study on critical C-RAN technologies, White paper. 2015.

[62] Y. Polyanskiy. Channel coding: non-asymptotic fundamental limits. Ph.D. thesis, Princeton university, 2010.

[63] Y. Polyanskiy, H.V. Poor, and S. Verdú. Channel coding rate in the finite blocklength regime. *IEEE Trans. Inf. Theory*, 56(5):2307–2359, May 2010.

[64] D. Qiao, Y. Wu, and Y. Chen. Massive MIMO architecture for 5G networks: Co-located, or distributed? In *11th International Symposium on Wireless Communications Systems*, pages 192–197, Aug. 2014.

[65] R. Sassioui, E. Pierre-Doray, L. Szczecinski, B. Pelletier. Modelling decoding errors in HARQ.

[66] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan. Odessa: Enabling interactive perception applications on mobile devices. In *Proc. of the 9th International Conf. on Mobile Systems, Applications, and Services*, pages 43–56.

[67] M. Razaviyayn, M. Hong, and Z. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, Jun. 2013.

[68] M. Razaviyayn, M. Sanjabi, and Z.-Q. Luo. A stochastic successive minimization method for nonsmooth nonconvex optimization with applications to transceiver design in wireless communication networks. *arXiv*, Jul. 2013.

[69] A. W. Rihaczek. *Principles of High-Resolution Radar.* New York: Wiley, 1969.

[70] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Cáceres, M. Ben-Yehuda, W. Emmerich, and F. Galán. The reservoir model and architecture for open federated cloud computing. *IBM J. Res. Dev.*, 53(4):535–545, Jul. 2009.

[71] P. Rost and A. Prasad. Opportunistic hybrid ARQ-enabler of centralized-RAN over nonideal backhaul. *IEEE Wireless Commun. Letters*, 3(5):481–484, Oct. 2014.

[72] B. G. Ryder. Constructing the call graph of a program. *IEEE Trans. Softw. Eng.*, 3(3):216–226, May 1979.

[73] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo. Communicating while computing: Distributed mobile cloud computing over 5G heterogeneous networks. *IEEE Signal Process. Mag.*, 16(1):369–392, Nov 2014.

[74] S. Sardellitti, G. Scutari, and S. Barbarossa. Joint optimization of radio and computational resources for multicell mobile cloud computing. *CoRR*, abs/1412.8416, Dec. 2014.

[75] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23, Oct.-Dec. 2009.

[76] I. Stanojev, O. Simeone, and Y. Bar-Ness. Performance analysis of collaborative hybrid-ARQ incremental redundancy protocols over fading channels. In *Proc. IEEE Signal Processing Advances in Wireless Communications*, pages 1–5, Cannes, France, Jul. 2006.

[77] T. M. Cover and J. A. Thomas. *Element of Information Theory.* John Wiley & Sons, 2006.

[78] A. M. Tulino and S. Verdu. *Random Matrix Theory and Wireless Communications.* Now Publishers Inc, 2004.

[79] P. K. Varshney. *Distributed Detection and Data Fusion.* Springer, 1997.

[80] R. Viswanathan and P. K. Varshney. Distributed detection with multiple sensors: Part I – Fundamentals. *Proc. of the IEEE*, 85(1):54–63, Jan. 1997.

[81] R. Wang, H. Hu, and X. Yang. Potentials and challenges of C-RAN supporting multi-RATs toward 5G mobile networks. *IEEE Access*, 2:1187–1195, Sep. 2014.

[82] D. Wubben, P. Rost, J.S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis. Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through Cloud-RAN. *IEEE Signal Process. Mag.*, 31(6):35–44, Nov. 2014.

[83] K. Yang, S. Ou, and H.-H. Chen. On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications. *IEEE Commun. Mag.*, 46(1):56–63, Jan. 2008.

[84] W. Yang, G. Durisi, T. Koch, and Y. Polyanskiy. Quasi-static MIMO fading channels at finite blocklength. *CoRR*, abs/1311.2012, 2013.

[85] R. Zamir and M. Feder. On lattice quantization noise. *IEEE Trans. Info. Th.*, 42(4):1152–1159, Jun. 1996.

[86] W. Zhang, Y. Wen, and D.O. Wu. Collaborative task execution in mobile cloud computing under a stochastic wireless channel. *IEEE Trans. Wireless Commun.*, 14(1):81–93, Jan. 2015.

[87] L. Zhou, Z. Yang, J. J. P. C. Rodrigues, and M. Guizani. Exploring blind online scheduling for mobile cloud multimedia services. *IEEE Wireless Communications*, 20(3):54–61, Jun. 2013.