Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Ingegneria Elettronica, Telecomunicazioni e Tecnologie dell'Informazione

Ciclo XXX

Settore Concorsuale: 09/F2 - TELECOMUNICAZIONI

Settore Scientifico Disciplinare: ING-INF/03 - TELECOMUNICAZIONI

TITOLO TESI

Heterogeneous wireless networks for smart cities

Presentata da: Alex Calisti

Coordinatore Dottorato **Supervisore**

Dr. Ing. Alessandro Vanelli Coralli Dr. Ing. Gianni Pasolini

Esame finale anno 2018

Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Electronics Engineering, Telecommunications and Information Technology

Ciclo XXX

Settore Concorsuale: 09/F2 - TELECOMUNICAZIONI

Settore Scientifico Disciplinare: ING-INF/03 - TELECOMUNICAZIONI

TITOLO TESI

Heterogeneous wireless networks for smart cities

Presentata da: Alex Calisti

Coordinatore Dottorato

Dr. Ing. Alessandro Vanelli Coralli

Supervisore

Dr. Ing. Gianni Pasolini

Esame finale anno 2018

UNIVERSITÀ DEGLI STUDI DI BOLOGNA

Department of Electrical, Electronic and Information Engineering (DEI)

Ph.D programme in Electronics Engineering, Telecommunications And Information Technology

XXX Cycle

Heterogeneous wireless networks for smart cities

Candidate:
Alex Calisti

Supervisor:
Dr. Eng. Gianni Pasolini

Co-Supervisor:
Prof. Eng. Davide Dardari
Dr. Eng. Barbara Mavì Masini

Contents

List of Tables

List of Figures

List of acronyms

ADC	Analog-to-Digital Converter
AWGN	Additive White Gaussian Noise
BSS	Basic Service Set
ccdf	complementary cumulative distribution function
CDF	Cumulative Distribution Function
CSK	Color Shift Keying
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSVN	Crowd Sensing Vehicular Network
DAC	Digital-to-Analog Converter
DSRC	Dedicated Short-Range Communications
DVB-T	Digital Video Broadcasting-Terrestrial
ED	Energy Detector
ESD	Energy Spectral Density
EU	European Union
FIM	Fisher Information Matrix
FIR	Finite Impulse Response
FOV	Field Of View
FL	Flooding
FT	Fourier Transform
FTC	Finite-Time Consensus
FTCM	Finite-Time Consensus with Memory
GDOP	Geometric Dilution Of Precision
GF	Galois Field
GFA	Greedy Forwarding
GLRT	Generalized Likelihood Ratio Test
GP	Gaussian process
GPS	Global Positioning System
HPA	High Power Amplifier
IID	Independent Identically Distributed
ILP	Ideal Low Pass

IMU	Inertial Measurement Unit
IoV	Internet of Vehicles
LED	Light Emitting Diode
LMS	Least Mean Square
LOS	Line-Of-Sight
LRT	Likelihood Ratio Test
LS	Least Squares
LSCR	Leave-out Sign-dominant Correlation Regions
LSI	Linear Space Invariant
LTE	Long Term Evolution
MAC	Medium Access Control
MC	Metropolis Consensus
MF	Matched Filter
MFN	Multi Frequency Network
ML	Maximum Likelihood
MSE	Mean Square Error
NC	Network Coding
NEMSE	Normalized Estimation Mean Square Error
NHTSA	National Highway Traffic Safety Administration
NL	nonlinear
NLOS	Non-Line-Of-Sight
OBU	On Board Unit
OCR	On-Channel Repeater
OFDM	Orthogonal Frequency Division Multiplexing
OOK	On Off Keying
O-QPSK	Offset-Quadrature Phase Shift Keying
OWC	Optical Wireless Communication
PDF	Probability Density Function
PEB	Position Error Bound
PHY	Physical
PPP	Poisson Point Process
PSD	Power Spectral Density
PSP	Poisson Sampling Process
RF	Radio Frequency
RMSE	Root Mean Square Error

RSS	Received Signal Strength
RSU	Road Side Unit
RTT	Round-Trip Time
RV	Random Variable
SFN	Single Frequency Network
SINR	Signal to Noise and Interference Ratio
SNR	Signal to Noise Ratio
SPS	Sign Perturbed Sums
SV	Smart Vehicle
TAS	Tagged and Aggregated Sums
US	United States
V2R	Vehicle-to-Roadside
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad hoc NETwork
VLC	Visible Light Communication
VN	Vehicular Network
VPPM	Variable Pulse-Position Modulation
VSN	Vehicular Sensor Network
VVLN	Vehicular Visible Light Network
WAVE	Wireless Access in Vehicular Environment
WSN	Wireless Sensor Network

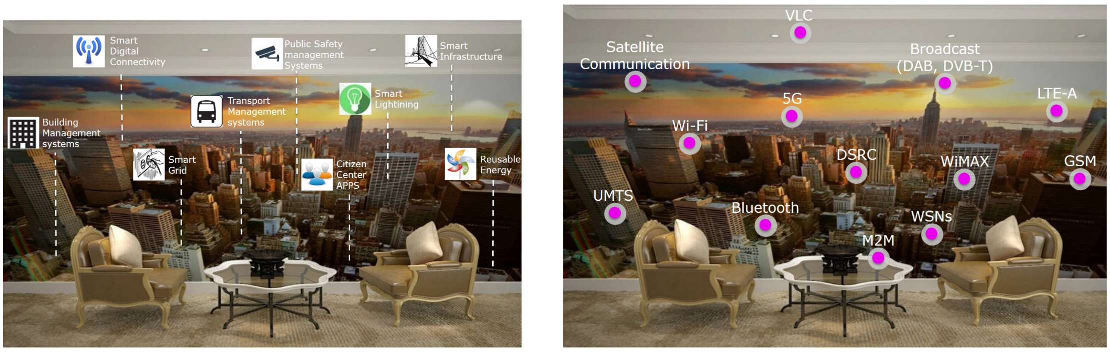
Introduction

In the near future, a world of smart cities is envisioned, in which everyone has one or more smart devices (such as smartphones, tablets, wearables) with sensing and communication capabilities, while many other smart objetcs are present everywhere to compose the so called Internet of Things (IoT).

Even now, devices equipped with sensors and communication interfaces have become ubiquitous. These devices can be used to collect and share data in order to derive maps or infer information on some parameter of interest, such as temperature, pollution or vehicular traffic. This technique is known as **Crowdsensing**.

The information sensed and collected by these devices are helpful for many applications (Figure 1(a)): cars could exchange information to optimize the traffic management in vehicular networks, safety devices could store and improve the quality of life in modern urban areas, sensors could exchange information to localize a given target.

Wireless communications are enabling this smart city paradigms (Figure 1(b)), where people, knowledge, devices and information are networked for the growth of society, life, and business. This scenario opens new challenges to wireless network designers, along with new performance metrics, coverage and privacy needs, as well as the need for a tighter integration of different networks. This is the fundamental concept of **Heterogeneous Networks** (Figure 1).



(a) Different applications.

(b) Cooperation between available networks.

Figure 1: Heterogeneous Networks for the future Smart Cities

The availability of high rate communication devices equipped with many different technologies will allow, in the next future, the provision of many services, which will dramatically increase the amount of data traffic. A deep investigation on the performance levels provided by the technologies that are envisioned to support such services is thus needed.

This kind of investigation requires the adoption of an integrated approach which jointly considers all aspects that affects the performance level experienced by users, starting from and the physical level (the presence of interference, noise, fading, the adopted modulation technique, etc.) up to the higher layers (the adoption of error correcting or revealing codes, the medium access control strategy, the error resilience strategies, etc.).

This approach has been the guide line throughout the whole research activity performed during the three years of my Ph.D. course, whose results are reported in this thesis.

My research project aims at studying new advanced techniques to allow cooperation and coexistence among heterogeneous devices and data. The vast information collected by devices can be used to enable a variety of new services addressing safety, traffic management, smart navigation, pollution measurements, urban surveillance, forensic investigations, and Internet access. The following topics will be addressed:

- Multi-modal heterogeneous networks and optimal management of resources.

- Optimal retrieval of relevant data and optimal routing toward the final users: connectivity, information sharing, cloud computing, and name data networking (NDN).

- Cooperative wireless telecommunication networks enabling smart mobility: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Low latency physical (PHY) and medium access control (MAC) techniques (OFDMA, CSMA, non orthogonal waveforms).

- Integrated radio-access technologies enabling the 5G networked society: new ideas toward an efficient handling of a very large number of devices with widely varying requirements. Design of advanced PHYs for the coexistence of heterogeneous data flows (high rate flows and several low rate flows) still maintaining a high efficiency.

The research project aims at formulating and solving in a unified framework these problems with both analytical and simulation-based approaches. The activity has been developed within the group leaded by Prof. Andrisano, by taking advantage of the experience developed in the field of wireless communications applied to sensor networks, vehicular networks, signal and data processing, and of the available simulation platform or preliminary test.

This activity has been devoted, in particular, to the investigation of the performance achievable by WSNs (Chapters 1, 2, 3, 4) and VNs (Chapter 5).

The introduction of echo cancellers in on-channel repeaters is also considered and, in particular, its performance in terms of stability problems arising from the joint effect of echo estimation errors and the quantization of the echo-cancelling filter taps is analyzed (Chapter 6).

A more theoretical investigation has been, furthermore, carried out on the performance of different information diffusion algorithms and a novel information distribution strategy, called tagged and aggregated sums (TAS), specifically designed to support the exchange of information between devices in a network, was also experimentally investigated in different network topologies (Chapter 2 and Chapter 3).

During my Ph.D, I also investigated the possibility to enable vehicle-to-vehicle (V2V) communication through Visible Light Communication (VLC) also when jointly adopted with Dedicated Short Range Communication (DSRC) for data acquisition in vehicular networks (Chapter 5).

The thesis is organized as follows.

- Focusing on the concept of Heterogeneous Networks in which sensor nodes have usually an irregular deployment, in Chapter 1 a brief overview on the concept of random sampling is presented. With the goal to introduce the reader to the concept of sampling in WSNs and provide an answer to the following questions i) is it preferable to have few accurate samples or many inaccurate ones? ii) when the capacity-per-volume or the sensor lifetime dominate the estimation accuracy of a large WSN?

  The answer to these questions has also led to the publication of a conference paper [1].

- In Chapter 2, I focus my attention on the question "what about the accuracy of the estimations?". In many applications, a simple point estimate of the parameter of interest is not sufficient. Therefore, a confidence region must be computed in order to assess its uncertainty. To this aim, sensor nodes must exchange their data, which could results in a significant burden for the communication network. In order to compute the confidence region associated with the estimated data, reducing the impact of the information exchange phase, I have introduced a new data dissemination strategy called Tagged and Aggregation Sums (TAS), which aggregates information in an efficient way before its dissemination. During this work, many simulation activities through Matlab have been performed and several implementations on an actual scenario were also investigated programming 52 nodes that are a part of Data Sensing and Processing Testbed (DataSens) of the European Laboratory of Wireless Communications for the Future Internet (EUWIN) in Bologna. The results of this work provide the level of accuracy of the estimation with a reduced amount of exchanged data.

  This work has led to the publication of a technical report [2], a conference paper [3] and a journal paper [4].

- Remaining in the framework of distributed computation in WSNs, in Chapter 3, I investigate average consensus strategies. I consider several information diffusion strategy (TAS, Flooding, Network Coding, Consensus Metropolis, Finite-Time consensus) and introduce a new information diffusion and distillation algorithm that, after a finite number of rounds, allows the computation of the average consensus at each node. Also in this work, many simulation activities through Matlab have been performed in order to evaluate the capability of the algorithms to achieve the average consensus. The outcomes of performance investigations, carried out considering unstructured random networks, tree networks, and clustered networks, show that the new introduced algorithm is very well behaving when operated on unstructured random network topologies, whereas TAS outperforms its competitors when structured networks are considered, either tree or clustered networks.

  This work originated a paper currently submitted to the ICASSP 2018 conference. Future work will be dedicated to the generalization of average consensus for which a journal paper is under construction.

- Once understood the meaning of random sampling and investigated the distributed computation in terms of confidence region and average consensus. In Chapter 4, I also investigate the challenging situation of networks operating with intermittent connectivity in Delay Tolerant Networks (DTNs). In DTNs agents are mobile nodes and communication is established only between closely located nodes producing frequent link disruptions and network topology reconfiguration. This time-varying nature exposes DTNs to infiltrations by potentially malicious nodes, who may attempt to perturb the DTN behaviour (Byzantine attacks). I assumed that the network behaviour is perturbed both by nodes with defective sensors and by nodes performing Byzantine attacks. An algorithm for distributed faulty detection (DFD) in DTNs has been introduced. This work has been done in order to determine whether the Distributed Faulty Node Detection (FND) algorithm, proposed in previous work, is robust against the introduction of Byzantine nodes and understand how to adjust the algorithm parameters to minimize the effects of the Byzantine attack. The DFD algorithm is executed considering node inter-contact times taken from real databases provided by our own experiments conducted at the EuWin platform at University of Bologna. I realized the real database in Bologna using the EuWin platform. Students of the university of Bologna have been equipped of a device during a break of an academic course, then they have spent the break as usual while each device counted the number of meetings with other devices. In the simulation, one is interested in the inter-contact trace, i.e., which pair of agents have a meeting at which time. Results show the robustness of the DFD algorithm and the way to optimize the choice of the algorithm parameters to minimize the effects of Byzantine attacks.

This work has led to a conference paper [5].

- The limited RF bandwidth shared among several applications for an ever increasing amount of data pushes researchers to look at new technological solutions. An actual option are the large, unlicensed, and uncongested bands enabled by VLC. VLC offers a great opportunity to exploit optical communications and to enable new applications. Specifically, since the acquisition of environmental information from vehicles is expected to overload cellular networks, I investigated the possibility to use VLC technology jointly to short range vehicle-to-vehicle and vehicle-to-roadside wireless communication technologies, such as IEEE 802.11p. Based on this, In Chapter 5 I investigated on the possibility to enable vehicle-to-vehicle (V2V) communication through VLC also when jointly adopted with dedicated short range communication (DSRC) for data acquisition in vehicular networks. I studied the impact of VLC between vehicles for data exchanging and cellular network offload exploiting traffic lights as road side units to collect information toward a remote control center. I adopted the simulation platform for heterogeneous interworking networks (SHINE), developed at Wilab, to provide realistic results in terms of communication system performance, introducing the possibility to simulate the IEEE 802.15.7 standard from the application layer down to the physical layer. A cooperative algorithm to adaptively select the best available V2V communication technology has been also proposed. Simulation results showed the significant improvement obtained by the use of VLC and DSRC Heterogeneous Networks.

  This work has led to a conference paper [6] and a journal paper [7].

- In Chapter 6 the most relevant impairment experienced by OnChannel Repeaters (OCRs), that is, the coupling-channel between the transmitting and receiving antennas, is investigated. The echoes generated critically influence the overall system behavior, with harmful effects on the signal quality and, above all, pose a threat on the system integrity. Usually, echo cancellers are adopted to remove unwanted coupling contributions. I investigate the OCR stability problems arising from the joint effect of echo estimation errors and the quantization of the echo-cancelling filter taps. The probability of OCR instability was analytically derived for different channel models and OCR settings. In particular, I derived an analytical expression for the upper bound of the probability of instability as a function of estimation SNR and number of quantization bits.

  This work has led to a conference paper [8].

Chapter 1

Random Sampling

1.1 Introduction

Self-organizing WSN has attracted considerable attention in the last four years [9]. The possibility to create a network infrastructure composed of low-cost, small-size and energy-limited sensing devices (sensors) has created a large number of applications in smart home, environmental monitoring, crowdsensing and internet of thing.

For the classical regular sampling, the well-known Whittaker-Kotelnikov-Shannon sampling theorem states that a signal can be perfectly reconstructed from its samples provided that the sampling frequency is almost twice its bandwidth. For the irregular sampling, a theorem of Landau [10, 11] establishes necessary conditions on samples density for the perfect estimation. In the case of a random sampling in time, the estimation accuracy is usually evaluated in terms of MSE [12]. In particular, if the sample positions are the output of a stationary Poisson Point Process (PPP), Marvasti [13] shows that the reconstruction is still possible via ILP filtering provided that the average samples density is higher than twice the signal bandwidth, and that the spectrum of the estimated signal is that of the original signal immersed in a white noise floor.

Recently, the extension of Marvasti's result to a multidimensional domain has gained interest due to its application on WSN for environmental monitoring [14–19]. The assumption that nodes in a WSN are deployed according to a homogeneous PPP (homogeneous in $\mathbb{R}^d$ corresponds to stationary in $\mathbb{R}$) is widely adopted [20–23]. However, many works do not consider that the ILP filtering is no longer optimal for irregular sampling, as widely known in the one-dimension domain in case of jitter [24–26]. Moreover, in a realistic sensing scenario, the distortion due to measurement errors and sensors energy consumption have to be considered [27–29].

The estimation of spatial processes from sparse sensing nodes is fundamental for many applications, including environmental monitoring and crowd-sourcing.

Based on this, I analyzed the impact of measurement errors on the estimation of a finite-energy signal sampled by a set of sensors randomly deployed in a finite d-dimensional space according to homogeneous PPP. The analytical expressions of both the estimated signal energy spectral density and the NEMSE are obtained. The optimal linear space invariant (LSI) interpolator is derived by jointly taking into account random sampling, measurement errors and energy consumption finding an expression in $\mathbb{R}^d$ that is consistent to that found in the time domain for the stationary Poisson sampling process. An asymptotic analysis for sensors density high with respect to the signal bandwidth is given for scenarios subject to estimation energy constraint. The NEMSE is derived for large WSNs with limitations in the capacity-per-volume constraint and in the battery duration. In particular, the estimated signal ESD and NEMSE are derived as functions of important parameters such as samples density and distortion. These results aim to answer the following question in multidimensional signal estimation: *given a certain amount of energy spent for the estimation, is it better to have few accurate samples or many inaccurate ones?* However, in an actual WSN, the most relevant estimation energy constraint is not on the whole network, but on each sensor, due to the battery duration limitation. In addition, a constraint on the capacity of each sensor for sending the samples to the interpolation entity has to be taken into account. Therefore, another relevant question is: *when the estimation accuracy of a large WSN is dominated by the capacity-per-volume and when by the sensor lifetime?* By modeling the communication channel between each nodes and the interpolator as an erasure channel, and by considering the capacity-per-volume as a constraint for a large WSN, I derived a simple analytical expression for the NEMSE as a function of both the estimation rate and the capacity-per-volume.

1.2 Signal Estimation

Consider the signal $z(\mathbf{x}) \in \mathbb{C}$ with support $\mathcal{A} \subseteq \mathbb{R}^d$, finite energy E_z, and ESD $\mathcal{E}_z(\boldsymbol{\nu})$, $\triangleq \mathcal{F}\left\{\int_{\mathbb{R}^d} z(\mathbf{x})z^{\dagger}(\mathbf{x}-\boldsymbol{\tau})d\boldsymbol{\tau}\right\}$, where $\mathbf{x}, \boldsymbol{\nu} \in \mathbb{R}^d$. Let $\mathcal{F}\{\cdot\}$ be the d-dimensional Fourier transformation [30]. The homogeneous Poisson sampling process can be expressed as

$$\mathcal{P}(\mathbf{x}) = \sum_{n \in \mathcal{N}(\Pi)} \delta(\mathbf{x} - \mathrm{x}_n) \tag{1.1}$$

where $\mathcal{N}(\Pi)$ is the index set of the homogeneous PPP [31] with intensity ρ and $\delta(\cdot)$ indicates the generalized Dirac delta function in $\mathbb{R}^d$. It is known [32] that

$$\mathbb{E}\{\mathcal{P}(\mathbf{x})\} = \rho \tag{1.2a}$$

$$\mathbb{E}\{\mathcal{P}(\mathbf{x})\mathcal{P}(\mathbf{x}-\boldsymbol{\tau})\} = \rho^2 + \rho\delta(\boldsymbol{\tau}) \ . \tag{1.2b}$$

The sampled signal with measurement errors is given by

$$\mathrm{z}_{\epsilon}(\mathbf{x}) = \sum_{n \in \mathcal{N}(\Pi_{\mathcal{A}})} \hat{\mathrm{z}}_n \delta(\mathbf{x} - \mathrm{x}_n) \tag{1.3}$$

where $N(\Pi_{\mathcal{A}})$ is the index set of the points of Π falling in $\mathcal{A}$. Process $\mathcal{P}(\mathbf{x})\mathbf{1}_{\mathcal{A}}(\mathbf{x})$ inside the signal support $\mathcal{A}$ and $\mathbf{1}_{\mathcal{A}}(\mathbf{x})$ denotes the indicator function equal to 1 for $\mathbf{x} \in \mathcal{A}$ and 0 otherwise. For $\mathbf{x} \notin \mathcal{A}$, the n-th sample affected by measurement is

$$\hat{\mathsf{z}}_n = z(\mathsf{x}_n) + \mathsf{e}_n \tag{1.4}$$

where the $\mathsf{e}_n\,'s$ are zero-mean independent random variables (RVs) with variance σ_n^2, independent of $\mathcal{P}(\mathbf{x})$. The distortion due to measurement errors is defined as

$$D \triangleq \mathbb{E}\left\{ \sum_{n \in \mathcal{N}(\Pi_{\mathcal{A}})} |\mathsf{e}_n|^2 \right\}. \tag{1.5}$$

Consider the estimation performed by an LSI interpolator $\theta(\mathbf{x})$ with Fourier transform $\Theta(\boldsymbol{\nu})$. In the d-dimensional domain, the term space-invariant takes the place of the usual time-invariant in the one-dimension domain. The estimated signal is given by

$$\hat{\mathsf{z}}(\mathbf{x}) = (\mathsf{z}_\epsilon * \theta)(\mathbf{x}) = \sum_{n \in \mathcal{N}(\Pi_{\mathcal{A}})} \hat{\mathsf{z}}_n \theta(\mathbf{x} - \mathsf{x}_n). \tag{1.6}$$

Two metrics are employed to evaluate the signal estimation accuracy: the ESD of the estimated signal (1.6) and the NEMSE. The former is

$$\mathcal{E}_{\hat{\mathsf{z}}}(\boldsymbol{\nu}) \triangleq \mathcal{F}\left\{ \int_{\mathbb{R}^d} \mathbb{E}\left\{ \hat{\mathsf{z}}(\mathbf{x})\hat{\mathsf{z}}^\dagger(\mathbf{x} - \boldsymbol{\tau}) \right\} d\boldsymbol{\tau} \right\} = \mathbb{E}\left\{ |\hat{\mathsf{Z}}(\boldsymbol{\nu})|^2 \right\} \tag{1.7}$$

with $\hat{\mathsf{Z}}(\boldsymbol{\nu}) \triangleq \mathcal{F}\{\hat{\mathsf{z}}(\mathbf{x})\}$. The latter is

$$\epsilon_{\mathrm{S}} \triangleq \frac{\int_{\mathbb{R}^d} |z(\mathbf{x}) - \hat{\mathsf{z}}(\mathbf{x})|^2}{E_z} \tag{1.8}$$

where the expectation is with respect the measurement errors and the samples positions. To perform an asymptotic analysis for large sample intensity, I also introduce the following quantities normalized to the signal bandwidth-per-dimension B_z: the (normalized) spatial frequency $\breve{\boldsymbol{\nu}} \triangleq \boldsymbol{\nu}/2B_z$, the Poisson sampling process intensity $\breve{\rho} \triangleq \rho/(2B_z)^d$, distortion $\breve{D} \triangleq \frac{D}{E_z(2B_z)^d}$, signal ESD $\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) \triangleq \frac{(2B_z)^d}{E_z}\mathcal{E}_z\left(2B_z\breve{\boldsymbol{\nu}}\right)$, and estimated signal ESD $\breve{\mathcal{E}}_{\hat{\mathsf{z}}}(\breve{\boldsymbol{\nu}}) \triangleq \frac{(2B_z)^d}{E_z}\mathcal{E}_{\hat{\mathsf{z}}}\left(2B_z\breve{\boldsymbol{\nu}}\right)$. For finite energy signals with infinite band in $\mathbb{R}^d$, I consider, for normalization purpose, the extension of the well-known Gabor's bandwidth to $\mathbb{R}^d$, i.e. $B_z \triangleq \sqrt{\int_{\mathbb{R}^d} \frac{|\boldsymbol{\nu}|^2\mathcal{E}_z(\boldsymbol{\nu})d\boldsymbol{\nu}}{E_z}}$.

Lemma 1 (Optimal LSI Interpolator) *The transfer function of the LSI which minimizes the NEMSE defined in* (1.8) *is given by*

$$\Theta(\boldsymbol{\nu}) = \frac{\rho\mathcal{E}_z(\boldsymbol{\nu})}{\rho^2\mathcal{E}_z(\boldsymbol{\nu}) + \rho(E_z + \sigma_M^2|\mathcal{A}|)} = \frac{\mathcal{E}_z(\boldsymbol{\nu})}{\rho\mathcal{E}_z(\boldsymbol{\nu}) + E_z\left(1 + \frac{D}{E_z\rho}\right)}. \tag{1.9}$$

Proof: By extending the Wiener filtering theory [33, 34] it can be shown that the LSI interpolator minimizing to $\mathbb{R}^d$ (1.8) results in

$$\Theta(\boldsymbol{\nu}) = \frac{\mathcal{E}_{z,\mathsf{z}_\epsilon}(\boldsymbol{\nu})}{\mathcal{E}_{\mathsf{z}_\epsilon}(\boldsymbol{\nu})} \tag{1.10}$$

where $\mathcal{E}_{z,\mathsf{z}_\epsilon}(\boldsymbol{\nu}) \triangleq \mathcal{F}\left\{R_{z,\mathsf{z}_\epsilon}(\boldsymbol{\tau})\right\}$ and $\mathcal{E}_{\mathsf{z}_\epsilon}(\boldsymbol{\nu}) \triangleq \mathcal{F}\left\{R_{\mathsf{z}_\epsilon}(\boldsymbol{\tau})\right\}$ with $R_{z,\mathsf{z}_\epsilon}(\boldsymbol{\tau}) \triangleq \int_{\mathbb{R}^d} \mathbb{E}\left\{z(\mathbf{x})\mathsf{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} d\mathbf{x}$ and $R_{\mathsf{z}_\epsilon}(\boldsymbol{\tau}) \triangleq \int_{\mathbb{R}^d} \mathbb{E}\left\{\mathsf{z}_\epsilon(\mathbf{x})\mathsf{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} d\mathbf{x}$. In the sense of distributions, we have from (1.6) that

$$\mathsf{z}_\epsilon(\mathbf{x}) = z(\mathbf{x}) \sum_{n\in\mathcal{N}(\Pi)} \delta(\mathbf{x}-\mathsf{x}_n) + \sum_{n\in\mathcal{N}(\Pi_{\mathcal{A}})} \mathsf{e}_n \delta(\mathbf{x}-\mathsf{x}_n) \tag{1.11}$$

where the first term follows by the definition of support of $z(\mathbf{x})$. From (1.40a), (1.40b), (1.11), and the independence between e_n and $\mathbf{x}_n$, it follows that

$$\begin{aligned}
\mathbb{E}\left\{\mathsf{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} &= \rho z^\dagger(\mathbf{x}-\boldsymbol{\tau}) \\
\mathbb{E}\left\{\mathsf{z}_\epsilon(\mathbf{x})\mathsf{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} =& \mathbb{E}\left\{z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})\mathcal{P}(\mathbf{x})\mathcal{P}(\mathbf{x}-\boldsymbol{\tau})\right\} \\
&+ \mathbb{E}\left\{\sum_{n\in\Pi(\mathcal{A})}\sum_{k\in\Pi(\mathcal{A})} \mathsf{e}_n\mathsf{e}_k^\dagger \delta(\mathbf{x}-\mathsf{x}_n)\delta(\mathbf{x}-\boldsymbol{\tau}-\mathsf{x}_k)\right\} \\
=& z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})\mathbb{E}\left\{\mathcal{P}(\mathbf{x})\mathcal{P}(\mathbf{x}-\boldsymbol{\tau})\right\} \\
&+ \mathbb{E}\left\{\sum_{n\in\Pi(\mathcal{A})} |\mathsf{e}_n|^2 \delta(\mathbf{x}-\mathsf{x}_n)\delta(\mathbf{x}-\boldsymbol{\tau}-\mathsf{x}_n)\right\} \\
&= z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})[\rho^2 + \rho\delta(\boldsymbol{\tau})] \\
+&\mathbb{E}\left\{\sum_{n\in\mathcal{N}(\Pi_{\mathcal{A}})} \sigma_n^2 \delta(\mathbf{x}-\mathsf{x}_n)\delta(\mathbf{x}-\boldsymbol{\tau}-\mathsf{x}_n)\right\}
\end{aligned}$$

which lead to

$$R_{z,\mathsf{z}_\epsilon}(\boldsymbol{\tau}) = \rho \int_{\mathbb{R}^d} z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})d\mathbf{x} \tag{1.12a}$$

$$R_{\mathsf{z}_\epsilon}(\boldsymbol{\tau}) = [\rho^2 + \rho\delta(\boldsymbol{\tau})] \int_{\mathbb{R}^d} z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})d\mathbf{x} + D\delta(\boldsymbol{\tau}) \,. \tag{1.12b}$$

By Fourier transforming (1.57) in $\mathbb{R}^d$ we obtain

$$\mathcal{E}_{z,\mathsf{z}_\epsilon}(\boldsymbol{\nu}) = \rho\mathcal{E}_z(\boldsymbol{\nu}) \tag{1.13a}$$

$$\begin{aligned}
\mathcal{E}_{\mathsf{z}_\epsilon}(\boldsymbol{\nu}) &= \mathcal{E}_z(\boldsymbol{\nu}) * [\rho^2\delta(\boldsymbol{\nu}) + \rho] + D \\
&= \rho^2\mathcal{E}_z(\boldsymbol{\nu}) + E_z\left(\rho + \frac{D}{E_z}\right)
\end{aligned} \tag{1.13b}$$

which, together with (1.10), provides (1.9).

Remark 1 *By using normalized quantities, for* $\breve{\rho} \to +\infty$*, the optimal LSI interpolator in* (1.9) *tends to* $\frac{1}{\rho}\mathbf{1}_{\mathcal{B}_z}(\boldsymbol{\nu})$ *where* $\mathbf{1}_{\mathcal{B}_z}(\boldsymbol{\nu})$ *denotes the indicator function equal to* 1 *for* $\boldsymbol{\nu} \in \mathcal{B}_z$ *and* 0 *otherwise. It means that the ILP filter considered in [14, 16, 17] is an asymptotic optimal choice when the samples intensity which is much higher than the signal band cardinality in* $\mathbb{R}^d$*.*

Remark 2 *In the particular case of* $D = 0$ *and* $d = 1$, (1.9) *reduces to the optimal linear time invariant interpolator for stationary Poisson sampling process found by Leneman in [24] once the ESD is replaced by the power spectral density.*

Corollary 1 (Normalized Estimated Signal ESD) *When the optimal LSI interpolator in* (1.9) *is employed, the normalized ESD of the estimated signal is given by*

$$\breve{\mathcal{E}}_{\hat{z}}(\breve{\boldsymbol{\nu}}) = \breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) \left[\frac{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}})}{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) + \frac{1}{\breve{\rho}}\left(1 + \frac{\breve{D}}{\breve{\rho}}\right)} \right] . \tag{1.14}$$

Proof: From (1.6) and (1.7), the ESD of the estimated signal $\hat{z}(\boldsymbol{\nu})$ results in $\mathcal{E}_{\hat{z}}(\boldsymbol{\nu}) = |\Theta(\boldsymbol{\nu})|^2 \mathcal{E}_{z_\epsilon}(\boldsymbol{\nu})$, which from (1.59b) and (1.9) becomes (1.14) in terms of normalized quantities.

$$\mathcal{E}_{\hat{z}}(\boldsymbol{\nu}) = \mathcal{E}_z(\boldsymbol{\nu}) \left[\frac{\mathcal{E}_z(\boldsymbol{\nu})}{\mathcal{E}_z(\boldsymbol{\nu}) + \frac{E_z}{\rho}\left(1 + \frac{D}{E_z \rho}\right)} \right] .$$

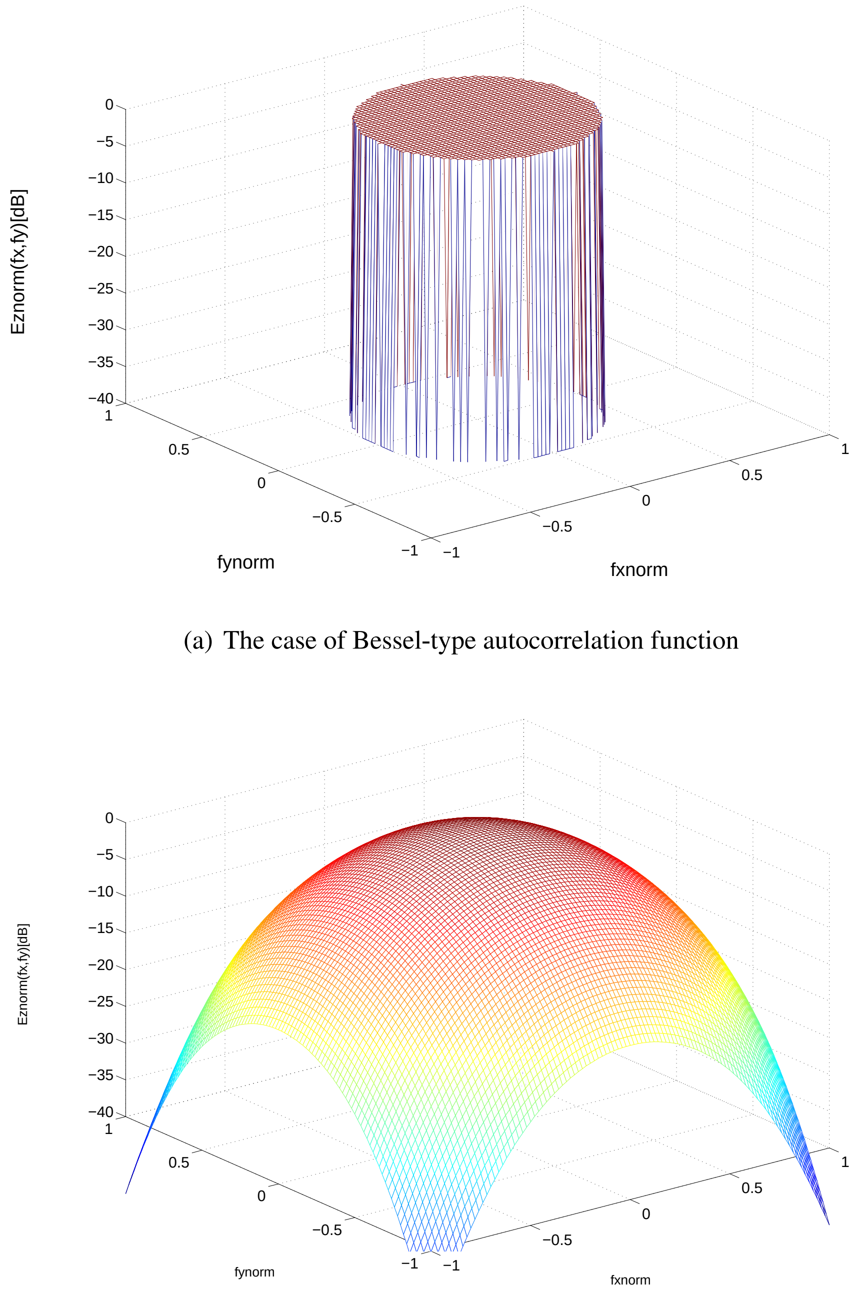
The normalized version is readily shown to result in (1.14).

Remark 3 *While the ILP interpolator causes a worse floor on the estimated signal ESD [13],* (1.14) *shows that the optimal ILP interpolator introduces a scaling factor.*

I now introduce two examples.

Example 1 (Bessel-type autocorrelation function) *Consider a signal* $z(\mathrm{x})$ *(*$\mathrm{x} \in \mathbb{R}^2$*) with a Bessel-type autocorrelation function such that its normalized ESD results in* $\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) = \frac{4}{\pi}\mathbf{1}_{C_0}(\breve{\boldsymbol{\nu}})$*, where* C_0 *denotes the* 2*-dimensional ball centered in the origin with radius* $1/2$*. The corresponding normalized estimated signal ESD according to* (1.14) *is depicted in Fig. 1.1(a).*

Example 2 (Gaussian-type autocorrelation function) *Consider a signal* $z(\mathrm{x})$ *(*$\mathrm{x} \in \mathbb{R}^2$*) with a Gaussian-type autocorrelation function, such that its normalized ESD results in* $\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) = \frac{4}{\pi}e^{-4|\breve{\boldsymbol{\nu}}|^2}$ *(i.e., an infinite band signal where the standard deviation is considered as the practical bandwidth per dimension for normalization purpose). The corresponding estimated signal normalized ESD according to* (1.14) *is depicted in Fig. 1.1(b).*



(a) The case of Bessel-type autocorrelation function

(b) The case of Gaussian autocorrelation function

Figure 1.1: Estimated signal normalized ESD in $\mathbb{R}^2$ for $\breve{\rho} = 10^2$ and $\breve{D} = 10^2$.

Note that, in both the examples, the typical noise floor due to homogeneous Poisson random sampling and measurement errors does not arise when using the optimal LSI interpolator (1.9).

Corollary 2 (NEMSE) *When the optimal LSI interpolator in* (1.9) *is employed, the NEMSE results in*

$$\epsilon_S = 1 - \int_{\mathbb{R}^d} \frac{\breve{\mathcal{E}}_z^2(\breve{\boldsymbol{\nu}})}{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) + \frac{1}{\breve{\rho}}\left(1 + \frac{\breve{D}}{\breve{\rho}}\right)} d\breve{\boldsymbol{\nu}} \,. \tag{1.15}$$

Proof: By using the fundamental isometry presented in [12], the NEMSE corresponding to the optimal LSI interpolator results in

$$\epsilon_{\mathrm{S}} = 1 - \frac{1}{E_z} \int_{\mathbb{R}^d} \frac{|\mathcal{E}_{z,\mathbf{z}_\epsilon}(\boldsymbol{\nu})|^2}{\mathcal{E}^{\dagger}_{\mathbf{z}_\epsilon}(\boldsymbol{\nu})} d\boldsymbol{\nu}$$

which, from (1.59a) and (1.59b), provides (1.15) in terms of normalized quantities.

Remark 4 *From* (1.15)*, the distortion effect can be described by an equivalent diminished normalized intensity*

$$\rho_D \triangleq \frac{\rho}{1 + \frac{D}{\rho E_z}} \tag{1.16}$$

or, in the terms of normalized intensity

$$\breve{\rho}_{\breve{D}} \triangleq \frac{\breve{\rho}}{1 + \frac{\breve{D}}{\breve{\rho}}} . \tag{1.17}$$

Example 3 (NEMSE for ILP ESD Signals) *For signals of the example 1, with an ILP signal with band $\mathcal{B}_z$ in $\mathbb{R}^d$, i.e. normalized ESD $\mathcal{E}_z(\boldsymbol{\nu}) = \frac{E_z}{(2B_z)^d}\mathbf{1}_{\mathcal{B}_z}(\boldsymbol{\nu})$, $\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) = \mathbf{1}_{\breve{\mathcal{B}}_z}(\breve{\boldsymbol{\nu}})$ (where $|\breve{\mathcal{B}}_z| = 1$ $\breve{\mathcal{B}}_z = \{\breve{\boldsymbol{\nu}} : 2B_z\breve{\boldsymbol{\nu}} \in \mathcal{B}_z\}$), the NEMSE results in $\epsilon_S = \left(\frac{1}{\breve{\rho}_{\breve{D}}}\right) / \left(1 + \frac{1}{\breve{\rho}_{\breve{D}}}\right)$.*

$$\begin{aligned} \epsilon_S &= 1 - \int_{\breve{\mathcal{B}}_z} \frac{1}{1 + \frac{1}{\breve{\rho}}\left(1 + \frac{\breve{D}}{\breve{\rho}}\right)} d\breve{\boldsymbol{\nu}} \\ &= \frac{\frac{1}{\breve{\rho}} + \frac{\breve{D}}{\breve{\rho}^2}}{1 + \frac{1}{\breve{\rho}} + \frac{\breve{D}}{\breve{\rho}^2}} \,. \end{aligned} \tag{1.18}$$

The ϵ_{S} as a function of $\breve{\rho}$ is shown in Fig. 1.2.

Example 4 (NEMSE for Gaussian ESD Signals) *For signals of the example 2, $\mathcal{E}_z(\boldsymbol{\nu}) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{|\breve{\boldsymbol{\nu}}|^2}{2\sigma^2}}$,*

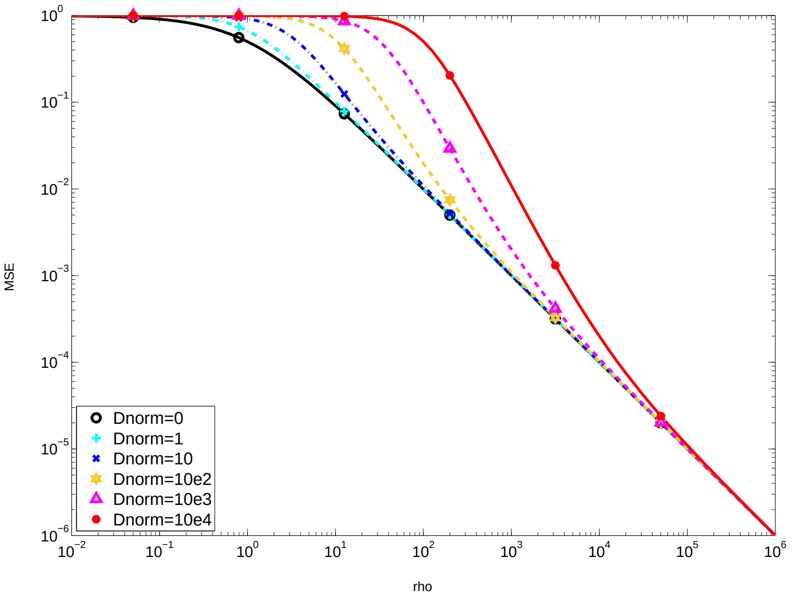


Figure 1.2: ILP ESD signals: NEMSE as a function of normalized sampling intensity

the Gabor bandwidth results in $B_z = \sqrt{d}\sigma$. *Thus* $\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) = \sqrt{\frac{2d}{\pi}}^{d} e^{-2d|\breve{\boldsymbol{\nu}}|^2}$.

$$\begin{aligned}
\epsilon_S &= 1 - \int_{\mathbb{R}^d} \frac{\frac{1}{(2\pi)^d} e^{-|\breve{\boldsymbol{\nu}}|^2}}{\frac{1}{(2\pi)^{d/2}} e^{-\frac{|\breve{\boldsymbol{\nu}}|^2}{2}} + \frac{1}{\breve{\rho}}\left(1+\frac{\breve{D}}{\breve{\rho}}\right)} d\breve{\boldsymbol{\nu}} \\
&= 1 - \int_{\mathbb{R}^d} \frac{e^{-|\breve{\boldsymbol{\nu}}|^2}}{(2\pi)^{d/2} e^{-\frac{|\breve{\boldsymbol{\nu}}|^2}{2}} + \frac{(2\pi)^d}{\breve{\rho}}\left(1+\frac{\breve{D}}{\breve{\rho}}\right)} d\breve{\boldsymbol{\nu}} \\
&= 1 - \frac{\sqrt{2d}^d S_d(1)}{(\pi)^{d/2}} F_d(\breve{\rho}_{\breve{D}}) \int_r \frac{e^{-r^2}}{(2\pi)^{d/2} e^{-\frac{r^2}{2}} + \frac{(2\pi)^d}{\breve{\rho}}\left(1+\frac{\breve{D}}{\breve{\rho}}\right)} r^{d-1} dr
\end{aligned} \tag{1.19}$$

where $S_d(1) = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})}$ *is the area of the d-dimensional unitary hyper-sphere*

$$S_d(1) = \frac{2\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}\right)} \tag{1.20}$$

and

$$F_d(R) \triangleq \int_0^R \frac{e^{-r^2}}{e^{-\frac{r^2}{2}} + \frac{(2\pi)^{d/2}}{\breve{\rho}}\left(1 + \frac{\breve{D}}{\breve{\rho}}\right)} r^{d-1} dr \tag{1.21}$$

which results in $F_d(\breve{\rho}_{\breve{D}}) \triangleq \int_0^{+\infty} \frac{e^{-4dr^2}}{e^{-2dr^2} + \frac{1}{\breve{\rho}_{\breve{D}}}\left(\frac{\pi}{2d}\right)^{d/2}} r^{d-1} dr$.

For $d = 2$, (1.19) *can be expressed in the closed form and the NEMSE results in* $\epsilon_S = \frac{\pi}{4\breve{\rho}_{\breve{D}}} \ln\left(1 + \frac{4\breve{\rho}_{\breve{D}}}{\pi}\right)$.

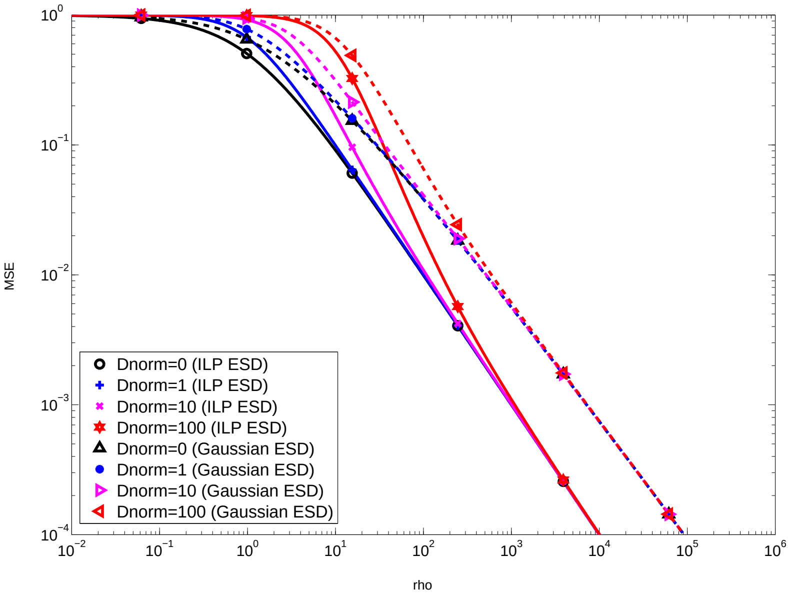


Figure 1.3: NEMSE as a function of normalized sampling intensity

Fig. 1.3 shows the NEMSE as a function of the normalized sampling intensity $\breve{\rho}$ for different values of the normalized distortion for both examples 3 and 4. It can be noticed that the distortion effects vanish for $\breve{\rho}$ approaching infinity. The different asymptote in the case of Gaussian ESD is due to its infinite signal bandwidth.

1.3 Constraints

Several constraints have been evaluated in order to better analyze this topic.

1.3.1 Estimation Energy Constraint

Consider the energy spent by the n-th sensor for the estimation is own $1/\sigma_n^2$ trough the hardware-dependent constant κ_{M} expressed in $\left[Ju^2\right]$ where u is the measurement unit for the signal (see Table 1.3.1). For $\sigma_n^2 = \sigma_{\mathrm{M}}^2$ for all n, The overall average energy spent in the measurement process results in

$$E_{\mathrm{M}} = \mathbb{E}\left\{ \sum_{n\in\Pi(\mathcal{A})} \frac{\kappa_{\mathrm{M}}}{\sigma_n^2} \right\} . \tag{1.22}$$

Table 1.1: Table of the main quantities related to an large WSN for signal estimation (u is the measurement unit for the signal).

Quantity	Significance	Unit
$z(\mathbf{x})$	Signal to be reconstructed	u
$\mathbf{x}$	Spatial coordinate in $\mathbb{R}^d$	m
$\boldsymbol{\nu}$	Spatial frequency coordinate in $\mathbb{R}^d$	m^{-1}
E_z	Energy of the signal	$u^2 \cdot m^d$
$\mathcal{E}_z(\boldsymbol{\nu})$	ESD of $z(\mathbf{x})$	$u^2 \cdot m^{2d}$
$\lvert\mathcal{A}\rvert$	Cardinality of $z(\mathbf{x})$ domain in $\mathbb{R}^d$	m^d
B_z	Bandwidth-per-dimension of $z(\mathbf{x})$	m^{-1}
D	Distortion due to sensing errors	u^2
E_{M}	Overall estimation energy	J
T	Time duration of the sensing	s
σ_{M}^2	Measurement error variance for each sensor	u^2
W_{s}	Power consumption for each sensor	W
κ_{M}	Proportional constant between $W_{\mathrm{s}}T$ and $1/\sigma_{\mathrm{M}}^2$	$J \cdot u^2$
c	Capacity-per-volume	$\frac{\text{symbol/ch.use}}{m^d}$
$\breve{\cdot}$	Normalized version of a quantity	

In the simple case $\sigma_n^2 = \sigma_{\mathrm{M}}^2$ for all n, it results $E_{\mathrm{M}} = c_0 \frac{\rho|\mathcal{A}|}{\sigma_{\mathrm{M}}^2}$. Since the distortion are equal to $D = \rho|\mathcal{A}|\sigma_{\mathrm{M}}^2$, it results

$$D = c_0 \frac{\rho^2 |\mathcal{A}|^2}{E_{\mathrm{M}}}. \tag{1.23}$$

The overall estimation energy and the distortion correspond to $E_{\mathrm{M}} = \kappa_{\mathrm{M}} \frac{\rho|\mathcal{A}|}{\sigma_{\mathrm{M}}^2}$ and $D = \rho|\mathcal{A}|\sigma_{\mathrm{M}}^2$, respectively. It follows

$$\breve{D} = \kappa_{\mathrm{M}} \frac{\rho^2 |\mathcal{A}|^2}{E_{\mathrm{M}} E_z (2B_z)^d} = \kappa_{\mathrm{M}}\, \breve{\rho}^2\, \frac{|\mathcal{A}|^2 (2B_z)^d}{E_{\mathrm{M}} E_z}\ . \tag{1.24}$$

By substituting (1.24) in (1.15), provides

$$\epsilon_{\mathrm{S}} = 1 - \int_{\mathbb{R}^d} \frac{\breve{\mathcal{E}}_z^2(\breve{\boldsymbol{\nu}})}{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) + \frac{1}{\breve{\rho}} + \kappa_{\mathrm{M}} \frac{|\mathcal{A}|^2 (2B_z)^d}{E_{\mathrm{M}} E_z}} d\breve{\boldsymbol{\nu}}\,. \tag{1.25}$$

Equation (1.25) shows that the NEMSE is decreasing with respect to the energy spent for the estimation E_{M} and increasing with respect to the signal domain Lebesgue measure $|\mathcal{A}|$ and to the signal band Lebesgue measure $(2B_z)^d$. The fact that (1.25) is also decreasing with respect the signal energy E_z is simply due to the normalization choice in (1.8).

Remark 5 *Given the overall estimation energy E_M,* (1.25) *shows that having more samples (higher $\breve{\rho}$) with higher measurement error variance σ_M^2 reduces the NEMSE with respect to the* *case of fewer samples with higher precision. This is thank to the employment of the optimal LSI interpolator* (1.9)*.*

Remark 6 *For every finite set of samples, the NEMSE is lower bounded by* $\epsilon_S > 1 - \int_{\mathbb{R}^d} \frac{\breve{\mathcal{E}}_z^2(\breve{\boldsymbol{\nu}})}{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) + \kappa_M \frac{|\mathcal{A}|^2 (2B_z)^d}{E_M E_z}} d\breve{\boldsymbol{\nu}}$*. Note that such a bound is asymptotically approached for both the normalized samples intensity and the normalized distortion approaching the infinity.*

1.3.2 Network Capacity Constraint

Consider now large WSN with constraints in the capacity of each sensor of sending the samples to the interpolation entity. Here I study the effect of network capacity constraint when a large WSN is employed to reconstruct the signal $z(\mathbf{x})$ (expressed in unit) in $\mathcal{A} \in \mathbb{R}^2$. Assume that the sensor positions are the output of a homogeneous PPP with intensity λ and that each sensor each sensor transmits the interpolation with probability q. Thus, the sampling point process results in (1.1) with $\rho = q\lambda$, that is

$$\breve{\rho} = \frac{q\lambda}{(2B_z)^d} \, . \tag{1.26}$$

1. Capacity-per-unit volume Constraint.

   A general and simple way to model the network capability to collect data and forward them to the interpolation entity is to assume the WSN can guarantee a certain capacity-per-volume c. Such a value is a function of network bandwidth and protocols (MAC, routing, physical layer, etc.). For homogeneous Poisson distribution of the sensors positions, the capacity of the channel between the n-th sensor and the interpolator for large WSN can be approximated as

   $$C_n = \frac{C}{|\Pi_{\mathcal{A}}|} \approx \frac{c|\mathcal{A}|}{\mathbb{E}\{|\Pi_{\mathcal{A}}|\}} = \frac{c}{\lambda} \tag{1.27}$$

   where C is the overall capacity in $\left[symbols/channeluse\right]$ and $|\cdot|$ denotes the cardinality. If the channel between the n-th sensor and the interpolator is modeled as an erasure channel, it results

   $$C_n = 1 - \epsilon \tag{1.28}$$

   where ϵ is the erasure probability, which is considered equal for all sensors. From (1.27) and (1.28) it follows that the sample availability can be written as

   $$q \triangleq 1 - \epsilon = C_n \approx \frac{c}{\lambda} \, . \tag{1.29}$$

   Since $0 \leq C \leq 1$ and $|\Pi_{\mathcal{A}}| \geq 1$ (for each PPP realization, a WSN is obviously constituted by at least one sensor), it is $0 \leq q \leq 1$.

2. Sensor Lifetime Constraint.

   Consider all sensors have the same power consumption W_{s} such that the measurement error variance results in

$$\sigma_{\mathrm{M}}^2 = \frac{\kappa_{\mathrm{M}}}{W_{\mathrm{s}} T}, \quad \forall n \tag{1.30}$$

   where T is the estimation process duration in seconds. By substituting (1.30) in (1.5) we obtain $D = \rho |\mathcal{A}| \frac{\kappa_{\mathrm{M}}}{W_{\mathrm{s}} T}$, thus

$$\frac{\breve{D}}{\breve{\rho}} = \frac{\kappa_{\mathrm{M}} |\mathcal{A}|}{E_z W_{\mathrm{s}} T} \,. \tag{1.31}$$

3. NEMSE for a large WSN

   By substituting (1.29) in (1.26) we obtain

$$\breve{\rho} \approx \frac{c}{(2B_z)^d} \tag{1.32}$$

   that highlights how in a large WSN, the effectively available samples intensity depends on the capacity-per-volume (and no longer on the sensors intensity). From (1.15), (1.31), and (1.32), we obtain

$$\epsilon_{\mathrm{S}} = 1 - \int_{\mathbb{R}^d} \frac{\breve{\mathcal{E}}_z^2(\breve{\boldsymbol{\nu}})}{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) + \frac{1}{\breve{c}}\left(1 + \frac{1}{\breve{T}}\right)} d\breve{\boldsymbol{\nu}} \tag{1.33}$$

   It can be observed that (1.33) depends on two WSN parameters, the capacity per unit volume normalized to the signal band

$$\breve{c} \triangleq \frac{c}{(2B_z)^d} \tag{1.34}$$

   and the normalized battery lifetime

$$\breve{T} \triangleq \frac{E_z W_{\mathrm{s}} T}{\kappa_{\mathrm{M}} |\mathcal{A}|}. \tag{1.35}$$

Remark 7 *If the signal is rigorously band-limited, from* (1.33) *and for large $\breve{c}$ it is*

$$\epsilon_S = \frac{1}{\breve{c}}\left(1 + \frac{1}{\breve{T}}\right) + o\left(\frac{1}{\breve{c}}\right) \,. \tag{1.36}$$

Remark 8 *I remark that* (1.36) *holds for signals which are band-limited in strict sense only. It is easy to notice that it does not holds for finite-energy signals with infinite bandwidth such as, e.g., that of the example in Sec. 4.*

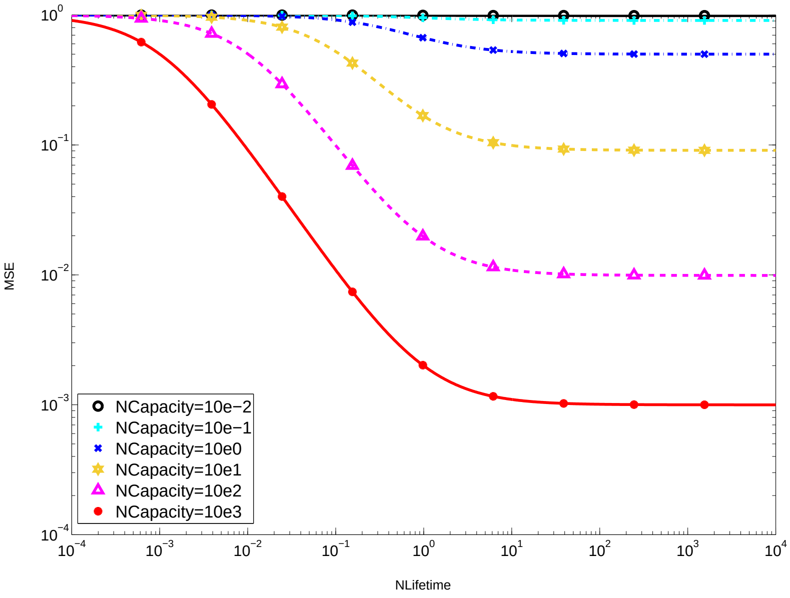


Figure 1.4: Large WSN for signal estimation: MSE as a function of normalized capacity and lifetime

Example 5 *In the case of Bessel-type autocorrelation function for the signal, it results* $\epsilon_S = \frac{1}{\breve{c}}\left(1+\frac{1}{\breve{T}}\right)/\left[1+\frac{1}{\breve{c}}\left(1+\frac{1}{\breve{T}}\right)\right]$. *Fig. 1.4 shows the NEMSE as a function of the normalized estimation time for different values of normalized capacity. It can be seen the capacity-limited region for large* $\breve{T}$*, corresponding to the horizontal asymptotes, and the energy-limited region for low* $\breve{T}$.

Remark 9 *For infinite normalized estimation time, the asymptotical MSE is given by*

$$\epsilon_S^{(\infty)} \triangleq \lim_{\breve{T}\to+\infty} \epsilon_S = 1 - \int_{\mathbb{R}^d} \frac{\breve{\mathcal{E}}_z^2(\breve{\boldsymbol{\nu}})}{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) + \frac{1}{\breve{c}}} d\breve{\boldsymbol{\nu}}\ .$$

4. Asymptotic MSE for Infinite Battery Duration

   For $\frac{E_z T}{\kappa_{\mathrm{M}}|\mathcal{A}|} \longrightarrow +\infty$ it results

$$\lim_{\frac{E_z T}{\kappa_{\mathrm{M}}|\mathcal{A}|}\to+\infty} \epsilon_{\mathrm{S}} = 1 - \int_{\mathbb{R}^d} \frac{\breve{\mathcal{E}}_z^2(\breve{\boldsymbol{\nu}})}{\breve{\mathcal{E}}_z(\breve{\boldsymbol{\nu}}) + \frac{(2B_z)^d}{c}} d\breve{\boldsymbol{\nu}}\ . \tag{1.37}$$

5. Asymptotic Reconstruction MSE for Band-limited Signals

   For $\frac{1}{\breve{c}} \longrightarrow 0$, the asymptotic NEMSE (1.37) for a band-limited signal can be written as

$$\lim_{\frac{1}{\breve{c}}\to 0} \epsilon_{\mathrm{S}}^{(\infty)} = \frac{1}{\breve{c}} + o\left(\frac{1}{\breve{c}}\right)\ . \tag{1.38}$$

1.4 In the presence of noise

As intuitive, the accuracy of the field estimate is affected by the possible inaccuracies of the punctual estimates used for its derivation. Moreover, the errors affecting the samples provided by sensor nodes could significantly compromise the accuracy of the whole field estimate. It follows that the availability of an accuracy index of the punctual estimate provided by each node could help the reconstruction: In fact, the interpolation procedure could take into account the possibly different reliabilities of the punctual estimates in order to get a more accurate field estimate, e.g., by giving more importance to the more accurate points. Motivated by this consideration, I suggested and analytically investigated a strategy aimed at minimizing the NEMSE by properly selecting the punctual estimate used for the field reconstruction. This investigation is carried out in the general case of sensor nodes irregularly distributed in the spatial domain. Here, in particular, I assume that sensor nodes are deployed in the scenario of interest according to the well known and widely adopted homogeneous PPP [20–23]. In such a scenario both the interpolation task and the analytical investigation are more complicated than in the case of regular sampling. Yet, this is the most likely, hence realistic, scenario for an actual WSN.

Consider the deterministic spatial field $z(\mathbf{x}) \in \mathbb{C}$ with support $\mathcal{A} \subseteq \mathbb{R}^d$ and finite energy E_z. The ESD of $z(\mathbf{x})$ is given by $\mathcal{E}_z(\boldsymbol{\nu}) = \mathcal{F}\left\{\int_{\mathbb{R}^d} z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})d\mathbf{x}\right\}(\boldsymbol{\nu})$, where $\mathcal{F}\{\cdot\}$ denotes the d-dimensional Fourier transform [30]. I considered a poisson sampling process $\mathcal{P}(\mathbf{x})$ given by

$$\mathcal{P}(\mathbf{x}) = \sum_{n\in\mathcal{N}(\Pi)} \delta(\mathbf{x} - \mathsf{x}_n) \tag{1.39}$$

where $\mathcal{N}(\Pi)$ is the index set of the homogeneous PPP Π with intensity λ_0 and x_n is the nth sampling point. The mean and the autocorrelation of $\mathcal{P}(\mathbf{x})$ are respectively given by [32]

$$\mathbb{E}\{\mathcal{P}(\mathbf{x})\} = \lambda_0 \tag{1.40a}$$

$$\mathbb{E}\{\mathcal{P}(\mathbf{x})\mathcal{P}(\mathbf{x}-\boldsymbol{\tau})\} = \lambda_0^2 + \lambda_0\delta(\boldsymbol{\tau})\ . \tag{1.40b}$$

Assume that $z(\mathbf{x})$ is sampled in $\mathcal{A}$ according to the sampling process

$$\mathcal{P}_{\mathcal{A}}(\mathbf{x}) \triangleq \mathcal{P}(\mathbf{x})\mathbf{1}_{\mathcal{A}}(\mathbf{x}) = \sum_{n\in\mathcal{N}(\Pi\cap\mathcal{A})} \delta(\mathbf{x} - \mathsf{x}_n) \tag{1.41}$$

where $\mathbf{1}_{\mathcal{A}}(\mathbf{x})$ denotes the indicator function that is 1 for $\mathbf{x} \in \mathcal{A}$ and 0 elsewhere. The value of $z(\mathbf{x})$ at each sampling position $\mathsf{x}_n \in \Pi \cap \mathcal{A}$ is known to the estimator as

$$\hat{\mathsf{z}}_n = z(\mathsf{x}_n) + \mathsf{e}_n \tag{1.42}$$

where $\{\hat{\mathsf{z}}_n\}$ denote the punctual estimates of $\{z(\mathsf{x}_n)\}$ and the punctual estimation errors $\{\mathsf{e}_n\}$ constitute a set of zero mean independent random variables with variances $\{\sigma_n^2\}$. Consider the n-th measurement error variance σ_n^2, this variances of the punctual estimation errors are considered as a set of independent identically distributed (IID) RVs with probability density function (PDF) $f_{\sigma^2}(\cdot)$. Assume that, for each punctual estimate $\hat{\mathsf{z}}_n$, the estimator knows also the corresponding estimation error variance σ_n^2, which quantifies its reliability. In this case the field reconstruction can be carried out discarding punctual estimates with poor accuracy, that is, such that the corresponding variance of the estimation error exceeds a given value σ_{th}^2. Formally, the set of sampling points after such a selection is given by

$$\Pi_{\mathcal{A}} \triangleq \{\mathsf{x}_n \in \Pi \cap \mathcal{A} \ : \ \mathbb{E}\left\{|\hat{\mathsf{z}}_n - z(\mathsf{x}_n)|^2\right\} \leq \sigma_{\text{th}}^2\} \tag{1.43}$$

The sampling process resulting from (1.43) is

$$\mathcal{P}_{\mathcal{A}}^{\star}(\mathbf{x}) \triangleq \sum_{n \in \mathcal{N}(\Pi_{\mathcal{A}})} \delta(\mathbf{x} - \mathsf{x}_n) = \sum_{n \in \mathcal{N}(\Pi \cap \mathcal{A})} \mathsf{a}_n \delta(\mathbf{x} - \mathsf{x}_n) \tag{1.44}$$

where each a_n is defined as a binary RV that is 0 if $\sigma_n^2 > \sigma_{\text{th}}^2$ and 1 if $\sigma_n^2 \leq \sigma_{\text{th}}^2$. Since $\{\sigma_n^2\}$ are IID RVs, the same is true also for $\{\mathsf{a}_n\}$. It is, moreover, $\text{Prob}\{\mathsf{a}_n = 1\} = q(\sigma_{\text{th}}^2)$ and $\text{Prob}\{\mathsf{a}_n = 0\} = 1 - q(\sigma_{\text{th}}^2)$, regardless of n, where

$$q(\sigma_{\text{th}}^2) = F_{\sigma^2}(\sigma_{\text{th}}^2) = \int_0^{\sigma_{\text{th}}^2} f_{\sigma^2}(x)dx \tag{1.45}$$

with $F_{\sigma^2}(\cdot)$ denoting the cumulative distribution function (CDF) of σ_n^2. The process $\Pi^\star$ resulting from Π by retaining only those sampling points for which $\sigma_n^2 \leq \sigma_{\text{th}}^2$ is still a homogeneous PPP, with intensity $q\lambda_0$. Thus, the sampling process defined as

$$\mathcal{P}^{\star}(\mathbf{x}) \triangleq \sum_{n \in \mathcal{N}(\Pi^\star)} \delta(\mathbf{x} - \mathsf{x}_n) \tag{1.46}$$

results in a homogeneous Poisson sampling process (PSP) with stochastic mean and autocorrelation function respectively given by (1.40a) and (1.40b) with $q(\sigma_{\text{th}}^2)\lambda_0$ in place of λ_0. Note that (1.44) can be re-written as

$$\mathcal{P}_{\mathcal{A}}^{\star}(\mathbf{x}) = \mathbf{1}_{\mathcal{A}}(\mathbf{x}) \sum_{n \in \mathcal{N}(\Pi^\star)} \delta(\mathbf{x} - \mathsf{x}_n) = \mathbf{1}_{\mathcal{A}}(\mathbf{x})\mathcal{P}^{\star}(\mathbf{x}) \tag{1.47}$$

The sampled estimated spatial field known to the interpolator can be written as

$$\mathsf{z}_\epsilon(\mathbf{x}) = \sum_{n \in \mathcal{N}(\Pi_{\mathcal{A}})} \hat{\mathsf{z}}_n \delta(\mathbf{x} - \mathsf{x}_n) = \mathsf{z}_{\mathcal{P}^\star}(\mathbf{x}) + \mathsf{e}_{\mathcal{P}^\star}(\mathbf{x}) \tag{1.48}$$

where from (1.44), (1.46), and (1.47)

$$\mathsf{z}_{\mathcal{P}^\star}(\mathbf{x}) \triangleq \sum_{n \in \mathcal{N}(\Pi_{\mathcal{A}})} z(\mathsf{x}_n)\delta(\mathbf{x} - \mathsf{x}_n) = z(\mathbf{x})\mathcal{P}^{\star}(\mathbf{x}) \tag{1.49}$$

and

$$\mathsf{e}_{\mathcal{P}^{\star}}(\mathbf{x}) \triangleq \sum_{n\in\mathcal{N}(\Pi_{\mathcal{A}})} \mathsf{e}_n|_{\sigma_n^2\leq\sigma_{\text{th}}^2}\delta(\mathbf{x}-\mathsf{x}_n)\,. \tag{1.50}$$

From the Bayes theorem, the CDF of $\sigma_n^2|_{\sigma_n^2\leq\sigma_{\text{th}}^2}$ results in

$$\begin{aligned} F_{\sigma_n^2|_{\sigma_n^2\leq\sigma_{\text{th}}^2}}(x) &= \text{Prob}\{\sigma_n^2 \leq x|\sigma_n^2 \leq \sigma_{\text{th}}^2\} \\ &= \frac{\text{Prob}\{\sigma_n^2 \leq x, \sigma_n^2 \leq \sigma_{\text{th}}^2\}}{\text{Prob}\{\sigma_n^2 \leq \sigma_{\text{th}}^2\}} \\ &= \frac{u(\sigma_{\text{th}}^2 - x)F_{\sigma^2}(x) + u(x-\sigma_{\text{th}}^2)q}{q(\sigma_{\text{th}}^2)} \end{aligned} \tag{1.51}$$

where $u(x)$ is the Heaviside step function equal to 0 if $x < 0$ and 1 if $x \geq 0$. By deriving (1.51) in the sense of distributions, we obtain that the variance of $\mathsf{e}_n|_{\sigma_n^2\leq\sigma_{\text{th}}^2}$ is distributed as

$$f_{\sigma^2|_{\sigma^2\leq\sigma_{\text{th}}^2}}(x) = \frac{u(\sigma_{\text{th}}^2 - x)f_{\sigma^2}(x)}{q(\sigma_{\text{th}}^2)}\,. \tag{1.52}$$

1.4.1 Signal Reconstruction

The signal estimated via LSI filtering is given by

$$\hat{\mathsf{z}}(\mathbf{x}) \triangleq (\mathsf{z}_\epsilon * \theta)(\mathbf{x}) = \sum_{n\in\mathcal{N}(\Pi_{\mathcal{A}})} \hat{\mathsf{z}}_n\theta(\mathbf{x}-\mathsf{x}_n) \tag{1.53}$$

where $\theta(\mathbf{x}) : \mathbb{R}^d \longrightarrow \mathbb{R}$ is the interpolating function. This means that the ESD acts as a Kernel for the signal, i.e., Kernel-based interpolation where the Kernel is optimized to the signal structure. The estimation accuracy is evaluated in terms of the NEMSE defined as [12, 14, 16, 17]

$$\epsilon_{\text{S}} \triangleq \frac{\mathbb{E}\left\{\int_{\mathbb{R}^d} |\hat{\mathsf{z}}(\mathbf{x}) - z(\mathbf{x})|^2 d\mathbf{x}\right\}}{E_z}\,. \tag{1.54}$$

Define $R_{z,\mathsf{z}_\epsilon}(\boldsymbol{\tau}) \triangleq \int_{\mathbb{R}^d} \mathbb{E}\left\{z(\mathbf{x})\mathsf{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} d\mathbf{x}$ and $R_{\mathsf{z}_\epsilon}(\boldsymbol{\tau}) \triangleq \int_{\mathbb{R}^d} \mathbb{E}\left\{\mathsf{z}_\epsilon(\mathbf{x})\mathsf{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} d\mathbf{x}$. It can be shown that the optimal LSI interpolator transfer function $\Theta(\boldsymbol{\nu}) \triangleq \mathcal{F}\left\{\theta(\mathbf{x})\right\}(\boldsymbol{\nu})$ results in

$$\Theta(\boldsymbol{\nu}) = \frac{\mathcal{E}_{z,\mathsf{z}_\epsilon}(\boldsymbol{\nu})}{\mathcal{E}_{\mathsf{z}_\epsilon}(\boldsymbol{\nu})} \tag{1.55}$$

with the corresponding NEMSE given by

$$\epsilon_{\text{S}} = 1 - \frac{1}{E_z}\int_{\mathbb{R}^d} \frac{|\mathcal{E}_{z,\mathsf{z}_\epsilon}(\boldsymbol{\nu})|^2}{\mathcal{E}_{\mathsf{z}_\epsilon}^\dagger(\boldsymbol{\nu})} d\boldsymbol{\nu} \tag{1.56}$$

where $\mathcal{E}_{z,\mathrm{z}_\epsilon}(\boldsymbol{\nu}) \triangleq \mathcal{F}\left\{R_{z,\mathrm{z}_\epsilon}(\boldsymbol{\tau})\right\}$ and $\mathcal{E}_{\mathrm{z}_\epsilon}(\boldsymbol{\nu}) \triangleq \mathcal{F}\left\{R_{\mathrm{z}_\epsilon}(\boldsymbol{\tau})\right\}$. From (1.48), (1.49), and (1.50), by exploiting the independence between $\mathrm{e}_n|_{\sigma_n^2 \leq \sigma_{\mathrm{th}}^2}$ and $\Pi_{\mathcal{A}}$, and the zero mean of $\mathrm{e}_n|_{\sigma_n^2 \leq \sigma_{\mathrm{th}}^2}$, we obtain

$$\begin{aligned}\mathbb{E}\left\{\mathrm{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} &= \mathbb{E}\left\{\mathrm{z}_{\mathcal{P}^\star}^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} = \lambda_{\mathrm{th}} z^\dagger(\mathbf{x}-\boldsymbol{\tau}) \\ \mathbb{E}\left\{\mathrm{z}_\epsilon(\mathbf{x})\mathrm{z}_\epsilon^\dagger(\mathbf{x}-\boldsymbol{\tau})\right\} &= z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})[\lambda_{\mathrm{th}}^2 + \lambda_{\mathrm{th}}\delta(\boldsymbol{\tau})] \\ &+\mathbb{E}\left\{\sum_{n\in\mathcal{N}(\Pi_{\mathcal{A}})} \sigma_n^2|_{\sigma_n^2\leq\sigma_{\mathrm{th}}^2}\delta(\mathbf{x}-\mathrm{x}_n)\delta(\mathbf{x}-\boldsymbol{\tau}-\mathrm{x}_n)\right\}\end{aligned}$$

where, for what concerns $\mathcal{P}^\star(\mathrm{x})$, $\lambda_{\mathrm{th}} \triangleq q(\sigma_{\mathrm{th}}^2)\lambda_0$ has taken the place of λ_0 in (1.40). Thus

$$R_{z,\mathrm{z}_\epsilon}(\boldsymbol{\tau}) = \lambda_{\mathrm{th}} \int_{\mathbb{R}^d} z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})d\mathbf{x} \tag{1.57a}$$

$$R_{\mathrm{z}_\epsilon}(\boldsymbol{\tau}) = [\lambda_{\mathrm{th}}^2 + \lambda_{\mathrm{th}}\delta(\boldsymbol{\tau})] \int_{\mathbb{R}^d} z(\mathbf{x})z^\dagger(\mathbf{x}-\boldsymbol{\tau})d\mathbf{x} + D\delta(\boldsymbol{\tau}) \tag{1.57b}$$

where the term

$$D \triangleq \mathbb{E}\left\{\sum_{n\in\mathcal{N}(\Pi_{\mathcal{A}})} \sigma_n^2|_{\sigma_n^2\leq\sigma_{\mathrm{th}}^2}\right\} \tag{1.58}$$

represents the contribution caused by (punctual) estimation errors. The Fourier transform of (1.57) in $\mathbb{R}^d$ leads to

$$\mathcal{E}_{z,\mathrm{z}_\epsilon}(\boldsymbol{\nu}) = \lambda_{\mathrm{th}}\mathcal{E}_z(\boldsymbol{\nu}) \tag{1.59a}$$

$$\mathcal{E}_{\mathrm{z}_\epsilon}(\boldsymbol{\nu}) = \lambda_{\mathrm{th}}^2\mathcal{E}_z(\boldsymbol{\nu}) + E_z\left(\lambda_{\mathrm{th}} + \frac{D}{E_z}\right) \tag{1.59b}$$

that, substituted in (1.55) and (1.56), provide respectively

$$\Theta(\boldsymbol{\nu}) = \frac{\mathcal{E}_z(\boldsymbol{\nu})}{\lambda_{\mathrm{th}}\mathcal{E}_z(\boldsymbol{\nu}) + E_z\left(1 + \frac{D}{\lambda_{\mathrm{th}}E_z}\right)} \tag{1.60}$$

and

$$\epsilon_{\mathrm{S}} = 1 - \frac{1}{E_z}\int_{\mathbb{R}^d} \frac{|\mathcal{E}_z(\boldsymbol{\nu})|^2}{\mathcal{E}_z(\boldsymbol{\nu}) + \frac{E_z}{\lambda_{\mathrm{th}}}\left(1 + \frac{D}{\lambda_{\mathrm{th}}E_z}\right)}d\boldsymbol{\nu}\,. \tag{1.61}$$

Eq. (1.60) shows that the optimal LSI interpolation requires the knowledge of both λ_{th} and D. Eq. (1.61) clearly highlights, instead, the dependence of the normalized estimation mean square error on both the distortion introduced by punctual estimation errors, through D, and the intensity of the resulting (after removing unreliable punctual estimates) Poissonian random sampling, through λ_{th}. As expected, in the limit conditions $D \longrightarrow 0$ and $\lambda_{\mathrm{th}} \longrightarrow +\infty$ it results $\epsilon_{\mathrm{S}} \longrightarrow 0$. In the following I provide a closed form expression of D, then I derive the optimal threshold σ^2_{th}, that leads to the minimization of ϵ_{S}. From (1.52) we have

$$\begin{aligned}\mathbb{E}\left\{\sigma^2_n|_{\sigma^2_n \le \sigma^2_{\text{th}}}\right\} &= \int_0^{+\infty} \frac{u(\sigma^2_{\text{th}} - x) f_{\sigma^2}(x)}{q(\sigma^2_{\text{th}})} x dx \\ &= \frac{1}{q(\sigma^2_{\text{th}})} \int_0^{\sigma^2_{\text{th}}} x f_{\sigma^2}(x) dx = \frac{\mu(\sigma^2_{\text{th}})}{q(\sigma^2_{\text{th}})} \qquad (1.62)\end{aligned}$$

where $\mu(x) \triangleq \int_0^x \xi f_{\sigma^2}(\xi) d\xi$, independent of n. Thus, due to the stochastic independence between $\sigma^2_n|_{\sigma^2_n \le \sigma^2_{\text{th}}}$ and $\Pi_{\mathcal{A}}$, (1.58) results in

$$\begin{aligned}D &= \mathbb{E}\left\{\sum_{n \in \mathcal{N}(\Pi_{\mathcal{A}})} \frac{\mu(\sigma^2_{\text{th}})}{q(\sigma^2_{\text{th}})}\right\} = \frac{\mu(\sigma^2_{\text{th}})}{q(\sigma^2_{\text{th}})} \mathbb{E}\left\{|\mathcal{N}(\Pi_{\mathcal{A}})|\right\} \\ &= \frac{\mu(\sigma^2_{\text{th}})}{q(\sigma^2_{\text{th}})} \lambda_{\text{th}} |\mathcal{A}| \qquad (1.63)\end{aligned}$$

where the last equation follows from the PPP properties [31]. By substituting (1.63) in (1.61) we get

$$\epsilon_{\text{S}} = 1 - \frac{1}{E_z} \int_{\mathbb{R}^d} \frac{|\mathcal{E}_z(\boldsymbol{\nu})|^2}{\mathcal{E}_z(\boldsymbol{\nu}) + \frac{E_z}{\lambda_{\text{eq}}(\sigma^2_{\text{th}})}} d\boldsymbol{\nu}\,. \qquad (1.64)$$

where

$$\lambda_{\text{eq}}(\sigma^2_{\text{th}}) \triangleq \lambda_0 \; q(\sigma^2_{\text{th}}) \left[1 + \frac{\mu(\sigma^2_{\text{th}})|\mathcal{A}|}{q(\sigma^2_{\text{th}}) E_z}\right]^{-1} \qquad (1.65)$$

can be considered as an equivalent sampling intensity. Thus, to minimize the NEMSE we need to minimize the scaling factor

$$\frac{1}{q(\sigma^2_{\text{th}})} + \frac{|\mathcal{A}|}{E_z} \frac{\mu(\sigma^2_{\text{th}})}{q^2(\sigma^2_{\text{th}})}\,. \qquad (1.66)$$

By deriving (1.66) with respect to σ^2_{th} we obtain

$$-\frac{q'(\sigma^2_{\text{th}})}{q^2(\sigma^2_{\text{th}})} + \frac{|\mathcal{A}|}{E_z}\left[\frac{\mu'(\sigma^2_{\text{th}})}{q^2(\sigma^2_{\text{th}})} - 2\frac{q'(\sigma^2_{\text{th}})}{q^3(\sigma^2_{\text{th}})} \mu(\sigma^2_{\text{th}})\right] \qquad (1.67)$$

where

$$q'(\sigma^2_{\text{th}}) \triangleq \frac{d\; q(\sigma^2_{\text{th}})}{d\sigma^2_{\text{th}}} = f_{\sigma^2}(\sigma^2_{\text{th}}) \qquad (1.68a)$$

$$\mu'(\sigma^2_{\text{th}}) \triangleq \frac{d\; \mu(\sigma^2_{\text{th}})}{d\sigma^2_{\text{th}}} = \sigma^2_{\text{th}}\; f_{\sigma^2}(\sigma^2_{\text{th}})\,. \qquad (1.68b)$$

The derivative (1.67) is zero iff

$$\mu'(\sigma^2_{\text{th}}) - 2\frac{q'(\sigma^2_{\text{th}})}{q(\sigma^2_{\text{th}})}\mu(\sigma^2_{\text{th}}) = \frac{E_z}{|\mathcal{A}|}q'(\sigma^2_{\text{th}})\,. \tag{1.69}$$

By (1.68), condition (1.69) results to be equivalent (for $f_{\sigma^2}(\sigma^2_{\text{th}}) > 0$) to

$$\sigma^2_{\text{th}} - \frac{E_z}{|\mathcal{A}|} = 2\frac{\mu(\sigma^2_{\text{th}})}{q(\sigma^2_{\text{th}})}\,. \tag{1.70}$$

Thus, the optimal threshold is implicitly given by

$$\sigma^2_{\text{th}} = \frac{E_z}{|\mathcal{A}|} + 2\frac{\int_0^{\sigma^2_{\text{th}}} x f_{\sigma^2}(x)dx}{\int_0^{\sigma^2_{\text{th}}} f_{\sigma^2}(x)dx}\,. \tag{1.71}$$

1.4.2 Case Study: Example of Measurement Errors Variance Statistics

As far as the statistics of the punctual estimation error variance is concerned, I consider two significant cases of PDF, the one bounded and the other unbounded.

1. Chi-Squared Distribution

   Assume that the punctual estimate provided by each n-th sensor is derived as the mean of K measurements (the spatial field to be estimated is assumed time-invariant for the whole measurement interval)

   $$\hat{\mathsf{z}}_n = \frac{1}{K}\sum_{k=1}^{K} \mathsf{y}_{nk} \tag{1.72}$$

   where each measurement y_{nk} is affected by an additive white Gaussian noise with power σ^2_{N} (that is assumed to be independent of the sensor position). It is therefore

   $$\mathsf{y}_{nk} = z(\mathsf{x}_n) + \mathsf{e}_{n,k} \tag{1.73}$$

   where $\{\mathsf{e}_{n,k}\}$ are IID zero-mean Gaussian RVs with variance σ^2_{N}, for all n. By substituting (1.73) in (1.72) we obtain (1.42) with $\mathsf{e}_n \sim \mathcal{N}\left(0, \frac{\sigma^2_{\text{N}}}{K}\right)$, that is, $\sigma^2_n = \sigma^2 = \sigma^2_{\text{N}}/K$. For K sufficiently large, each sensor can also get and estimate the variance of e_n as

   $$\mathbb{E}\left\{|\mathsf{e}_n|^2\right\} \approx \frac{1}{K}\sum_{k=1}^{K} |\mathsf{e}_{n,k}|^2 \approx \frac{1}{K}\sum_{k=1}^{K} |\mathsf{y}_{nk} - \hat{\mathsf{z}}_n|^2\,. \tag{1.74}$$

   This estimate is also provided to the interpolator as an accuracy index of $\hat{\mathsf{z}}_n$. Define the normalized errors

   $$\breve{\mathsf{e}}_{n,k} \triangleq \frac{\mathsf{e}_{n,k}}{\sigma_{\text{N}}} \tag{1.75}$$

that result in zero mean Gaussian RVs with unitary variance for all n. Thus the variable

$$\mathsf{s}_n \triangleq \sum_{k=1}^{K} |\breve{\mathsf{e}}_{n,k}|^2 \tag{1.76}$$

is distributed as a central chi-square of order K for all n

$$\mathsf{s}_n \sim f_{\chi^2(K)}(x) = \frac{1}{2^{\frac{K}{2}}\Gamma(\frac{K}{2})} x^{\frac{K}{2}-1} e^{-\frac{x}{2}} \tag{1.77}$$

From (1.74) we have

$$\sigma_n^2 = \mathbb{E}\left\{|\mathsf{e}_n|^2\right\} \approx \frac{\sigma_{\mathrm{N}}^2}{K} \sum_{k=1}^{K} |\breve{\mathsf{e}}_{n,k}|^2 = \frac{\sigma_{\mathrm{N}}^2}{K}\mathsf{s}_n \tag{1.78}$$

that, by (1.77), is distributed as

$$\begin{aligned} f_{\sigma^2}(x) &= \frac{K}{\sigma_{\mathrm{N}}^2} f_{\chi^2(K)}\left(\frac{K\,x}{\sigma_{\mathrm{N}}^2}\right) \\ &= \frac{K}{\sigma_{\mathrm{N}}^2\, 2^{\frac{K}{2}}\Gamma\left(\frac{K}{2}\right)} \left(\frac{K\,x}{\sigma_{\mathrm{N}}^2}\right)^{\frac{K}{2}-1} e^{-\frac{K\,x}{2\sigma_{\mathrm{N}}^2}} \end{aligned} \tag{1.79}$$

for all n where $\Gamma(x)$ denotes the Euler Gamma function. Thus we get

$$\begin{aligned} q(\sigma_{\mathrm{th}}^2) &= \int_0^{\sigma_{\mathrm{th}}^2} \frac{K}{\sigma_{\mathrm{N}}^2} f_{\chi^2(K)}\left(\frac{K\,x}{\sigma_{\mathrm{N}}^2}\right) dx \\ &= \int_0^{\frac{K\sigma_{\mathrm{th}}^2}{\sigma_{\mathrm{N}}^2}} f_{\chi^2(K)}(\breve{\mathrm{x}}) d\breve{\mathrm{x}} = F_{\chi^2(K)}\left(\frac{K\,\sigma_{\mathrm{th}}^2}{\sigma_{\mathrm{N}}^2}\right) \\ &= \frac{\gamma\left(\frac{K}{2}, \frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_{\mathrm{N}}^2}\right)}{\Gamma\left(\frac{K}{2}\right)} \end{aligned} \tag{1.80}$$

and

$$\begin{aligned} \mu(\sigma_{\mathrm{th}}^2) &= \int_0^{\sigma_{\mathrm{th}}^2} \frac{K}{\sigma_{\mathrm{N}}^2} f_{\chi^2(K)}\left(\frac{K\,x}{\sigma_{\mathrm{N}}^2}\right) x dx \\ &= \frac{\sigma_{\mathrm{N}}^2}{K} \int_0^{\frac{K\sigma_{\mathrm{th}}^2}{\sigma_{\mathrm{N}}^2}} \breve{\mathrm{x}}\; f_{\chi^2(K)}(\breve{\mathrm{x}}) d\breve{\mathrm{x}} \\ &= \frac{\sigma_{\mathrm{N}}^2}{K} \frac{1}{2^{\frac{K}{2}}\Gamma(\frac{K}{2})} \int_0^{\frac{K\sigma_{\mathrm{th}}^2}{\sigma_{\mathrm{N}}^2}} \breve{\mathrm{x}}^{\frac{K}{2}} e^{-\frac{\breve{\mathrm{x}}}{2}} d\breve{\mathrm{x}} \\ &= \frac{2\sigma_{\mathrm{N}}^2}{K} \frac{\gamma\left(\frac{K}{2}+1, \frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_{\mathrm{N}}^2}\right)}{\Gamma\left(\frac{K}{2}\right)}. \end{aligned} \tag{1.81}$$

where $\gamma(s,x) \triangleq \int_0^x t^{s-1}e^{-t}dt$. Thus, the equivalent sampling intensity results from (1.65) in

$$\lambda_{\mathrm{eq}}(\sigma_{\mathrm{th}}^2) = \lambda_0 \frac{\left[\frac{\gamma\left(\frac{K}{2},\frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_{\mathrm{N}}^2}\right)}{\Gamma\left(\frac{K}{2}\right)}\right]}{\left[1+\frac{2\sigma_{\mathrm{N}}^2}{K}\frac{|\mathcal{A}|}{E_z}\frac{\gamma\left(\frac{K}{2}+1,\frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_{\mathrm{N}}^2}\right)}{\gamma\left(\frac{K}{2},\frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_{\mathrm{N}}^2}\right)}\right]}\,. \tag{1.82}$$

By exploiting that $\frac{\gamma(s+1,x)}{\gamma(s,x)} = s - \frac{x^s e^{-x}}{\gamma(s,x)}$... By defining the normalized quantities $\breve{\sigma}_{\mathrm{th}}^2 \triangleq \sigma_{\mathrm{th}}^2/\sigma_{\mathrm{N}}^2$, $\breve{\lambda}_{\mathrm{eq}}(\breve{\sigma}_{\mathrm{th}}^2) \triangleq \lambda_{\mathrm{eq}}(\sigma_{\mathrm{N}}^2\breve{\sigma}_{\mathrm{th}}^2)/\lambda_0$ and the single measurement SNR

$$\rho_{\mathrm{M}} \triangleq \frac{E_z}{|\mathcal{A}|\sigma_{\mathrm{N}}^2} \tag{1.83}$$

the (1.82) becomes

$$\breve{\lambda}_{\mathrm{eq}}(\breve{\sigma}_{\mathrm{th}}^2) = \frac{\left[\frac{\gamma\left(\frac{K}{2},\frac{K}{2}\breve{\sigma}_{\mathrm{th}}^2\right)}{\Gamma\left(\frac{K}{2}\right)}\right]}{\left[1+\frac{1}{\rho_{\mathrm{M}}}\frac{2}{K}\frac{\gamma\left(\frac{K}{2}+1,\frac{K}{2}\breve{\sigma}_{\mathrm{th}}^2\right)}{\gamma\left(\frac{K}{2},\frac{K}{2}\breve{\sigma}_{\mathrm{th}}^2\right)}\right]}\,. \tag{1.84}$$

It is immediate that, as expected

$$\lim_{\breve{\sigma}_{\mathrm{th}}^2\to\infty} \breve{\lambda}_{\mathrm{eq}}(\breve{\sigma}_{\mathrm{th}}^2) = \frac{1}{1+\frac{1}{\rho_{\mathrm{M}}}} \tag{1.85}$$

that is consistent with the result in [1] once the variance σ_n^2 reduces to the deterministic value $\mathbb{E}\left\{\sigma_n^2\right\} = \sigma_{\mathrm{N}}^2$ (according to (1.78)).

2. Uniform Distribution

   Consider

$$f_{\sigma^2}(x) = \frac{1}{2\overline{\sigma}^2}\mathrm{rect}\left(\frac{x-\overline{\sigma}^2}{2\overline{\sigma}^2}\right) \tag{1.86}$$

where $\mathrm{rect}(x)$ denotes the rectangular function that is equal to 1 for $|x| \leq 1/2$ and to 0 elsewhere. By (1.86) we get

$$q(\sigma_{\mathrm{th}}^2) = \frac{\sigma_{\mathrm{th}}^2}{2\overline{\sigma}^2}u(2\overline{\sigma}^2-\sigma_{\mathrm{th}}^2)+u(\sigma_{\mathrm{th}}^2-2\overline{\sigma}^2) \tag{1.87a}$$

$$\mu(\sigma_{\mathrm{th}}^2) = \frac{\sigma_{\mathrm{th}}^4}{4\overline{\sigma}^2}u(2\overline{\sigma}^2-\sigma_{\mathrm{th}}^2)+\overline{\sigma}^2u(\sigma_{\mathrm{th}}^2-2\overline{\sigma}^2)\,. \tag{1.87b}$$

Thus, for $0 \leq \sigma_{\mathrm{th}}^2 \leq 2\overline{\sigma}^2$, the equivalent sampling intensity resulting from (1.65) is

$$\lambda_{\mathrm{eq}}(\sigma_{\mathrm{th}}^2) = \frac{\lambda_0}{\overline{\sigma}^2}\left(\frac{2}{\sigma_{\mathrm{th}}^2}+\frac{|\mathcal{A}|}{E_z}\right)^{-1} \tag{1.88}$$

that is always increasing with σ_{th}^2.

3. Sensor Defaillance Model

Consider a probability p_ϵ of having a measurement problem. Thus

$$f_{\sigma^2}(x) = (1-p_\epsilon)\frac{K}{\sigma_{\mathrm{N}}^2} f_{\chi^2(K)}\left(\frac{K\,x}{\sigma_{\mathrm{N}}^2}\right) + p_\epsilon \frac{K}{\sigma_\epsilon^2} f_{\chi^2(K)}\left(\frac{K\,x}{\sigma_\epsilon^2}\right) \tag{1.89}$$

with $\sigma_\epsilon^2 >> \sigma_{\mathrm{N}}^2$. Thus

$$q(\sigma_{\mathrm{th}}^2) = (1-p_\epsilon)\frac{\gamma\left(\frac{K}{2}, \frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_{\mathrm{N}}^2}\right)}{\Gamma\left(\frac{K}{2}\right)} + p_\epsilon \frac{\gamma\left(\frac{K}{2}, \frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_\epsilon^2}\right)}{\Gamma\left(\frac{K}{2}\right)} \tag{1.90}$$

$$\begin{aligned}\mu(\sigma_{\mathrm{th}}^2) = (1-p_\epsilon)\frac{2\sigma_{\mathrm{N}}^2}{K}\frac{\gamma\left(\frac{K}{2}+1, \frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_{\mathrm{N}}^2}\right)}{\Gamma\left(\frac{K}{2}\right)} \\ + p_\epsilon \frac{2\sigma_\epsilon^2}{K}\frac{\gamma\left(\frac{K}{2}+1, \frac{K\,\sigma_{\mathrm{th}}^2}{2\sigma_\epsilon^2}\right)}{\Gamma\left(\frac{K}{2}\right)}.\end{aligned} \tag{1.91}$$

By defining $\epsilon \triangleq \frac{\sigma_\epsilon^2}{\sigma_{\mathrm{N}}^2}$ we have the following normalized equivalent sampling intensity

$$\breve{\lambda}_{\mathrm{eq}}(\breve{\sigma}_{\mathrm{th}}^2) = \frac{\left[\frac{(1-p_\epsilon)\gamma\left(\frac{K}{2},\frac{K}{2}\breve{\sigma}_{\mathrm{th}}^2\right)+p_\epsilon\gamma\left(\frac{K}{2},\frac{K}{2}\frac{\breve{\sigma}_{\mathrm{th}}^2}{\epsilon}\right)}{\Gamma\left(\frac{K}{2}\right)}\right]}{\left[1+\frac{1-p_\epsilon}{\rho_{\mathrm{M}}}\frac{2}{K}\frac{\gamma\left(\frac{K}{2}+1,\frac{K}{2}\breve{\sigma}_{\mathrm{th}}^2\right)}{\gamma\left(\frac{K}{2},\frac{K}{2}\breve{\sigma}_{\mathrm{th}}^2\right)}+\frac{p_\epsilon\,\epsilon}{\rho_{\mathrm{M}}}\frac{2}{K}\frac{\gamma\left(\frac{K}{2}+1,\frac{K}{2}\frac{\breve{\sigma}_{\mathrm{th}}^2}{\epsilon}\right)}{\gamma\left(\frac{K}{2},\frac{K}{2}\frac{\breve{\sigma}_{\mathrm{th}}^2}{\epsilon}\right)}\right]}. \tag{1.92}$$

1.4.3 Case Study: Examples of Signal with flat ESD

As for PDF, I consider an example of signal ESD in $\mathbb{R}^2$. Consider a spatial field $z(\mathbf{x})$ with flat ESD

$$\mathcal{E}_z(\boldsymbol{\nu}) = \frac{E_z}{|\mathcal{B}_z|}\mathbf{1}_{\mathcal{B}_z}(\boldsymbol{\nu}) \tag{1.93}$$

where the set $\mathcal{B}_z$ is the band in $\mathbb{R}^d$. From (1.64), by extending the result in [26] to $\mathbb{R}^d$, the NEMSE results in

$$\epsilon_{\mathrm{S}} = \frac{1}{1+\frac{\lambda_{\mathrm{eq}}(\sigma_{\mathrm{th}}^2)}{|\mathcal{B}_z|}}. \tag{1.94}$$

The NEMSE as a function of $\lambda_0/|\mathcal{B}_z|$ and $\sigma_{\mathrm{th}}^2/\sigma_{\mathrm{N}}^2$ can be obtained by substituting (1.82) in (1.94).

1.5 Conclusions

In conclusion, I analyzed the estimation of a finite-energy signal from its samples affected by measurement errors and scattered in $\mathbb{R}^d$ according to a homogeneous PPP. The expression of the optimal LSI interpolator in the MSE sense has been derived and verified that such an expression in $\mathbb{R}^d$ includes a result previously known in the literature as a special case.

When the optimal interpolator is used, the effect of both the random sampling and the measurement errors on the estimated signal ESD is an attenuation of the original signal ESD instead of a noise floor as for the case of ILP interpolator. Moreover, the effect of the distortion due to measurement errors on the NEMSE is shown to be equivalent to that of a reduction of samples intensity, which can be compensated by increasing the number of nodes inside the sampling area.

If a constraint in the overall estimation energy is imposed, I verified that an increasing number of sensors leads to a decreasing NEMSE in spite of the corresponding increasing measurement error for each sensor. For the case when the energy constraint is imposed on each sensor due to the battery lifetime limitation, as usual in the large WSN scenario, I derived a simple but useful expression for the NEMSE as a function of the estimation time and the capacity-per-volume.

Random sampling in presence of noise has been also investigated.

Chapter 2

Consensus Algorithms for Distributed Parameter Estimation in Wireless Sensor Networks

2.1 Introduction

A WSN can be defined as a network of sensing devices, denoted as nodes, which can exchange the gathered information through wireless links. Nodes are usually low cost, small size and energy-limited sensing devices. They can be stationary or mobile and are usually organized into a network. Those sensing devices are deployed to collaborate in performing a common task. Examples may be the monitoring of an environmental parameter (*e.g.*, temperature or pressure [9, 35, 36]), the detection of a binary event [37], the estimation of a spatial field [16], the estimation of the coordinates of a signal source, etc. Depending on the specific task requirements (fault tolerance, privacy issues, energy constraints), either a centralized or a distributed approach can be adopted. In a centralized setup a central unit collects all the information and completes the task, whereas in a distributed setup the nodes exchange information and accomplish the task locally. As far as the centralized estimation of physical parameters is concerned, maximum likelihood (ML) or least squares (LS) estimators [38] can be adopted, both working under the hypothesis of having all the required observations available at one central unit. However, the scarce robustness to central unit failures and poor network scalability have brought to consideration distributed approaches. For instance, recursive weighted LS estimation has been considered [39, 40], alongside a consensus-based algorithm that allows incorporating information from neighbor nodes in the local estimate. A similar approach is taken within the Bayesian framework [41–43], where consensus-based distributed Kalman filtering is proposed.

Whatever the adopted processing strategy, either centralized or distributed, in many applications a simple point estimate of the parameter vector of interest is not sufficient if not associated with a confidence region to assess the estimation uncertainty. Classically, the estimation accuracy is investigated using Cramér-Rao-like bounds [44–47]. Confidence regions can also be derived as a by-product of distributed Kalman filtering [42, 43]. Nevertheless, strong assumptions on the measurement noise (typically Gaussian) are necessary and most of the techniques provide only approximate, possibly asymptotically tight, confidence regions.

In centralized setups, provided that the regression model is linear, the derivation of confidence regions in the non-asymptotic regime is possible using the results in [28, 48–52]. The Leave-out Sign-dominant Correlation Regions (LSCR) method [48, 49] and the sign perturbed sums (SPS) method [28, 50] allow the central unit to derive, from a finite set of measurements, *guaranteed*, *non-asymptotic* confidence regions with prescribed confidence levels around the LS estimate of the parameter vector. Differently from Cramér-Rao-like bounds, the SPS [28, 50, 51] method, defines exact confidence regions under mild conditions on the distribution of the measurement noise even with a low number of measurements. Provided that the regression model is linear and that the measurement noise samples are independently and symmetrically distributed, the SPS method allows the derivation, from a *finite* set of measurements, of confidence regions with *prescribed confidence levels* around the LS estimate of the parameter vector. Efficient centralized characterization of confidence regions can be obtained using interval analysis [52]. Initially proposed for centralized estimation, SPS has been shown in [29] to be amenable to distributed estimation in WSNs.

This work proposed distributed solutions, based on SPS and suited to a wide variety of sensor networks, for distributed in-node evaluation of non-asymptotic confidence regions as defined by SPS. For that purpose, the nodes share their local information with each other and the confidence region computation is performed locally. The purpose of the network is to make each node capable of computing locally the confidence region of the estimate of unown parameters with the lowest impact on network traffic. Three information diffusion approaches (data flooding and parallel in-node processing, distributed processing via average consensus, and mixed flooding+consensus) have been considered in [29] to provide each node with the information allowing a distributed computation of the confidence region. In all cases, the information diffusion strategy, in addition to the network topology, determines the amount of data exchanged, which needs to be limited. In this regard, I introduced a novel information diffusion strategy, named TAS. It exploits the peculiarities of the SPS method, leading to a reduction of the amount of information to be exchanged among nodes and, at the same time, it is sufficiently general to be applied to any network topology. It is compared with classical *general purpose* information diffusion strategies, such as flooding [35, 53] and consensus algorithms [41], in terms of generated traffic load as well as of confidence region volume/traffic trade-off. Performance predictions, simulation and experimental results are provided, in terms of required traffic load, for various topologies, extending preliminary results presented in [29].

2.2 Non-asymptotic confidence regions

For the reader's convenience, the most significant symbols introduced in the following and their meaning are reported in Table 2.1.

Consider some spatial field described by the following parametric model [27]

$$y(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\varphi}^T(\mathbf{x})\, \boldsymbol{\theta}, \tag{2.1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is some vector of experimental conditions (time, location...) under which the field is observed, $\boldsymbol{\varphi}(\mathbf{x})$ is the regressor function, and $\boldsymbol{\theta}$ is a vector of unknown parameters. Measurements are taken by a network of n sensor nodes, spread at random locations $\mathbf{x}_i \in \mathbb{R}^{n_x}$, $i = 1, \ldots, n$. Node i collects the scalar measurement y_i according to the local measurement model

$$y_i = y(\mathbf{x}_i, \boldsymbol{\theta}^*) + w_i = \boldsymbol{\varphi}_i^T \boldsymbol{\theta}^* + w_i, \tag{2.2}$$

where $\boldsymbol{\varphi}_i = \boldsymbol{\varphi}(\mathbf{x}_i)$ is the $n_\mathrm{p} \times 1$ regressor vector at $\mathbf{x}_i$; $\boldsymbol{\theta}^*$ is the true value of the deterministic $n_\mathrm{p} \times 1$ parameter vector, which is only known to belong to the subset $\Theta \subset \mathbb{R}^{n_\mathrm{p}}$; w_i represents the measurement noise at Node i. As in [50], the random variables with realizations w_i, $i = 1 \ldots, n$ are assumed to be statistically independent and to follow a symmetrical distribution. Also in other works ([51]), no symmetry condition is considered and the random measurement sequence is only assumed to form an exchangeable sequence of random variables. This work readily extends to this alternative assumption. Deterministic regressors $\boldsymbol{\varphi}_i$ are considered here, but this work may be extended to the case of random exogenous regressors, *i.e.*, regressors $\boldsymbol{\varphi}_i$s that are independent of the noise terms. I consider the worst case in which the value of $\boldsymbol{\varphi}_i$ is assumed known only by Node i. Moreover, I assume that there exists $n' < n$ such that for all subsets of indexes $\mathcal{I} \subset \{1, \ldots, n\}$ with $|\mathcal{I}| \geqslant n'$, the regressors are such that $\det \mathbf{Q}_\mathcal{I} \neq 0$, where

$$\mathbf{Q}_\mathcal{I} = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T. \tag{2.3}$$

In what follows, $\mathbf{Q}_{\{1,\ldots,n\}}$ is denoted by $\mathbf{Q}_n$ and results in

$$\mathbf{Q}_n = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\varphi}_i \boldsymbol{\varphi}_i^T. \tag{2.4}$$

The purpose of the network is to make each node capable of computing locally the confidence region of the estimate of $\boldsymbol{\theta}^*$ with the lowest impact on network traffic. This proposed approach readily extends to vector fields in which the measurement is a vector, as well as to vectors

Table 2.1: Table of symbols and related meanings

	Linear regression system
n_p	dimension of the parameter vector
Θ	parameter space ($\Theta \subset \mathbb{R}^{n_\mathrm{p}}$)
$\boldsymbol{\theta}$	vector belonging to the parameter space Θ
$\boldsymbol{\theta}^*$	true value of the $n_\mathrm{p} \times 1$ parameter vector
$\widehat{\boldsymbol{\theta}}$	least squares estimate of $\boldsymbol{\theta}^*$
$\mathbf{x}_i$	location of Node i
$\boldsymbol{\varphi}_i$	regressor vector at $\mathbf{x}_i$;
y_i	measurement collected by Node i
	SPS variables
m	amount of sums considered by the SPS method
$a_{j,k}$	realizations of independent random signs
$\mathbf{Q}_n$	SPS normalization matrix
$\mathrm{s}_0(\boldsymbol{\theta})$	unperturbed sum
$\mathrm{s}_j(\boldsymbol{\theta})$	$m-1$ sign perturbed sums ($j = 1, ..., m-1$)
$\boldsymbol{\Sigma}_q$	non-asymptotic confidence region
	TAS information diffusion algorithm
$\mathbf{t}_r^{(k)}$	tag vector to be transmitted by Node k in round r
$\mathbf{d}_r^{(k)}$	dataset to be transmitted by Node k in round r
d_TAS	size of the dataset transmitted by TAS
$\boldsymbol{\delta}_{i,j,...}$	dataset with sums involving data from Nodes i, j, ...
$\boldsymbol{\delta}_\mathrm{F}^k$	dataset at Node k after final wrap-up
$n_\mathrm{TAS}^\mathrm{GT}$	amount of data transmitted by TAS in a generic tree
$n_\mathrm{TAS}^\mathrm{BT}$	amount of data transmitted by TAS in a binary tree
$n_\mathrm{TAS}^\mathrm{CN}$	amount of data transmitted by TAS in a clustered network
	Flooding information diffusion algorithm
d_F	size of the dataset transmitted by FL
$n_\mathrm{FL}^\mathrm{GT}$	amount of data transmitted by FL in a generic tree
$n_\mathrm{FL}^\mathrm{BT}$	amount of data transmitted by FL in a binary tree
$n_\mathrm{FL}^\mathrm{CN}$	amount of data transmitted by FL in a clustered network
	Network setup
n	number of nodes in the network
$\mathcal{N}(k)$	set of neighbors of node k
$\lambda(\ell)$	number of nodes at Level ℓ (tree network)
$\overline{\lambda}(\ell)$	number of nodes with no children at level ℓ (tree netw.)
L	number of levels of the tree network (excluding the root)
n_c	number of clusters in the clustered network
n_i^c	number of nodes (clusterhead included) in the i-th cluster

of measurements, provided that the noise components of each vector are independent and symmetrically distributed. This extension is not considered here for the sake of keeping a lighter notation. The centralized SPS method [28, 50] assumes all measurements and regressors to be known at the central processing unit. It defines an exact confidence region around the least squares estimate $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}^*$, obtained as the solution of the normal equations $\sum_{k=1}^{n} \varphi_k \left(y_k - \varphi_k^T \boldsymbol{\theta}\right) = \mathbf{0}$, that is the derivative of the MSE estimator, set equal to zero. For that purpose, as in [50], consider the *unperturbed sum* as the following function over Θ

$$\mathrm{s}_0(\boldsymbol{\theta}) = \mathbf{Q}_n^{-1/2} \sum_{k=1}^{n} \varphi_k \left(y_k - \varphi_k^T \boldsymbol{\theta}\right) \tag{2.5}$$

and the $m-1$ *sign-perturbed sums*, defined $\forall j = 1, \dots, m-1$ as the following functions over Θ

$$\mathrm{s}_j(\boldsymbol{\theta}) = \mathbf{Q}_n^{-1/2} \sum_{k=1}^{n} a_{j,k} \varphi_k \left(y_k - \varphi_k^T \boldsymbol{\theta}\right), \tag{2.6}$$

where $a_{j,k} \in \{\pm 1\}$ are realizations of independent random signs.[1] For each $\boldsymbol{\theta} \in \Theta$, one considers the elements of the set

$$\mathcal{Z}(\boldsymbol{\theta}) = \left\{ z_j(\boldsymbol{\theta}) = ||\mathrm{s}_j(\boldsymbol{\theta})||_2^2 \right\}_{j=0,1,\dots,m-1}, \tag{2.7}$$

and lists them in increasing order, giving rise to a permutation $\pi_{\boldsymbol{\theta}}(\cdot) : \{0, \dots, m-1\} \longrightarrow \{0, \dots, m-1\}$. One defines the set

$$\boldsymbol{\Sigma}_q = \left\{ \boldsymbol{\theta} \in \Theta \mid \pi_{\boldsymbol{\theta}}(0) \leq m - 1 - q \right\} \tag{2.8}$$

which contains all $\boldsymbol{\theta} \in \Theta$ for which the rank of $z_0(\boldsymbol{\theta})$ in the ordering is among the $m - q$ smallest, with $q = 1, ..., m - 1$.

In [28, 50], it was proven that

$$\mathrm{Prob}(\boldsymbol{\theta}^* \in \boldsymbol{\Sigma}_q) = 1 - \frac{q}{m}. \tag{2.9}$$

As a consequence $\boldsymbol{\Sigma}_q$ is a non-asymptotic confidence region with *exact* confidence level $1 - q/m$. The values of q and m may be chosen to get the requested confidence level of the confidence region $\boldsymbol{\Sigma}_q$ for the estimate $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}^*$. An extension of the SPS method is presented in [51], which considers that $\pi_{\boldsymbol{\theta}}$ is one of the $m!$ possible permutations on $\mathcal{Z}(\boldsymbol{\theta})$. Letting Π_k be a set of k permutations, the set

$$\boldsymbol{\Sigma}_k = \left\{ \boldsymbol{\theta} \in \Theta \mid \pi_{\boldsymbol{\theta}} \in \Pi_k \right\} \tag{2.10}$$

[1] A random sign is a symmetric ±1 value random variable taking both values with the same probability.

is defined, which allows one getting confidence regions such that

$$\mathrm{Prob}(\boldsymbol{\theta}^* \in \boldsymbol{\Sigma}_k) = \frac{k}{m!}. \tag{2.11}$$

Notice that (2.9) and (2.11) are equivalent for $k = m! - q(m-1)!$. The main advantage of the extension of SPS in [51] over that in [50] is that in the former the resolution of the confidence level is $1/m!$, while in the latter it is $1/m$. For example, with the approach in [51], confidence regions for levels $\{100\%, 96\%, \dots, 62.5\%, \dots, 8.3\%, 4.2\%\}$ may be theoretically defined for $m = 4$, whereas confidence regions only for levels $\{100\%, 75\%, 50\%, 25\%\}$ are defined in [50]. This difference may appear to be interesting when SPS is used in a distributed version, where small values of m are of interest, to limit communication costs. Nevertheless, our experiments show that with the approach in [51], when choosing $k \geqslant m! - (m-1)!$, the confidence regions are not necessarily compact. Non-asymptotic confidence regions as defined in [50] may be outer-approximated using ellipsoids, as in [50], boxes, or union of non-overlapping boxes as in [52]. In the following, the distributed computation of $\boldsymbol{\Sigma}_k$ is addressed considering different information diffusion strategies.

2.3 Information Diffusion Algorithms

This section describes the distributed computation of confidence regions as defined by the SPS algorithm [50]. Concurrent procedures for information diffusion applicable to any network topology are considered. The purpose is that each node collects the largest amount of information with the lowest amount of data exchanged in the network so that it is able to compute locally the confidence region of the LS estimate for any $\boldsymbol{\theta}^*$. Before entering into the details of our investigation, a detailed description of the different roles played by the *physical*, *logical*, and *processing* elements that affect the performance of the investigated strategies is needed. The *physical* element of a WSN is given by the deployment of nodes in the given scenario, that defines the network layout. On this regards, the only condition I assume is that all nodes can communicate with each other, either with single or multi-hop links. Given the network layout, a routing protocol is typically applied, which defines the *logical topology* of the network, that is, the set of paths and directions that data can flow through. On top of the same network layout, in fact, different kinds of *logical topologies* can be created, either hierarchical (tree topology, cluster topology. . .) or flat, depending on the routing protocol that defines, in other words, the possible *information paths* for the given deployment of nodes. Finally, the information diffusion strategies investigated in this chapter concern the *processing* elements. In fact, they deal with the way the information is managed (aggregated and/or fused) by a node before being transmitted to the next one(s) according to the *logical topology*. A node can transmit,

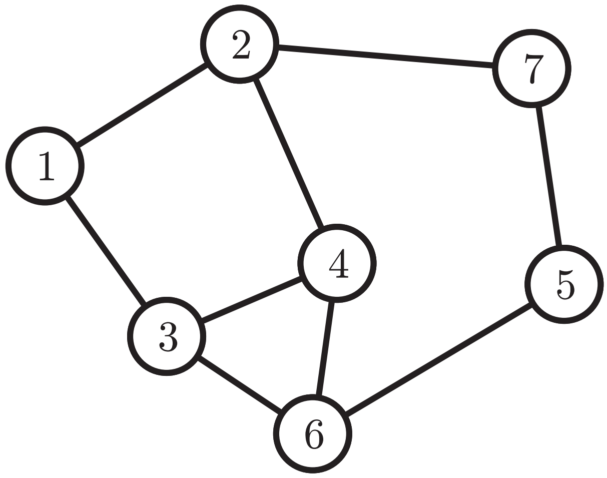


Figure 2.1: Toy network example

for instance, either elementary data (as done by FL) or a processed version of data (as done by consensus schemes and the proposed TAS algorithm). Obviously, given a fixed logical network topology, it is always possible to design an ad-hoc information diffusion algorithm that provides the best performance. However, I am interested in designing procedures that are not tailored to any specific network configuration. The TAS algorithm proposed in this chapter is meant as a topology-agnostic information diffusion strategy, thus being a general-purpose solution. For this reason, the FL algorithm, which is topology-agnostic as well, is its natural term of comparison. Both information diffusion strategies are here meant to provide each node with the information needed to locally compute the confidence region as defined by SPS. In what follows, I assume that data transmitted by Node k, $k = 1, \dots, N$ is directly received by its neighbours, that is, by the nodes within its communication range, whose indices are in the set $\backslash(k)$. By convention, $k \notin \backslash(k)$. I consider, moreover, a fully connected mesh network, where each node can directly (single hop) or indirectly (multi-hop) communicate with any other node. The behavior of the algorithms will be illustrated on the toy network represented in Figure 2.1, where circles represent network nodes and edges between two nodes indicate that they are able to communicate. For each algorithm the evolution of the amount of information available at a node k is described by a table $\mathbf{R}^{(k)}$.

2.3.1 Flooding algorithm

FL will be used as a benchmark [35,53]. When implemented to support the SPS algorithm, *pure* FL works as follows: during the first round, Node k broadcasts its own privy pair (φ_k, y_k), and receives data from its neighbors, as dictated by the logical topology. On successive rounds, Node k will also broadcast any previously received pair (φ_i, y_i), $i \neq k$ along with its own. In particular, at round r Node k transmits a packet $(\mathbf{t}_r^{(k)}, \mathbf{d}_r^{(k)})$, in which the *tag vector* $\mathbf{t}_r^{(k)}$ indicates the indices of the nodes whose measurements are present in the packet, whereas the data $\mathbf{d}_r^{(k)}$ contain the measurements and the corresponding regressors $\{(\varphi_j, y_j)\}$, $\forall j \in \mathbf{t}_r^{(k)}$.

Usually, in order to reduce the amount of transmitted information, actual implementations of flooding (*e.g.*, AODV [54]) do not retransmit already transmitted data. In the following I will always refer to such *enhanced* algorithm, that will be simply denoted as flooding.

In this case, Node i is referenced in the tag vector $\mathbf{t}_r^{(k)}$ iff

1. the pair (φ_i, y_i) is available at Node k at round $r - 1$,

2. the pair (φ_i, y_i) has never been broadcast by Node k.

At round $r = 1$, Node k transmits data $\mathbf{d}_1^{(k)}$ consisting of

$$d_\mathrm{F} = n_\mathrm{p} + 1 \tag{2.12}$$

real values, corresponding to its measurement and n_p regressors. The dimension of data $\mathbf{d}_r^{(k)}$ broadcast by Node k at successive rounds (that is, for $r > 1$) is an integer multiple of d_F, possibly zero. The transmission cost d_TAG for the tag vector depends on the way it is represented, *e.g.*, as a list of integers, in which case it is of variable length with r, or a constant-size vector of binary flags. The latter is considered in this work. As a consequence, the communication cost of the tag vector is of n binary values per communication round.

Ideally, transmission rounds are repeated until all nodes collect all the information, e.g., by checking whether the tag vector is full of ones. Upon completion, each node is able to compute (2.5) and (2.6), for any $\boldsymbol{\theta}$, and to locally derive the confidence region using the full set of data. In practice, transmission rounds may stop due to information diffusion delay constraints, or when all nodes do not detect any transmitted information from their neighbors over a given time interval.

In the latter cases, the local confidence region characterization may be performed on a reduced, possibly different across nodes, set of data.

Example 6 *Table 2.2 describes the evolution of the information collected by Node $k = 1$ in the network depicted in Figure 2.1, when FL is implemented. Before any transmission has taken place,* i.e.*, for $r = 0$, Node 1 only knows its own measurement and regressor, (φ_1, y_1).*

During the transmission round $r = 1$, Node 1 broadcasts data $\mathbf{d}_1^{(1)} = (\varphi_1, y_1)$. It receives data $\mathbf{d}_1^{(2)} = (\varphi_2, y_2)$ and $\mathbf{d}_1^{(3)} = (\varphi_3, y_3)$ from Nodes 2 and 3 respectively, thus learning measurements and regressors of Nodes 2 and 3.

*In round $r = 2$, Node 1 broadcasts $\mathbf{d}_2^{(1)} = \{(\varphi_i, y_i)\}_{i\in\{2,3\}}$. Moreover it receives data generated at Nodes 1, 4, and 7, forwarded by Node 2, (*i.e.*, it receives $\mathbf{d}_2^{(2)} = \{(\varphi_i, y_i)\}_{i\in\{1,4,7\}}$)*

Round	From Node	Data	Tag vector						
0	1	(φ_1, y_1)	1	0	0	0	0	0	0
1	2	(φ_2, y_2)	0	1	0	0	0	0	0
	3	(φ_3, y_3)	0	0	1	0	0	0	0
2	2, 3	(φ_4, y_4)	0	0	0	1	0	0	0
	3	(φ_6, y_6)	0	0	0	0	0	1	0
	2	(φ_7, y_7)	0	0	0	0	0	0	1
3	2, 3	(φ_5, y_5)	0	0	0	0	1	0	0

Table 2.2: Table $\mathbf{R}^{(1)}$ of available information at Node $k = 1$ when FL is used in the network of Figure 2.1

*and the data generated at Nodes 1, 4, and 6, forwarded by Node 3 (*i.e., $\mathbf{d}_2^{(3)} = \{(\varphi_i, y_i)\}_{i\in\{1,4,6\}}$*). Therefore, at the end of round $r = 2$, Node 1 discovers the measurements of Nodes 4, 6 and 7.*

In round $r = 3$, Node 1 broadcasts $\mathbf{d}_3^{(1)} = \{(\varphi_i, y_i)\}_{i\in\{4,6,7\}}$, and receives data generated at Nodes 3, 5, and 6, forwarded by Node 2, i.e., $\mathbf{d}_3^{(2)} = \{(\varphi_i, y_i)\}_{i\in\{3,5,6\}}$*, as well as data from Nodes 2 and 5, forwarded by Node 3,* i.e., $\mathbf{d}_3^{(3)} = \{(\varphi_i, y_i)\}_{i\in\{2,5\}}$*. Therefore, at the end of round $r = 3$, Node 1 discovers the measurement of Node 5.*

If the network is connected, and provided that sufficient transmission rounds are allowed, the FL algorithm diffuses the whole set of data to each node. The computation of the confidence region is accomplished locally using the centralized SPS algorithm. The locally computed confidence regions will be equal only in case there is agreement on the random signs realizations $\{a_{j,k}\}$ used to compute the sign perturbed sums (2.6), as well as on the random quantities (permutations or random perturbations, [50, 51]) used to resolve ties.

This agreement can be easily accomplished without additional transmission costs by the sharing of the seed of the random generators of the nodes.

2.3.2 Tagged and aggregated sums (TAS) algorithm

The TAS algorithm is based on the following consideration. Expanding (2.5) and (2.6) one gets,

$$\mathrm{s}_0(\boldsymbol{\theta}) = \mathbf{Q}_n^{-1/2}\left(\sum_{k=1}^{n}\varphi_k y_k - \left(\sum_{k=1}^{n}\varphi_k\varphi_k^T\right)\boldsymbol{\theta}\right) \tag{2.13}$$

$$\mathrm{s}_j(\boldsymbol{\theta}) = \mathbf{Q}_n^{-1/2}\left(\sum_{k=1}^{n}a_{j,k}\varphi_k y_k - \left(\sum_{k=1}^{n}a_{j,k}\varphi_k\varphi_k^T\right)\boldsymbol{\theta}\right). \tag{2.14}$$

The evaluation of (2.13) and (2.14) for any value of $\boldsymbol{\theta} \in \Theta$ does not necessarily require the knowledge of each term in the sums but rather of

$$\delta_{1\ldots n} = \left\{ \underbrace{\sum_{k=1}^{n} \varphi_k y_k}_{n_\mathrm{p} \text{ real values}}, \underbrace{\sum_{k=1}^{n} \varphi_k \varphi_k^T}_{n_\mathrm{p}^2 \text{ real values}}, \underbrace{\left\{ \sum_{k=1}^{n} a_{j,k} \varphi_k y_k \right\}_{j=1}^{m-1}}_{n_\mathrm{p}(m-1) \text{ real values}}, \underbrace{\left\{ \sum_{k=1}^{n} a_{j,k} \varphi_k \varphi_k^T \right\}_{j=1}^{m-1}}_{n_\mathrm{p}^2(m-1) \text{ real values}} \right\} \tag{2.15}$$

The main idea of the TAS algorithm is to propagate data structures similar to (2.15), composed of *partial sums* not necessarily ranging from $k = 1$ to n, but covering a subset of $\{1, \ldots, n\}$. At each transmission round, Node k generates and transmits partial sums built from data previously received from neighbors and stored in $\mathbf{R}^{(k)}$. The main challenge of the TAS algorithm is to determine a way to organize the content of the transmitted partial sums so that each node is able, after the termination of the transmission phase, to build the complete sums (2.15), or to compute partial sums with the maximum number of elements using the received partial sums. The main advantage of TAS is that the transmitted data sets are of constant size, and do not increase in size with the transmission round as it happens in FL. The size d_TAS of the dataset is obtained recalling the amount of data of its components, reported in (2.15):

$$d_\mathrm{TAS} = m \left(n_\mathrm{p} + n_\mathrm{p} \frac{n_\mathrm{p} + 1}{2} \right) \tag{2.16}$$

The evaluation of d_TAS takes into account the fact that $\varphi_k \varphi_k^T$ is symmetric[2]. Note that the size of the dataset is fixed, *independently* of the number of elements in the partial sums. As in FL, the tag vector has to be transmitted along with the data set at each transmission round. Notice that, with the representation chosen in this work, the transmission cost of the tag vector in the FL and TAS algorithms is the same.

The TAS algorithm, whose structure is reported in Algorithm 1, consists of six phases, namely, i) initialization, ii) reception, iii) distillation, iv) aggregation, v) transmission, and vi) wrap-up. The detailed description of each phase is reported hereafter, while the corresponding pseudo codes are in Section 2.3.3.

i) *Initialization phase*, see Algorithm 2. As in the FL protocol, the transmitted packet is formed by a data set and by a tag vector. During the initialization phase, Node k, $\forall k \in \{1, \ldots, n\}$

[2] Since $\sum_{k=1}^{n} \varphi_k \varphi_k^T$ is symmetric, instead of transmitting all its $\boldsymbol{n}_\mathrm{p}^2$ elements, it is sufficient to transmit $\boldsymbol{n}_\mathrm{p}$ values for the diagonal plus $\boldsymbol{\sum_{d=1}^{n_\mathrm{p}-1} d = \frac{n_\mathrm{p}(n_\mathrm{p}-1)}{2}}$ values for the upper (or lower) part, that gives $n_\mathrm{p} \frac{n_\mathrm{p}+1}{2}$. The same holds for the $(m-1)$ terms $\left\{ \sum_{k=1}^{n} a_{j,k} \varphi_k \varphi_k^T \right\}_{j=1}^{m-1}$.

Algorithm 1 TAS algorithm

```
Initialization
for r = 1 to MaxRound do
    Reception
    Distillation
    Aggregation
    Transmission
end for
Wrap-up
```

creates the packet $\left(\mathbf{t}_1^{(k)}, \mathbf{d}_1^{(k)}\right)$ to be sent in round $r = 1$. The tag vector $\mathbf{t}_1^{(k)}$ flags only Node k.

$$\begin{array}{rl} \mathbf{t}_1^{(k)} = & [0,\ \ldots,\ \underset{\substack{\uparrow \\ \text{k-1}}}{0},\ \underset{\substack{\uparrow \\ \text{k}}}{1},\ \underset{\substack{\uparrow \\ \text{k+1}}}{0},\ \ldots,\ 0] \\ & \quad\ \ \ldots \qquad\qquad\qquad\qquad \ldots \end{array} \tag{2.17}$$

The data set $\mathbf{d}_1^{(k)}$ contains the local quantities related to Node k

$$\mathbf{d}_1^{(k)} = \left\{\varphi_k y_k, \{\varphi_k \varphi_k^T\}, \{a_{j,k} \varphi_k y_k\}_{\forall j}, \{a_{j,k} \varphi_k \varphi_k^T\}_{\forall j}\right\}. \tag{2.18}$$

After initialization, the reception, distillation, aggregation, and transmission phases are sequentially repeated until a termination condition is met (*e.g.*, until a given number of rounds have been completed, as in Algorithm 1).

ii) *Reception phase*, see Algorithm 3. At each round r, Node k collects the messages containing the partial sums transmitted by its neighbors (according to the given logical topology), whose set is denoted by $\mathcal{N}(k)$.

iii) *Distillation phase*, see Algorithm 4. At the end of the reception phase of round r, Node k compares the incoming tag vectors $\mathbf{t}_r^{(j)}$, $j \in \mathcal{N}(k)$ to the previously received ones, to detect whether the packets received at round r contain new information. If it appears that a part of the data referenced in $\mathbf{t}_r^{(j)}$ have been previously received, these redundant data are removed from the corresponding partial sum and $\mathbf{t}_r^{(j)}$ is updated accordingly, see Lines 3 to 6. The resulting partial sums are then stored in $\mathbf{R}^{(k)}$. The same procedure is applied to already stored partial sums, see Lines 7 to 9. This phase reduces the number of contributors to each partial sum, so that the different partial sums can be more easily recombined, in the following *aggregation phase*, with each contributor counted no more than once.

Example 7 (Distillation phase) *Consider again the network of Figure 2.1 and the evolution of* $\mathbf{R}^{(1)}$ *given in Table 2.3. As in FL, for* $r = 0$*, Node 1 only holds its own data and forms partial sums from these data stored in*

$$\delta_1 = \left\{\varphi_1 y_1, \{\varphi_1 \varphi_1^T\}, \{a_{j,1} \varphi_1 y_1\}_{\forall j}, \{a_{j,1} \varphi_1 \varphi_1^T\}_{\forall j}\right\}.$$

Round	From Node	Data	Tag vector						
0	1	δ_1	1	0	0	0	0	0	0
1	2	δ_2	0	1	0	0	0	0	0
	3	δ_3	0	0	1	0	0	0	0
2	2	$\delta_{4,7}$	C	0	0	1	0	0	1
	3	$\delta_{4,6}$	C	0	0	1	0	1	0
3	2	$\delta_{5,6}$	0	0	C	0	1	1	0
	3	δ_5	0	C	0	0	1	0	0

Table 2.3: Table $\mathbf{R}^{(1)}$ for Node $k = 1$ using TAS in the network of Figure 2.1; C indicates elements that have been removed from the tag vector and partial sums during the distillation phase

During round $r = 1$, Node 1 broadcasts these partial sums and receives partial sums formed with the privy data from Node 2 and partial sums formed with the privy data from Node 3. During round $r = 2$, Node 1 receives a packet containing partial sums combining data from Nodes 1, 4, and 7, forwarded by Node 2, as well as a packet containing partial sums combining data from Nodes 1, 4, and 6, forwarded by Node 3. The content of these two packets is stored in $\mathbf{R}^{(1)}$, after having removed the contribution related to Node 1 from each previously received partial sum (this is indicated by a C in the tag vector in Table 2.3). Node 1 thus gets

$$\delta_{4,6} = \Bigg\{ \sum_{k\in\{4,6\}} \varphi_k y_k, \sum_{k\in\{4,6\}} \varphi_k \varphi_k^T, \\ \Bigg\{ \sum_{k\in\{4,6\}} a_{j,k}\varphi_k y_k \Bigg\}_{\forall j}, \Bigg\{ \sum_{k\in\{4,6\}} a_{j,k}\varphi_k \varphi_k^T \Bigg\}_{\forall j} \Bigg\} \tag{2.19}$$

and $\delta_{4,7}$. At the end of round $r = 3$, Node 1 receives a packet with partial sums combining data from Nodes 3, 5, and 6, forwarded by Node 2, as well as a packet with partial sums combining data from Nodes 2 and 5, forwarded by Node 3.

iv) *Aggregation phase*, see Algorithm 5. To create the packet to be broadcast at round r, Node k aggregates the partial sums available in $\mathbf{R}^{(k)}$ at round $r-1$ and which were *not* previously aggregated. This is done by summing the available partial sums to produce $\mathbf{d}_r^{(k)}$ and merging the related tag vectors to produce $\mathbf{t}_r^{(k)}$. In order to avoid duplication of terms in the sums, rows i and j of $\mathbf{R}^{(k)}$ can be merged in $\left(\mathbf{t}_r^{(k)}, \mathbf{d}_r^{(k)}\right)$ iff the intersection of i-th and j-th row tag vectors is empty.

Round	From Node	Data	Tag vector						
0	2	δ_2	0	1	0	0	0	0	0
1	1	δ_1	1	0	0	0	0	0	0
	4	δ_4	0	0	0	1	0	0	0
	7	δ_7	0	0	0	0	0	0	1
2	1	δ_3	0	C	1	0	0	0	0
	4	δ_6	0	C	C	0	0	1	0
	7	δ_5	0	C	0	0	1	0	0

Table 2.4: Table $\mathbf{R}^{(2)}$ for Node $k = 2$ using TAS in the network of Figure 2.1; C indicates elements that have been removed from the tag vector and partial sums during the distillation phase

If this condition is not met, only the row with smallest index is aggregated in a transmitted packet.

Example 8 (Aggregation phase) *Consider the evolution of* $\mathbf{R}^{(2)}$ *for Node 2 given in Table 2.4. At the end of round* $r = 1$*, Node 2 holds partial sums related to the data from Nodes 1, 2, 4, and 7, stored in* δ_1*,* δ_2*,* δ_4*, and* δ_7*. A packet containing* δ_2 *has already been transmitted in round* $r = 1$*. The other tag vectors do not intersect, as a consequence, the aggregated sums will involve* δ_1*,* δ_4*, and* δ_7*.*

The distillation phase facilitates the aggregation and wrap-up phases. Moreover, it allows to get sparser tag vectors, which may then be more efficiently combined.

v) *Transmission phase*, see Algorithm 6. The message obtained at the end of the aggregation phase is broadcast to all neighbor nodes. After the last transmission phase, the objective for Node k is the computation of the local confidence region, using the data collected so far and aggregated in the final partial sum $\delta_F^{(k)}$, evaluated in the wrap-up phase. The information diffusion process stops for Node k when it has collected all the information from other nodes or, more realistically, when a certain time has expired.

vi) *Wrap-up phase*, see Algorithm 7. The wrap-up phase can be performed by a node whenever it needs to compute the confidence region during or at the end of the information diffusion process. For that purpose, Node k evaluates a linearly weighted sum $\delta_F^{(k)} = \sum_l \widehat{b}_l^{(k)} \delta_l^{(k)}$, where $\delta_l^{(k)}$ contains the partial sums at the l-th row of $\mathbf{R}^{(k)}$ and $\widehat{\mathbf{b}}^{(k)}$ is a vector of weights. The non-zero entries of $\widehat{\mathbf{b}}^{(k)}$ select the rows of $\mathbf{R}^{(k)}$ to be combined in the partial sums.

To obtain $\widehat{\mathbf{b}}^{(k)}$, consider the tag matrix $\mathbf{T}^{(k)}$ of $\mathbf{R}^{(k)}$, with elements $t_{l,i}^{(k)}$, with l and i denoting the row and column indexes, respectively. If $\mathbf{T}^{(k)}$ is of full rank n, then $\mathbf{R}^{(k)}$ contains a contribution from all nodes of the network and as in network coding, one may retrieve each individual contribution via Gaussian elimination performed on $\mathbf{T}^{(k)}$ and proceed at the considered node in the same way as for the centralized SPS.

A second case is when $n^{(k)}$ columns of $\mathbf{T}^{(k)}$ contain 1s and the rank of $\mathbf{T}^{(k)}$ is also equal to $n^{(k)}$. In this case, only $n^{(k)}$ nodes have contributed to the partial sums stored in the rows of $\mathbf{R}^{(k)}$. Since $\mathbf{T}^{(k)}$ is of rank $n^{(k)}$, it is again possible to recover via Gaussian elimination the individual contributions of a subset $\mathcal{I}$ of $n^{(k)}$ out of the n nodes. Provided that[3] $n^{(k)} \geqslant n'$, $\mathbf{Q}_{\mathcal{I}}$ will be invertible and one will be able to obtain a LS estimate and its corresponding confidence region from a subset of $n^{(k)}$ data. When $n^{(k)} < n'$, more rounds have to be performed.

The last case to be considered is when $n^{(k)}$ columns of $\mathbf{T}^{(k)}$ contain 1s and the rank of $\mathbf{T}^{(k)}$ is strictly less than $n^{(k)}$. In that case, one may try to search the solution of the following constrained optimization problem

$$\widehat{\mathbf{b}}^{(k)} = \arg\max_{\mathbf{b}} \ \mathbf{b}^{T}\mathbf{T}^{(k)}\mathbf{1}, \tag{2.20}$$

with the constraints

$$c_i^{(k)} = \sum_l b_l \, t_{l,i}^{(k)} \in \{0,1\}, \quad i = 1, 2, \ldots, n. \tag{2.21}$$

$$\det \sum_l b_l \left(\sum_{k \in \mathbf{t}_l^{(k)}} \varphi_k \varphi_k^T \right) \neq 0. \tag{2.22}$$

The constraints (2.21) are related to the presence indicator of the quantities associated to Nodes $i = 1, \ldots, n$. Imposing $c_i^{(k)} \in \{0,1\}$ in (2.21) ensures that all measurements contribute similarly to the final sign perturbed sums, with some measurements possibly not contributing at all. In the latter case, one obtains a confidence region associated to the LS estimate of θ^* involving only the corresponding subset of sensor measurements. Since local quantities in (2.14) cannot contribute more than once, to keep their independence, one should have $c_i^{(k)} \in \{0,1\}$. The constraint (2.22) is introduced to allow the computation of an approximation of $\mathbf{Q}_n^{-1/2}$ relying on possibly less than n terms. The constrained integer programming problem (2.20)-(2.22) is NP-hard in general. If the constraint (2.22) is verified only *a posteriori*, one gets a linear cost function and (2.21) can be formulated as quadratic equality constraints. A further relaxation of (2.21) can be considered imposing only that $c_i^{(k)} \in [0,1]$. One gets then a linear programming problem, easier to solve, but that may provide a solution quite far from that of the original integer programming problem. More precisely, if for the solution, $c_i^{(k)} \in \,]0,1[$, the i-th measurement

[3] Remember that $n' < n$ is such that for all subsets of indexes $\mathcal{I} \subset \{1, \ldots, n\}$ with $|\mathcal{I}| \geqslant n'$, the regressors are such that $\det \mathbf{Q}_{\mathcal{I}} \neq 0$.

will not contribute with a unit weight. One obtains at the best a weighted LS estimate of $\boldsymbol{\theta}^*$ and its associated confidence region, and not the original LS estimate from equally-weighted data. An alternative sub-optimal wrap-up algorithm is provided in Section 2.3.3, which is less energy demanding owing to the lower computational effort required. The idea is closely related to that of the aggregation algorithm. The main difference is that in the wrap-up algorithm, the rows of $\mathbf{R}^{(k)}$ are first sorted by decreasing order of the weight of the rows of the tag matrix $\mathbf{T}^{(k)}$. The idea is to perform the aggregation starting with the partial sums to which a maximum number of nodes have contributed. The gap between the solution provided by this heuristic algorithm and the one obtained by solving (2.20)-(2.22) can be upper-bounded by considering the number of column $n^{(k)}$ of $\mathbf{T}^{(k)}$ containing 1s. Since $n^{(k)}$ represents the number of different nodes that have contributed to one of the partial sums stored in $\mathbf{R}^{(k)}$, the optimal wrap up performed solving (2.20)-(2.22) cannot aggregate data from more than $n^{(k)}$ nodes. The gap is thus less than the difference between $n^{(k)}$ and the number of aggregated data from different nodes, *i.e.*, the number of 1s in the final aggregated tag vector. In any case, before starting the final wrap-up, a node should have a matrix $\mathbf{T}^{(k)}$ such that $n^{(k)} \geqslant n'$ to have a chance wrapping-up data from enough nodes to get an invertible matrix $\mathbf{Q}_{\mathcal{I}}$. A relaxed version of the optimization problem (2.20-2.22), considering only the constraints (2.21) can be easily solved by linear programming. Once a solution has been found, one may verify whether (2.22) is satisfied. If it is not the case, (2.20) and (2.21) can be supplemented with additional inequality constraints to exclude the previously found solution and search for a new solution. When Phases ii) to v) have not been sufficiently iterated, it may happen that no satisfying solution to the optimization problem (2.20)-(2.22) can be found. Once a satisfying solution has been found, Node k can locally compute an exact confidence region based on $\boldsymbol{\delta}_F^{(k)}$, from which the following quantities are evaluated

$$\widetilde{\mathbf{s}}_0^{(k)}(\boldsymbol{\theta})=\widetilde{\mathbf{Q}}^{-1/2}\sum_{i=1}^{n} c_i^{(k)}\varphi_i\Big(y_i - \varphi_i^T\boldsymbol{\theta}\Big) \tag{2.23}$$

$$\widetilde{\mathbf{s}}_j^{(k)}(\boldsymbol{\theta})=\widetilde{\mathbf{Q}}^{-1/2}\sum_{i=1}^{n} c_i^{(k)} a_{j,i}\varphi_i\Big(y_i-\varphi_i^T\boldsymbol{\theta}\Big)\ \forall j=1,\ldots,m-1, \tag{2.24}$$

with

$$\widetilde{\mathbf{Q}} = \frac{1}{\sum_{i=1}^{n} c_i^{(k)}}\sum_{i=1}^{n} c_i^{(k)}\varphi_i\varphi_i^T. \tag{2.25}$$

Various confidence regions may then be defined and evaluated from (2.23) and (2.24). Note that (2.23) is the set of normal equations that would be obtained in a centralized context, considering a weighted least-squares estimator, with a diagonal weight matrix $\mathbf{C}^{(k)} = \operatorname{diag}\left(c_1^{(k)},\ldots,c_n^{(k)}\right)$. Similarly, (2.24) is the sign perturbed sum that would be obtained when considering weighted least-squares. In [28] it is shown that the confidence region, obtained considering (2.23) and

(2.24) in (2.8) is also a non-asymptotic confidence region. Reaching completion of the information diffusion algorithm entails that the $c_{k,i}$ are all equal to one, (FL) or equal to $1/N$ (consensus) or comprised between 0 and 1 (TAS algorithm). This implies that with a flooding or consensus approach it is always possible to have an asymptotic (in time) thus ensuring equivalence with the centralized scenario. In case of truncation, instead, the $c_{k,i}$ fall in the interval $[0,1]$, their values depending on the applied information diffusion procedure: In case that the TAS or a consensus approach are applied they might take any value in $[0,1]$, otherwise, with flooding, only 0 and 1 are possible values. while for the TAS algorithm this property depends on the network topology. If several satisfying solutions for (2.20-2.21) have been found, the one maximizing (2.22) should be selected to get the smallest confidence region, as in D-optimal experiment design [55].

Remark 10 *The TAS algorithm is inspired from network coding [56,57]. The main difference is that Node k does not need to recover, by means of Gaussian elimination, the privy data of all nodes, but the decoding of their partial sums suffices.*

Remark 11 *The efficiency of TAS with respect to FL comes from the fact that the size d_{TAS} of the data sets exchanged does not increase as the number of rounds does, as it happens in FL.*

2.3.3 TAS Pseudo Code

The pseudo-codes for each phase of the TAS algorithm are reported in Algorithms 2 to 7. The TAS algorithm is run similarly at each node of the network. The superscript $^{(k)}$ is thus omitted to lighten notations. All variables are assumed to be global.

Algorithm 2 Initialization

```
▷Get local sensor measurement◁
$y_k \leftarrow$ PerformMeasurement

▷Format data and transmit to neighbors◁
create tag vector $\mathbf{t}$ according to (2.17)
create data vector $\boldsymbol{\delta}$ according to (2.18)
TransmitToNeighbors $(\mathbf{t}, \boldsymbol{\delta})$

▷Initialize $\mathbf{R}$ with local infos◁
$\mathbf{R}.\mathbf{T} = \mathbf{t}$
$\mathbf{R}.\mathbf{D} = \boldsymbol{\delta}$
```

Algorithm 3 Reception

```
▷Get node indexes from which packets are received◁
idx ← GetNodeIdx

▷Update reception structure Rx with the tags and partial sums received from neighbors stored in $Node(i).\mathbf{t}$ and $Node(i).\boldsymbol{\delta}$◁
for i=1 to length(idx) do
    $\mathbf{Rx}.\mathbf{T} \leftarrow [\mathbf{Rx}.\mathbf{T}; Node(\mathbf{idx}(i)).\mathbf{t}]$
    $\mathbf{Rx}.\mathbf{D} \leftarrow [\mathbf{Rx}.\mathbf{D}; Node(\mathbf{idx}(i)).\delta]$
end for
```

Algorithm 4 Distillation

```
▷Distillation of new and already stored infos◁
for lx=1 to NbRows(Rx.T) do
    for l=1 to NbRows(R.T) do
        if $\mathbf{R}.\mathbf{T}(l) \subset \mathbf{Rx}.\mathbf{T}(lx)$ then
▷Clear received packet from already stored data◁
            $\mathbf{Rx}.\mathbf{T}(lx) \leftarrow \mathbf{Rx}.\mathbf{T}(lx) - \mathbf{R}.\mathbf{T}(l)$
            $\mathbf{Rx}.\mathbf{D}(lx) \leftarrow \mathbf{Rx}.\mathbf{D}(lx) - \mathbf{R}.\mathbf{D}(l)$
        end if
        if $\mathbf{Rx}.\mathbf{T}(lx) \subset \mathbf{R}.\mathbf{T}(l)$ then
▷Clear already stored data from received data◁
            $\mathbf{R}.\mathbf{T}(l) \leftarrow \mathbf{R}.\mathbf{T}(l) - \mathbf{Rx}.\mathbf{T}(lx)$
            $\mathbf{R}.\mathbf{D}(l) \leftarrow \mathbf{R}.\mathbf{D}(l) - \mathbf{Rx}.\mathbf{D}(lx)$
        end if
    end for

▷Any distilled received data is appended to R ◁
    if $\mathbf{Rx}.\mathbf{T}(lx) \neq \mathbf{0}$ then
        $\mathbf{R}.\mathbf{T} \leftarrow [\mathbf{R}.\mathbf{T}; \mathbf{Rx}.\mathbf{T}(lx)]$
        $\mathbf{R}.\mathbf{D} \leftarrow [\mathbf{R}.\mathbf{D}; \mathbf{Rx}.\mathbf{D}(lx)]$
    end if
end for

▷Clear reception structure of current node◁
clear Rx
```

Algorithm 5 Aggregation

```
▷ Perform aggregation of Tags and partial sums. Using boolean flag vector Agd, already
  aggregated infos are no more considered for aggregation in subsequent rounds ◁
t ← 0 ▷Initialize aggregated tag vector◁
δ ← 0 ▷Initialize aggregated data vector◁
for l=1 to NbRows(R.T) do
    if Agd(l) = false then
        if R.T(l) ∩ t = 0 then
            t ← t + R.T(l)
            δ ← δ + R.D(l)
            Agd(l) = true ▷l-th row of R.T flagged as aggregated◁
        end if
    end if
end for
```

Algorithm 6 Transmission

```
if t ≠ 0 then
    TransmitToNeighbors (t, δ)
end if
```

Algorithm 7 Wrap-up

```
▷Sorts lines of R by decreasing weight of lines of R.T◁
▷ Perform aggregation of tags and partial sums.◁
t ← 0 ▷Initialize wrapped-up tag vector◁
δ ← 0 ▷Initialize wrapped-up data vector◁
for l=1 to NbRows(R.T) do
    if R.T(l) ∩ t = 0 then
        t ← t + R.T(l)
        δ ← δ + R.D(l)
    end if
end for
```

2.3.4 Consensus algorithm

Given that the SPS algorithm does not require the single terms appearing in (2.13) and (2.14) but rather their sum, a possibility to compute (2.13) and (2.14) in a distributed way, is using an average consensus algorithm [58–61], converging to (2.15), as proposed in [29]. For this information diffusion strategy, $\mathbf{R}^{(k)}$ is always composed of a single row, storing the consensus state vector. Further details can be found in [29,58–61]. Consensus algorithms will be considered in the numerical results section, anyway I will not put more emphasis since they showed a poor performance in terms of generated traffic load and convergence speed, as investigated in [29].

2.4 Theoretical Analysis on various network topologies

In this section, the amount of transmitted data for distributed confidence region characterization is analyzed for both FL and TAS. Their performances are compared on different *logical* topologies, with particular reference to generic trees, that is trees with an arbitrary number of children for each node (Section 2.4.1), binary trees (Section 2.4.2) and clustered networks (Section 2.4.3), that are the most commonly used topologies in practical applications [9]. Section 2.5 considers also completely unstructured networks.

Recall that d_{F}, given by (2.12), denotes the numbers of real-valued scalars (possibly quantized) that a single data (measurement and vector of regressors) is composed of when the FL algorithm is used. With the FL algorithm, a packet usually contains several data, and thus an integer multiple of d_{F} scalars. Similarly, d_{TAS}, given by (2.16), is the fixed amount of (possibly quantized) real-valued scalars that are carried by a packet transmitted by a given node when considering the TAS algorithm.

The transmission cost of the tag vector, consisting of n binary values, is the same across transmission rounds, and whatever the information diffusion strategy.

2.4.1 Tree Topology

The tree topology is one of the most common logical topology encountered in WSNs. It might be the consequence of a particular physical deployment of nodes or the result of a spanning tree routing procedure. Usually, tree topologies resulting from routing algorithms specifically designed for WSNs introduce some constraints in the way data travel, according to energy saving strategies. For instance, only nodes at a single level of the tree may be allowed to transmit during each round and nodes belonging to that level can communicate only with nodes belonging to the successive level [62], as all the other nodes are in *sleep state*. For this reason, the generic tree topology addressed in this section will be investigated assuming that a message broadcast by a node in the forward phase is only exploited by its parent. This hypothesis will be removed in Section 2.4.2, addressing the particular case of binary trees, that discusses also what happens when children nodes can overhear transmissions carried out by their parents.

Consider now a generic tree topology, *i.e.*, a tree where each node has an arbitrary, yet known, number of children, possibly zero. Denote with $\lambda(\ell)$ the number of nodes at Level ℓ and with $\overline{\lambda}(\ell)$ the number of nodes at Level ℓ that have no children, with ℓ ranging from $\ell = 0$ (the root) to $\ell = L$ (the leaves). Of course $\lambda(0) = 1$, since the tree is single rooted. The total number of nodes

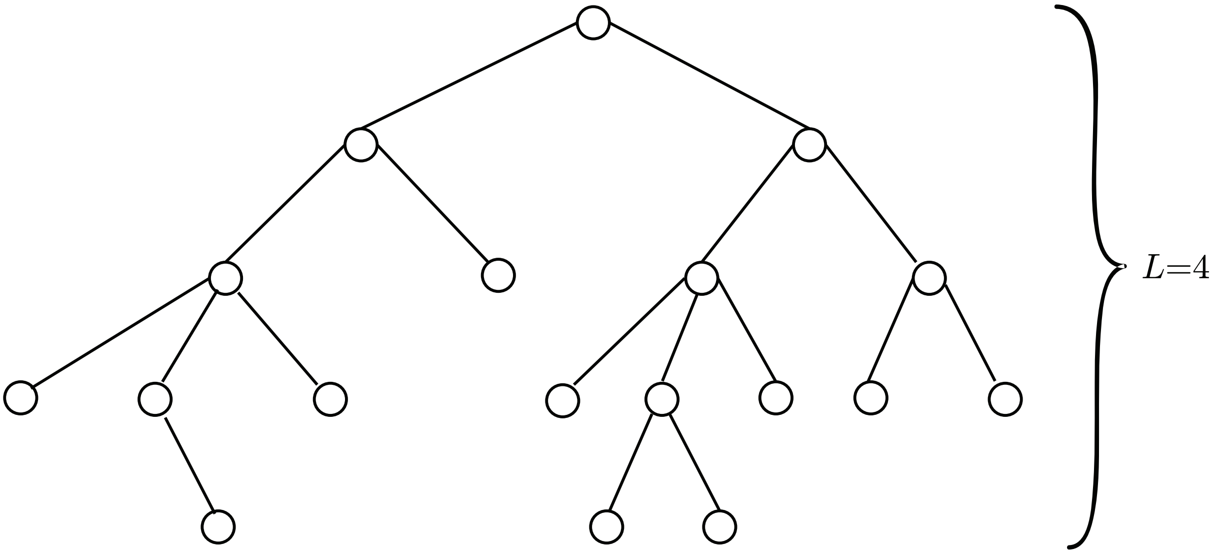


Figure 2.2: Generic tree topology with $L = 4$, where $\lambda(0) = 1$, $\lambda(1) = 2$, $\lambda(2) = 4$, $\bar{\lambda}(2) = 1$, $\lambda(3) = 8$, $\bar{\lambda}(3) = 6$, $\lambda(4) = \bar{\lambda}(4) = 3$.

forming the network is therefore $n = \sum_{\ell=0}^{L} \lambda(\ell)$. An example of these networks is depicted in Figure 2.2.

FL algorithm

The amount of data that needs to be transmitted in the forward phase from Level L to Level $L-1$ is $f_{L,L-1} = \lambda(L)\, d_{\mathrm{F}}$. When $1 \leqslant \ell < L$, this amount, from Level ℓ to Level $\ell-1$, is $f_{\ell,\ell-1} = (\lambda(L) + \cdots + \lambda(\ell))\, d_{\mathrm{F}}$. In the backward phase, the amount of data that needs to be transmitted from Level 0 to Level 1 is $b_{0,1} = nd_{\mathrm{F}}$. When $1 \leqslant \ell < L$, from Level ℓ to Level $\ell+1$, it is $b_{\ell,\ell+1} = \left(\lambda(\ell) - \overline{\lambda}(\ell)\right) n\, d_{\mathrm{F}}$.

Finally, the amount of data that has to be transmitted with the FL algorithm to share all data between nodes in the network is

$$\begin{aligned} n_{\mathrm{FL}}^{\mathrm{GT}} &= \left(\lambda(L) + (\lambda(L) + \lambda(L-1)) + \cdots + \sum_{\ell=1}^{L} \lambda(\ell)\right) d_{\mathrm{F}} \\ &\qquad + n\, d_{\mathrm{F}} + \sum_{\ell=1}^{L-1} \left(\lambda(\ell) - \overline{\lambda}(\ell)\right) n d_{\mathrm{F}} \\ &= L n\, d_{\mathrm{F}} - \left(\lambda(0) + (\lambda(0) + \lambda(1)) + \cdots + \sum_{\ell=0}^{L-1} \lambda(\ell)\right) d_{\mathrm{F}} \\ &\qquad + n^2 d_{\mathrm{F}} - \lambda(L)\, n\, d_{\mathrm{F}} - \left(\sum_{\ell=0}^{L-1} \overline{\lambda}(\ell)\right) n\, d_{\mathrm{F}}. \end{aligned} \tag{2.26}$$

TAS algorithm

In the forward phase, the TAS distillation and aggregation phases take place after each transmission round. The data reaching the root corresponds to the elements required to evaluate the unperturbed and perturbed sums that would be obtained in a centralized version of the algorithm. This way of operating ensures thus an exact retrieval of the entire sums (2.5) and (2.6). In the backward phase, this information is spread over the tree without any further processing. As already mentioned, all data packets have a constant size d_{TAS}.

The amount of data to be transmitted in the forward direction from Level ℓ to Level $\ell - 1$ is $\lambda(\ell)\, d_{\text{TAS}}$. In the backward direction, from Level ℓ to Level $\ell + 1$, it is $\left(\lambda(\ell) - \overline{\lambda}(\ell)\right) d_{\text{TAS}}$, since nodes without children do not transmit further. Accounting for both phases, one gets

$$\begin{aligned} n_{\text{TAS}}^{\text{GT}} &= \left(\sum_{\ell=1}^{L} \lambda(\ell)\right) d_{\text{TAS}} + \sum_{\ell=0}^{L-1} \left(\lambda(\ell) - \overline{\lambda}(\ell)\right) d_{\text{TAS}} \\ &= (2n-1)d_{\text{TAS}} - \lambda(L)d_{\text{TAS}} - \left(\sum_{\ell=0}^{L-1} \overline{\lambda}(\ell)\right) d_{\text{TAS}} . \end{aligned} \tag{2.27}$$

Starting from the general expressions (2.26) and (2.27), in Section 2.4.2 I investigate the amount of data transmitted by FL and TAS in the significant case of binary trees.

2.4.2 Binary Tree Topology

Consider a *single-rooted complete binary tree* with $L + 1$ levels, ranging from the root at level $\ell = 0$ to the leaves at level $\ell = L$. In this case,

$$\lambda(\ell) = 2^{\ell} , \tag{2.28}$$

$$\overline{\lambda}(\ell) = 0 \quad \text{for } \ell = 0, 1, \ldots, L - 1 , \tag{2.29}$$

$$n = \sum_{\ell=0}^{L} \lambda(\ell) = 2^{L+1} - 1 . \tag{2.30}$$

FL algorithm

Using (2.28), (2.29), and (2.30) in (2.26), the amount of data transmitted by FL in a generic tree can be specialized for the binary tree case. Given (2.28), (2.30) and (2.29), the amount of data transmitted by FL in a generic tree, given by (2.26), can be specialized for the binary tree case Starting from Level L, each node of that level broadcasts its own local data. Then, parent nodes process the received data with their own and broadcast the appropriate packet to their parents. This process (forward phase) is repeated until the root is reached. During the backward phase, the tree is then traveled from Level 0 to Level L. Nodes participate only in (at most two) rounds of transmission involving the level they belong to. On these tree topologies, tag vectors may be avoided. Usually, tree topologies resulting from routing algorithms specifically designed for WSNs introduce some constraints in the way data travel according to energy saving strategies. For instance, during each transmission round a single level of the tree may be active and nodes belonging to that level can comunicate only with nodes belonging to the successive or previous level [62]. For this reason two variants of the FL algorithm will be considered. In the forward phase, one assumes that a message broadcast by a node is only exploited by its parent, or by its parent *and* its two children. The two variants are denoted FL-U and FL-B for unidirectional and bidirectionnal processing (when children also process the message). Clearly, a message broadcast by a node may be overheard by many more nodes. The amount of data to be transmitted in the forward phase by nodes from

- Level L to Level $L-1$ is $f_{L,L-1} = 2^L d_{\mathrm{F}}$, since there are 2^L nodes at Level L, each transmitting its own data;

- Level $L-1$ to Level $L-2$ is $f_{L-1,L-2} = 2^{L-1}(2+1)\,d_{\mathrm{F}} = 2^{L-1}\left(2^2-1\right)d_{\mathrm{F}}$, since there are 2^{L-1} nodes at Level $L-1$, each of which broadcasts its own data plus the data it received from its two children;

- Level ℓ to Level $\ell-1$ is $f_{\ell,\ell-1} = 2^{\ell}\left(2^{L-\ell}+\cdots+1\right)d_{\mathrm{F}} = 2^{\ell}\left(2^{L-\ell+1}-1\right)d_{\mathrm{F}}$, since there are 2^{ℓ} nodes at Level ℓ, each of which broadcasts its own data plus the data it received from its two children;

- Level 1 to Level 0 is $f_{1,0} = 2\left(2^{L-1}+\cdots+1\right)d_{\mathrm{F}} = 2\left(2^{L}-1\right)d_{\mathrm{F}}$.

In the backward phase, the amount of data to be transmitted by nodes from

- Level 0 to Level 1 is $b_{0,1} = \left(2^{L+1}-1\right)d_{\mathrm{F}}$, since the root has to transmit the data collected by all nodes to its children;

- Level ℓ to Level $\ell+1$ is $b_{\ell,\ell+1} = 2^{\ell}\left(2^{L+1}-1\right)d_{\mathrm{F}}$, since there are 2^{ℓ} nodes at Level ℓ;

- Level $L-1$ to Level L is $b_{L-1,L} = 2^{L-1}\left(2^{L+1}-1\right)d_{\mathrm{F}}$.

Finally, the amount of data that has to be transmitted with the FL-U algorithm to share all data between nodes in the network is

$$\begin{aligned} N_{\mathrm{FL}}^{\mathrm{BT}} &= \left(\sum_{\ell=1}^{L} 2^{\ell}\left(2^{L-\ell+1}-1\right) + \sum_{\ell=0}^{L-1} 2^{\ell}\left(2^{L+1}-1\right)\right) d_{\mathrm{F}} \\ &= \left(L2^{L+1} - 2\left(2^{L}-1\right) + \left(2^{L+1}-1\right)\left(2^{L}-1\right)\right) d_{\mathrm{F}} \\ &= \left(2^{2L+1} + \left(L-\frac{5}{2}\right)2^{L+1} + 3\right) d_{\mathrm{F}}. \end{aligned} \tag{2.31}$$

Using (2.30) in (2.31), one gets

$$\begin{aligned} n_{\mathrm{FL}}^{\mathrm{BT}} &= \left(\frac{(n+1)^2}{2}+\left(\log_2(n+1)-\frac{7}{2}\right)(n+1)+3\right)d_{\mathrm{F}} \\ &\simeq \frac{(n+1)^2}{2}d_{\mathrm{F}}. \end{aligned} \tag{2.32}$$

for n sufficiently large.

If we remove the hypothesis that nodes enter in a *sleep state* at the end of their transmission round (thus allowing bidirectional communications), it is true that a message transmitted by a node in the forward phase can be processed also by its children. This property can be used in the backward phase by FL (denoted in this case FL-B) to reduce the amount of data to propagate. In this case (2.26) boils down to here, a message transmitted by a node in the forward phase is processed by its parent and by its children. In this backward phase, this property is used by the FL-B algorithm to reduce the amount of data to propagate backwards. In the backward phase, accounting for the data heard by children of a nodes broadcasting to their parents, the amount of data to be transmitted by nodes from

- Level 0 to Level 1 is $b_{0,1}=\left(2^{L+1}-1\right)d_{\mathrm{F}}$, since the root has to transmit the data collected by all nodes to its children;
- Level 1 to Level 2 is $b_{1,2}=2\left(\left(2^{L+1}-1\right)d_{\mathrm{F}}-\left(2^{L}-1\right)d_{\mathrm{F}}\right)=2^{L+1}\left(2^{1}-1\right)d_{\mathrm{F}}$, since there are 2 nodes at Level 1 and each node at Level 2 has already received all data from $\left(2^{L}-1\right)$ nodes brodcast in the forward phase;
- Level ℓ to level $\ell+1$ is $b_{\ell,\ell+1}=2^{\ell}\left(\left(2^{L+1}-1\right)d_{\mathrm{F}}-\left(2^{L-\ell+1}-1\right)d_{\mathrm{F}}\right)=2^{L+1}\left(2^{\ell}-1\right)d_{\mathrm{F}}$, since there are 2^{ℓ} nodes at Level ℓ and each nodes at Level $\ell+1$ has already received data from $\left(2^{L-\ell+1}-1\right)$ nodes broadcast in the forward phase;
- Level $L-1$ to level L is $b_{L-1,L}=2^{L-1}\left(\left(2^{L+1}-1\right)d_{\mathrm{F}}-\left(2^{2}-1\right)d_{\mathrm{F}}\right)=2^{L+1}\left(2^{L-1}-1\right)d_{\mathrm{F}}$.

Finally, the amount of data that has to be transmitted with the FL-B algorithm to share all data between nodes in the network is

$$\begin{aligned} n_{\mathrm{FL\text{-}B}}^{\mathrm{BT}} &= \left(\sum_{\ell=1}^{L}2^{\ell}\left(2^{L-\ell+1}-1\right)+\left(2^{L+1}-1\right)+\sum_{\ell=1}^{L-1}2^{L+1}\left(2^{\ell}-1\right)\right)d_{\mathrm{F}} \\ &= \left(L2^{L+1}-2^{L+1}+2+\left(2^{L+1}-1\right)+2^{L+1}\left(2^{L}-2-(L-1)\right)\right)d_{\mathrm{F}} \\ &= \left(2^{L+1}\left(2^{L}-1\right)+1\right)d_{\mathrm{F}}. \end{aligned} \tag{2.33}$$

Using again (2.30) in (2.33), one gets

$$\begin{aligned} n_{\text{FL-B}}^{\text{BT}} &= \left((n+1)\,\frac{1}{2}\,(n-1)+1\right) d_{\text{F}} \\ &= \frac{n^2+1}{2} d_{\text{F}} && (2.34) \\ &= \frac{(n+1)^2}{2} d_{\text{F}} - n\, d_{\text{F}}. && (2.35) \end{aligned}$$

One observes that $n_{\text{FL}}^{\text{BT}} > n_{\text{FL-B}}^{\text{BT}}$. As expected, accounting for data overheard by children in the forward phase reduces the amount of data to be transmitted. For large networks, however, both (2.32) and (2.35) scale quadratically in n, thus making the bidirectional tree not convenient, as it is more power consuming.

TAS algorithm

The amount of data transmitted by TAS in the binary tree case can be derived using (2.28), (2.29), and (2.30) in (2.27), thus obtaining

$$n_{\text{TAS}}^{\text{BT}} = \frac{3}{2}\,(n-1)\, d_{\text{TAS}}. \tag{2.36}$$

The amount of data to be transmitted in the forward direction from Level ℓ to level $\ell - 1$ is $2^{\ell} d_{\text{TAS}}$, since there are 2^{ℓ} nodes at Level ℓ, each broadcasting a packet of size d_{TAS}. Similarly, in the backward direction, the amount of data to be transmitted from Level $\ell - 1$ to level ℓ is $2^{\ell-1} d_{\text{TAS}}$. Accounting for both phases, the amount of data that has to be transmitted with the TAS algorithm to allow each node to evaluate a confidence region is

$$\begin{aligned} n_{\text{TAS}}^{\text{BT}} &= \sum_{\ell=1}^{L} 2^{\ell} d_{\text{TAS}} + \sum_{\ell=0}^{L-1} 2^{\ell} d_{\text{TAS}} \\ &= \left(2^{L+1} - 2 + 2^{L} - 1\right) d_{\text{TAS}} \\ &= \frac{3}{2}\,(N-1)\, d_{\text{TAS}}. && (2.37) \end{aligned}$$

With the TAS algorithm $n_{\text{TAS}}^{\text{BT}}$ scales thus linearly with n.

Comparison

When comparing (2.32), (2.35), and (2.37), asymptotically, the TAS algorithm is the most efficient, since the amount of data to be exchanged on the network scales linearly with the number of nodes n, where it scales in n^2 with the other algorithms. Nevertheless, for small values of n, the fact that $d_{\text{TAS}} > d_{\text{F}}$ can make the TAS algorithm less efficient.

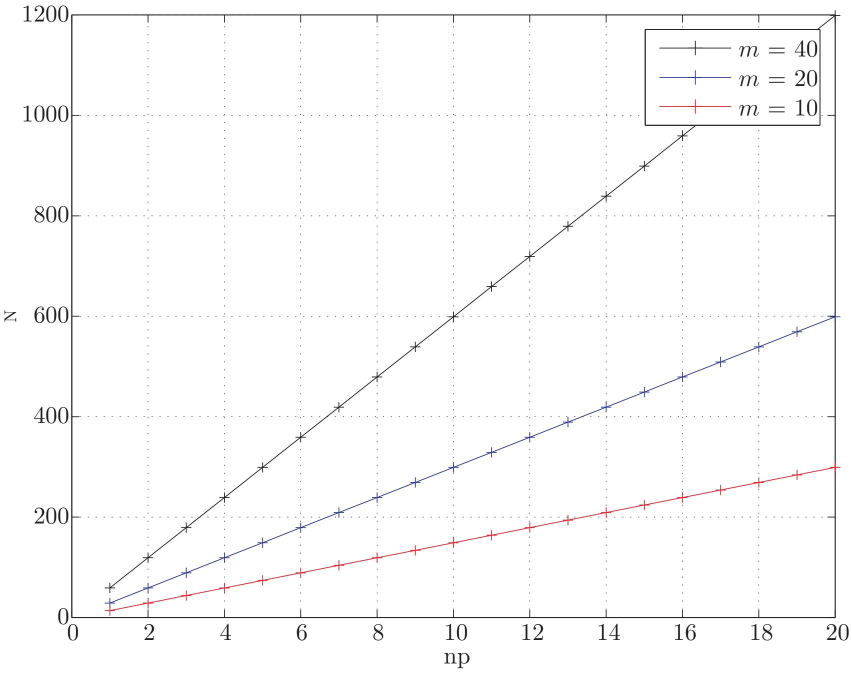


Figure 2.3: Critical value $n^*_{\text{TAS>FL}}$, as a function of n_{p}, on binary trees, for several values of m.

On a binary tree, TAS is more efficient than FL-B when

$$(3n - 3)d_{\text{TAS}} < (n^2 + 1)d_{\text{F}}.$$

Using (2.16) and (2.12) one obtains the following condition

$$\left(n^2 + 1\right) K_1 - 3n + 3 > 0, \tag{2.38}$$

where

$$K_1 = \frac{n_p + 1}{\left(n_p + n_p\frac{n_p+1}{2}\right) m}.$$

For sufficiently large n, (2.38) is always satisfied, for all n_p and m. Moreover, when n is larger than

$$n^*_{\text{TAS>FL}} = \frac{3 + \sqrt{9 - 4K_1\left(3 + K_1\right)}}{2K_1}, \tag{2.39}$$

TAS is more efficient than FL. Figure 2.3 represents $n^*_{\text{TAS>FL}}$ as a function of n_p, considering $m = 10$, $m = 20$, and $m = 40$. The behaviour is not exactly linear, but when n_p grows large, $K_1 \approx \frac{2}{n_p m}$ and $n^*_{\text{TAS>FL}} \approx \frac{3}{2} n_p m$.

2.4.3 Clustered Topology

Consider a clustered network, formed by n nodes, structured on a single level of hierarchy, as depicted in Figure 2.4. The network is hence assumed to be divided into n_c clusters. The i-th cluster comprises a random number of nodes n_i^c, including the clusterhead, that is the special node responsible for aggregating the local data of its sons. The subnetwork formed by clusterheads is considered to be fully connected: Clusterheads can directly communicate with each other. Moreover, each node in a cluster is assumed to directly communicate with its clusterhead (and vice-versa).

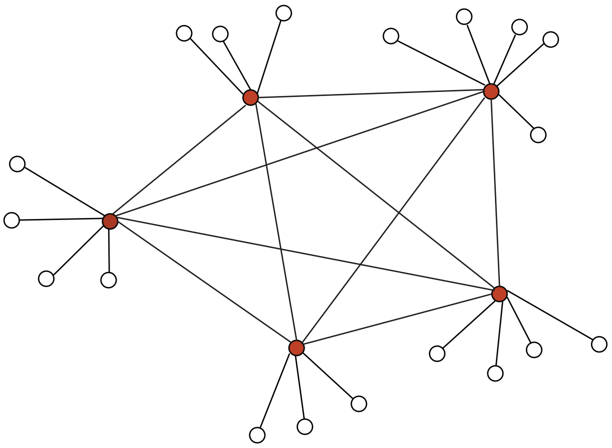


Figure 2.4: A clustered topology. Clusterheads are indicated in red.

FL algorithm

All nodes in a cluster can overhear broadcast transmissions operated by the corresponding clusterhead. Therefore, the amount of data to be transmitted when employing the FL algorithm is

$$\begin{aligned} n_\mathrm{FL}^\mathrm{CN} &= \left((n - n_\mathrm{c}) + n + (n_\mathrm{c} - 1)\, n\right) d_\mathrm{F} \\ &= (n - n_\mathrm{c} + n_\mathrm{c} n)\, d_\mathrm{F}. \end{aligned} \tag{2.40}$$

This is because all nodes, apart from clusterheads, initially transmit their local information to clusterheads, leading to $(n - n_\mathrm{c}) d_\mathrm{F}$ transmitted scalar data. Then clusterheads broadcast the received data and their own, thus forming a total flow of $n d_\mathrm{F}$ scalar data. At this point, all nodes in each cluster are completely informed about data related to their respective cluster. Finally, there is a backward transmission during which each clusterhead is transmitting towards its cluster all the $n d_\mathrm{F}$ scalar data except the ones that it previously transmitted, leading to further $(n_\mathrm{c} - 1)\, n d_\mathrm{F}$ transmitted scalars, composed of n_c clusterheads transmitting not n, but $(n - n_i^\mathrm{c}) d_\mathrm{F}$ scalar data, *i.e.*, a total of $\sum_{i=1}^{n_\mathrm{c}} \left(n - n_i^\mathrm{c}\right) d_\mathrm{F} = (n_\mathrm{c} - 1)\, n d_\mathrm{F}$.

TAS algorithm

On this topology, the TAS algorithm transmission phases can be organized as follows. At the beginning, each node, with the exception of clusterheads, transmits the partial sums calculated with its own data, corresponding to d_{TAS} real values per node. Then each clusterhead aggregates the local data of all nodes in its cluster. Successively, clusterheads transmit to all other clusterheads their aggregated data. Since the network of clusterheads is fully connected, a single broadcast transmission for each of the clusterheads suffices for all clusterheads being capable to construct the completely aggregated data. The amount of scalar data, that has to be transmitted, is thus

$$n_{\text{TAS}}^{\text{CN}} = ((n - n_{\text{c}}) + n_{\text{c}} + n_{\text{c}})\, d_{\text{TAS}} = (n + n_{\text{c}})\, d_{\text{TAS}}.$$

This accounts for the initial $n - n_{\text{c}}$ transmissions and the subsequent actions of clusterheads, that should broadcast to each other the partially aggregated data and then broadcast, towards nodes forming their cluster, the completely aggregated data.

Comparison

TAS is better than FL when $n_{\text{TAS}}^{\text{CN}} < n_{\text{FL}}^{\text{CN}}$, *i.e.*, when

$$\begin{aligned}(n - n_{\text{c}} + n_{\text{c}} n)\, d_{\text{F}} - (n + n_{\text{c}})\, d_{\text{TAS}} &> 0 \\ \left(1 + \frac{n_{\text{c}}(n-2)}{n + n_{\text{c}}}\right) \frac{d_{\text{F}}}{d_{\text{TAS}}} &> 1. \end{aligned} \tag{2.41}$$

With n sufficiently large, one has

$$\left(1 + \frac{n_{\text{c}}(n-2)}{n + n_{\text{c}}}\right) \frac{d_{\text{F}}}{d_{\text{TAS}}} \approx (n_{\text{c}} + 1)\frac{d_{\text{F}}}{d_{\text{TAS}}}.$$

This implies that TAS is better than FL when

$$n_{\text{c}} > \frac{d_{\text{TAS}}}{d_{\text{F}}} - 1.$$

Remark 12 *In Section 2.3 I indicated that the TAS algorithm proposed in this paper is meant as a topology-agnostic information diffusion strategy. Of course, given the network topology, specialized information diffusion strategies can be designed, likely providing better performance. For instance, in the case of the clustered topology here considered, one could imagine a mixed FL+TAS approach in which, during the first transmission phase, each node of a cluster conveys to the clusterheads d_F data, composed by its privy data with no aggregation (as done by FL). Then, the tagged and aggregated sums are evaluated by the clusterheads, that make data circulate as dictated by TAS. In this case, the amount of scalar data that has to be transmitted is*

$$n_{FL+TAS}^{CN} = (n - n_c)\, d_F + 2 n_c d_{TAS},$$

which is always lower than n_{TAS}^{CN}. Morevover one has $n_{FL+TAS}^{CN} < n_{FL}^{CN}$ as soon as $n > 2\frac{d_{TAS}}{d_F}$.

2.5 Simulation Results

In this section, all simulations results have been obtained considering sensor nodes randomly deployed over a square of side of one measurement unit. The nodes transmit information over lossless links (*i.e.*, no transmission errors and no packet collisions), while confidence regions have been evaluated with the interval analysis techniques described in [52] and the Intlab library [63] for interval computations. Data are generated considering the model (2.1), with randomly generated parameters and regressors using realizations of independent zero-mean unit variance Gaussian variables. The noise corrupting data is also zero-mean Gaussian, with a variance adjusted to get a signal-to-noise ratio of 15 dB. First, one numerically investigates the effect on the shape of the confidence region of information diffusion of a part of the measurements. To this

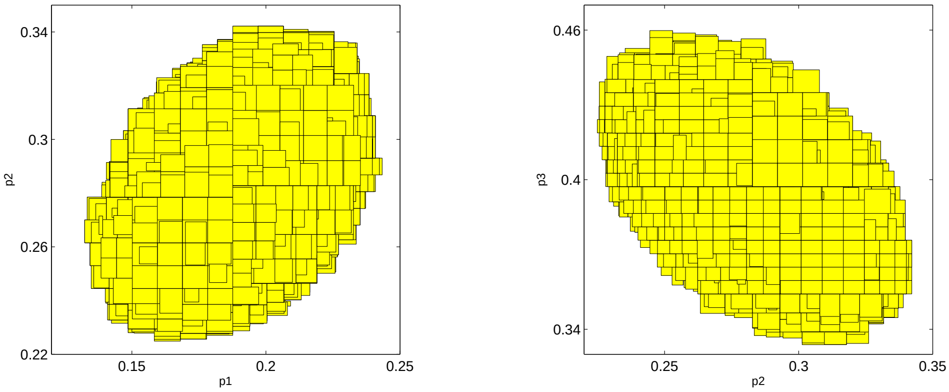


Figure 2.5: Projections of a 90% confidence region computed at node 1 after 4 consensus iterations. A random unstructured network of $n = 100$ nodes is considered.

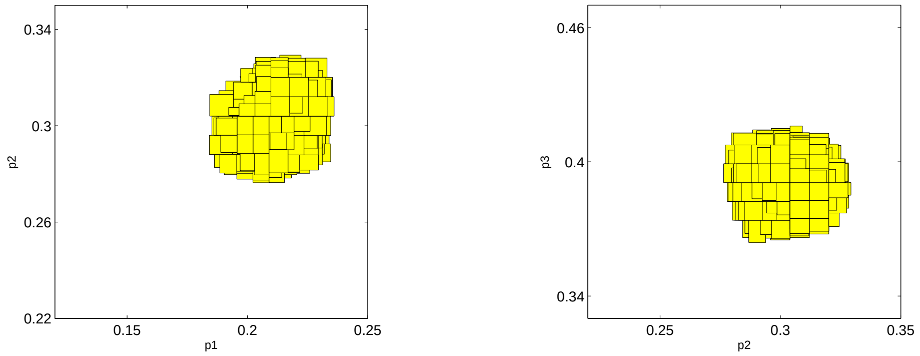


Figure 2.6: Projections of a 90% confidence region computed at node 1 after 30 consensus iterations. A random unstructured network of $n = 100$ nodes is considered.

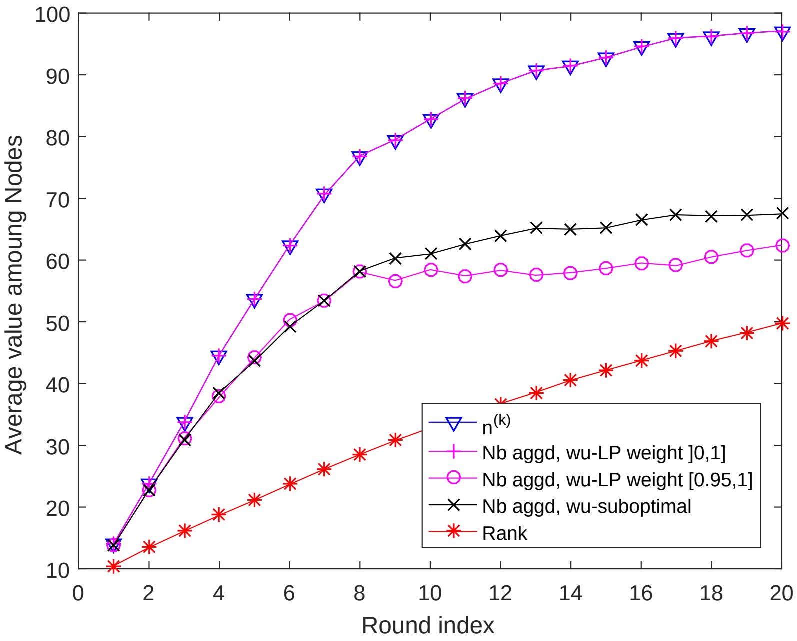


Figure 2.7: Behavior of the TAS algorithm with a random unstructured topology, as a function of the round index.

purpose, I instantiate a random unstructured network of $n = 100$ nodes, uniformly distributed over a unit area, and consider a true parameter value $\boldsymbol{\theta}^* = [\theta_1, \theta_2, \theta_3] = [0.2, 0.3, 0.4]$. The inter-node communication range is set to $d_{\text{comm}} = \sqrt{\frac{\log_2 n}{2n}}$. According to [64], this range guarantees almost sure connectivity of a network of n nodes, deployed on a finite area. A truncated Metropolis consensus algorithm [29, 58, 60] is considered for the distributed computation of confidence regions. Similar results may be obtained also for the other information diffusion strategies, considered in Section 2.3. Figs. 2.5 and 2.6 show the confidence region computed at node 1 after 4 and 30 iterations, respectively, projected over the planes (θ_1, θ_2) and (θ_2, θ_3). As expected, the volume of the confidence region reduces with the number of rounds and measurements collected. One should observe, however, that conditionally on the available information the confidence level is the same in the two cases. This has to be borne in mind also in the remainder of this section: Any truncation affected confidence region always keeps the same level of confidence as the one of the confidence region that would be obtained after a complete gathering of the information.

2.5.1 Behavior of the TAS algorithm

One considers here a random unstructured topology to see how information propagates within the network with the TAS algorithm. Figure 2.7 describes the evolution as a function of the number of rounds of the average rank of the tag matrices, the average number of data wrapped-up with the suboptimal wrap-up described in Algorithm 7, and that obtained using linear programming. With the latter approach, two plots are reported, one is showing the average number of data contributing to the final sum with a weight within the interval $[0.95, 1]$, the second is the average number of data contributing, whatever their strictly positive weight. Finally, the average value of $n^{(k)}$ is provided. Averages are taken over all nodes. For the considered simulation, a network of $n = 100$ nodes is investigated. The corresponding graph is connected with an average node connectivity of 6.38 and a diameter of 13.

One observes first that the average rank increases slower than $n^{(k)}$. The sum of the contributions of all nodes may thus be obtained before obtaining each individual contribution. Second, the wrap-up via linear programming is able to collect most of the data, even if their weight is not necessarily one in the final sum. The suboptimal wrap-up algorithm performs somewhat worse than the wrap-up via linear programming, but is able to gather an amount of data close to that contributing with a weight close to 1 in the wrap-up using linear programming. Moreover, all these quantities increase fast in the first rounds and slower after several rounds. This is due to the fact that at the beginning, each packet contains new information, whereas packets in the last rounds contain only limited new information. Moreover, the aggregation phase has more difficulties to aggregate tag vectors received in the last rounds, which contains already many contributions from different nodes and are likely to contain at least partly similar contributions. When the network is more structured, this phenomenon does not appear and the aggregation can be performed more efficiently.

Considering the diameter of the network, with a FL algorithm, without packet size limitation, all data would have reached all nodes in 13 rounds. On this unstructured topology, TAS is clearly less efficient, since with the suboptimal wrap-up, about 65% of the data have been gathered, whereas with wrap-up using linear programming, between 60% of the data are contributing with a weight close to 1 and 90% with a non-zero weight.

However, on such unstructured topology, the performance of the TAS algorithm may be improved considering variants of the aggregation phase, which may aggregate packets even if they have common contributions. This requires integer tag vectors. The distillation and wrap-up phases have to be adopted accordingly. This variant of TAS is close to network coding, but again, one is interested in the sum of the contributions of each node, and not in each contribution.

2.5.2 TAS vs FL

In order to compare the TAS and the FL algorithms, I consider random trees and random unstructured topologies, with the same order of magnitude in terms of number of nodes. For what concerns the analysis on random trees, I build a spanning tree on top of a random unstructured network, setting the inter-node communication range $d_{\text{comm}} = \sqrt{\frac{\log_2 n}{2n}}$. According to [64], this range guarantees almost sure connectivity of a network of n nodes, deployed on a unit area. For each n (see the horizontal axis in Figure 2.8), 100 connected network realizations are instantiated. TAS and FL are compared in terms of the required number of data to be transmitted in each network realization. The success rate of TAS is the percentage of network realizations that proved favorable to TAS, *i.e.*, for which fewer measurements need to be exchanged to get all data reaching all nodes of the network.

Figure 2.8 shows this success rate as a function of n, for several values of n_{p}. As foreseen in the theoretical analysis in Section 2.4, there always exists a threshold value of n, depending on n_{p}, above which the TAS outperforms the FL algorithm, *i.e.*, the percentage closes to 100%.

I now investigate the trade-off between the confidence region volume and the amount of data transmitted by each node. Figure 2.9 shows the average volume of 90% confidence region as a function of the average amount of data that is communicated by each node. The volume and data amount are averaged across all nodes and across 100 random tree realizations, while simulation parameters are set to $n_{\text{p}} = 2$, $q = 1$, $n = 100$ and $m = 10$. Figure 2.9 helps in determining the amount of data that needs to be transmitted by each node on average to obtain a given confidence region average volume. One can observe that the TAS algorithm outperforms the FL to achieve meaningful small volume values, in terms of the average amount of data transmitted by each node.

Similar results can be obtained on clustered networks. The number of clusters is set to $n_c = 20$ and the average number of per cluster nodes is set to $\mathbb{E}[N_i^{\text{c}}] = 7$ (the parameter dimension is $n_{\text{p}} = 2$, while $q = 1$ and $m = 10$). In particular, Fig. 2.10 shows the average volume of the confidence region, across nodes and 100 clustered network realizations. Here the number of computed pairs volume-amount of data is much lower than that of random trees, due to the fewer transmission rounds. The average amount of data transmitted by each node, needed to obtain meaningful small volumes, is lower when employing the TAS algorithm, as it was on random trees. Finally, consider a random unstructured network, setting $n = 100$ and $n_{\text{p}} = 3$. As shown in Figure 2.11, the FL algorithm behaves better than TAS, providing lower volume values for the same amount of data. For comparison, it is also shown how both the FL and the TAS algorithm outperform the state-of-the-art consensus algorithms, independently of the considered consensus matrix (Metropolis [58] or Perron [41]).

This section confirms the general behavior that was highlighted in Section 2.4: On structured

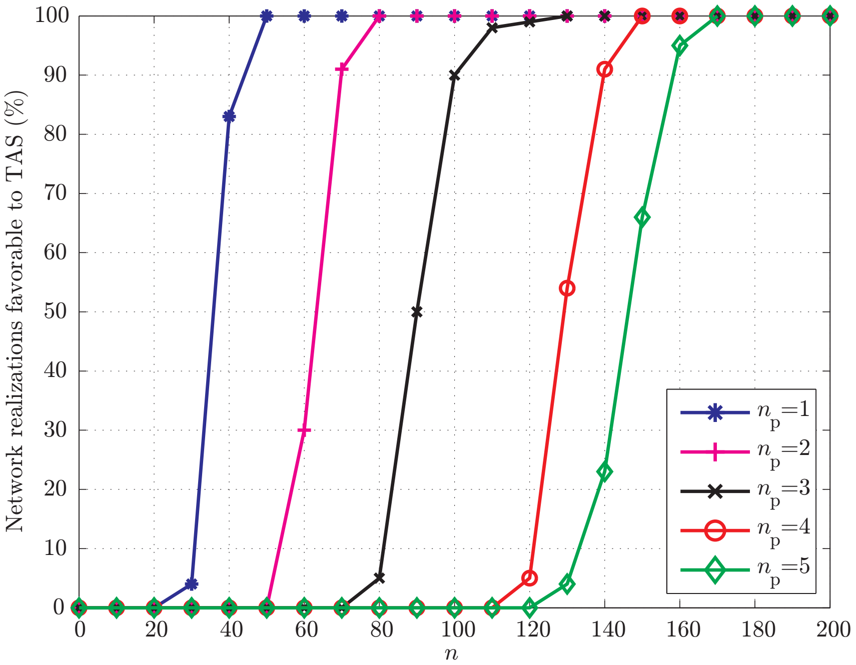


Figure 2.8: Percentage of network realizations favorable to TAS, in terms of required data exchanges, compared to FL, as a function of the number of nodes forming a random tree topology for different values of n_p. 100 random tree realizations are considered for each value of n.

topologies, such as random trees and clustered networks, there is an advantage in employing the TAS algorithm when the network dimension is sufficiently large. On unstructured networks of comparable size, the FL produces the best results, but, in any case, the absolute amount of data transmitted by each node is much larger than in structured networks. This suggests the adoption of structured networks, together with the TAS algorithm for the distributed computation of confidence regions, when the network traffic load for data diffusion is particularly critical.

2.6 Experimental results

This section describes the practical implementation of both TAS and FL on the commercial sensor nodes $EMB-Z2530PA$ [65] deployed in a real scenario. This implementation allows to account for the impact of the MAC layer.

2.6.1 Devices Characterization

Embit sensor nodes $EMB-Z2530PA$ were used to investigate the performance of the above-described algorithms in actual scenarios. $EMB-Z2530PA$ incorporates a temperature sensor, an

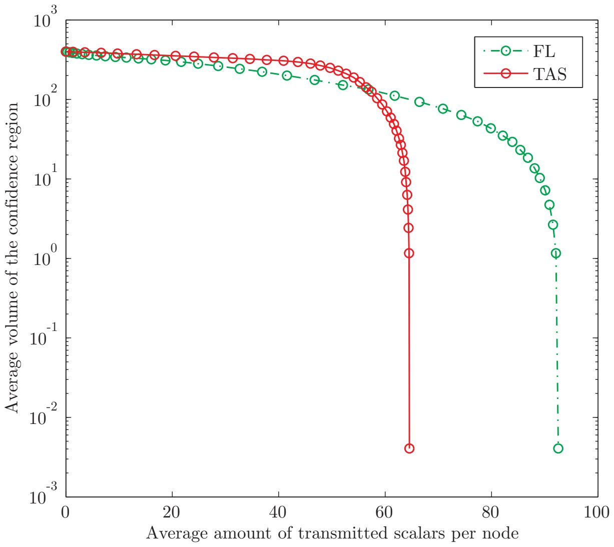


Figure 2.9: Average volume, across nodes and 100 random tree realizations, of the 90% confidence region. Simulation parameters are set to $n = 100, n_{\mathrm{p}} = 2, q = 1$, and $m = 10$.

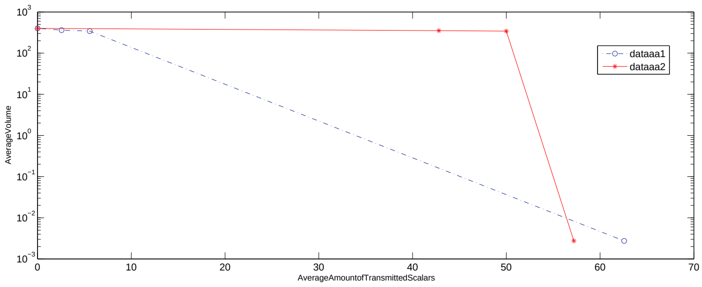


Figure 2.10: Average volume, across nodes and 100 clustered network realizations, of the 90% confidence region. Simulation parameters are set to $n_{\mathrm{p}} = 2, q = 1, n_c = 20$, and $m = 10$.

IEEE 802.15.4 ZigBee communication device and a Texas Instruments $CC2530$ microcontroller (with 8 Kbyte of RAM and 256 Kbyte Flash memory) that controls all operations. It combines high performance, small dimensions and low cost. The block diagram of the $EMB - Z2530PA$ is shown in Figure 2.12 [65]. The RF front end includes a power amplifier that allows an output power up to 20 dBm. Along with the low noise amplifier of the receiving section, that allows a sensitivity of -105 dBm, it provides a coverage distance up to 500 meters in line of sight.

The radio module is controlled by the RF core, which is composed by a modulator, a demodulator, a finite state machine (that controls the transceiver state and most of the dynamically controlled analog signals), an automatic gain control (to adjust the gain of the low noise amplifier), a frame filtering and source matching, a frequency synthesizer (that generates the carrier signal), a command strobe processor (that process all commands issued by the CPU), and a RAM

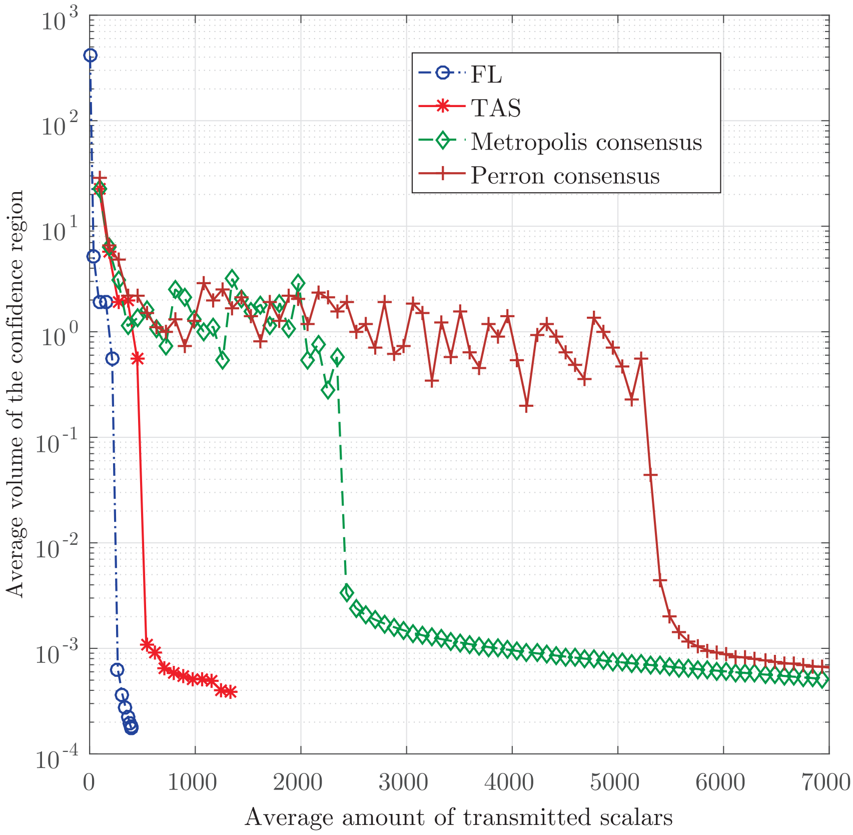


Figure 2.11: Average volume, across nodes, of the 90% confidence region. A random unstructured network of 100 nodes is considered.

memory. According to the IEEE 802.15.4 standard, carrier frequencies from 2394 MHz to 2507 MHz are supported. IEEE 802.15.4 specifies 16 channels, 5 MHz apart, within the 2.4 GHz band, numbered from 11 to 26. The center frequency of a channel is given by [66] $f = 2405 + 5\,(k - 11)$ MHz with $k = 11 \ldots 26$. The modulated signal is an offset-quadrature phase shift keying (O-QPSK) with half-sine chip shaping, in which each chip is shaped as a half-sine, transmitted alternately in phase and in quadrature channels with one-half chip period offset. The transmission register can hold 128 bytes. The carrier sense multiple access with collision avoidance (CSMA/CA) protocol is used to share the access to the medium. A node listens to the channel before transmission to determine whether someone else is transmitting or not. If the channel is idle the node can transmits, otherwise a back-off algorithm starts. The back-off time is randomly chosen in a range between zero and $W - 1$, where W is the contention window length. After each unsuccessful transmission the back-off windows size is doubled up to a maximum value; once the back-off window size reaches its maximum value it will stay at that value until it is reset after a given number of transmission attempts. The Embit devices are easily programmable through the CC debugger provided by Texas Instruments (Figure 2.13). They can

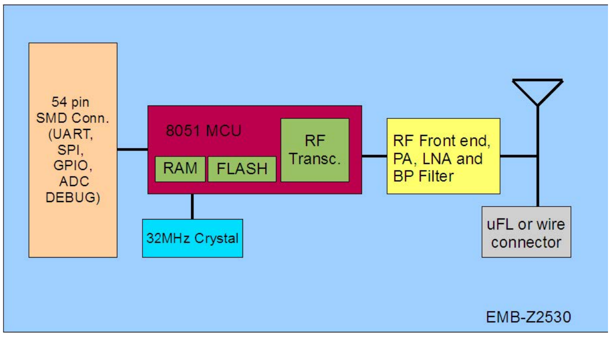

54 pin SMD Conn. (UART, SPI, GPIO, ADC DEBUG)
8051 MCU
RAM
FLASH
RF Transc.
RF Front end, PA, LNA and BP Filter
32MHz Crystal
uFL or wire connector
EMB-Z2530

Figure 2.12: Block diagram for the $EMB - Z2530PA$.

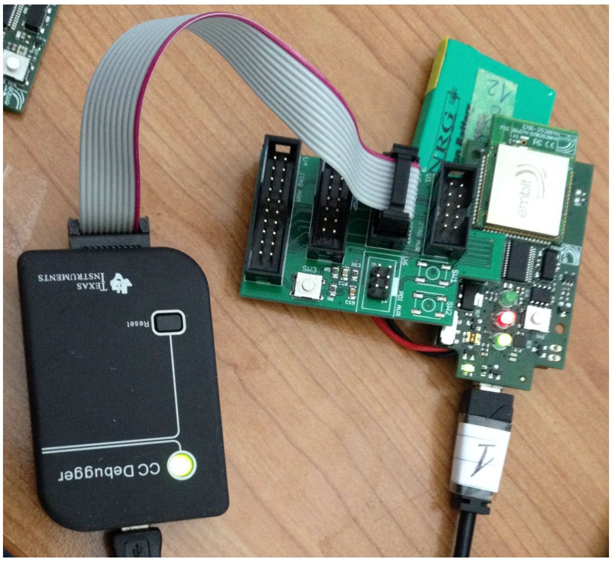


Figure 2.13: Representation $EMB - Z2530PA$ sensor with debugger.

be used in many applications, such as building automation, metering, industrial automation and healthcare.

2.6.2 Data Packets

Figure 2.14 shows the schematic view of IEEE 802.15.4 frame format. The synchronization header (SHR) consists of a preamble sequence followed by the start of frame delimiter (SFD), while the PHY header consist only of the frame length field. The frame control field, data sequence number and address information follow the frame length field, while at the end of the MPDU the frame check sequence is calculated over the MPDU following a polynomial definition. The transmission register can hold 128 bytes and, as we can see in Figure 2.14, considering that the 16 bytes address information, the maximum achievable payload is 102 bytes. Temperature measurements are carried out in our experimental setup in order to consider a real application.

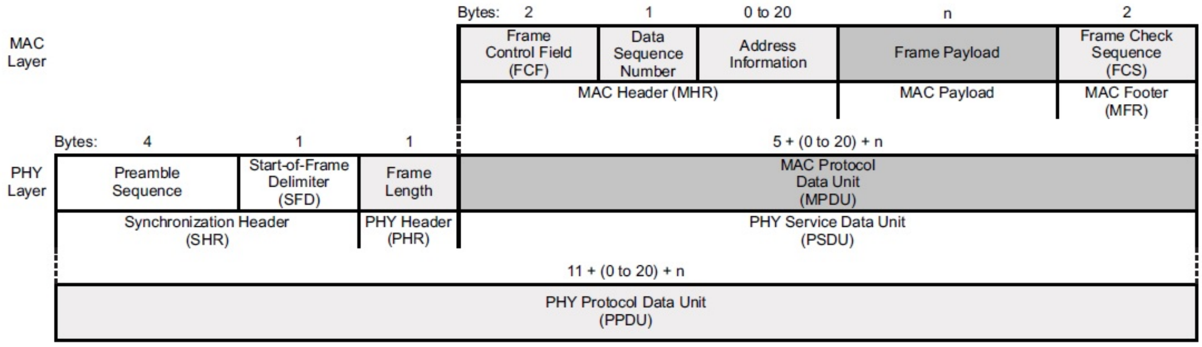


Figure 2.14: Schematic view of the IEEE 802.15.4 frame format.

The measured value is represented with 2 bytes.

FL data packet The FL algorithm generates at runtime a table of information available at each node. An information already transmitted by a node is never transmitted again by the same node. The data are stored with their respective tags, that represent the correspondent source node. The tag length depends on the number of sensors within the network (with a byte is possible to have up to 255 sensors). During a transmission round, the amount of new data (N_{Info}) with the relative tags (N_{TAGFL}) are transmitted, so at each round the number of transmitted bytes (N_{TxFL}) with $n_{\mathrm{p}} = 1$ results

$$N_{TxFL} = N_{Info} \left(2 + N_{TAGFL} \right). \tag{2.42}$$

TAS data packet The TAS aggregates the information to reduce the number of transmitted bytes. Also the TAS algorithm creates at runtime a table of information available at each node and an information already transmitted by a node is never transmitted again by the same node. At each transmission round, the TAS algorithm derives a partial sum with the new information contained in the table. Differently from the FL algorithm, the tag is created by setting ones in the position corresponding to the sensor index; with a byte is thus possible to cover up to 8 sensors. At each round, the amount of new data (N_{Info}) with the relative tags (N_{TAGTAS}) are aggregated and then transmitted, so at each round the number of transmitted bytes (N_{TxTAS}) with $n_{\mathrm{p}} = 1$ is

$$N_{TxTAS} = 2 + N_{TAGTAS}. \tag{2.43}$$

2.6.3 Preliminary Results

In this paragraph I evaluate the performance of the FL and TAS algorithms in a simple networks with 15 or 20 nodes. I considered a flat or a tree network topology.

1. Setting the same transmission time at each sensor

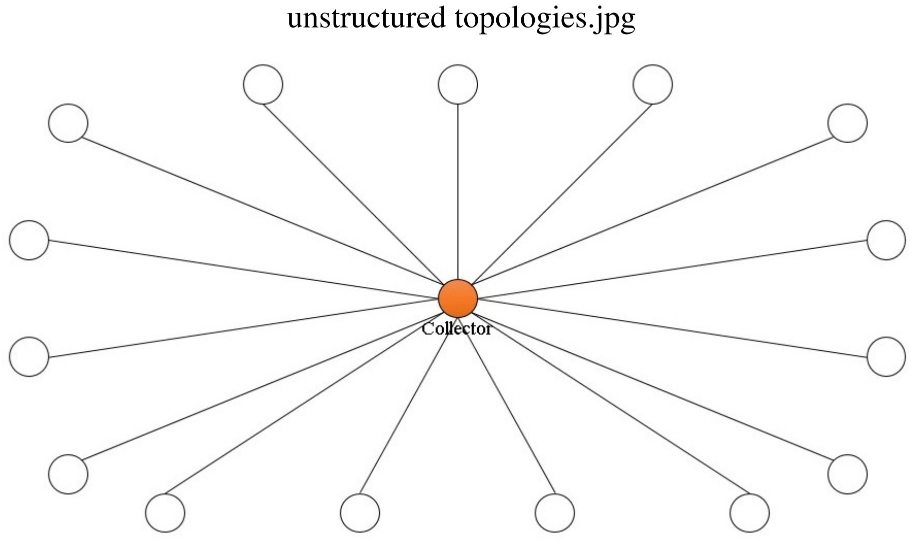


Figure 2.15: Representation of the collector rule in a random unstructured network.

In the first trial, the same transmission time was set in each sensor to investigate what happens if all sensors try to transmit at the same time in a small area.

Flat Topology

I set up a network, formed by n sensor nodes, structured on a single level of hierarchy. The network is fully connected so that each node can directly communicates with the others. One of these acts as a coordinator, that starts the information diffusion algorithm (either FL or TAS) and, at the end, collects the amount of packets transmitted/received by all nodes (Figure 2.15). In particular, the coordinator or collector sends a “start” packet to trigger either TAS or FL algorithm in each sensor for a finite number of rounds n_r. In the first round each sensor measures and transmits its data, whereas during the other rounds the sensors receive the information from the other nodes and execute the proper information diffusion algorithm (TAS or FL) before transmitting the next packet. As a first step, I evaluated the impact of the back-off algorithm with $n_\mathrm{r} = 10$. The behavior of the back-off algorithm has been controlled changing the maximum number of transmission retries, that ranges between 0 and 5, the initial length of the contention window $W_{min} = 2^{BE_{min}}$, with $BE_{min} \in [0, BE_{max}]$, and the maximum contention window size $W_{max} = 2^{BE_{max}}$, with $BE_{max} \in [3, 8]$. At the completion of the n_r rounds, the total amount of data received by each node is counted and transmitted to the collector. The procedure is repeated hundred times in order to derive the average metrics.

Figure 2.16 shows the percentage of information received by the nodes varying the back-off parameters with FL algorithm. I set the maximum number of transmission retries equal to

5; $BE_{min} = 0$, 3 or 4, and $BE_{max} = 8$. As we can see, collisions highly affect the algorithm performance. In fact, even in the case with maximum number of transmission retries (5), $BE_{min} = 4$ and $BE_{max} = 8$, only 22% of the sensors received all data from their neighbors, thus showing that collisions are still significant. To further investigate this phenomenon, I derived the FL performance as a function of n, in the same condition previously assumed. As can be observed in Figure 2.17, FL algorithm works perfectly when $n < 8$, then number of collisions increases, deeply affecting the system performance. For larger values of n the amount of lost packet increases significantly. Also in TAS algorithm, collisions affect

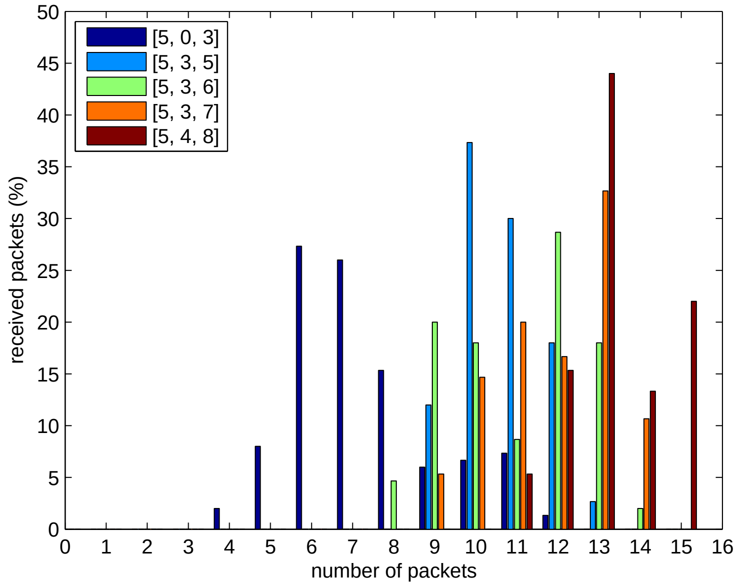


Figure 2.16: Percentage of information packets received. Experimental results of FL algorithm in unstructured topology of 15 nodes, varying the back-off parameters.

the results, but the use of aggregation of information received and the sends of a message with a predefined length at each round, the maximum number of bytes transmitted by each sensor is equal to the lengths of the TAS packet multiplied by the number of rounds. At each round the total number of information bytes sent is

$$Info_{sent} = Packet_{length} \cdot n = 4 \cdot 15 = 60. \tag{2.44}$$

Taking into account that the number of rounds is equal to ten, the total number of information bytes sent by the TAS algorithm will be equal to 600 bytes while if the FL algorithm performance is optimal, the total number of information bytes sent will be equal to 450.

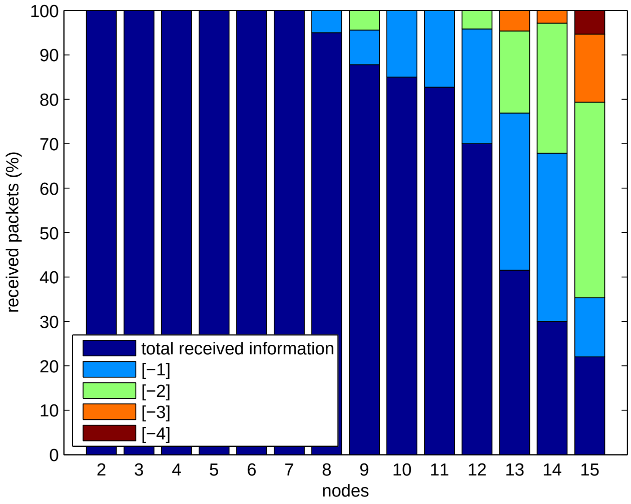


Figure 2.17: Percentage of received information packets. Experimental results of FL algorithm in unstructured topology varying the number of devices.

Comparing this results, it is possible to see that FL algorithm performs better than TAS in random unstructured network.

Tree Topology

Consider a predefined tree topology, a tree where each node has a number of sons defined a priori in order to better understand how the algorithms work in a tree. The number of nodes forming the network is considered equal to n and the levels in the tree are indicated by l, with $l \geq 0$. After the initialization the sensors will be connected to each other following the tree defined a priori, only at this point the algorithm will start. The information will be broadcast and after a reception filter the right father will receive the information. The number of rounds is kept at 10 to avoid problems to the execution of the algorithm, but using a tree with 3 or 4 levels, 2 and 3 rounds are enough for the execution of the system. At each round sensors send the information to their father until the information will reach the coordinator at level one. Some configuration with three or four levels have been created to investigate how the FL algorithm performs in a tree topology. In Figure 2.18 are shown four different tree with three levels, while in Figure 2.19 are shown four different tree with four levels. In the figures, sensors are represent by circles and the number written inside indicates the sensor index. The percentage written near each sensor indicates the percentage of information received with respect to the maximum possible amount of information that each sensor can collect since that the information collected by the collector is not sent back from the collector to the last level.

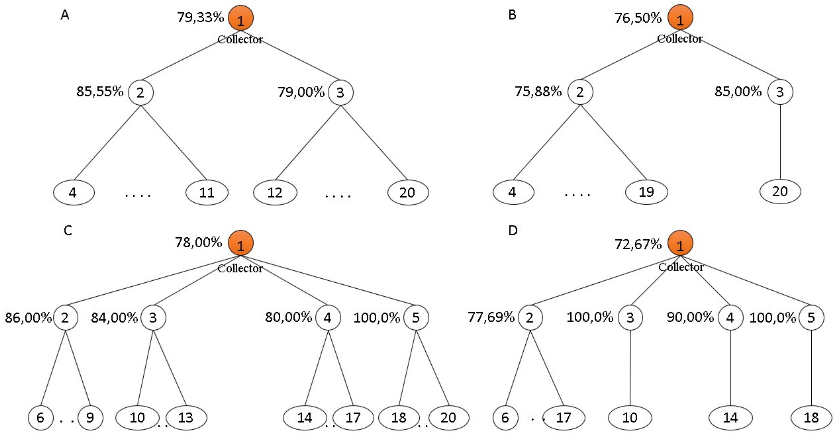


Figure 2.18: Examples of experimental results over tree topologies with three levels using FL algorithm.

As we can see from Figure 2.18 and Figure 2.19, the more the sensors in the tree are equally distributed, the more the effects of the collisions are masked and the collector can know more information about the topology. Also here, the impact of the back-off algorithm affects the performance of the FL algorithm but with the possibility to transfer information one level at time the performance of the system increases and the collector can know more information with respect to a random unstructured topology.

Results for different trees with three and four levels that evaluate the performance of TAS algorithm are shown in Figure 2.20 and Figure 2.21 respectively. As we can see the percentage of packets lost is greater compared to those seen with FL algorithm, this underlines the need of TAS algorithm to send more information than FL to obtain better performance and the fact that in a small tree FL performs better than TAS in terms of transmission byte (Figure 2.8). Obviously, very small trees are not usual in real implementations.

2. Without setting the same transmission time at each sensor

Subsequently, the transmission time was set differently in each sensor to consider a real implementation where sensors start to transmit information only if they have some information to send to their neighbors. Flat and tree topologies are investigated to evaluate the impact of the MAC protocol to the applications. The algorithms are performed 100 times to consolidate the results with 10 number of rounds varying the duration. The MAC parameters are set to standard value as number of retry equal to 5, $BE_{min} = 3$ and $BE_{max} = 5$.

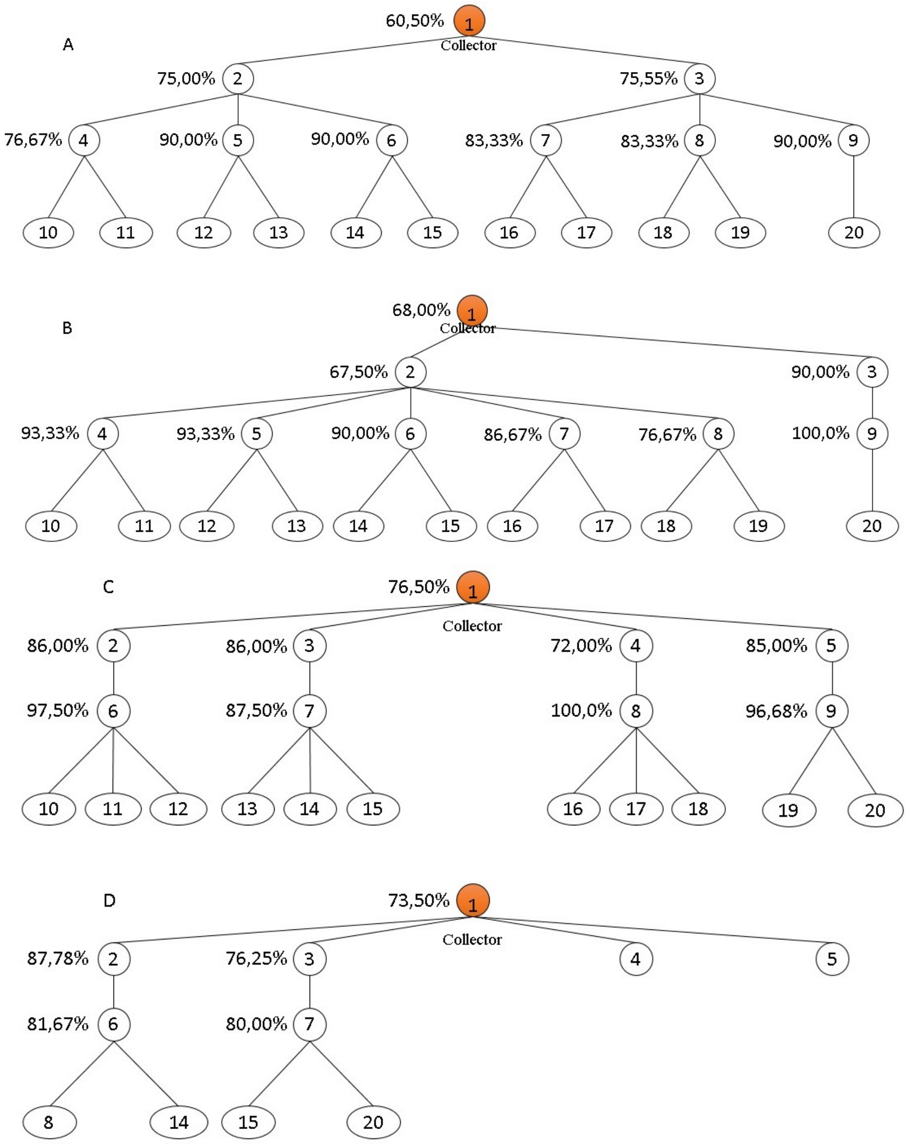


Figure 2.19: Examples of experimental results over tree topologies with four levels using FL algorithm.

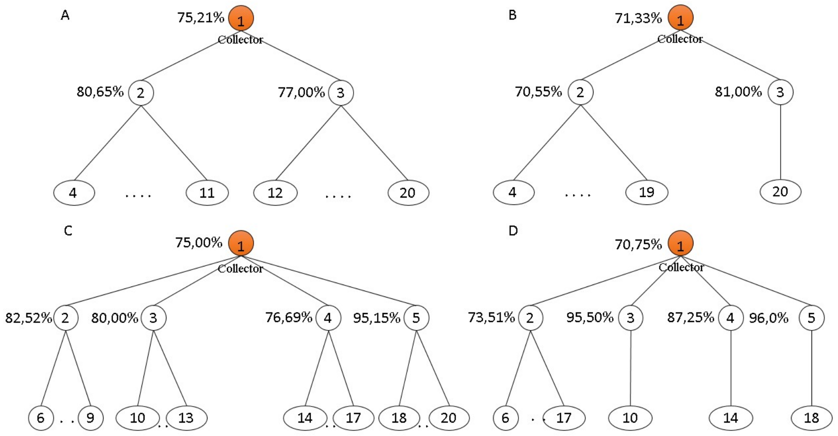


Figure 2.20: Examples of experimental results over tree topologies with three levels using TAS algorithm.

Flat Topology

As before, I consider a network formed by $n = 15$ sensor nodes, structured on a single level of hierarchy. The network is fully connected so that each node can directly communicate with the others, one of these acts as a coordinator. To investigate if something changes with respect to what previously seen, I derided the FL performance in this new implementation varying the duration of the rounds. As can be observed in Figure 2.22 the FL algorithm works perfectly even with 15 nodes until the duration of a round is greater than 0.5 seconds and the number of collisions increases, decreasing the duration, deeply affecting the system performance. As before, it is possible also to see the percentage of sensors that have received fewer packets respect the total number.

Also in TAS algorithm, collisions affect the results if the duration of the round is less than 0.5 seconds, and also now, taking into account that the number of rounds is 10, the total number of information bytes sent by the TAS algorithm will be equal to 600 bytes while if the FL algorithm performance is satisfactory, the total number of information bytes sent will be equal to 450. Comparing this results, it is possible to see that the FL algorithm performs better than the TAS algorithm in a random unstructured network.

Tree Topolgy

Now we can consider a random tree topology, a tree where each node has a number of children that can vary. The number of nodes forming the network is considered equal to $n = 20$ and the levels in the tree are indicated by l, with $l \geq 0$. After the initialization the sensors will be connected to each other starting from the coordinator at the first up

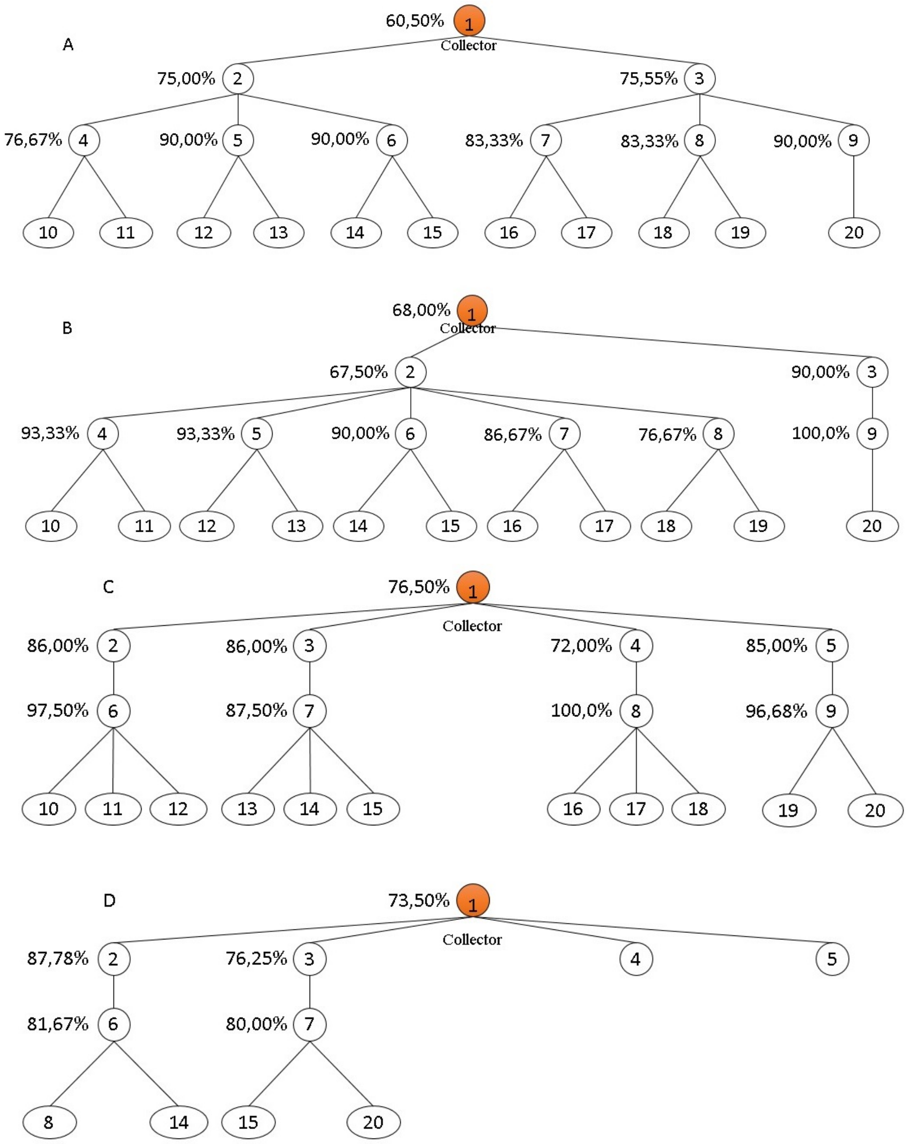


Figure 2.21: Examples of experimental results over tree topologies with four levels using TAS algorithm.

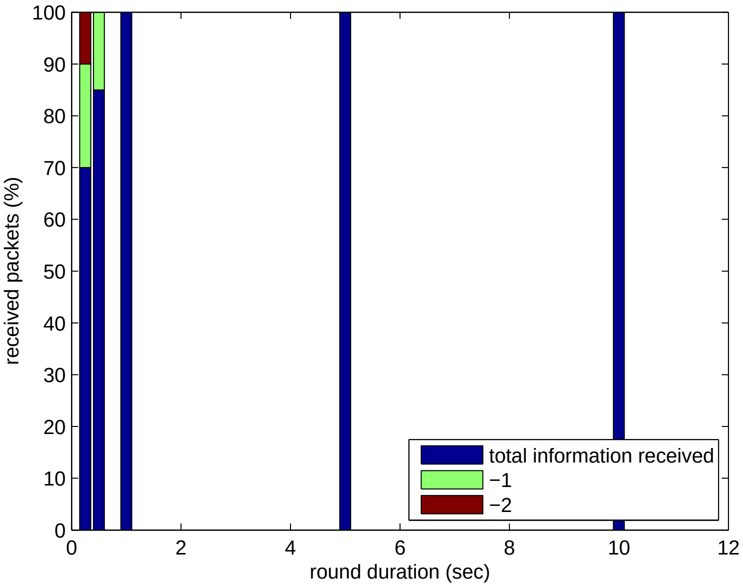


Figure 2.22: Percentage of received information as a function of round duration. Experimental results of FL algorithm in unstructured topology varying the round duration.

to the last level. Each node will ask for some children and only the first sensors that will send the request of aggregation will be accepted, at this point the algorithm will start. The information will be broadcast and after a reception filter the right father will receive the information. The information will be exchanged starting from the lower level up to the first (coordinator) and they will be forwarded inversely from the coordinator up to the leaves in order to reach all the nodes. In Figure 2.23 I show the average amount of information needed to evaluate the confidence region implemented in Matlab simulation. Figure 2.24 shows the cumulative distribution function of the transmitted information in byte of FL and TAS algorithms over different trees of 20 nodes. Result are shown with a round duration of 0.25 and 1 second. This figure shows in what extent the MAC layer impacts the system and results with 0.25 seconds of duration are smaller than the other because not all the information related to the other sensor has been received and then transmitted. Comparing Figure 2.23 and Figure 2.24, we can observe that TAS algorithm need to send more information with respect to the FL algorithm to obtain better performance and in a small tree FL performs better than TAS in terms of transmission byte as demonstrated and evaluate with Matlab in Section 2.5.

3. Evaluation in an actual scenario

The performance of FL and TAS algorithms has also been tested in a complex actual

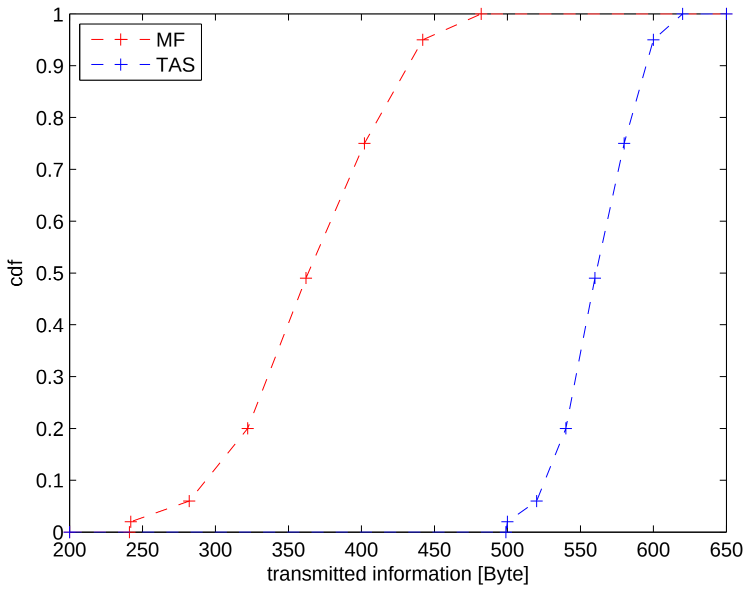


Figure 2.23: Cumulative distribution function versus the average number of transmission information needed for the evaluation of distributed non-asymptotic confidence region computation of FL and TAS algorithms simulated with matlab in different trees of 20 nodes.

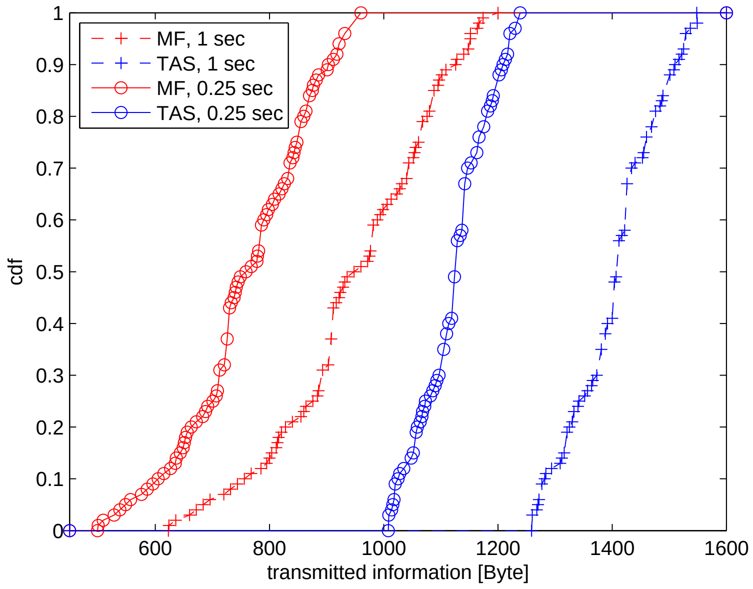


Figure 2.24: Cumulative distribution function versus the number of transmitted information. Comparison of the experimental results of FL and TAS algorithms in different trees of 20 nodes with a round time equal to 1 and 0.25 seconds.

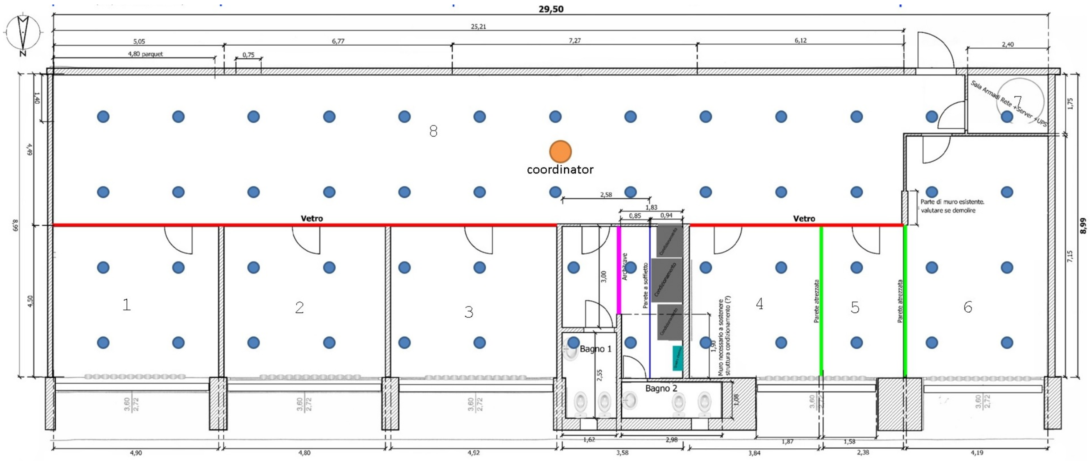


Figure 2.25: Sensors positions at WiLab laboratory.

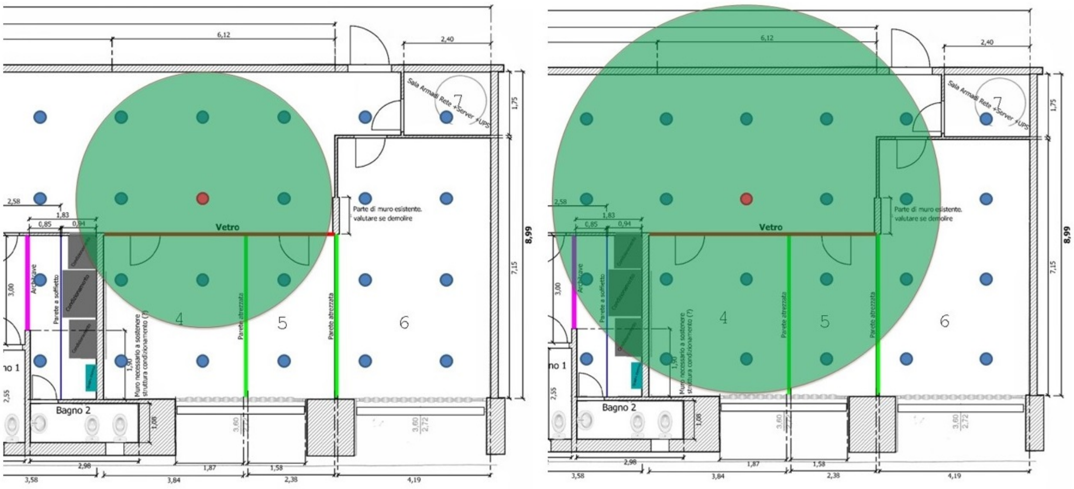


Figure 2.26: Impact of transmission range on the number of neighbors.

scenario. In particular, 52 nodes were placed in a $29.50 \cdot 9m^2$ small office area with furniture at the WiLab laboratory (Figure 2.25). Differently from the previous cases, where all sensors can reach each other, the communication range has been changed (properly setting the transmission power), in order to have a coverage in the order of 3 meters (low power) and 5 meters (high power). Adjacent nodes in the horizontal and vertical directions were placed at a distance of $2m$. The coordinator, working with the maximum allowable transmission power (20 dBm) in order to reach all the sensors, was placed in a central position. In Figure 2.25 the scenario layout is depicted, with the coordinator represented by an orange circle and the sensor nodes represented by blue circles.

In the first test, an unstructured network has been considered, with $n_\mathrm{r} = 20$ and $t_\mathrm{r} = 1s$.

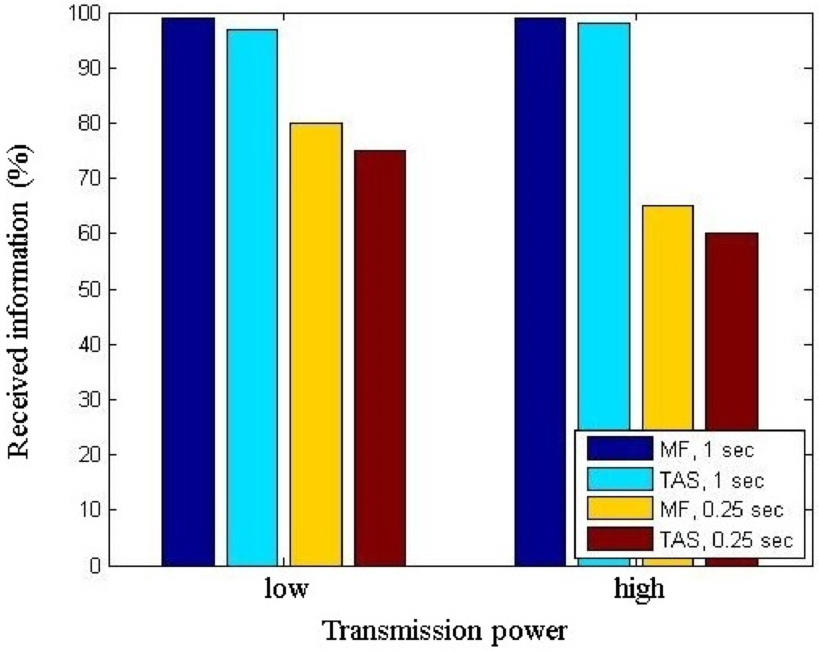


Figure 2.27: Experimental results of the average percentage of information received as a function of transmission range setting t_r at 1 and 0.25 seconds.

The performance of FL and TAS has been evaluated in both cases of low and high power conditions. Of course, different transmission ranges change the number of neighbors of each node and impact the measured performance. Figure 2.26 shows that with a transmission range in the order of 3 meters the number of neighbors is 8, on average, whereas this values increases to 17 with a coverage range in the order of 5 meters. Figure 2.27 shows the average percentage of information received as a function of transmission range setting different value of round duration. As we can see, a reduction of the round duration leads to reduction in the amount of received information. As expected, the small coverage case provides better performance, owing to the reduced number of neighbors and, therefore, of collisions.

Evaluation over random trees are also taken into account by adding to each sensor a random number of children. As we have seen in the evaluation of the FL performance, the first results are taken by setting $n_r = 20$ and $t_r = 1s$. Simulation results are present in Figure 2.28, which shows the cumulative distribution function of the transmitted information in byte of the FL and TAS algorithms over different trees of 52 nodes. Experimental results, instead, are shown in Figure 2.29 with t_r of 0.25 and 1 second, varying the transmission range. Only results with low transmission power are shown because the simulation at high transmission power are very closed to that with $t_r = 1s$. Figure 2.29 shown how the MAC layer impacts

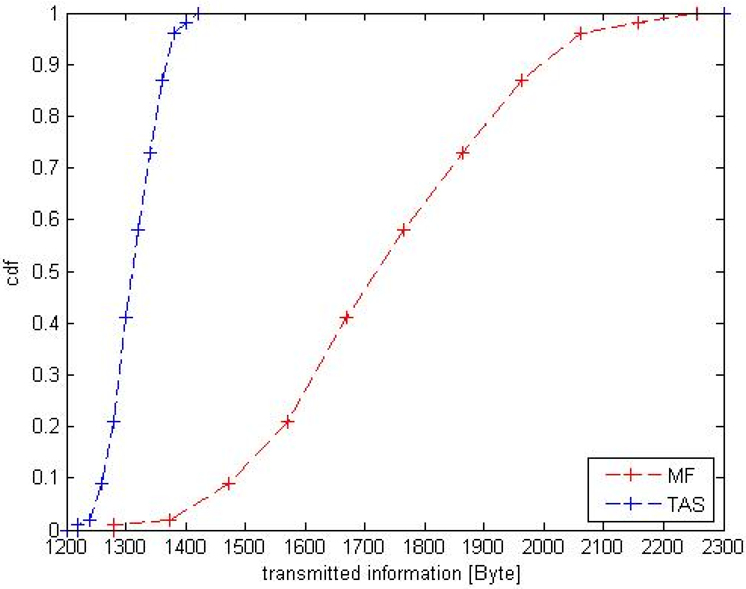


Figure 2.28: Cumulative distribution function versus the average number of transmission information needed for the evaluation of distributed non-asymptotic confidence region computation of FL and TAS algorithms simulated with matlab in different trees of 52 nodes.

the system and in results with 0.25 seconds of duration, collisions occur, and the average number of packets transmitted is smaller than the other because not all the information related to the other sensor has been received. Comparing Figure 2.28 and Figure 2.29, we can observe that TAS needs to send less information respect the FL algorithm to obtain better performance. In a structured topology TAS performs better than FL in terms of transmission byte as theoretically demonstrated in Section 2.4.

2.6.4 Experimental Results in Real Scenario

After some preliminary observations, I Analyzed FL and TAS performance in practical implementations.

1. Experimental setup: Network topologies

   Two network topologies have been considered, namely:

   - **flat network**, where nodes can directly communicate with their neighbors, that is, with the nodes within their coverage region;
   - **random tree**, where the tree structure is randomly established by the nodes themselves at each run.

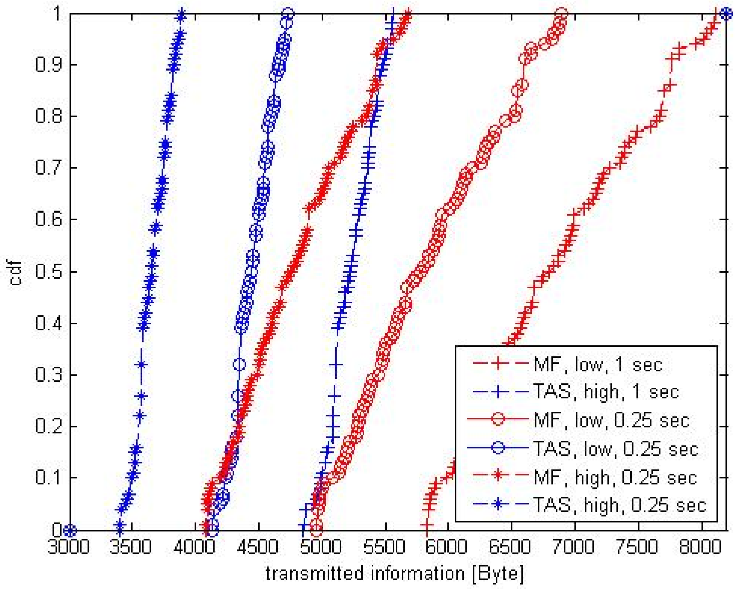


Figure 2.29: Cumulative distribution function versus the total information transmitted from all the sensors. Experimental results in tree topology varying the transmission power and t_{r}.

For the sake of conciseness here I do not present any result concerning clustered networks, as they can be seen as a derivation of generic trees. For flat networks and random trees, the transmission power and the positions of nodes are managed in order to vary their connectivity level. In particular, for each network topology different measurement campaigns were carried out, varying the level of transmit power (the same for all nodes) in order to control the average (over the n nodes of the network) number n_{n} of neighbors of each node.

In the tree topology case, our aim is twofold: 1) to check whether our analytical methodology is able to capture the behavior of TAS and FL in a real network and 2) to compare the actual (that is, measured) performance of TAS and FL.

No comparison between analytical and experimental results will be possible, instead, for flat networks, because their totally arbitrary shape prevents any analytical modeling of the amount of exchanged data. The objective, in this case, is to compare the performance of TAS and FL even when operated with such networks.

2. Experimental setup: Network setup and data management

For both topologies, a network coordinator is introduced for monitoring and network setting purposes without compromising the distributed nature of the algorithms. At the beginning, the coordinator sends a *start* message that triggers the network setup (in the tree topology case), and the information diffusion algorithm, either FL or TAS.

For the tree topology, the tree construction starts from the root (level 0), which randomly selects the number n_{ch} of its children with uniform (discrete) distribution in $[1, 2, ..., n_{\text{max}}]$. Provided that a sufficient number of nodes is available within the coverage range of the root, n_{ch} of them are selected as its children. Otherwise, all (thus less than n_{ch}) available nodes are joined to level 1. The same procedure is repeated by each node of level 1 and then iterated for all levels, until all nodes join the tree. Once the network has been established, the information diffusion algorithm, either FL or TAS, is started, beginning from the leaves up to the root and then in the opposite direction. In our experimental setup the information transmitted by a node to its father is not overheard by its children.

In the flat network case, instead, no network-setup phase is needed. Hence either the FL or TAS execution is triggered as soon as the *start* packet is received.

For FL, each payload contains the amount of data transmitted, measurements and regressors, and a unique tag vector that identifies the contributing nodes. For TAS, payloads contain partial aggregated sums and a tag vector indicating the contributing nodes. In the proposed implementation, the tag consists of a vector of d_{TAG} bits, with 1s at the positions corresponding to the indexes of the contributing nodes. Since the same tag is used for TAS and FL algorithms, the difference in the transmission cost only depends on the amount of data transmitted.

3. Experimental setup: Time axis management

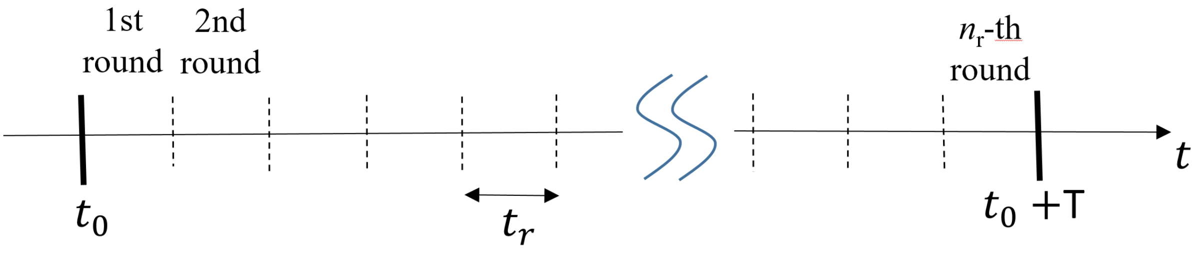


Figure 2.30: Time scheduling management

At the beginning of each measurement period of duration T, the coordinator awakens all nodes, initiating the network activity. Whatever the network topology, the measurement period is divided into n_{r} rounds of equal duration $t_{\text{r}} = \frac{T}{n_{\text{r}}}$ (Figure 2.30).

In the flat network case each node performs a measurement and, during each round, attempts to transmit its data (measurements and corresponding regressors for FL, or aggregated sums for TAS) and the data it has received from neighbours and not yet transmitted.

In the tree network case, instead, nodes are allowed to transmit only during the round pertaining to the level they belong to. Data (measurements and corresponding regressors for FL or aggregated sums for TAS) are then exchanged beginning from the leaves up to the root and then in the opposite direction.

To emulate the time jitter in nodes operations caused by local clocks drift in a distributed network as well as to prevent all nodes from simultaneously accessing the channel, thus congesting the medium access control (MAC), each node defers the measurement phase, and therefore also the beginning of the information diffusion algorithm, by locally choosing a random delay $\Delta_i \in [0, t_\mathrm{r}]$, with $i = 1, 2, ..., n$.

All nodes stop data dissemination once n_r rounds have been completed. The coordinator collects then the amount of data transmitted/received by each node to allow an analysis of the behavior of the TAS and FL algorithms.

4. Results

A network of $n = 52$ nodes equipped with temperature sensors has been considered. The transmission power and the position of each node are chosen so that each node has an average number of neighbours n_n ranging from 2 to 33.

Simple temperature measurements are performed. The temperature θ^* is assumed constant in the area where the nodes are deployed. The corresponding measurement model is $y_i = \varphi_i \theta^* + w_i$, where φ_i is known by each node,[4] and θ^* is the parameter to estimate. Thus $n_\mathrm{p} = 1$ and the data to be transmitted by the FL algorithm are collections of pairs (φ_i, y_i), consisting in this case of $d_\mathrm{FL} = 2$ real values (which may be quantized). For the SPS algorithm, one chooses $m = 10$, and $q = 1$ to be able to characterize 90% confidence regions according to (2.9). Therefore, the amount of data transmitted at each round by TAS is $d_\mathrm{TAS} = 20$ real values (which may also be quantized) and remains constant.

The measurement period is taken as $T = 2$ s. n_r ranges from 2 to 30 and therefore t_r varies from 1 s down to 67 ms. The parameters adopted for our experimental campaign are summarized in Table 2.5.

Given the network topology (either generic tree or flat network) and for each chosen setup (transmit power, n_r), I performed the measurement campaign over 100 network realizations and I derived the average (over the 100 resulting networks) amount of information received by each node and the average amount of data transmitted in the whole network.

[4] Here, for simplicity, I choose $\varphi_i = 1 \; \forall i$. However, this choice does not affect the outcomes of our investigations. With a larger number of sensors it would be possible to estimate also spatial variations of the temperature, but the simple example here considered is enough for the purpose of this paper.

Parameter	Symbol	value
Number of nodes	n	52
Maximun number of children (tree topology only)	$n_{\max}$	5
Measurement period	T	2 s
Number of rounds	n_{r}	$\{2, 3, \ldots, 30\}$
Number of neighbours	n_{n}	$\{2, 4, 8, 17, 33\}$
Number of parameters to be estimated	n_{p}	1
Number of sign perturbed sums	m	10
Size of data sets with FL	d_{F}	2 Bytes
Size of data sets with TAS	d_{TAS}	20 Bytes
Size of the tag vector (both TAS and FL)	d_{TAG}	7 Bytes

Table 2.5: Parameters of the experimental setup

(a) Flat network

Figure 2.31 shows the average proportion (expressed in percentage) of data reaching a given node in a flat topology for various n_{r} and n_{n}.

The value of n_{r} that maximizes the average amount of received data depends on n_{n}. For low values of n_{r}, the performance is limited by the constraint on the maximum number of allowed hops (that is coincident with n_{r}), that might not be sufficient for a particular data to reach all nodes in the network, especially for low degrees of connectivity n_{n}. On the contrary, for large values of n_{r} the performance is limited by the MAC, as a small t_{r} increases the collision probability.

From the same figure one can also see that better performance is obtained when the network is characterized by a low degree of connectivity n_{n} provided that a sufficiently high number of rounds can be allocated within the measurement period. In fact, large n_{n}, *i.e.*, high power levels, generate more interference among nodes that leads the MAC to collapse. This suggests that a proper power control strategy able to keep n_{n} at minimum values to keep connectivity is beneficial both for network performance as well as to save energy.

FL and TAS perform similarly in all conditions, hence they are equivalent considering

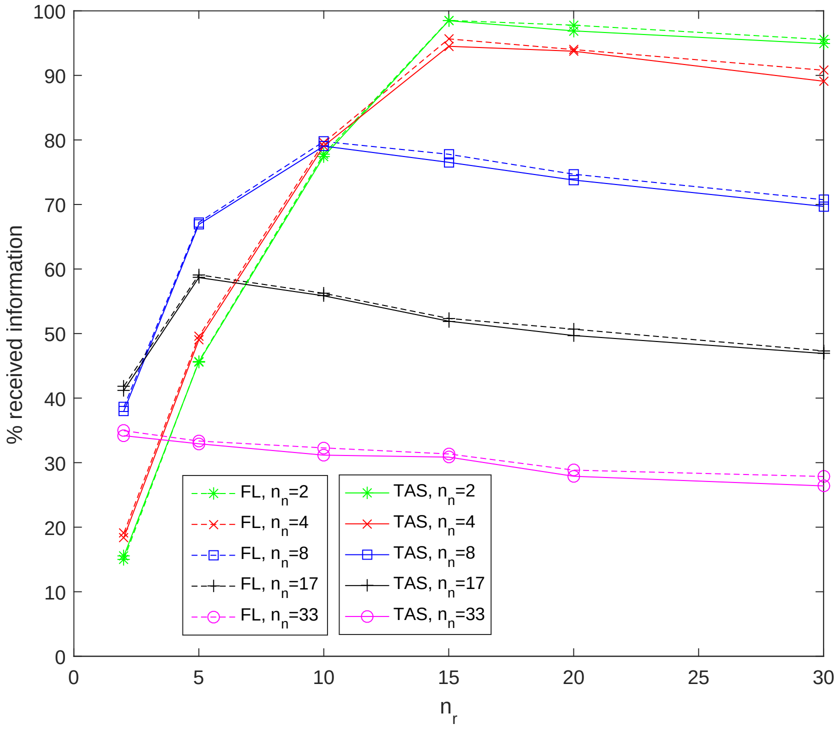


Figure 2.31: Flat network: average proportion of the total information received by nodes as a function of n_r for various n_n. The legend entries and the curves in the right-hand part of the figure are in the same order.

only the amount of received information. They differ, instead, in terms of amount of transmitted information, as seen in Table 2.6, which reports the average amount (over 100 network realizations) of transmitted data (scalars) within the whole network in the case $n_\mathrm{r} = 15$.

When operated in a flat topology, FL outperforms TAS as it requires a lower amount of transmitted information. With such topology, in fact, the information efficiently diffuses within the network, up to the maximum extent permitted by the transmission power and without back and forth paths (that occur, instead, in the tree topology), hence the aggregation carried out by TAS is not sufficient to compensate for the larger value of d_TAS with respect to d_FL.

(b) Generic tree topology

Figure 2.32 provides for the tree topology the average proportion (expressed in percentage) of data reaching a given node as a function of n_r for various n_n. Here it can be noticed a limited sensitivity of the optimum value of n_r to n_n, as the average

Neighbors	FL Experimental	TAS Experimental
2	5236	14337
4	5171	12770
8	4197	8770
17	2832	4860
33	1705	2561

Table 2.6: Flat network: Average amount of transmitted data (scalars) within the whole network in the case $n_r = 15$.

number of children of each node only slightly depends on the connectivity degree. In fact, for the tree topology in this example I upper bounded by $n_{\max} = 5$ the number of children of each node to avoid the generation of 'fat' trees. Therefore, for a given node only a fraction of its neighbors are actually involved in data diffusion. As a consequence, increasing the number of neighbors n_n does not increase the amount of information diffused, but determines higher levels of interference and packet collisions. This makes power control less critical in tree topologies with respect to flat topologies. In general, better data dissemination is observed when n_r is large compared to flat topologies since transmissions happen level by level and only a small part of the network tries to access the channel at the same time. On the contrary, with small values of n_r, data disseminates only to a limited part of the network due to the depth of the tree which may be larger than $1 + n_r/2$. In fact, With n_r rounds the maximum number of levels of a tree that allows a complete dissemination of data from the leaves up to the root and back is $1 + n_r/2$. Similarly to the flat topology, even in this case FL and TAS are very similar in terms of amount of received data. Table 2.7 reports the average amount of transmitted data within the whole network when $n_r = 15$. Now, TAS outperforms FL when operated on a tree topology.

Table 2.7 also compares the analytical outcomes, derived feeding (2.26) and (2.27) with the parameters corresponding to each network realization and averaging over all realizations, and the respective averages of experimental results. When the number of neighbors is small ($n_n = 2, 4, 8$) a good agreement between analysis and measurements is observed both for TAS and FL. The experimental values are always less than the analytical ones because, as can be observed in Figure 2.32, the amount of received information never reaches 100%, even in the considered case of $n_r = 15$. This phenomenon is emphasized as n_n increases ($n_n = 17, 33$), which further reduces the amount of received data (Figure 2.32) and hence the amount of data transmitted by nodes with respect to the ideal (no collisions) situation described by the analysis. We can conclude, therefore, that the analytical framework can be usefully exploited

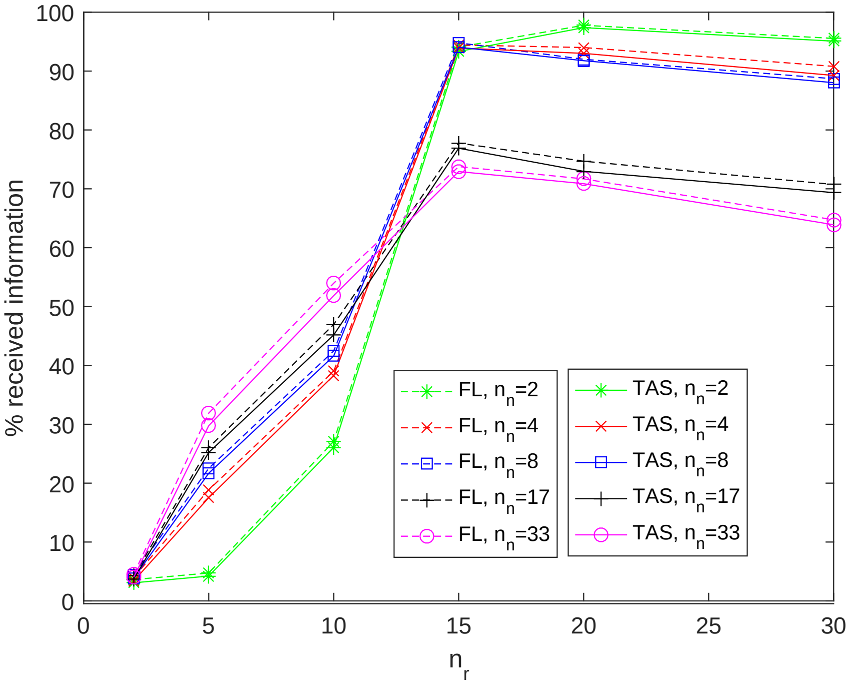


Figure 2.32: Generic tree topology: Average percentage of information received by a node. The legend entries and the curves in the right-hand part of the figure are in the same order.

to provide performance predictions in non congested networks and a performance bound in MAC limited networks.

To evaluate the influence of the proportion of measurements received by each node on the quality of the confidence region that can be derived, a temperature measurement has been performed by each of the n nodes of the network. For different target proportions $\rho \in [0, 1]$ of measurements reaching some node of the network, 100 random selections of a subset of measurements have been considered and a $90\ \%$ confidence region evaluation with $m = 10$ and $q = 1$ has been performed. Figure 2.33 describes the evolution of the average width of the 90 % confidence region as a function of the proportion of measurements collected by a given node. Figure 2.33 (right) shows that the width decreases approximately as $1/\sqrt{\rho n}$, which is consistent with what is observed when maximum-likelihood estimation is carried out assuming an additive Gaussian noise [38], although this hypothesis on the noise is not considered here. From Figures 2.31 or 2.32 and 2.33, one may deduce the width of the confidence interval that may be obtained with FL or TAS, when not all measurements have reached some node. One can for example see that even if only $80\ \%$ of the measurements have reached a node, the width of the confidence region is only $10\ \%$

Neighbors	FL analytic	FL exp.	TAS analytic	TAS exp.
2	2179	2047	1427	1330
4	2144	2022	1420	1331
8	2087	1978	1409	1322
17	1802	1400	1353	1042
33	1705	1256	1334	972

Table 2.7: Generic tree topology: Average proportion of transmitted data reaching a given node as a function of n_{r} for various n_{n}.

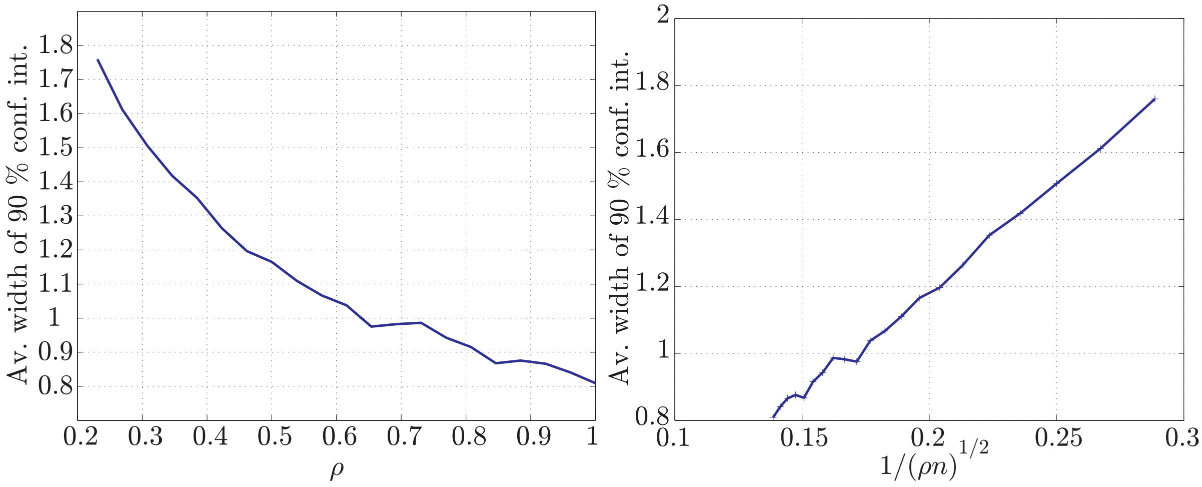


Figure 2.33: Average width of the 90 % confidence region as a function of the proportion ρ (left) of measurements collected by a node with FL or TAS and as a function $1/\sqrt{\rho n}$ (right).

larger than that obtained from all measurements. This means that if one tolerates evaluating a confidence region from a reduced subset of the data, the constraints on the data dissemination duration may be significantly relaxed, with beneficial effects in terms of time and energy savings.

2.7 Conclusions

This work has investigated the distributed evaluation of non-asymptotic confidence regions at each node in a sensor network. I proposed and investigated a novel information diffusion strategy, namely TAS, especially designed for the distributed evaluation of non-asymptotic confidence regions in WSNs with the SPS approach. The TAS algorithm has been designed to efficiently exploit the peculiarities of the distributed evaluation of confidence regions via SPS. In this Chapter I have presented the TAS algorithm and its comparison with other information diffusion algorithms on structured and unstructured topologies and also demonstrating that, even in presence of truncated information diffusion, the level of confidence remains the same as in the centralized non truncated case. Simulation results provide a characterization of the trade-off for the achievable average confidence region volume as a function of the required amount of data that each node should transmit on average. The contributions show that, on structured networks, the proposed TAS algorithm is able to outperform the FL when the network dimension is sufficiently high, this independently of the specific dimension of the parameter space, as corroborated by the theoretical and numerical analyzes, as well as by an experimental setup.

Chapter 3

Information Diffusion Algorithms for Average Consensus Evaluation over WSNs

3.1 Introduction

The state of a spatial distributed physical phenomenon of interest has to be monitored through a network of sensors (nodes). Each node has in general a limited (local) and noisy visibility of the phenomenon but it is interested in obtaining an information about the entire (or partial) state. Measurements taken from different nodes can be in general correlated. If present, a central unit can collect all measurements from nodes, estimate the state of the phenomenon and redistribute the result to all nodes. Unfortunately, for many applications this solution is not sufficiently scalable and robust against failures. Completely distributed solutions, where all nodes become aware of the global state of the physical phenomenon through cooperation without a central processing (distributed estimation), represent a viable and interesting alternative. To this purpose, the intuitive approach of sharing the local measurements among nodes and let each node perform an equivalent central processing is one possibility, but it might not be practicable when the number nodes is large because of the consequent high communication and computational overhead (not scalable). Therefore cooperation approaches exploiting consensus strategies, in which the consensus in reached on the phenomenon global state, are of particular interest. The main advantages are: no routing scheme is necessary, less signalling overhead (for routing), scalability. As will be detailed shortly, possible objectives are:

- let all nodes get an estimate of the phenomenon state (stationary phenomenon)
- let all nodes track the state of a non-stationary phenomenon
- let all nodes get an estimate of the phenomenon statistics (learning)

- let all nodes be aware of the spatial exceedable level map depending of the phenomenon state
- let each node be aware of a different partial version of the global state (e.g., the phenomenon evolution within its surrounding area).

Usually the performance of distributed schemes will be compared with the centralized counterparts used as benchmarks. Potential applications are spectrum sensing - cognitive radio, radio cartography and distributed estimation in general using WSNs.

The state of the observed phenomenon is a sequence of unknown vector variables $x_i = [x_i(1), ..., x_i(M)]^T$, where i is the discrete time index and M is the state space dimension. We can consider three different models for x_i:

1. Constant Signal Vector

   The state vector x_i is constant, i.e., $x_i = \theta$, with $\theta = [\theta_1, ..., \theta_M]^T$.

2. Stationary Signal Vector

   The sequence x_i is composed of independent RVs drawn from the PDF $f(x)$. The PDF might be known or not a priori. A more complex scenario could consider time-correlated RVs.

3. Nonstationary Signal Vector

   The x_i sequence belongs to a non-stationary random process. I restrict the attention to first order Markov chain processes, for which the state transition PDF $f(x_i|x_{i-1})$ is given. The PDF $f(x_i|x_{i-1})$ may be known, partially known, or unknown at node level. As a particular case, I will also consider the Gauss-Markov model under which it is $x_i = Ax_{(i-1)} + Bv_{(i-1)}$. Where A and B are known matrices and v_i is a vector of (independent) Gaussian RVs.

At time i we observe x_i through a set $z_i = \left[z_i^1, ..., z_i^N\right]^T$ of N measurements. The N_{th} measurement z_i^N is taken by the N_{th} node. Therefore, N represents the number of nodes composing the monitoring network. It is possible to distinguish between

- Linear Observations

  In this case each node observes a noisy linear combination of the state $x_i : z_i^k = H_k x_i + r_i^k$ where r_i^k is the measurement noise (in general Gaussian) and independent on x_i, and matrix H_k accounts for the visibility of x_i from node k. In general, for varying topologies, H_k could be time-variant. I will consider H_k to be known to all nodes. A particular case of interest is when $M = N$ and $H_k = (0, 0, ..., 1, ..., 0, 0)$, where the ones is the k_{th} element of the diagonal. In such as case, there is a correspondence between the location of the node and the k_{th} element x_i^k of the phenomenon state x_i.

- Nonlinear Observations

  This is the most general case in which the observation model is specified through the generic global PDF $f(z_i|x_i)$ that is here assumed to be independent of the time index i (time-stationarity). As further simplifying hypothesis, it is possible to consider the current measurement z_i conditionally independent of all past measurements $f(z_i|x_i, z_{i:i-1}) = f(z_i|x_i)$ given the current state x_i, where

$$z_{(1:i-1)} = \left[\left(z_1^T, ..., z_{i-1}^T\right)\right]^T . \tag{3.1}$$

  In general the global PDF $f(z_i|x_i)$ is not available to nodes and the k_{th} node knows only its local PDF $f(z_i^k|x_i^k)$. In case of independent observations it is

$$f(z_i|x_i) = \prod_{k=1}^{N} f(z_i^k|x_i). \tag{3.2}$$

Generally, in this kind of applications, we are in the presence of a slow state evolution if the network speed is much higher than the time interval between two consecutive observations. In other words, in between consecutive observations there is always enough time for the consensus algorithm to converge. On the contrary, we are in the presence of a fast state evolution when the convergence time for the consensus and the temporal dynamics of the monitored phenomenon are comparable. In this case, consensus and innovative schemes must be adopted.

The Consensus among different entities is a fundamental feature of distributed systems, as it is the prerequisite for complex tasks such as distributed coordination of autonomous agents, network synchronization and localization in wireless sensor networks. Consensus schemes represent efficient fully distributed approaches to let nodes in a wireless network agree on a specific quantity without paying the price of the large overhead caused by routing algorithms [67, 68]. In the literature, several average consensus algorithms have been proposed and proved to converge, either asymptotically or in finite time, see *e.g.*, [58, 69, 70]. These schemes are iterative: at each round, each node exchanges with its neighbors a suitable combination of data collected in the previous round until the convergence to the consensus is reached [69].

Since the cost of memory is typically lower than the cost of communication, it is expected that the exploitation of data collected from neighbors during all previous rounds, and not only during the latest round, would increase the degrees of freedom in reaching the consensus, even in a finite time.

In this regard, my work introduces a novel algorithm which is capable of achieving the distributed average consensus in a finite amount of time. The new consensus algorithm, named Finite-Time Consensus with Memory (FTCM), exploits the on-board memory of network nodes.

Moreover, I proposed an adaptation to the distributed average consensus problem of the TAS algorithms, which was introduced in [4] and presented in the previous Chapter for the distributed confidence region evaluation. The performance of both algorithms is compared with that of standard or recently introduced algorithms for distributed average consensus computation.

The algorithms have been investigated in unstructured and structured topologies through simulations. The performance of both algorithms are investigated through simulation and compared with state-of-the-art approaches.

3.2 Problem Formulation and Average Consensus Algorithms

Consider a sensor network with N nodes described by an undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ represents the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Denote with $\mathbf{A} = \left[a_{ij}\right] \in \{0, 1\}^{N \times N}$ the network adjacency matrix, with entries $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and zero otherwise. Assume that at time $t = 0$, each node $i \in \mathcal{V}$ has performed some measurement $x_i(0)$. Let $\mathbf{x}(0) = [x_1(0), x_2(0), \ldots, x_N(0)]^T$ denote the $N \times 1$ vector of all measurements taken by the N sensor nodes. Communications between Node i and Node j are possible only provided that they are linked by a direct edge $(i, j) \in \mathcal{E}$. In this regards, denote by $\mathcal{N}_i$ the set of neighbors of node $i \in V$. We consider that $i \notin \mathcal{N}_i$. Let $x_i(0)$ be a real scalar, which is sampled by Node i a $t = 0$. Since I am dealing with sensor networks, I assume without loss of generality that the initial values $x_i(0)$, $i = 1, 2, \ldots, N$, represents the measurements taken by the nodes. The distributed average consensus problem refers to the situation where all nodes seek to compute the average

$$\overline{x}_N = \frac{1}{N} \left(x_1(0) + x_2(0) + \ldots + x_N(0) \right). \tag{3.3}$$

of the $x_i(0)$s.

In all cases I assume that each node knows the network adjacency matrix $\mathbf{A}$, that means that all nodes are aware of the network topology. Such full knowledge of $\mathbf{A}$ is not strictly needed for all algorithms investigated in the following, as some of them require less information (that nevertheless can be extracted from $\mathbf{A}$). This assumption is well suited to static networks, in which the network discovery phase is rarely executed by nodes. In this case, the corresponding signaling overhead could be negligible with respect to the amount of side information that should be transmitted by the different consensus algorithms when $\mathbf{A}$ is not available.

Unless otherwise specified, we additionally assume that the information content transmitted by a node at each round corresponds to a single scalar (as usual in consensus algorithms), even though the analysis can be easily extended to vector measurements.

3.2.1 Finite Time consensus

The FTC algorithm has been introduced in [70] to allow the distributed computation of (3.3) in a finite amount of rounds.

Each node i at round t holds a scalar $x_i(t)$ initialized at $t = 0$ with its own measurement $x_i(0)$. At each round, each node i transmits $x_i(t)$ to its neighbors and updates its value as follows

$$x_i(t+1) = w_{ii}(t)x_i(t) + \sum_{j\in\mathcal{N}_i} w_{ij}(t)x_j(t), \tag{3.4}$$

where the weights $w_{ij}(t)$ are real scalars that change in time according to

$$\begin{cases} w_{ij}(t) = \beta_t\omega_{ij}, & i \neq j \\ w_{ii}(t) = \alpha_t + \beta_t\omega_{ii}. \end{cases} \tag{3.5}$$

In (3.5) α_t and β_t are properly chosen scalar coefficients depending on the network topology. In [70] it is assumed, moreover, that $\omega_{ij} = \omega_{ji}$ are known and provided as a part of the network topology, and

$$\omega_{ii} = -\sum_{j\in\mathcal{N}_i} \omega_{ij}. \tag{3.6}$$

As proven in [70], (3.3) may be evaluated within a finite amount of iterations, whose minimum is given by $K - 1$, where K is the number of different eigenvalues of

$$\mathbf{\Omega} = \begin{bmatrix} \omega_{11} & \dots & \omega_{1N} \\ \vdots & \vdots & \vdots \\ \omega_{N1} & \dots & \omega_{NN} \end{bmatrix}.$$

A matrix $\mathbf{\Omega}$ that satisfies (3.6) and has positive weights $\omega_{ij} > 0$ for $i \neq j$ can be seen as the negative Laplacian matrix of the underlying graph [70].

With these assumptions, one may rewrite (3.4) as

$$x_i(t+1) = \beta_t \left[\left(\frac{\alpha_t}{\beta_t} + \omega_{ii} \right) x_i(t) + \sum_{j\in\Omega_i} \omega_{ij}x_j(t) \right]. \tag{3.7}$$

Then, introducing $x(t) = [x_1(t), ..., x_N(t)]^T$, (3.7) becomes

$$x_i(t+1) = \beta_t \left[\left(\frac{\alpha_t}{\beta_t} I + \Omega \right) x(t) \right] \rightarrow x_i(T) = \prod_{t=0}^{T-1} \beta_t \left(\frac{\alpha_t}{\beta_t} I + \Omega \right) x(0), \tag{3.8}$$

where I is the $N \times N$ identity matrix.

Consider the coefficients g_t and f_t defined as

$$\begin{aligned} g_t &= \frac{\alpha_t}{\beta_t} \\ f_{t+1} &= \beta_0\beta_1\cdots\beta_t \end{aligned} \tag{3.9}$$

it is possible to write (3.8) as

$$x(T) = f_T \prod_{t=0}^{T-1} (g_t I + \Omega)\, x(0). \tag{3.10}$$

The aim in [70] is to show that the average consensus (3.4) can be computed exactly and in finite time if there exist a non-negative integer T, and coefficients f_T and $g_0, \ldots, g_{T-1}$ that satisfy

$$P = f_T \prod_{t=0}^{T-1} (g_t I + \Omega)\,. \tag{3.11}$$

In order to achieve the distributed average consensus, Theorem 2 in [70] shows that the minimum number of iterations required to compute the distributed average consensus is $K-1$ (with K different eigenvalues of ω). Theorem 3, instead, defines the coefficients required to compute the average consensus operator in $T = K-1$, that are:

$$\begin{aligned} g_t &= -\lambda_{t+2}, \quad 0 \le t \le K-2 \\ f_{K-1} &= \frac{(-1)^{K-1}}{\lambda_2 \cdots \lambda_K}. \end{aligned} \tag{3.12}$$

- Average Consensus: Solutions

  Many solution are possible in order to solve (3.12), in the following, three of them are listed

  1. we know that $f_{t+1} = \beta_0\beta_1\cdots\beta_t$ by definition, so at time $t = K-1$ we get $f_{K-1} = \beta_0\beta_1\cdots\beta_{K-2}$. We know by (3.12) that $f_{K-1} = \frac{(-1)^{K-1}}{\lambda_2\cdots\lambda_K}$ and so we get that $\beta_0\beta_1\cdots\beta_{K-2} = \frac{(-1)^{K-1}}{\lambda_2\cdots\lambda_K}$. The simplest way, to solve (3.12) is to set $\beta_t = 1 and \alpha_t = -\lambda_{t+2} for 0 \le t < K-2$. Since $\beta_0\beta_1\cdots\beta_{K-2} = \frac{(-1)^{K-1}}{\lambda_2\cdots\lambda_K}$ and $\beta_t = 1$ for $0 \le t < K-2$ we get $\beta_{K-2} = \frac{(-1)^{K-1}}{\lambda_2\cdots\lambda_K}$ and then evaluate the last value (α_{K-2}) in order to solve (3.12). We obtain

$$\begin{aligned} \alpha_t &= -\lambda_{t+2} \quad 0 \le t < K-2 \\ \beta_t &= 1 \quad 0 \le t < K-2 \\ \alpha_{K-2} &= g_{K-2}\beta_{K-2} = -\lambda_K\beta_{K-2} \\ \beta_{K-2} &= \frac{(-1)^{K-1}}{\lambda_2\cdots\lambda_K} \end{aligned} \tag{3.13}$$

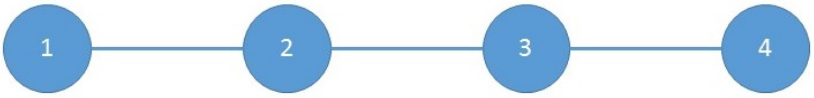


Figure 3.1: Toy Network.

2. we know that $f_{t+1} = \beta_0\beta_1 \cdots \beta_t$ by definition, so at time $t = K-1$ we get $f_{K-1} = \beta_0\beta_1 \cdots \beta_{K-2}$. We know by (3.12) that $f_{K-1} = \frac{(-1)^{K-1}}{\lambda_2 \cdots \lambda_K}$ and so we get that $\beta_0\beta_1 \cdots \beta_{K-2} = \frac{(-1)^{K-1}}{\lambda_2 \cdots \lambda_K}$. This give us the possibility to set all the value of β_t constant and equal to $\beta_t = \sqrt[K-1]{\frac{(-1)^{K-1}}{\lambda_2 \cdots \lambda_K}}$ and select the relative value of α_t by (3.12) and (3.9), obtaining:

$$\begin{aligned} \beta_t &= \sqrt[K-1]{\frac{(-1)^{K-1}}{\lambda_2 \cdots \lambda_K}} \quad 0 \leq t \leq K-2 \\ \alpha_t &= g_t\beta_t = -\lambda_{t+2}\beta_t \quad 0 \leq t \leq K-2 \end{aligned} \tag{3.14}$$

3. It is possible to rewrite $\beta_0\beta_1 \cdots \beta_{K-2} = \frac{(-1)^{K-1}}{\lambda_2 \cdots \lambda_K}$ as $\beta_{K-2} = \frac{(-1)^{K-1}}{\beta_0\lambda_2\beta_1\lambda_3 \cdots \beta_{K-3}\lambda_{K-1}\lambda_K}$. Then, selecting $\beta_t = \frac{1}{\lambda_{t+2}}$ for $0 < t < K-2$ in order to make the multiplications equal to one ($\beta_t\lambda_{t+2} = 1$) and avoid the multiplication between the all eigenvalue that could increase possible mathematical instability. In the last value of β_t must be take into account also the sign ($(-1)^{K-1}$) given by f_t in (3.12).

$$\begin{aligned} \alpha_t &= -1 \quad 0 \leq t < K-2 \\ \beta_t &= \frac{1}{\lambda_{t+2}} \quad 0 \leq t < K-2 \\ \alpha_{K-2} &= g_{K-2}\beta_{K-2} = -\lambda_K\beta_{K-2} \\ \beta_{K-2} &= \frac{(-1)^{K-1}}{\lambda_K} \end{aligned} \tag{3.15}$$

- Toy Network

In order to help the reader understand the FTC, the average consensus has been evaluated in a simple toy network of Figure 3.1. The adjacency matrix of this toy network is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Therefore, Ω results

$$\Omega = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}.$$

Ω has 4 different eigenvalues (K), $\lambda_1 = 0$, $\lambda_2 = -0.5858$, $\lambda_3 = -2$ and $\lambda_4 = -3.4142$, then in [70], is possible to reach the average Consensus in $T = K - 1 = 3$ rounds.

Based on (3.9) and (3.12), we know that

$$\begin{aligned} g_0 &= \frac{\alpha_0}{\beta_0} = 0.5858; \\ g_1 &= \frac{\alpha_1}{\beta_1} = 2; \\ g_2 &= \frac{\alpha_2}{\beta_2} = 3.4142; \\ f_3 &= \beta_0\beta_1\beta_2 = \frac{1}{4}, \end{aligned} \tag{3.16}$$

and, using the first solution (3.13), it is possible to determine the value of the coefficients as

$$\begin{aligned} \beta_0 &= 1 \\ \alpha_0 &= 0.5858 \\ \beta_1 &= 1 \\ \alpha_1 &= 2 \\ \beta_2 &= \frac{1}{4} \\ \alpha_2 &= 0.85 \end{aligned} \tag{3.17}$$

At first round ($t = 0$), the status of each Node i ($x_i(0)$) is equal to its own measure, then we can write

$$\begin{aligned} x_1(0) &= x_1(0) \\ x_2(0) &= x_2(0) \\ x_3(0) &= x_3(0) \\ x_4(0) &= x_4(0). \end{aligned} \tag{3.18}$$

Since $\beta_0 = 1$ and $\alpha_0 = 0.5858$, the weights matrix [(3.5)] at $t = 0$ is

$$W(0) = \begin{bmatrix} -0.41 & 1 & 0 & 0 \\ 1 & -1.41 & 1 & 0 \\ 0 & 1 & -1.41 & 1 \\ 0 & 0 & 1 & -0.41 \end{bmatrix}.$$

Then, at round $t = 1$ the status of nodes are updating based on

$$x_i(1) = w_{ii}(0)x_i(0) + \sum_{j \in \mathcal{N}_i} w_{ij}(0)x_j(0), \tag{3.19}$$

that is (3.4) applied at $t = 1$. Therefore the status of nodes become

$$\begin{aligned} x_1(1) &= -0.41x_1(0) + x_2(0) \\ x_2(1) &= -1.41x_2(0) + x_1(0) + x_3(0) \\ x_3(1) &= -1.41x_3(0) + x_2(0) + x_4(0) \\ x_4(1) &= x_4(1) = -0.41x_4(0) + x_3(0). \end{aligned} \tag{3.20}$$

Since $\beta_1 = 1$ and $\alpha_1 = 2$, the weights matrix [(3.5)] at $t = 1$ is

$$W(1) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Subsequently, at round $t = 2$ the status of nodes become ((3.4))

$$\begin{aligned} x_1(2) &= 0.59x_1(0) - 0.41x_2(0) + x_3(0) \\ x_2(2) &= -0.41x_1(0) + 2x_2(0) - 1.41x_3(0) + x_4(0) \\ x_3(2) &= x_1(0) - 1.41x_2(0) + 2x_3(0) - 0.41x_4(0) \\ x_4(2) &= x_2(0) - 0.41x_3(0) + 0.59x_4(0) \end{aligned} \tag{3.21}$$

and the weights matrix [(3.5)]

$$W(2) = \begin{bmatrix} 0.6 & \frac{1}{4} & 0 & 0 \\ \frac{1}{4} & 0.35 & \frac{1}{4} & 0 \\ 0 & \frac{1}{4} & 0.35 & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} & 0.6 \end{bmatrix}$$

Then, at last round ($t = 3$), since we have 4 different eigenvalues, then $K - 1 = 3$, each Node i reach the consensus

$$\begin{aligned} x_1(3) &= \frac{x_1(0)}{4} + \frac{x_2(0)}{4} + \frac{x_3(0)}{4} + \frac{x_4(0)}{4} \\ x_2(3) &= \frac{x_1(0)}{4} + \frac{x_2(0)}{4} + \frac{x_3(0)}{4} + \frac{x_4(0)}{4} \\ x_3(3) &= \frac{x_1(0)}{4} + \frac{x_2(0)}{4} + \frac{x_3(0)}{4} + \frac{x_4(0)}{4} \\ x_4(3) &= \frac{x_1(0)}{4} + \frac{x_2(0)}{4} + \frac{x_3(0)}{4} + \frac{x_4(0)}{4}. \end{aligned} \tag{3.22}$$

3.2.2 Metropolis Consensus

Also in this case it is assumed that, starting from its own measurement $x_i(0)$, each node i updates its scalar value $x_i(t)$ at round t according to the same linear iterative algorithm (3.4) adopted by FTC. In this case a possible choice for the weights is

$$w_{ij}(t) = \begin{cases} \frac{1}{\max\{d(i),d(j)\}}, & \text{if } i \neq j,\ j \in \mathcal{N}_i \\ 1 - \sum_{k \in \mathcal{N}_i} w_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \tag{3.23}$$

where $d\,(i) = |\mathcal{N}_i|$ and $d\,(j) = |\mathcal{N}_j|$ denote the degrees of Node i and Node j, respectively. These are the so-called local-degree weights, since they only depend on the degrees of the two incident nodes. The local-degree weights guarantee *asymptotic* convergence provided that the graph is not bipartite [58].

3.2.3 Flooding

One straightforward method to compute the average consensus is to make all measurements available to all nodes by means of the FL information diffusion algorithm [35, 53]. At round $t = 0$, Node i broadcasts its measurement $x_i(0)$ and receives the measurements $x_j(0), j \in N_i$, of its neighbors. On successive rounds, Node i broadcasts previously received measurements that it did not transmit before. In principle, the identifiers of the nodes that generated each scalar should be transmitted as well; however, since I assume that each node knows the adjacency matrix $\mathbf{A}$, the identifiers can be derived by receiving nodes autonomously at each round. Moreover, also the number of scalars transmitted by each node might vary at each round. After a number of rounds equal to the diameter of the network, each node knows the measurements of all the nodes, so the average consensus can be computed.

3.2.4 TAS

The main idea of the TAS algorithm [4] is to propagate at each round *partial sums* $\sum_i x_i(0)$ of node measurements, with i not necessarily ranging the whole set of nodes, but covering a subset of $\{1, \ldots, N\}$. TAS was originally introduced in [4] as an efficient information diffusion algorithm for the distributed computation of confidence regions. Here I proposed an adaptation of TAS, described hereafter, to the distributed average consensus problem.

TAS consists of six phases, namely, i) initialization, ii) reception, iii) distillation, iv) aggregation, v) transmission, and vi) wrap-up. After initialization, the reception, distillation, aggregation, and transmission phases are sequentially repeated until all information needed to compute the average consensus is available. The wrap-up phase is then executed for the consensus computation.

i) *Initialization phase.* At round $t = 0$ each node i takes its measurement $x_i(0)$ and stores it in the first position of a vector $\mathbf{m}^{(i)}$, that represents the node's local memory. At round $t = 0$ it is, therefore, $\mathbf{m}^{(i)} = [x_i(0)]$.

ii) *Reception phase.* At each round $t \geq 1$ Node i collects the messages containing the partial sums transmitted by its neighbors.

iii) *Distillation phase.* At each round $t \geq 1$ Node i detects whether the partial sums received from its neighbors contain the contribution of measurements not received before. The node can perform this task because it knows the network topology, hence it knows which measurements contribute to the partial sums received from its neighbors. If it appears that a received partial sum contains already known contributions, these contributions are removed, so that a partial sum with a lower number of terms is obtained. The partial sums resulting from distillation, provided they are different from 0, are then appended to $\mathbf{m}^{(i)}$:

$$\mathbf{m}^{(i)} = [x_i(0), \sum_{n \in \mathcal{N}_i^{(1)}} x_n(0), \ldots, \sum_{n \in \mathcal{N}_i^{(k)}} x_n(0)]^T,$$

where $\mathcal{N}_i^k$ are different subsets of $\subset \mathcal{N}$. The same procedure is now applied to partial sums already stored in $\mathbf{m}^{(i)}$, whose contributors is compared with those of freshly received and distilled partial sums. This phase reduces the number of contributors to each partial sum stored in $\mathbf{m}^{(i)}$, so that the different partial sums can be more easily recombined, in the subsequent *aggregation phase*, with each contributor counted no more than once. The length of $\mathbf{m}^{(i)}$ possibly increases at each round and already stored scalars, apart from the node's privy measurement $x_i(0)$, could change as a consequence of the distillation operation.

iv) *Aggregation phase.* At each round $t \geq 1$, in order to create the scalar to be broadcast at round t, Node i aggregates the scalars available in $\mathbf{m}^{(i)}$ at round $t-1$ which were *not* previously aggregated, thus creating a new partial sum.

v) *Transmission phase.* At round $t = 0$, Node i transmits its own measurement $x_i(0)$, whereas at round $t \geq 1$ it broadcasts the new partial sum computed at the end of the aggregation phase. The information diffusion process stops for Node i when the average consensus can be computed through a proper combination of scalars stored in $\mathbf{m}^{(i)}$, or when a certain time has expired.

vi) *Wrap-up phase* The wrap-up phase is performed by each node to compute the average consensus at the end of the information diffusion process. A more detailed description of each phase can be found in [4]. The main advantage of TAS with respect to FL is that the transmitted data sets are of constant size, independently of the transmission round. The transmission rounds are repeated until all nodes are able to compute (3.3) combining the partial sums stored in $\mathbf{R}^{(i)}$.

3.2.5 Network Coding

In my work, I also considered the possibility of using NC for Average Consensus Evaluation over WSNs. In the following, a brief introduction to the Galois field is presented before moving into the details of NC.

- Galois Field

  Let start to three basic algebraic structures that are shown in Figure 3.2:

  Group: is a set of element G together with an operator $\circ$ which combines two elements of G. A group has the following proprieties:

  1. The group operation $\circ$ is closed. That is, for all $a, b \in G$, it holds that $a \circ b = c \in G$.
  2. The group operation is associative. That is, $a \circ (b \circ c) = (a \circ b) \circ c$, for all $a, b, c \in G$.
  3. There is an element $1 \in G$ called the neutral element or identity element, such that $a \circ 1 = 1 \circ a = a$, for all $a \in G$.
  4. For each $a \in G$ there exists an element $a^{-1} \in G$, called the inverse of a, such that $a \circ a^{-1} = a^{-1} \circ a = 1$.
  5. A group G is commutative (or abelian) if, $a \circ b = b \circ a$, for all $a, b \in G$.

  Ring: is a set of element R together with two binary operation that generalize the arithmetic operations of addition and multiplication. A Ring has the following proprieties:

  1. R is an abelian group under addition. $a + (b + c) = (a + b) + c$, for all $a, b, c \in R$; $a + b = b + a$, for all $a, b \in R$; There is an element 0 such that $a + 0 = a$, for all $a \in R$; For each $a \in R$ there exist $-a \in R$ such that $a - a = 0$.
  2. R is monoid under multiplication. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$, for all $a, b, c \in R$; There is an element 1 such that $a \cdot 1 = 1 \cdot a = a$, for all $a \in R$.
  3. Multiplication is distributive with respect to addition. $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$, for all $a, b, c \in R$; $(b + c) \cdot a = (b \cdot a) + (c \cdot a)$, for all $a, b, c \in R$.

  Field: is a set of elements F with the following proprieties:

  1. All elements of F form an additive group with the group operation $+$ and the neutral element 0.
  2. All elements of F except 0 form a multiplicative group with the group operation $\cdot$ and the neutral element 1.

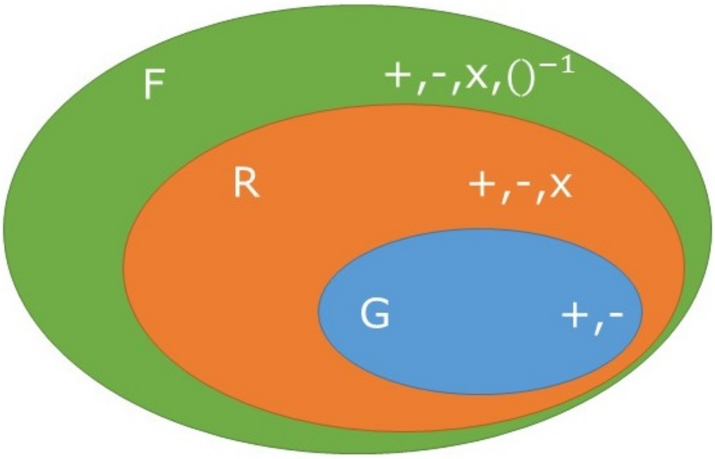


Figure 3.2: Basic algebraic structures.

3. When the two group operator are mixed, the distributivity law holds, $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$, for all $a, b, c \in R$.

In mathematics, a finite field or Galois field is a field that contains a finite number of elements. As with any field, a finite field is a set on which the operations of multiplication, addition, subtraction and division are defined and satisfy certain basic rules. The most common examples of finite fields are given by the integers *mod p* when p is a prime number. The number of elements of a finite field is called its order. A finite field of order q exists if and only if the order q is a prime power p^m. All fields of a given order are isomorphic. In a field of order p^m, adding p copies of any element always results in zero; that is, the characteristic of the field is p. In a finite field of order q, the polynomial $X^q - X$ has all q elements of the finite field as roots. The non-zero elements of a finite field form a multiplicative group. This group is cyclic, so all non-zero elements can be expressed as powers of a single element called a primitive element of the field.

Finite fields only exist if they have p^m elements, with p is prime and m integer.

1. There is a finite field with 11 elements called $GF(11)$;
2. There is a finite field with 81 elements called $GF(81) = GF(3^4)$;
3. There is a finite field with 256 elements called $GF(256) = GF(2^8)$;
4. There is not a finite field with 12 elements $(12 = 2^2 \cdot 3)$.

Two type of finite field exist, *Prime Fields* $GF(p^m)$ with $m = 1$ or *Extension Fields* $GF(p^m)$ with $m > 1$.

A finite field is a finite set on which the four operations multiplication, addition, subtraction and division are defined, satisfying the rules of arithmetic known as the field axioms. The simplest examples of finite fields are the fields of prime order: for each prime number p, the field GF(p) of order p may be constructed as the integers mod p. The elements of such a field may be represented by integers in the range 0,. . . ,p-1. The sum, the difference and the product are computed by taking the remainder by p of the integer result. The multiplicative inverse of an element may be computed by using the extended Euclidean algorithm.

Let F be a finite field. For any element $x \in F$ and any integer n, let us denote by $n \cdot x$ the sum of n copies of x. The least positive n such that $n \cdot 1 = 0$ must exist and is a prime number; it is called the characteristic of the field. If the characteristic of F is p, one can define multiplication of an element k of $GF(p)$ by an element x of $F(k, x) \mapsto k \cdot x$ by choosing an integer representative for k and using repeated addition. This multiplication makes F into a $GF(p)$-vector space. It follows that the number of elements of F is p^n for some integer n. For each element x in the field $GF(p)$ for a prime number p, one has $x^p = x$ (this may be proved as follows: the equality is trivially true for $x = 0$ and $x = 1$; one obtains the result for the other elements of $GF(p)$ by applying the above identity to x and 1, where x successively takes the values $1, 2, ..., p - 1 \; mod \, p$). This implies the equality $X^p - X = \prod_{a \in F}(X - a)$ for polynomials over GF(p). More generally, every element in $GF(p^m)$ satisfies the polynomial equation $x^{p^m} - x = 0$. Any finite field extension of a finite field is separable and simple. That is, if E is a finite field and F is a subfield of E, then E is obtained from F by adjoining a single element whose minimal polynomial is separable.

1. Prime Fields

   The element of a prime field $GF(p)$ are the integers $\{0, 1, . . . , p - 1\}$. Let us evaluate the addition, subtraction and multiplication, it results very simple in

$$\begin{cases} a + b \equiv c \;\; mod \, p \;\; a, b \in GF(p) = \{0, 1, . . . , p - 1\} \\ a - b \equiv d \;\; mod \, p \;\; a, b \in GF(p) = \{0, 1, . . . , p - 1\} \\ a \cdot b \equiv e \;\; mod \, p \;\; a, b \in GF(p) = \{1, . . . , p - 1\} \end{cases} \tag{3.24}$$

   For any $a \in GF(p) = \{0, 1, . . . , p - 1\}$, the inverse a^{-1} must satisfy the rule $a \cdot a^{-1} \equiv 1 \; mod \, p$. It can be computed by th extended Euclidean algorithm.

2. Extension Fields

The element of an extension field $GF(p^m)$ are polynomials $A(x) = a_{m-1}x^{m-1} + \ldots + a_1x^1 + a_0$; $A(x) \in GF(p^m)$, in which the coefficients $a_i \in GF(p) = \{0, 1, \ldots, p-1\}$. In Coding theory $A(x) \in GF(2^m)$ and $a_i \in GF(2) = \{0, 1\}$. For example, considering a finite field $GF(2^3)$ that lead in $A(x) = a_2x^2 + a_1x + a_0$ in which $a_i \in GF(2) = \{0, 1\}$. It follows that $A(x)$ can be represent as a 3-bit vector $(A(x) = a_2x^2 + a_1x + a_0 = (a_2, a_1, a_0))$ so it is possible to represent 8 different number in fact $GF(2^3) = GF(8)$ (3.25). The element of this field are

$$\begin{cases} a_2, a_1, a_0 & GF(8) = \{\ldots\} \\ 0,0,0 & 0 \\ 0,0,1 & 1 \\ 0,1,0 & x \\ 0,1,1 & x+1 \\ 1,0,0 & x^2 \\ 1,0,1 & x^2+1 \\ 1,1,0 & x^2+x \\ 1,1,1 & x^2+x+1 \end{cases} \tag{3.25}$$

- Introduction to Linear NC Linear network coding is a mathematical technique which may be used to improve a network throughput, efficiency and scalability, as well as resilience to attacks and eavesdropping. Instead of simply relaying the packets of information they receive, the nodes of a network take several packets and combine them together for transmission. This may be used to attain the maximum possible information flow in a network. It has been proven in theory that linear coding is enough to achieve the upper bound in multicast problems with one source [56]. However linear coding is not sufficient in general (e.g. multisource, multisink with arbitrary demands), even for more general versions of linearity such as convolutional coding and filter-bank coding [71]. In a linear network coding problem, a group of N nodes are involved in moving the data from S source nodes to sink nodes. Each node generates new packets which are linear combinations of earlier received packets, multiplying them by coefficients chosen from a finite field, typically of size $GF(2^S)$. Each node, n_k with indegree, $InDeg(n_k) = S$, generates a message x_k from the linear combination of received messages $\{M_i\}_{i=1}^{S}$ by the relation

$$x_k = \sum_{i=1}^{S} g_k^i M_i \tag{3.26}$$

where the values g_k^i are the coefficients selected from $GF(2^S)$. Note that, since operations are computed in a finite field, the generated message is of the same length as the original messages. Each node forwards the computed value x_k along with the coefficients, g_k^i. Sink nodes receive these network coded messages, and collect them in a matrix. The original messages can be recovered by performing Gaussian elimination on the matrix. In reduced row echelon form, decoded packets correspond to the rows of the form $e_i = [0 \ldots 0\,1\,0 \ldots 0]$.

In linear algebra, Gaussian elimination (row reduction) is an algorithm for solving systems of linear equations. It is usually understood as a sequence of operations performed on the corresponding matrix of coefficients. The process of row reduction makes use of elementary row operations, and can be divided into two parts. The first part reduces a given system to row echelon form, from which one can tell whether there are no solutions, a unique solution, or infinitely many solutions. The second part continues to use row operations until the solution is found. There are three types of elementary row operations which may be performed on the rows of a matrix:

- Swap the positions of two rows;
- Multiply a row by a nonzero scalar;
- Add to one row a scalar multiple of another.

If the matrix is associated to a system of linear equations, then these operations do not change the solution set. Therefore, if the goal is to solve a system of linear equations, then using these row operations could make the problem easier. For each row in a matrix, if the row does not consist of only zeros, then the left-most non-zero entry is called the leading coefficient (or pivot) of that row. So if two leading coefficients are in the same column, then a row operation (Add to one row a scalar) could be used to make one of those coefficients zero. Then by using the row swapping operation, one can always order the rows so that for every non-zero row, the leading coefficient is to the right of the leading coefficient of the row above. If this is the case, then matrix is said to be in row echelon form. So the lower left part of the matrix contains only zeros, and all of the zero rows are below the non-zero rows. An examples of a matrix in row echelon form is

$$\begin{bmatrix} 1 & c_1 & c_2 & c_3 & c_4 \\ 0 & 0 & 0 & 3 & c_5 \\ 0 & 0 & 0 & 0 & 5 \end{bmatrix} \tag{3.27}$$

A matrix is said to be in reduced row echelon form if furthermore all of the leading coefficients are equal to 1 (which can be achieved by multiply a row by a nonzero scalar), and in every column containing a leading coefficient, all of the other entries in that column

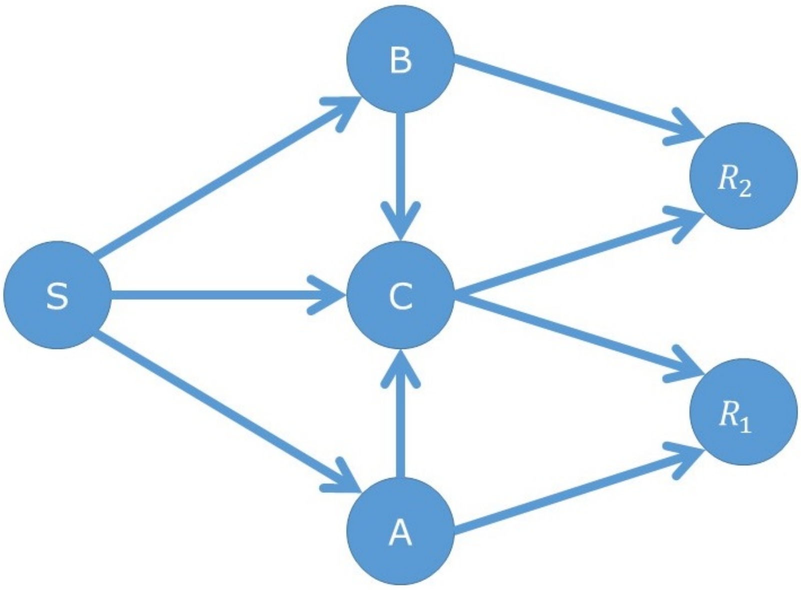


Figure 3.3: NC Toy Network.

are zero, as shown in (3.28)

$$\begin{bmatrix} 1 & c_1 & 0 & c_2 & 0 \\ 0 & 0 & 1 & c_3 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.28}$$

- Toy Network Consider a communication network in which certain source nodes multicast information to other nodes on the network in the multihop fashion where every node can pass on any of its received data to others. I am interested in how fast each node can receive the complete information. Allowing a node to encode its received data before passing it on, the question involves optimization of the multicast mechanism at the nodes. Among the simplest coding schemes is linear coding, which regards a block of data as a vector over a certain base field and allows a node to apply a linear transformation to a vector before pass it on. As example, consider the multicast of two bits b_1 and b_2 from a source (S) to two receivers (R_1, R_2) in the network of Figure 3.3

  A solution is to let the channels $SA,\ AC,\ AR_1$ carry b_1, $SB,\ BC,\ BR_2$ carry b_2 and $CR_1,\ CR_2$ carry the exclusive or $b_1 \oplus b_2$. Then, the node R_1 receives b_1 and $b_1 \oplus b_2$, from which the bit b_2 can be decoded. Similarly, the node R_2 receives b_2 and $b_1 \oplus b_2$, from which the bit b_1 can be decoded.

- Network Coding

  In NC, data packets are seen as vectors with elements in some Galois Field (GF) $\mathbb{F}$ [72,73].

Random linear NC can be adopted for distributed average consensus computation as follows. At each round the generic node i generates a new packet which is a linear combinations of packets received in the previous round, multiplying them by random coefficients $g_i^j \in \mathbb{F}$, with $j \in \mathcal{N}_i$:

$$x_i(t+1) = \sum_{j \in \mathcal{N}_i} g_i^j x_j(t), \tag{3.29}$$

where, $x_j(t)$ is a vector in $\mathbb{F}$ representing the data packet sent by node j at round t. Each node forwards the computed value $x_i(t+1)$ along with the coefficients g_i^j. In principle we could avoid their transmission provided that a common seed is adopted for random numbers generation. In any case this aspect is not discussed here. Each node receives network-coded messages, and collects them in a matrix. Provided that a sufficiently large amount of iterations is performed, the vector $\mathbf{x}(0)$ can be recovered by performing Gaussian elimination on the matrix whose rows are formed by all packets received by a node. Then, the average consensus can be immediately computed.

3.3 Finite-Time Consensus With Memory

The FTCM algorithm, which I introduce here as an original contribution, works as follows.

- At round $t = 0$ each node i broadcasts its measurement $x_i(0)$,
- at round $t = 1, 2, ..., R-1$ each node i stores in its internal memory the data $x_j(t-1), j \in N_i$, received from its neighbors in round $t-1$ and broadcasts their sum:

$$x_i(t) = \sum_{j \in \mathcal{N}_i} x_j(t-1). \tag{3.30}$$

Introducing $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$, and observing that $\mathbf{x}(1) = \mathbf{A}\mathbf{x}(0)$, $\mathbf{x}(2) = \mathbf{A}\mathbf{x}(1) = \mathbf{A}^2\mathbf{x}(0)$, it is possible to write in general

$$\mathbf{x}(t) = \mathbf{A}^t\mathbf{x}(0). \tag{3.31}$$

In the same round $t \geqslant 1$, each node i collects the data received at round $t-1$ from its neighbors, which is given by vector

$$\mathbf{y}_i(t) = \mathbf{I}_{\mathcal{N}_i}[\mathbf{x}(t-1)] = \mathbf{I}_{\mathcal{N}_i}\mathbf{A}^{t-1}\mathbf{x}(0) \tag{3.32}$$

where $\mathbf{I}_{\mathcal{N}_i}$ indicates the matrix composed of the rows of the identity matrix whose indexes are included in the set $\mathcal{N}_i$. Therefore, after t rounds, the total amount of data collected by Node i, including its own measurement $y_i(0) = x_i(0)$, can be stacked in the column vector

$$\mathbf{c}_i(t) = \begin{bmatrix} y_i(0) \\ \mathbf{y}_i(1) \\ \vdots \\ \mathbf{y}_i(t) \end{bmatrix} = \begin{pmatrix} \mathbf{e}_i^T \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A}^0 \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A} \\ \vdots \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A}^{t-1} \end{pmatrix} \mathbf{x}(0) = \mathbf{F}_i(t)\mathbf{x}(0), \tag{3.33}$$

where $\mathbf{e}_i$ is a column vector of all zeros except a one in the ith position to account for the fact that each node knows its own measurement. The problem is to understand whether after t rounds Node i is able to calculate (3.3) through a proper combination of the values of all data collected from its neighbors stored in vector $\mathbf{c}_i(t)$ (3.33). In other words, whether there exists a proper vector $\mathbf{w}_i$ of weights such that

$$\mathbf{c}_i(t)^T \mathbf{w}_i(t) = \overline{x}_N \tag{3.34}$$

To address this problem, introduce the matrices

$$\mathbf{B}_i(0) = \mathbf{e}_i^T \tag{3.35}$$

$$\mathbf{B}_i(t) = \mathbf{I}_{\mathcal{N}_i}\mathbf{A}^{t-1} \text{ for } t \geq 1. \tag{3.36}$$

being $\mathbf{e}_i$ a column vector of all zeros except a one in the ith position, to account for the fact that each node knows its own measurement. Similarly to what done before in (3.33), we stack $\mathbf{B}_i(0)$, $\mathbf{B}_i(1)$, ..., $\mathbf{B}_i(t)$ into

$$\mathbf{F}_i(t) = \begin{bmatrix} \mathbf{B}_i(0) \\ \mathbf{B}_i(1) \\ \vdots \\ \mathbf{B}_i(t) \end{bmatrix}^T . \tag{3.37}$$

which results to be a matrix of size $N \times M_i$. The weight vector $\mathbf{w}_i(t)$ can be found by solving

$$\frac{1}{N}\mathbf{e} = \mathbf{F}_i(t)\mathbf{w}_i(t) \tag{3.38}$$

which may also be written as

$$\mathbf{w}_i^T \begin{pmatrix} \mathbf{e}_i^T \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A}^0 \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A} \\ \vdots \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A}^{k-1} \end{pmatrix} = \frac{1}{N}\mathbf{e}^T$$

where $\mathbf{e}$ is a column vector of all ones. Equation (3.38) admits at least one solution if and only if

$$\operatorname{rank}(\mathbf{F}_i(t)) = \operatorname{rank}\left(\mathbf{F}_i(t)\middle|\mathbf{e}\right) \tag{3.39}$$

which is given by

$$\mathbf{w}_i(t) = \frac{1}{N}\mathbf{F}_i(t)^T \left(\mathbf{F}_i(t)\mathbf{F}_i(t)^T\right)^{-1} \mathbf{e}. \tag{3.40}$$

where the terms multiplying vector $\mathbf{e}$ represent the pseudo-inverse of $\mathbf{F}_i(t)$. This requires nevertheless that the $N \times N$ matrix $\mathbf{F}_i(t)\mathbf{F}_i(t)^T$ is invertible, *i.e.*, that $N \geqslant M_i$. But in that case, one may perform Gaussian elimination to recover each data produced by the nodes. In the case $M_i < N$, one may still have $\operatorname{rank}(\mathbf{F}_i(t)) = \operatorname{rank}\left(\mathbf{F}_i(t)\middle|\mathbf{e}\right)$. In that case, the solution is $\mathbf{w}_i(t)$ has to be such that

$$\begin{aligned} N\left(\mathbf{w}_i^T(t)\mathbf{F}_i^T(t) - \mathbf{e}^T\right)\left(\mathbf{w}_i^T(t)\mathbf{F}_i^T(t) - \mathbf{e}^T\right)^T &= 0 \\ N\left(\mathbf{w}_i^T(t)\mathbf{F}_i^T(t)\mathbf{F}\mathbf{w}_i(t) - 2\mathbf{w}_i^T(t)\mathbf{F}_i^T(t)\mathbf{e} + \mathbf{e}^T\mathbf{e}\right) &= 0 \\ \mathbf{F}_i^T(t)\mathbf{F}_i(t)\mathbf{w}_i(t) &= \mathbf{F}_i^T(t)\frac{\mathbf{e}}{N} \end{aligned}$$

thus

$$\mathbf{w}_i(t) = \left(\mathbf{F}_i^T(t)\mathbf{F}_i(t)\right)^{-1} \mathbf{F}_i^T(t)\frac{\mathbf{e}}{N}, \tag{3.41}$$

which only requires the $M_i \times M_i$ matrix to be invertible. If we denote by t_i the minimum number of rounds necessary to find at least one solution for Node i, the convergence time for the network, expressed in terms of rounds, is given by $R = \max_i t_i$. Since all nodes know the adjacency matrix $\mathbf{A}$, each node can compute in advance (with respect to the algorithm execution) its weights vectors $\mathbf{w}_i(t)$ $\forall t$ and the number of rounds needed for the convergence within the whole network. Moreover, (3.39) provides a condition that can be checked in advance to determine for each node i whether the consensus can be reached or not and at which round.

3.3.1 Controllability of System

Further works, related to the Consensus scheme and not relevant to the results that I present in Section 3.4, has been carried out during my Ph.D. In particular, I analyzed the relation with controllability system and try to extrapolate generic representation of Consensus algorithms.

Consider the case of a node sharing a single connection with another node in the network. In that case, $\mathbf{I}_{\mathcal{N}_i}$ consists of a single row. Assume that its index is j. Then $\mathbf{I}_{\mathcal{N}_i}\mathbf{A} = \mathbf{a}_j^T$,the j-th row of $\mathbf{A}$ and

$$\begin{pmatrix} \mathbf{e}_i^T \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A} \\ \vdots \\ \mathbf{I}_{\mathcal{N}_i}\mathbf{A}^{k-1} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_i^T \\ \mathbf{a}_j^T\mathbf{A}^0 \\ \vdots \\ \mathbf{a}_j^T\mathbf{A}^{k-2} \end{pmatrix}.$$

One has to compare

$$\text{rank}\begin{pmatrix} \mathbf{e}_i^T \\ \mathbf{a}_j^T\mathbf{A}^0 \\ \vdots \\ \mathbf{a}_j^T\mathbf{A}^{k-2} \end{pmatrix} \text{ and rank}\begin{pmatrix} \mathbf{e}_i^T \\ \mathbf{a}_j^T\mathbf{A}^0 \\ \vdots \\ \mathbf{a}_j^T\mathbf{A}^{k-2} \\ \mathbf{e}^T \end{pmatrix}.$$

Assuming that the links are symmetric, one has $\mathbf{A} = \mathbf{A}^T$. One has to compare the ranks of

$$\begin{pmatrix} \mathbf{e}_i & \mathbf{A}^0\mathbf{a}_j & \mathbf{A}^1\mathbf{a}_j & \dots & \mathbf{A}^{k-2}\mathbf{a}_j \end{pmatrix}$$

and that of

$$\begin{pmatrix} \mathbf{e}_i & \mathbf{A}^0\mathbf{a}_j & \mathbf{A}^1\mathbf{a}_j & \dots & \mathbf{A}^{k-2}\mathbf{a}_j & \mathbf{e} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_i & \mathbf{A}^1\mathbf{e}_j & \mathbf{A}^2\mathbf{e}_j & \dots & \mathbf{A}^{k-1}\mathbf{e}_j & \mathbf{e} \end{pmatrix}$$

This type of property is closely related to the controllability of systems. Consider a discrete-time system

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \tag{3.42}$$

Then the system is said to be controllable, *i.e.*, with a properly chosen sequence of $u(k)$, any value of $\mathbf{x}(k)$ may be reached iff the rank of the matrix

$$\begin{pmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{pmatrix} \tag{3.43}$$

is of full rank n.

3.4 Performance Comparison

In this section the performance of the previously introduced algorithms is compared on different network topologies, namely

- generic trees, *i.e.*, trees with a random number of children for each node. When operated on this topology, TAS and FL can exploit the network structure, hence they work on one level at a time: transmissions are only performed by the children (the leafs) at round $t = 0$, by their parents at round $t = 1$, and so on in subsequent rounds, up to the root and then in the opposite direction. The other algorithms require, instead, that each node transmits at each round.

- clustered networks with n_c clusters structured on a single level of hierarchy. Each cluster comprises a random number of nodes and a cluster head that is the special node responsible for aggregating the data of its children. We consider that each cluster head can directly communicate with each other. Moreover, each node in a cluster is assumed to directly communicate with its cluster head and vice-versa. When operated on this topology, FL, TAS and NC work one level at a time: transmissions are first performed by children in the first round, then by cluster heads in the second and third round (and also in subsequent rounds for NC only). FTC, FTCM and Metropolis Consensus (MC) require, instead, that each node transmits at each round.

- completely unstructured connected random networks, where each node can directly communicate by means of broadcast transmissions with its one-hop neighbors. In this case no specific network structure can be exploited by any algorithm, hence all of them require that each node transmits at each round.

1. Preliminary Results

   As preliminary results, I evaluated the performance of TAS and FTC in terms of amount of generated traffic and convergence speed. Simulations have been carried out considering 10 Monte Carlo different network realizations with $N = 100$ nodes each, where nodes are uniformly distributed in a 1×1 (normalized units) square area, each of them with a coverage range of 0.25.

   In the following, TAS and FTC are compared in terms of average amount of nodes that have evaluated the consensus with a relative error less than $\epsilon = 10^{-4}$, the number of round needed in order to reach the consensus or the maximum result (if not all nodes can reach the consensus) and the number of exchanged scalars within the networks.

   Table 3.1 shows the performance comparison in unstructured topologies. As we can see, TAS does not allow all nodes to reach consensus at every realization (it has some convergence problem as we have seen previously) but it reaches the consensus or its maximum performance with a short number of iteration and, thanks to the aggregation, with a low amount of transmitted scalars. With FTC, instead, all nodes reach the consensus but it takes a huge amount of rounds and needs an high amount of transmitted scalars.

   Different tests have been performed varying the characteristic of the network. In particular, it has been noticed that decreasing the number of nodes N or the average amount of neighbors (reducing the coverage range) the performance of TAS increases. Table 3.2 shows the results of 10 different realizations with 20 nodes and coverage range of 0.15. The performance of TAS is still better than the one of FTC but, in this case, all nodes reach the consensus also with TAS.

   Table 3.3 and Table 3.4 show the performance comparison in structured topologies. In this case all nodes reach the consensus. In Table 3.3 listed the performance of TAS and

FTC algorithms in Tree topologies. As seen in [4], TAS performs very good in structured topologies and there its performance outperforms the ones of FTC. Table 3.4, instead, shows the performance of the algorithms in clustered topologies with different cluster heads. Also in this case, TAS outperforms FTC in terms of amount of transmitted scalars even if the structure of the network decreases the gap between the algorithms.

Realization	Algorithm	Amount of nodes that reach $\bar{x}_N$	Convergence round	Amount of Transmitted Scalars
1	FTC	100	97	9700
	TAS	96	36	3669
2	FTC	100	98	9800
	TAS	100	26	2700
3	FTC	100	98	9800
	TAS	98	62	6169
4	FTC	100	94	9400
	TAS	100	17	1800
5	FTC	100	95	9500
	TAS	94	40	4057
6	FTC	100	95	9500
	TAS	96	23	2400
7	FTC	100	95	9500
	TAS	95	43	4334
8	FTC	100	98	9800
	TAS	99	36	3698
9	FTC	100	99	9900
	TAS	98	35	3589
10	FTC	100	97	9700
	TAS	100	26	2700

Table 3.1: TAS Vs. FTC preliminary results with $N = 100$ in 10 different unstructured topologies.

2. Results

   Simulations have been carried out considering 100 Monte Carlo different network realizations with $N = 100$ nodes each, where nodes are uniformly distributed in a 1×1 (normalized units) square area, each of them with a coverage range of 0.25. In this work we considered trees with 4 levels and clustered networks with $n_c = 8$. As far as the NC algorithm is concerned, numerical results have been obtained assuming that the coefficients g_i^j are selected from $\mathbb{F}$ =GF(2^1).

   Consensus accuracy vs traffic

   For a given network realization, we consider the relative consensus error at round t for Node i, $E^{(i)}(t) = \frac{|\widetilde{x}^{(i)}(t) - \overline{x}_N|}{|\overline{x}_N|}$, where $\widetilde{x}^{(i)}(t)$ is the estimate of $\overline{x}_N$ by node i at round t. In the following I assume that Node i achieves the consensus at round t if $E^{(i)}_{r-1} > 10^{-4}$ and $E^{(i)}(t) \leq 10^{-4}$.

Realization	Algorithm	Amount of nodes that reach $\bar{x}_N$	Convergence round	Amount of Transmitted Scalars
1	FTC	20	19	380
	TAS	20	11	205
2	FTC	20	19	380
	TAS	20	7	145
3	FTC	20	19	380
	TAS	20	9	183
4	FTC	20	19	380
	TAS	20	12	219
5	FTC	20	18	360
	TAS	20	6	130
6	FTC	20	15	300
	TAS	20	6	126
7	FTC	20	18	360
	TAS	20	8	156
8	FTC	20	19	380
	TAS	20	7	142
9	FTC	20	19	380
	TAS	20	8	152
10	FTC	20	17	340
	TAS	20	6	130

Table 3.2: TAS Vs. FTC preliminary results with $N = 20$ and coverage range of 0.15 in 10 different unstructured topologies.

For each algorithm and each network topology, the average relative error $\overline{E}(t)$ (at round t) is evaluated by averaging $E^{(i)}(t)$ over all network realizations. The evolution of the average relative error $\overline{E}(t)$ with the number of exchanged scalars is represented in Figure 3.4(a), Figure 3.5(a) and Figure 3.6(a) for the considered topologies.

Figure 3.4(a) refers to the random network topology, where the consensus is achieved (with a relative error less than $\epsilon = 10^{-4}$) by all considered algorithms except TAS. One observes, in particular, that FTCM outperforms all the considered competitors, with a significant gain in terms of amount of exchanged scalars (and ultimately, in terms of energy saved) with respect to MC, FL, and FTC. The MC algorithm shows a slow convergence, which is mainly due to the relatively low average connectivity degree. FTC achieves convergence in about 99 iterations. This is due to the fact that for a random topology, the number of different eigenvalues of the Laplacian matrix is likely to be equal to the number of nodes. Moreover, while other approaches show a smooth decrease of the estimation error, the relative error with FTC starts to increase and then to decrease before vanishing at the iteration where convergence occurs. This may be a critical issue when not enough communication iterations are available to complete convergence.

Figure 3.5(a) and Figure 3.6(a) are for tree and clustered networks, respectively. In such

Realization	Algorithm	Amount of nodes that reach $\bar{x}_N$	Convergence round	Amount of Transmitted Scalars
1	FTC	100	63	6300
	TAS	100	10	138
2	FTC	100	57	5700
	TAS	100	8	135
3	FTC	100	63	6300
	TAS	100	12	137
4	FTC	100	51	5100
	TAS	100	8	142
5	FTC	100	67	6700
	TAS	100	8	138
6	FTC	100	68	6800
	TAS	100	8	140
7	FTC	100	52	5200
	TAS	100	8	144
8	FTC	100	64	6400
	TAS	100	12	139
9	FTC	100	58	5800
	TAS	100	10	137
10	FTC	100	64	6400
	TAS	100	10	140

Table 3.3: TAS Vs. FTC preliminary results with $N = 100$ in 10 different tree topologies.

Realization	Clusterheads	Algorithm	Amount of nodes that reach $\bar{x}_N$	Convergence round	Amount of Transmitted Scalars
1	9	FTC	100	18	1800
		TAS	100	3	128
2	7	FTC	100	12	1200
		TAS	100	3	125
3	8	FTC	100	14	1400
		TAS	100	3	127
4	5	FTC	100	10	1000
		TAS	100	3	132
5	5	FTC	100	10	1000
		TAS	100	3	128
6	5	FTC	100	8	800
		TAS	100	3	126
7	4	FTC	100	8	800
		TAS	100	3	130
8	9	FTC	100	16	1600
		TAS	100	3	125
9	10	FTC	100	20	2000
		TAS	100	3	123
10	8	FTC	100	16	1600
		TAS	100	3	126

Table 3.4: TAS Vs. FTC preliminary results varying $n_c \in [4 - 10]$ with $N = 100$ in 10 different clustered topologies.

structured networks, all algorithms achieve consensus. TAS performs the best. FTCM still performs satisfactorily in the case of tree networks, see Figure 3.5(a), but appears less appealing when operated in clustered networks (Figure 3.6(a)). In this respect, we observe that NC and FL show a remarkable performance improvement when operated in clustered networks with respect to the case of tree networks, whereas FTCM experiences only a small improvement. Moving from tree to clustered networks, in fact, FL benefits from the reduced number of network levels to be traveled, while NC takes advantage of the fact that (differently from the tree case) the multiple transmissions needed for network decoding are performed only by the cluster heads. Similar conclusion to those done for the random topology hold for MC. For what concerns FTC, a faster convergence is observed, due to a reduction of the number of different eigenvalues of the Laplacian matrix of these structured topologies.

Convergence speed

Figure 3.4(b), Figure 3.5(b) and Figure 3.6(b) represent the average number $\overline{N}(t)$ of nodes that have evaluated the consensus with a relative error less than $\epsilon = 10^{-4}$ as a function of the iteration index. The average is evaluated over 100 network realizations for each topology.

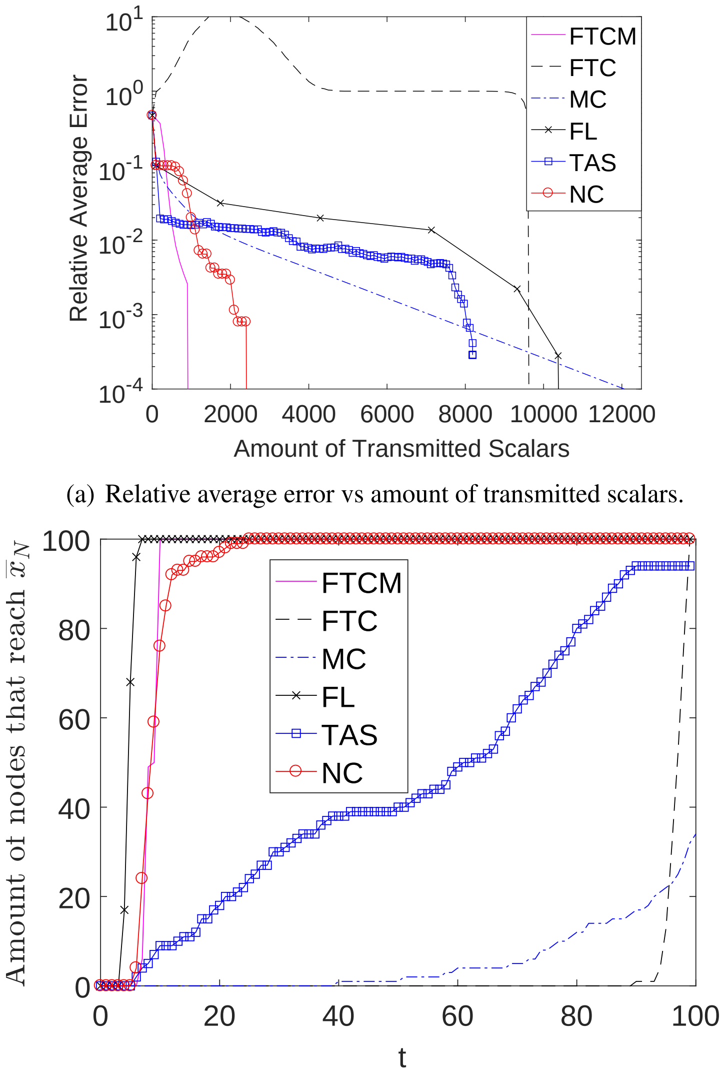
Figure 3.4(b) is for the random topology. FL performs the best: convergence is reached in a number of iterations equal to the diameter of the network. Nevertheless, the price of this rapid convergence is a relatively large amount of data that have be exchanged, see Figure 3.4(a).

The number of iterations required by FTCM and NC are very close to that of FL. FTCM achieves the best compromise between convergence speed and amount of exchanged data, see Figure 3.4(a). TAS, MC, and FTC are much more slower. With TAS, not all nodes are able to reach consensus.

In the case of structured networks, TAS becomes the best solution. It is as fast as FL, see Figures 3.5(b) and 3.6(b), but it requires the lowest amount of data exchanged, as already observed in Figure 3.5(a) and 3.6(a). Finally, in both cases of tree and clustered topologies, FTCM shows faster convergence speeds than MC, FTC, and NC.

3.5 Conclusions

In this work I introduced a novel algorithm, named FTCM, for the distributed evaluation of the average consensus, which exploits the node memory to facilitate the consensus evaluation. Moreover, I proposed an adaptation to the distributed average consensus problem of the TAS



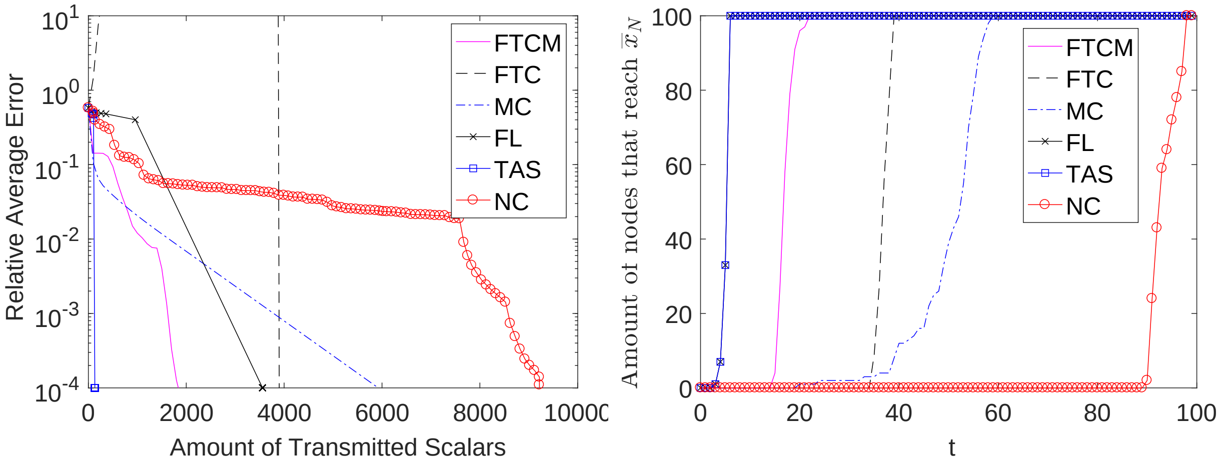
(a) Relative average error vs amount of transmitted scalars.

(b) Amount of nodes achieving consensus vs number of iterations.

Figure 3.4: Unstructured random networks

algorithm introduced in [4] for the distributed confidence region evaluation. The performance of the two algorithms, in terms of efficiency in the usage of network resources and convergence speed, has been compared with those of standard or recently introduced algorithms, such as Metropolis Consensus, Finite Time Consensus, Flooding and Network Coding. The outcomes of performance investigations, carried out considering unstructured random networks, tree networks, and clustered networks, show that FTCM is very well behaving when operated on unstructured random network topologies, whereas TAS outperforms its competitors when structured networks are considered, either tree or clustered networks.

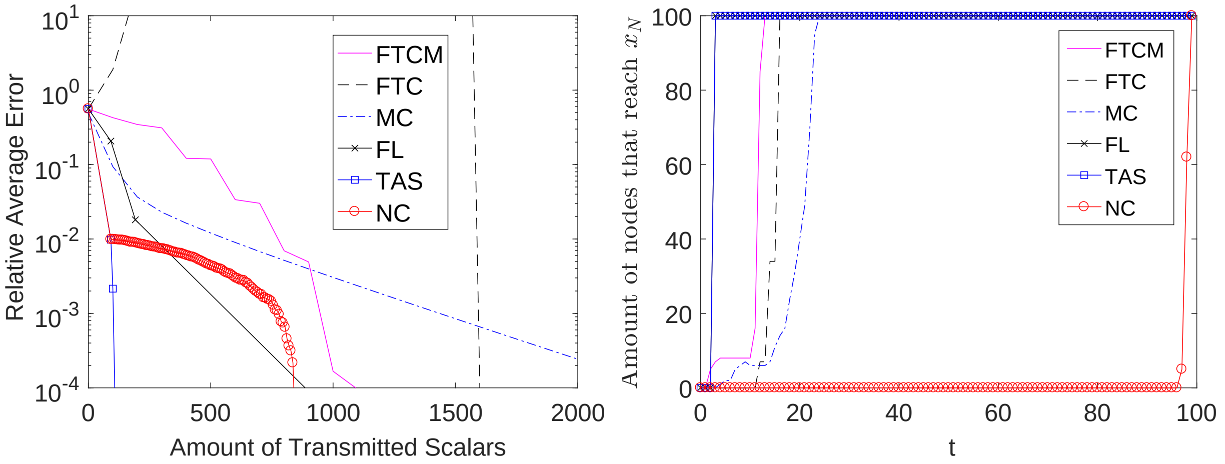
Future work discussed in Section 3.3.1 will be object of a future journal publication.



(a) Relative average error vs amount of transmitted scalars.

(b) Amount of nodes achieving consensus vs number of iterations.

Figure 3.5: Tree networks



(a) Relative average error vs amount of transmitted scalars.

(b) Amount of nodes achieving consensus vs number of iterations.

Figure 3.6: Clustered networks

Chapter 4

Distributed Faulty Node Detection under Byzantine Attack in DTNs

4.1 Introduction

Delay Tolerant Networks (DTNs) refer to the challenging situation of networks operating with intermittent connectivity [74]. This happens, for example, in Vehicular DTNs [75], where the nodes are moving vehicles and communication is established only between closely located agents. This produces frequent link disruptions and network topology reconfiguration. This time-varying nature exposes DTNs to infiltrations by potentially malicious nodes, which may attempt to perturb the DTN behavior. Threatens against the DTN integrity may come in the form of malware attacks [76], selfish behavior of nodes [77], Byzantine attacks [78], and so on. The absence of a central unit able to act as a certifying authority makes trust management in DTNs very difficult.

In this work I considered a DTN where nodes are equipped with sensors, collecting data used, *e.g.*, to estimate some physical phenomenon. I assume that the network behavior is perturbed by nodes with defective sensors and by nodes performing Byzantine attacks.

A sensor is called defective if it frequently reports erroneous measurements. This phenomenon may be due, *e.g.*, to the degradation of the equipment in time. The identification of nodes equipped with defective sensors is very important to save communication resources and to prevent erroneous measurements to pollute the estimates provided by the DTN. Distributed fault detection (DFD) is a well-investigated topic in Wireless Sensor Networks (WSNs), see [79–81] and references therein. The WSNs considered in the literature are usually dense and have a static topology. DFD in DTNs is made more challenging by the sparse and dynamic topology, and is much less investigated. In previous work [82], I proposed a distributed faulty node detection (DFD) and easily implementable algorithm allowing each node of a DTN to determine whether its own sensors are defective.

A basic assumption in [82] is that all the nodes in the DTN may not be misbehaving in other ways than carrying defective sensors. This paper investigates the performance of the DFD algorithm when the DTN is under Byzantine attack, *i.e.*, several nodes are fully controlled by an adversary. While the normal nodes perform the DFD algorithm to determine the status of their own sensors, the Byzantine nodes try to prevent the correct self-evaluation of normal nodes. This work aims to determine *i*) whether the DFD algorithm proposed in [82] is robust against the introduction of Byzantine nodes; *ii*) how to adjust the algorithm parameters to minimize the effects of the Byzantine attack. To answer these questions I extend the analysis in [82] by taking into account a portion of Byzantine nodes. Theoretical predictions are supported by simulation results obtained by using both an idealized node displacement model and traces from real databases. This provides insights on the way the parameters of the DFD algorithm should be adapted to minimize the impact of misbehaving nodes. Theoretical results are illustrated with simulations considering nodes with random displacements, as well as traces of node inter-contact times from real databases.

4.2 DFD algorithm under Byzantine attack

Consider a set $\mathcal{S}$ of moving nodes equipped with sensors. Assume that a subset $\mathcal{B} \subset \mathcal{S}$ of these nodes are controlled by an adversary and perform a Byzantine attack to disturb the behavior of the network, these nodes are referred in what follows to as “malicious”. The nodes in the set $\mathcal{N} = \mathcal{S} \setminus \mathcal{B}$ are *normal*. Let $\mathcal{D} \subset \mathcal{N}$ denote the subset of nodes which are not malicious but produce *outliers* due to their defective sensors. The *outliers* are measurements having statistical characteristics significantly different from normal measurements provided by good sensors. As a consequence, the *status* of Node i has three possible values $\theta_i(t) \in \Theta = \{0, 1, 2\}$, *i.e.*,

$$\theta_i(t) = \begin{cases} 0, & \text{if } i \in \mathcal{N} \setminus \mathcal{D}, \\ 1, & \text{if } i \in \mathcal{D}, \\ 2, & \text{if } i \in \mathcal{B}. \end{cases}$$

Assumes that the status of nodes remains constant during the algorithm, *i.e.*, $\theta_i(t) = \theta_i$, and that the nodes are initially not aware of their status and only nodes in $\mathcal{N}$ are willing to estimate their status. Let p_θ be the proportion of nodes with status $\theta \in \Theta$, with $p_0 + p_1 + p_2 = 1$. Nodes can exchange information only during the limited time interval in which they are in close proximity (within radio coverage). As in [82], nodes are assumed to be well-mixed and the time interval between two successive meetings of a given node is assumed to follow an exponential distribution with an inter-contact rate λ [83]. Moreover, one assumes that each meeting involves only two nodes. When more than two nodes meet at the same time instant, processing is performed pair-by-pair. During each meeting of a pair of nodes $(i, j) \in \mathcal{S}$ each node senses data $m_.$ with their own sensors and then may exchange these data. If Node i has received the data from Node j (*i.e.*, m_j), then a *local outlier detection test* (LODT) can be performed by Node i with outcome y_{ij}. Assume that the spatial and temporal correlation between data is such that only data sensed during the meeting of two nodes can be exploited by a LODT. Therefore, previously collected data are not used. The LODT yields $y_{ij} = 1$ if it detects the presence of at least an outlier among the data m_i and m_j, and $y_{ij} = 0$ otherwise. The LODT is not able to determine which sensor is producing outliers. Such situation occurs for example, when comparing few scalar measurements of the same physical quantity. The presence of an outlier is easily detected when the measurements are very different. Nevertheless, even if the difference is large, it is difficult to determine which measurement is an outlier. LODTs can take various forms, see [81]. In this paper, the LODT is characterized by the probabilities $q_{\theta_i\theta_j} = \mathbb{P}\left\{Y_{i,j} = 1 \mid \theta_i, \theta_j\right\}$, with $\theta_i \in \Theta$ and $\theta_j \in \Theta$. For example, q_{00} is the probability that an outlier is detected when data are provided by good sensors. One has $q_{\theta_i\theta_j} = q_{\theta_j\theta_i}$ as $y_{ij} = y_{ji}$ (symmetry). One further assumes that $q_{00} < q_{01} = q_{10} \leqslant q_{11}$, which is reasonable, since the outcome of a LODT is more likely to be 1 as the number of outliers involved increases.

I introduce now the DFD algorithm presented in [82] and then discusses the behavior of misbehaving nodes.

In the DFD algorithm [82], each node manages two counters $c_{\mathrm{m},i}(t)$ and $c_{\mathrm{d},i}(t)$ with $c_{\mathrm{m},i}(0) = c_{\mathrm{d},i}(0) = 0$. Using $c_{\mathrm{m},i}(t)$, Node i counts the number of LODTs that it has performed. Using $c_{\mathrm{d},i}(t)$, Node i counts the number of LODTs resulting in the detection of outliers, *i.e.*, $y_{i\cdot} = 1$. Consider ν as a constant decision threshold, Node i sets its own estimate $\widehat{\theta_i}\,(t) = 1$ if $c_{\mathrm{d},i}(t)/c_{\mathrm{m},i}(t) \geqslant \nu$. Otherwise, it sets $\widehat{\theta_i}\,(t) = 0$. Only the nodes with $\widehat{\theta}\,(t) = 0$ can send their data to the nodes met at time t. Each node performs a LODT and updates its counters only when it has received some data from another node. For example, assume that Node i with $\widehat{\theta_i}\,(t) = 1$ meets Node j at time t. Node i still takes measurements, but it does not send these data to Node j. If $\widehat{\theta_j}\,(t) = 0$, then Node i can receive the data from Node j and perform a LODT. To simplify the analysis, one has chosen to consider the evolution of $c_{\mathrm{m},i}(t)$ and $c_{\mathrm{d},i}(t)$ over a sliding time window containing the time instants of the last M meetings during which Node i has performed a LODT. Algorithm 8 summarizes the proposed DFD technique for an arbitrary normal Node $i \in \mathcal{N}$.

To disturb the behavior of Algorithm 8, a Byzantine Node b may set $\widehat{\theta_b}\,(t) = 0$, $\forall t \geqslant 0$, so that it always indicates to the encountered nodes that it is well behaving and that it trusts its sensors. Then Node b may transmit some artificial data to mislead the other nodes. Two types of behavior are considered in what follows. *B1)* Node b always transmits random quantities to the encountered nodes. These random data are usually outliers. Therefore, q_{20} and q_{21} are close to 1. *B2)* Node b performs a measurement m_b and always waits for the data m_i coming from the encountered Node i. If m_i is close to m_b then it is likely that Node i is carrying good sensors.

Algorithm 8 Sliding-Window DFD algorithm for Node $i \in \mathcal{N}$.

```
1. Initialize $t_i^0 = 0$, $\widehat{\theta}_i\left(t_i^0\right) = 0$, $c_{\mathrm{m},i}(t_i^0) = c_{\mathrm{d},i}(t_i^0) = 0$, $\iota = 1$, and $\mu = 0$.
2. Do $\widehat{\theta}_i(t) = \widehat{\theta}_i\left(t_i^{\iota-1}\right)$, $c_{\mathrm{m},i}(t) = c_{\mathrm{m},i}\left(t_i^{\iota-1}\right)$, $c_{\mathrm{d},i}(t) = c_{\mathrm{d},i}\left(t_i^{\iota-1}\right)$, and $t = t + \delta t$ until the $\iota$-th
   meeting occurs at time $t_i^\iota$ with Node $j^\iota \in \mathcal{S}$.
3. Perform local measurement of data $m_i\left(t_i^\iota\right)$.
4. If $\widehat{\theta}_i\left(t_i^\iota\right) = 0$, then transmit $m_i\left(t_i^\iota\right)$ to Node $j^\iota$.
5. If data $m_{j^\iota}$ have been received from Node $j^\iota$, then
   (a) $\mu = \mu + 1$. Perform a LODT with outcome $y_i^\mu$.
   (b) Update $c_{\mathrm{m},i}$ and $c_{\mathrm{d},i}$ as
       $\begin{cases} c_{\mathrm{m},i}(t_i^\iota) = \min\{\mu, M\} \\ c_{\mathrm{d},i}(t_i^\iota) = \sum_{m=\max\{1,\mu-M+1\}}^{\mu} y_i^m \end{cases}$   (4.1)
   (c) Update $\widehat{\theta}_i$ as follows
       $\widehat{\theta}_i(t_i^\iota) = \begin{cases} 1 & \text{if } c_{\mathrm{d},i}(t_i^\iota)/c_{\mathrm{m},i}(t_i^\iota) \geqslant \nu, \\ 0 & \text{else} \end{cases}$   (4.2)
6. $\iota = \iota + 1$. Go to 2.
```

To introduce confusion, Node b does not send m_b, but sends a significantly different quantity to Node i. If m_i is very different from m_b, it is likely that Node i is carrying a defective sensor. To increase confusion, Node b transmits to Node i a quantity similar to m_i. In this case q_{20} is close to 1 and q_{21} is close to 0.

Define the triple $\mathbf{x}_i(t) = \left(\theta_i, c_{\mathrm{m},i}(t), c_{\mathrm{d},i}(t)\right)$ to represent the state of Node $i \in \mathcal{N}$. The evolution of the state of Node i, conditioned by its status θ_i, follows a Markov model. In particular, there are two chains as $\theta \in \{0, 1\}$. In order to simplify the notations, let $c_{\mathrm{m},i}(t) = \ell$ and $c_{\mathrm{d},i}(t) = k$. At time t, among the nodes with status $\theta \in \{0, 1\}$, denote $X_\theta^{\ell,k}(t)$ as the proportion of nodes in state $\mathbf{x}(t) = (\theta, \ell, k)$. The state transition probabilities of nodes are evaluated in Section 4.2.1. Then the evolution of $X_\theta^{\ell,k}(t)$ are described in Section 4.2.2.

4.2.1 Transition probabilities

Define $\pi_\theta^{\delta_\mathrm{m},\delta_\mathrm{d}}$ as the transition probability from State (θ,ℓ,k) to State $(\theta,\ell+\delta_\mathrm{m},k+\delta_\mathrm{d})$. In case where $c_{\mathrm{m},i}(t)=\ell<M$, the counter $c_{\mathrm{m},i}(t)$ either increases or remains constant, thus $(\delta_\mathrm{m},\delta_\mathrm{d})\in\{(0,0),(1,0),(1,1)\}$. The only possibility leading to $\delta_\mathrm{m}=0$ is that Node J is not a Byzantine node and $\widehat{\theta}_J(t)=1$. Therefore, for any $\theta\in\{0,1\}$,

$$\begin{aligned}\pi_\theta^{0,0}(t,\ell,k)&=\sum_{\theta\in\{0,1\}}\mathbb{P}\{\theta_J=\theta\}\,\mathbb{P}\left\{\widehat{\theta}_J(t)=1|\theta_J=\theta\right\}\\&=p_0p^{01}(t)+p_1p^{11}(t),\end{aligned}\tag{4.3}$$

where $p_\theta=\mathbb{P}\{\theta_J=\theta\}$ by the assumption that the nodes are well mixed. One introduces

$$p^{\theta\widehat{\theta}}(t)=\mathbb{P}\left\{\widehat{\theta}_J(t)=\widehat{\theta}|\theta_J=\theta\right\},\tag{4.4}$$

which is the proportion of agents with status θ believing their status is $\widehat{\theta}$. Notice that $p^{\theta\widehat{\theta}}(t)$ can be obtained from $X_\theta^{\ell,k}(t)$ according to the decision rule (4.2), *i.e.*,

$$\begin{cases}p^{\theta0}(t)&=X_\theta^{0,0}(t)+\sum_{\ell,k:k/\ell<\nu}X_\theta^{\ell,k}(t).\\p^{\theta1}(t)&=\sum_{\ell,k:k/\ell\geqslant\nu}X_\theta^{\ell,k}(t).\end{cases}\tag{4.5}$$

A state transition occurs with $(\delta_\mathrm{m},\delta_\mathrm{d})=(1,1)$ when Node i with status $\theta_i=\theta$ meets Node J with $\widehat{\theta}_J(t)=0$ *and* when the LODT yields $y_i(t)=1$. The two events are independent, hence

$$\begin{aligned}\pi_\theta^{1,1}(t,\ell,k)&=\sum_{\phi\in\Theta}\mathbb{P}\left\{Y_{iJ}(t)=1,\theta_J=\phi,\widehat{\theta}_J(t)=0|\theta_i=\theta\right\}\\=&\sum_{\phi\in\Theta}\mathbb{P}\{\theta_J=\phi\}\,\mathbb{P}\left\{\widehat{\theta}_J(t)=0|\theta_J=\theta\right\}\cdot\\&\cdot\mathbb{P}\{Y_i(t)=1|\theta_i=\theta,\theta_J=\phi\}=\sum_{\phi\in\Theta}p_\phi q_{\theta\phi}p^{\phi0}(t).\end{aligned}\tag{4.6}$$

Since the Byzantine nodes with $\theta_b=2$ always indicate $\widehat{\theta}_b=0$, one may rewrite (4.6) as

$$\pi_\theta^{1,1}(t,\ell,k)=p_2q_{\theta2}+\sum_{\phi\in\{0,1\}}p_\phi q_{\theta\phi}p^{\phi0}(t).\tag{4.7}$$

Finally, $\pi_\theta^{1,0}(t,\ell,k)=\mathbb{P}\left\{Y_i(t)=0,\widehat{\theta}_J(t)=0|\theta_i=\theta\right\}$ is obtained similarly from (4.6)

$$\pi_\theta^{1,0}(t,\ell,k)=p_2(1-q_{\theta2})+\sum_{\phi\in\{0,1\}}p_\phi\left(1-q_{\theta\phi}\right)p^{\phi0}(t).\tag{4.8}$$

In the case where $c_{\mathrm{m},i}(t)=M$, one has $\delta_\mathrm{m}=0$ as the counter $c_{\mathrm{m},i}(t)$ reaches its maximum value. In Algorithm 8, μ is the number of LODTs performed by Node i up to time t. When $\mu\geqslant M$, only the last M LODT outcomes are considered: LODT outcomes y_i^m with $m \leqslant \mu - M$ are discarded. Consider the random event $\mathcal{E}_y(t) = \left\{Y_i^{\mu-M} = y \mid \sum_{m=\mu-M}^{\mu-1} Y_i^m = k\right\}$ in which $y \in \{0,1\}$. This event represents a situation where one knows that k LODTs were positive among the last M tests and the old LODT outcome that will be discarded once the new LODT outcome is available, also concluded in the presence of defective sensors. As discussed in [82], one has $\mathbb{P}\{\mathcal{E}_1(t)\} \approx k/M$ and $\mathbb{P}\{\mathcal{E}_0(t)\} \approx 1 - k/M$. Assume that the $(\mu - M)$-th LODT performed by Node i occurred at time $\tilde{t}$, then $y_i^{\mu-M}$ can also be denoted as $y_i\left(\tilde{t}\right)$ and $\delta_{\mathrm{d}} = y_i(t) - y_i\left(\tilde{t}\right) \in \{-1,0,1\}$. To have $(\delta_{\mathrm{m}}, \delta_{\mathrm{d}}) = (0,1)$, three independent events have to occur: 1) the encountered Node J has $\widehat{\theta}_J(t) = 0$; 2) $y_i(t) = 1$; 3) $y_i\left(\tilde{t}\right) = 0$, *i.e.*, $\mathcal{E}_0(t)$. The transition probability is then deduced using derivations similar to (4.6),

$$\pi_\theta^{0,1}(t,M,k) = \frac{M-k}{M}\left(p_2 q_{\theta 2} + \sum_{\phi\in\{0,1\}} p_\phi q_{\theta\phi} p^{\phi 0}(t)\right). \tag{4.9}$$

Consider then $(\delta_{\mathrm{m}}, \delta_{\mathrm{d}}) = (0,-1)$, similarly, one obtains,

$$\begin{aligned}&\pi_\theta^{0,-1}(t,M,k)\\ &= \frac{k}{M}\left(p_2\left(1-q_{\theta 2}\right) + \sum_{\phi\in\{0,1\}} p_\phi\left(1-q_{\theta\phi}\right) p^{\phi 0}(t)\right).\end{aligned} \tag{4.10}$$

Considering the last transition $(\delta_{\mathrm{m}}, \delta_{\mathrm{d}}) = (0,0)$. To obtain the expression of $\pi_\theta^{0,0}(t,M,k)$, one needs to introduce (4.9-4.10) into $\pi_\theta^{0,0}(t,M,k) = 1 - \pi_\theta^{0,1}(t,M,k) - \pi_\theta^{0,-1}(t,M,k)$.

4.2.2 Macroscopic evolution

With the transition probabilities discussed in Section 4.2.1 and the initial conditions

$$X_\theta^{0,0}(0) = 1, \text{ and } X_\theta^{\ell,k}(0) = 0, \forall \ell, k \neq 0, \tag{4.11}$$

the evolution of the various proportions $X_\theta^{\ell,k}(t)$ of nodes in the corresponding states can be obtained, see [82] for the detail. To simplify the equations, consider the function

$$Z_\theta^{\delta_{\mathrm{m}},\delta_{\mathrm{d}}}(\ell,k,t) = \begin{cases} X_\theta^{\ell,k}(t)\,\pi_\theta^{\delta_{\mathrm{m}},\delta_{\mathrm{d}}}(\ell,k), & \text{if } 0 \leqslant k \leqslant \ell \leqslant M,\\ 0, & \text{otherwise,}\end{cases} \tag{4.12}$$

then for any $\theta \in \{0,1\}$, one has

$$\begin{cases} \frac{dX_\theta^{\ell,k}}{dt} \overset{(a)}{=} \lambda \sum\limits_{\delta_{\mathrm{d}}\in\{0,1\}} \left(Z_\theta^{1,\delta_{\mathrm{d}}}(\ell-1,k-\delta_{\mathrm{d}},t) - Z_\theta^{1,\delta_{\mathrm{d}}}(\ell,k,t)\right)\\ \frac{dX_\theta^{M,k}}{dt} \overset{(b)}{=} \lambda \sum\limits_{\delta_{\mathrm{d}}\in\{-1,1\}} \left(Z_\theta^{0,\delta_{\mathrm{d}}}(M,k-\delta_{\mathrm{d}},t) - Z_\theta^{0,\delta_{\mathrm{d}}}(M,k,t)\right)\\ \qquad\qquad +\lambda \sum\limits_{\delta_{\mathrm{d}}\in\{0,1\}} Z_\theta^{1,\delta_{\mathrm{d}}}(M-1,k-\delta_{\mathrm{d}},t), \end{cases} \tag{4.13}$$

where (a) describes the evolution of the proportion of state components in the transient regime and (b) is for the permanent regime.

$$\begin{cases} \overline{p}^{00} = \sum_{k:k/M<v} \binom{M}{k} \left(\frac{p_0 q_{00} \overline{p}^{00} + p_1 q_{01} \overline{p}^{10} + p_2 q_{02}}{p_0 \overline{p}^{00} + p_1 \overline{p}^{10} + p_2}\right)^k \left(1 - \frac{p_0 q_{00} \overline{p}^{00} + p_1 q_{01} \overline{p}^{10} + p_2 q_{02}}{p_0 \overline{p}^{00} + p_1 \overline{p}^{10} + p_2}\right)^{M-k}, \\ \overline{p}^{10} = \sum_{k:k/M<v} \binom{M}{k} \left(\frac{p_0 q_{10} \overline{p}^{00} + p_1 q_{11} \overline{p}^{10} + p_2 q_{12}}{p_0 \overline{p}^{00} + p_1 \overline{p}^{10} + p_2}\right)^k \left(1 - \frac{p_0 q_{10} \overline{p}^{00} + p_1 q_{11} \overline{p}^{10} + p_2 q_{12}}{p_0 \overline{p}^{00} + p_1 \overline{p}^{10} + p_2}\right)^{M-k}. \end{cases} \tag{4.16}$$

4.3 Analysis of the Equilibrium

In this section, I investigate the asymptotic behavior of the DTN state equations (4.13). Algorithm 8 may drive $X_\theta^{\ell,k}$ to an equilibrium $\overline{X}_\theta^{\ell,k}$ at which the proportions of nodes in different states $X_\theta^{\ell,k}(t)$ do not vary any more. As a consequence, $p^{\theta 0}(t)$ defined in (4.5) also tends to an equilibrium $\overline{p}^{\theta 0}$.

4.3.1 Equilibrium of $X_\theta^{\ell,k}$

The results presented in this section are the extension of those in [82] by considering the affect of Byzantine attack.

Proposition 1 *Assume that the dynamic system described by (4.13) admits some equilibrium* $\overline{X}_\theta^{\ell,k}$*, then* $\overline{\mathbf{p}} = \left(\overline{p}^{00}, \overline{p}^{10}\right)$ *is the solution of* (4.16) *(at the top of the next page) and for any* $\theta \in \{0,1\}$ *and* $k \leqslant \ell$*,*

$$\overline{X}_\theta^{\ell,k} = \begin{cases} 0, & \forall \ell < M, \\ \binom{M}{k} \left(h_\theta\left(\overline{\mathbf{p}}\right)\right)^k \left(1 - h_\theta\left(\overline{\mathbf{p}}\right)\right)^{M-k}, & \ell = M, \end{cases} \tag{4.14}$$

where

$$h_\theta\left(\overline{\mathbf{p}}\right) = \frac{p_0 q_{\theta 0} \overline{p}^{00} + p_1 q_{\theta 1} \overline{p}^{10} + p_2 q_{\theta 2}}{p_0 \overline{p}^{00} + p_1 \overline{p}^{10} + p_2}. \tag{4.15}$$

Proposition (1) can be proved using derivations similar to those presented in [82]. Proposition (1) provides non-linear equations (4.16) that have to be satisfied by $\overline{\mathbf{p}}$. With the solutions of (4.16), the values of $\overline{X}_\theta^{M,k}$ at equilibrium can be easily deduced.

4.3.2 Approximations of the Equilibrium

Closed-form expressions for $\overline{p}^{00}$ and $\overline{p}^{10}$ are difficult to obtain from (4.16). Here, I introduce an approximation of (4.16) from which some insights may be obtained on the way v should be chosen to minimize the impact of the presence of misbehaving nodes. Since both $\overline{p}^{10}$ and $\overline{p}^{01}$ represent the proportions of nodes having wrong estimates of their status, the values of $\overline{p}^{10}$ and

$\overline{p}^{01}$ should be small. Thus one may consider the following approximations

$$\begin{aligned}\widetilde{h}_{\theta} &= \lim_{\left(\overline{p}^{00},\overline{p}^{10}\right)\to(1,0)} \frac{p_0 q_{\theta 0}\overline{p}^{00} + p_1 q_{\theta 1}\overline{p}^{10} + p_2 q_{\theta 2}}{p_0\overline{p}^{00} + p_1\overline{p}^{10} + p_2} \\ &= \frac{p_0 q_{\theta 0} + p_2 q_{\theta 2}}{p_0 + p_2}. \end{aligned} \tag{4.17}$$

Therefore, (4.16) may be rewritten as

$$\begin{cases} \widetilde{p}^{00} = \sum\limits_{k:k/M<\nu} \binom{M}{k} \left(\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^{M-k}, \\ \widetilde{p}^{10} = \sum\limits_{k:k/M<\nu} \binom{M}{k} \left(\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^{M-k}. \end{cases} \tag{4.18}$$

from which one deduces the approximate values $\widetilde{X}_{\theta}^{M,k}$ of $\overline{X}_{\theta}^{M,k}$

$$\begin{cases} \widetilde{X}_0^{M,k} = \binom{M}{k} \left(\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{00}+p_2 q_{02}}{p_0+p_2}\right)^{M-k}, \\ \widetilde{X}_1^{M,k} = \binom{M}{k} \left(\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^k \left(1-\frac{p_0 q_{10}+p_2 q_{12}}{p_0+p_2}\right)^{M-k}. \end{cases} \tag{4.19}$$

The quality of the approximation can be verified by checking whether there exists some value of ν that leads to both $\overline{p}^{00} \longrightarrow 1$ (or $\overline{p}^{01} \longrightarrow 0$) and $\overline{p}^{10} \longrightarrow 0$. Consider here a toy example: fix $M = 20$ and the LODT is such that $q_{00} = 0.05$ and $q_{10} = 0.8$. The Byzantine nodes have the behavior of type $B2$) with $p_{02} = 1$ and $p_{12} = 0$, which corresponds to the most serious attack. Consider $p_2 \in \{0, 0.01, 0.05, 0.1\}$ and $p_0 = p_1 = (1 - p_2)/2$ in all the cases, Figure 4.1 presents $\widetilde{p}^{10}$ as a function of $\widetilde{p}^{01}$, obtained for different values of $\nu \in [0, 1]$. One observes that the Byzantine nodes have limited influence on the performance of the DFD algorithm, except when p_2 reaches 10%. Nevertheless, if the values of M and ν are properly chosen, both $\widetilde{p}^{01}$ and $\widetilde{p}^{10}$ can be kept relatively small even in presence of 10% of Byzantine nodes. Figure 4.1 is also helpful to choose the value of ν in order to meet different performance requirements.

4.4 Numerical results

This section provides simulation results to illustrate the theoretical results above presented. At first, I consider results obtained considering nodes with an idealized displacement model. Real databases are then considered.

4.4.1 Idealized displacement model

Consider a DTN consisting of 1000 moving nodes, with their initial positions uniformly distributed over a unit square. Nodes randomly move within this square. Two nodes communicate

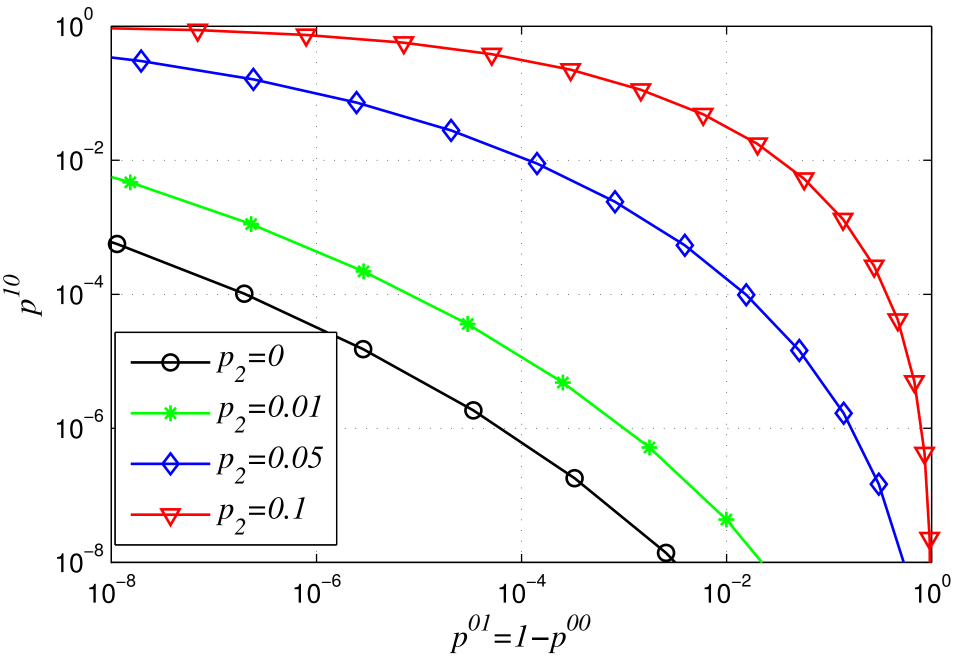


Figure 4.1: Approximate p^{10} as a function of approximate p^{01} at equilibrium, for various $v \in [0, 1]$ and $p_2 \in \{0, 0.01, 0.05, 0.1\}$.

only when their distance is less than their communication range r_0 at discrete time instants $k\Delta t$, $k = 1, 2 \ldots$. One assumes an idealized displacement model: the location of each agent at time $(k+1)\,t$ is independent of its previous location at time $k\Delta t$. The value of r_0 can be chosen to adjust the inter-contact probability during a time interval of duration Δt. Here, the inter-contact probability is taken as 0.33. Consider $N_\mathrm{b} = 50$ Byzantine nodes and $N_\mathrm{d} = 200$ nodes with defective sensors, which leads to $p_0 = 0.75$, $p_1 = 0.2$, and $p_2 = 0.05$. The characteristics of the LODT are $q_{00} = 0.05$, $q_{01} = 0.8$, and $q_{11} = 0.9$. Consider both types of Byzantine nodes: for the type B1), assume that $p_{02} = p_{12} = 1$; for the type B2), assume that $p_{02} = 1$ and $p_{12} = 0$. One also takes into account the situation where no Byzantine node is present, *i.e.*, $p_2 = 0$, in order to see the influence of Byzantine attack. In the latter case, one sets $N_\mathrm{d} = 211$ so that the ratio of p_0 and p_1 are close in all the situations. Figure 4.2 presents the evolution of p^{01} and p^{10} as functions of time, with $M = 15$ and $v = 0.4$. Recall that p^{01} is the proportion of normal nodes with good sensors that wrongly decide their sensors as defective and p^{10} is the proportion of normal nodes with defective sensors that wrongly decide their sensors as good. Compared with the situation where $p_2 = 0$, one observes that both p^{01} and p^{10} decrease slower when the Byzantine nodes are present. As expected, the attack of type B2) impact more the agents compared than that of type B1). Figure 4.3 shows a good match between the distribution of $X_\theta^{M,k}$ obtained by the end of the simulation and the approximation of $X_\theta^{M,k}$ using (4.19). In order to have a good performance of

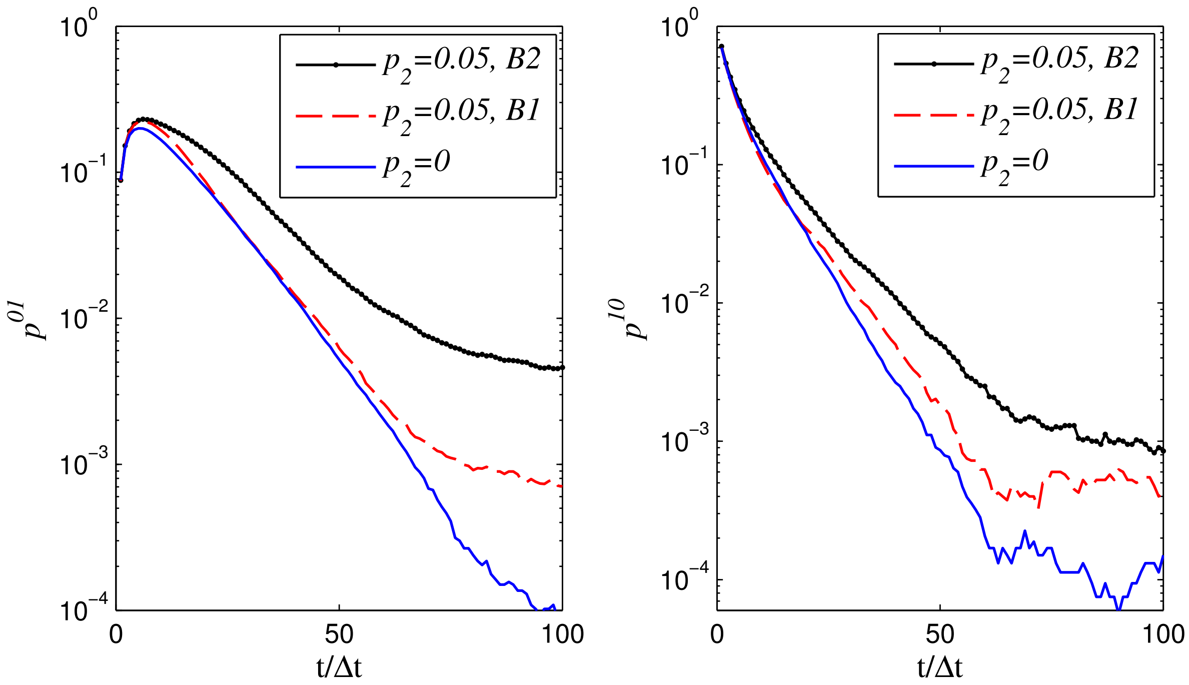


Figure 4.2: Evolution of p^{01} (left) and p^{10} (right), considering an idealized displacement model.

the DFD algorithm, the distributions of $X_0^{M,k}$ and $X_1^{M,k}$ should be as separate as possible. The main influence of the Byzantine attack is that it makes the two distributions closer. Nevertheless, the DFD algorithm still behaves in a satisfying way if the parameter ν is properly chosen using (4.16): in the simulations both p^{01} and p^{10} can be made less then 1%.

4.4.2 Simulation with real databases

In this section, the DFD algorithm is executed considering node inter-contact times taken from real databases provided by the Haggle Project [84] and by our own experiments conducted at the EuWin platform at University of Bologna. I realized the real database in Bologna using the EuWin platform. Students of the university of Bologna have been equipped of a device during a break of an academic course, then they have spent the break as usual while each device counted the number of meetings with other devices. In the simulation, one is interested in the inter-contact trace, *i.e.*, which pair of agents have a meeting at which time. I use the following databases:

- *Infocom05*, in which $N = 41$, lasted 3 days.

- *Bologna16,* in which $N = 34$, during the break of a course (which lasts about 17 minutes).

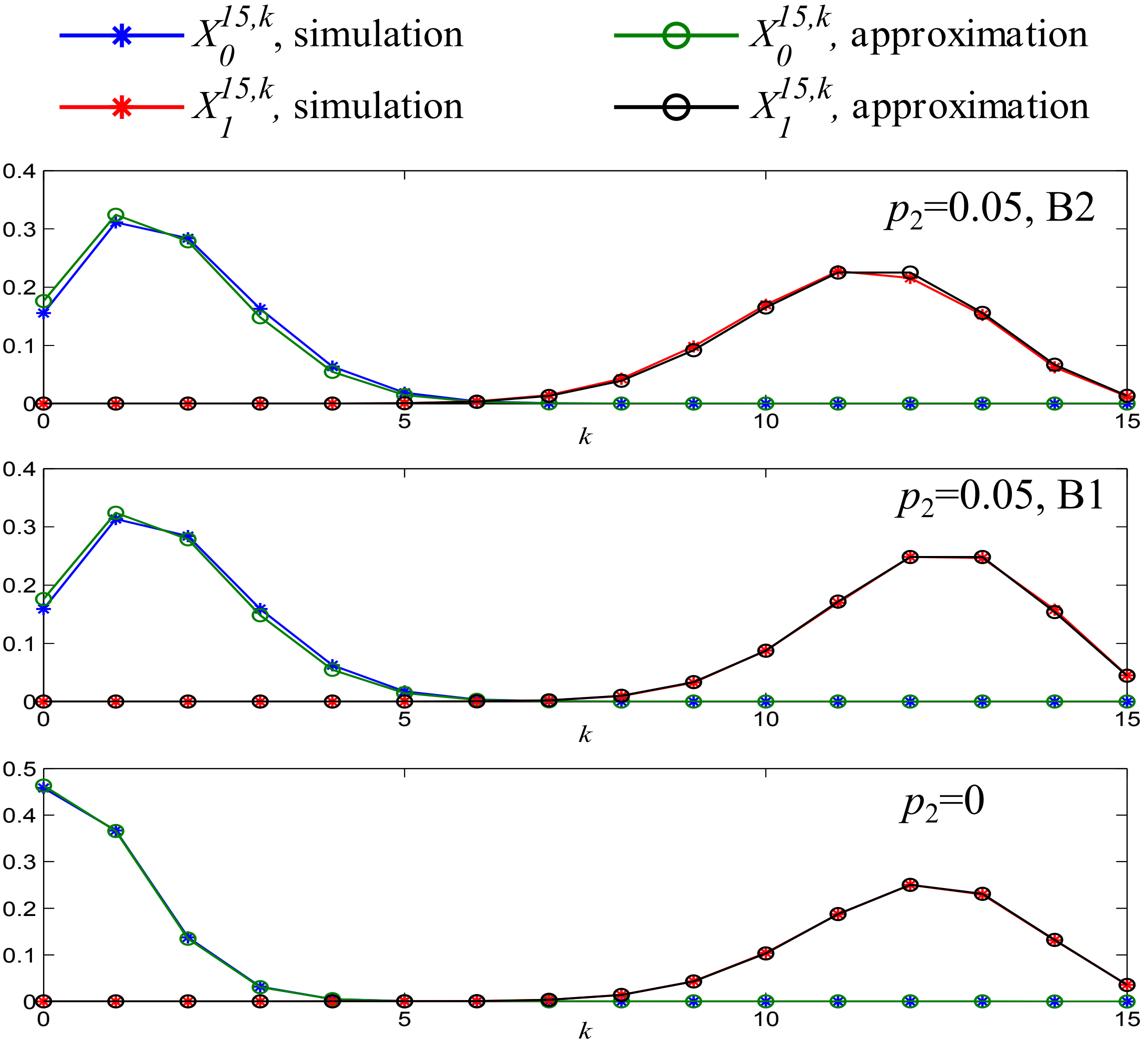


Figure 4.3: Comparison of $X_\theta^{15,k}$ at the equilibrium, when 5% of nodes perform a byzantine attack of type B2 (top), of type B1 (middle), and when there are no Byzantine nodes (bottom).

For each database, 500 Monte-Carlo simulations are performed. In each simulation, one randomly choose N_b nodes as Byzantine nodes and N_d nodes as the ones with defective sensors. The results are then averaged over these simulations. In *Infocom05*, one sets $N_\mathrm{b} = 2$ and $N_\mathrm{d} = 10$. In *Bologna16*, one sets $N_\mathrm{b} = 1$ and $N_\mathrm{d} = 6$. Consider the following parameters: $q_{00} = 0.05$, $q_{01} = 0.8$, $q_{11} = 0.9$, $p_{02} = 1$, $p_{12} = 0$, $M = 15$ and $\nu = 0.4$. At the top of Figure 4.4, the index of the active nodes (which have contact with the others) are presented at each time to show the frequency of the inter-contacts at different epochs. The evolution of p^{10} and p^{01} is plotted at the bottom of Figure 4.4. Interestingly, both p^{10} and p^{01} obtained by both databases decrease to 10^{-2} after a sufficient long time. The decreasing speed of p^{10} and p^{01} is highly related to the inter-contact rate (reflected by the density of points in the sub-figures at the top): using *Infocom05*, variations are significant at beginning of working hours; using Bologna16, p^{10} and p^{01} decrease significantly in the end as all the students came back to the class. Figure 4.5 represents the proportion of nodes in each state $X_\theta^{M,k}$ in the end of the simulation, obtained by using the databases *Infocom05* and *Bologna16*. The simulation results are compared with the approximation (4.19). One still finds that there is a good match by using the databases.

4.5 Conclusion

In this work, I investigated the impact of Byzantine attacks on the performance of a distributed faulty node detection algorithm in the context of delay tolerant networks. The aim of the algorithm is to make each normal node estimate the status of its own sensors, whereas some Byzantine nodes attempt to compromise the effectiveness of the algorithm. The affect of Byzantine attack on the equilibrium is analyzed theoretically, which is helpful to adjust the algorithm parameters in order to ensure the robustness of the DFD algorithm. Both ideal movement model and real databases have been considered in the simulations to illustrate the achieved results.

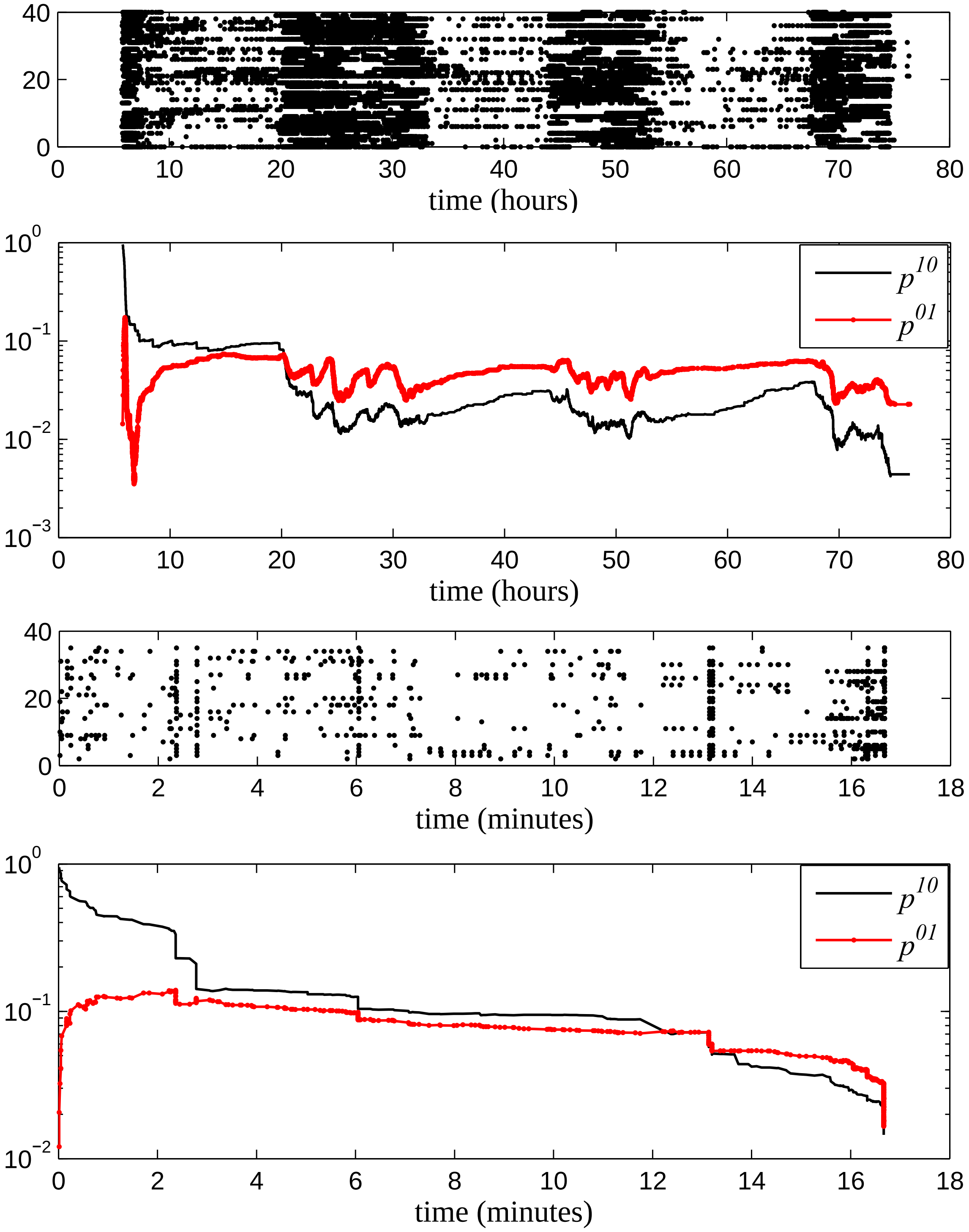


Figure 4.4: Indexes of active nodes (having met another node) at different time (top) and evolution of p^{10} and p^{01}, obtained using the *Infocom05* database (left) and the *Bologna16* database (right).

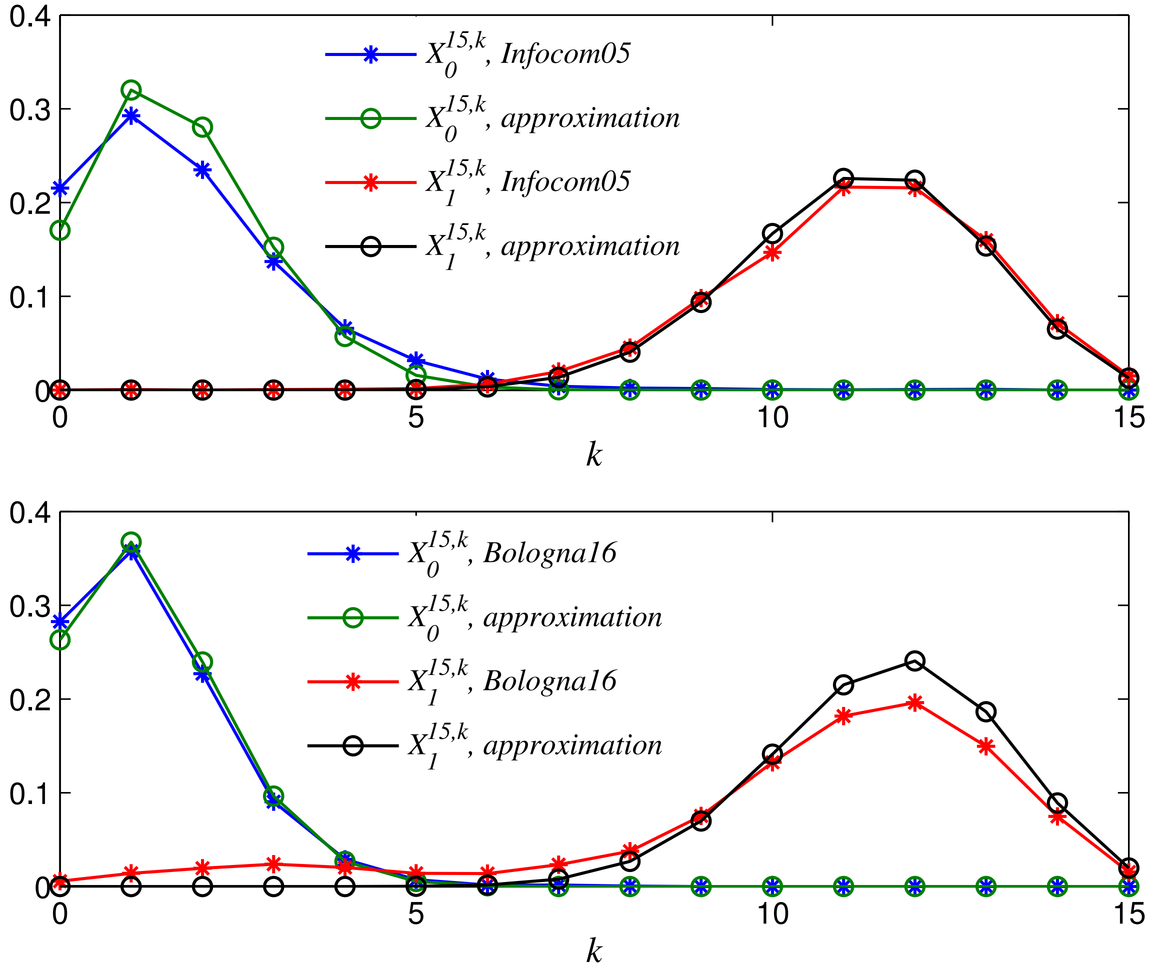


Figure 4.5: Values of $X_\theta^{15,k}$ the end of the simulation, as well as the theoretical values at equilibrium obtained from (4.19), obtained using the *Infocom05* database (left) and the *Bologna16* database (right).

Chapter 5

Visible Light Communication and its application to Vehicular Networks

5.1 Introduction

The paradigm of connected vehicles is moving from research to implementation, thus enabling new applications that start from safety improvement and widen to the so called Internet of Vehicles (IoV). The wide diffusion of mobile connected devices increases the data produced and consumed, highlighting the necessity of very high performance networks. In this context, Vehicular Sensor Networks (VSNs) may represent the future for the environmental monitoring. In fact, devices on board of vehicles, named on board units (OBUs), act as sensors, acquiring a variety of information such as position, speed, and acceleration of the vehicles themselves, but also pollution measurements, video recording of the environment, number of connected devices. This leads to the production of a large amount of information that can be exploited to enable a variety of new services addressing safety, traffic management, smart navigation, pollution measurements, urban surveillance, forensic investigations, and Internet access demanding for very high level performance networks with extremely low latency [85, 86].

In the next few years, connected vehicles will travel on the roads exchanging information with one another and with the infrastructure; the collaboration will permit safer travels, more efficient traffic management, and new services for drivers and passengers. The first steps towards this vision are being taken in many Countries around the World. In the United States (US), steps were taken around August 2014, when the National Highway Traffic Safety Administration (NHTSA), one of the main agencies in the field of transportation, issued an Advance Notice to proceed with standardization of vehicle to vehicle communication for light vehicles [87]. This means that new vehicles in the US will soon be equipped with the WAVE protocol suite for short range communications, based on IEEE 802.11p at the lower levels of the protocol stack [88, 89] and using the DSRC frequency bands. In the European Union (EU), even if there is still no

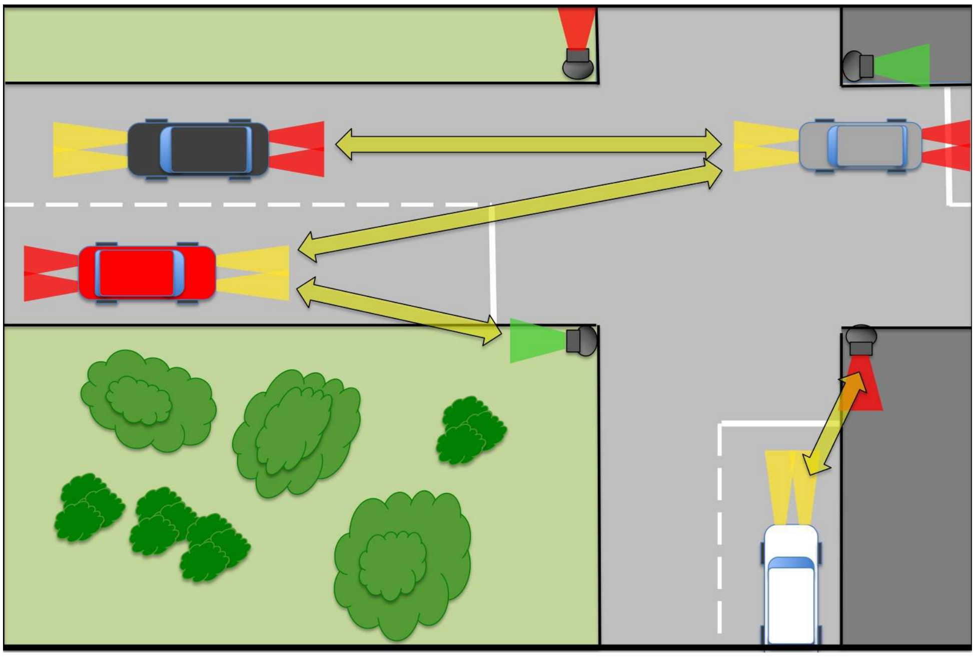


Figure 5.1: Vehicular visible light networks.

mandate from governments, important activities are being carried out. In particular, the so called Release 1 of the set of standards for cooperative intelligence transport systems (C-ITS) was issued in February 2014 by the European Committee for Standardization (CEN) and the European Telecommunications Standards Institute (ETSI) [90]. Differently from the US, various technologies are envisioned as enabler of connected vehicles, and particular attention is being posed on cellular networks. In the EU, the LTE technology can thus be considered as another key enabler of connected vehicles [91, 92].

The availability of wireless communications will enable the creation of vehicular networks with a wide range of new applications [93–97]. Great attention is obviously devoted to safety improvement, thanks to neighbor discovery and tracking and the immediate warning of critical events, like accidents in the proximity. In addition, connected vehicles will also form, with fixed RSUs as gateways, the so called IoV, with other data services that include traffic management improvement or entertainment applications.

The OBUs can be connected through different wireless technologies. Today, the OBUs, transmit their data through the cellular network [98, 99]. The wide spreading of this service is leading to high loads for the cellular network itself (that results almost sature in urban environments) and, consequently, to higher costs. A solution that could be soon available is the use of short range vehicle-to-vehicle (V2V) and vehicle-to-roadside (V2R) communications to offload part of the cellular transmissions [100, 101]. In Europe, for instance, the Release 1 of standards for cooperative intelligent transportation systems (C-ITS) has been completed, which derives from wireless access in vehicular environment (WAVE)/IEEE 802.11p in the 5.9 GHz band and adapts to European requirements [88, 89, 98].

Although presently the fight is tackled in the radio frequency (RF) band, with short range communications (with the IEEE 802.11p standard in the DSRC band) and cellular communications (mainly focusing on LTE), great interest is also devoted to the visible light spectrum. In fact, the great development made by light emitting diodes (LEDs) in the last years allows the implementation of a variety of applications based on VLC. This technology enables transmissions in free and unregulated channels with very wide bandwidths while consuming low power and without producing electromagnetic radiations and health problems. Energy saving, high data rate secure connectivity, safe communication in hazardous environments, services to passengers on aircrafts, underwater communications, location based services, and vehicular communications are only some examples of possible applications [102–105]. The enormous spread of LEDs and its huge communication potential, led in fact VLC to the introduction in the family of standards for wireless communications, by 2011, in the IEEE 802.15.7 specifications [106] which defines the physical (PHY) and MAC layers for optical wireless communications (OWCs) in the visible light spectrum. Exploiting the already mounting LED lights, VLC could be used in several application scenarios (such as underwater communications [107] or localization and tracking [108]), and also vehicles could be connected to each other to create the so called vehicular visible light networks (VVLNs) (a.k.a. V^2LC networks [109]), as represented in Figure 5.1.

Differently from RF, the visible light spectrum offers large portions of unlicensed and uncongested bands. In addition to the potentially high throughput guaranteed by the low congested frequencies, the large bandwidth, and the optimal spatial reuse, VLC is also characterized by a high directivity and a predictable channel; these aspects allow high accurate neighbors positioning without use of other technologies [110], reduce the sources of interference [6], and guarantee a high security level due to the inherently reserved channels [111, 112].

The high directivity also implies, however, the need for almost clear line of sight that limits the use of VLC to the applications where no obstacles must be overtaken and only single or multiple hops between vehicles that are traveling on the same road are needed. Besides pure VVLNs, anyway, VLC can be foreseen in heterogenous vehicular networks as an addition to the RF technologies to increase the overall capacity.

I focused my attention on VLC as the enabling technology for data exchanging between vehicles and between vehicles and traffic lights, here exploited as RSUs. Referring to delay tolerant applications, I aim to collect as much information as possible at the traffic lights and forward it to a remote control center. Specifically, I assume that vehicles try to transmit their data through the head or rear lights to the nearer traffic light either directly (if in visibility), or

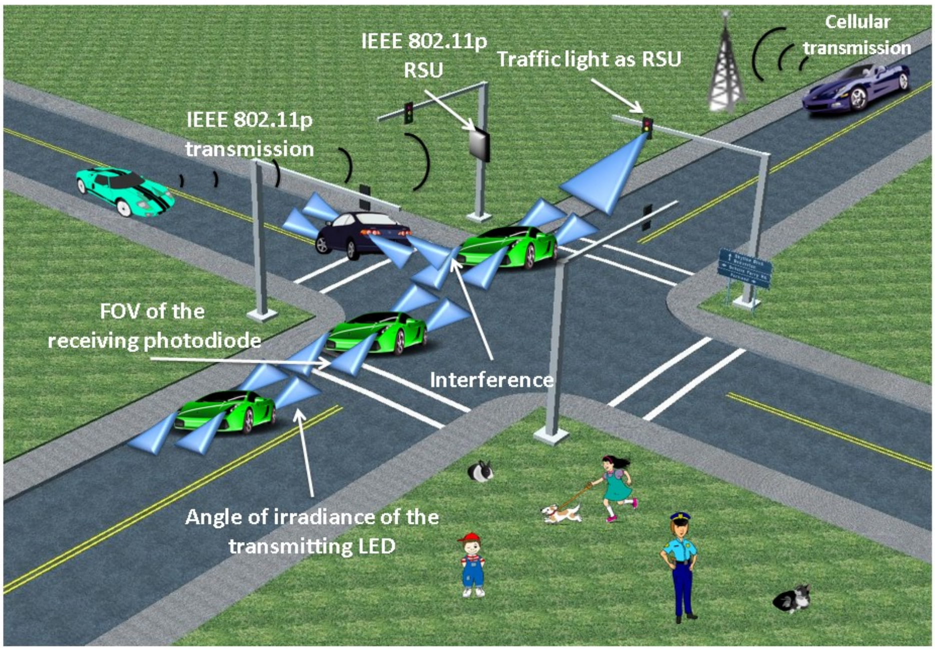

IEEE 802.11p RSU
Traffic light as RSU
Cellular transmission
IEEE 802.11p transmission
FOV of the receiving photodiode
Interference
Angle of irradiance of the transmitting LED

Figure 5.2: System model. Vehicles attempt to use short range communications based on IEEE 802.11p and/or VLC for data delivery at RSUs, which, for VLC, are represented by traffic lights. Background taken from http://www.lista.it/hve/scenario.htm.

through multi hop VLC networking. I remark that this work is among the first discussing the use of VLC in vehicular networks for purposes that are not limited to safety improvement.

In particular, I:

- Exploit VLC as the enabling technology for V2V and V2R communications aimed at data uploading in delay tolerant applications toward a remote control center.
- Investigate the feasibility of the proposed solution in terms of connectivity and evaluate the performance in terms of packets delivered, thus offloaded from the cellular infrastructure.
- Examine the impact of interference varying the transmitter and receiver characteristics, that is, assuming different transmitting distances and directivity degrees.

Therefore, the scope of this work is to introduce the paradigm of VVLNs and to highlight the improvement allowed by its integration in future heterogenous vehicular networks (Figure 5.2). To this aim, results are shown focusing on the example application of crowd sensing vehicular networks (CSVNs), where data collected by sensors on board of vehicles are delivered through single or multiple hops to RSUs, which act as gateways towards a remote control center. The strategies for the selection of the technology to be used is also discussed and a congestion-adaptive algorithm is proposed.

5.2 DSRC (WAVE/IEEE 802.11p)

WAVE defines the communication system architecture and the complementary set of services and interfaces for vehicular scenarios, whereas IEEE 802.11p describes the MAC and physical layer protocols at 5.9 GHz [88, 89]. Through the WAVE mode, this standard allows the transmission and reception of data frames with the wildcard basic service set (BSS) identity and without the need of belonging to a particular BSS. This feature enables very efficient communication-group setup without much of the overhead typically needed in nomadic IEEE 802.11a/g networks; it simplifies the BSS operations in a truly ad hoc manner for vehicular usage [88], and can be used by devices for a fast exchange of contextual data, including Global Positioning System (GPS) coordinates, direction, and speed. In particular, all OBUs periodically broadcast their identity and position in packets denoted as beacons, thus each OBU has a real time knowledge of all its neighbors.

The IEEE 802.11p amendment has been released in 2010 and several tests have been performed worldwide, even with thousands of vehicles. It is part of the WAVE protocol suite in the US and a slightly modified version denoted ITS-G5 is included in the cooperative-intelligent transport systems (CITS) in Europe. Given the large experimentation records and the large number of devices already available on the market, the main advantage of this technology is that it appears mature for a large scale deployment and still remains the main standard for V2V communications. Among the main concerns about IEEE 802.11p there are i) the possibly high level of errors under heavy traffic conditions, ii) the lack of clear plans for future enhancements of the standard, and iii) the need for deployment of completely new devices as RSUs. Regarding the technology, one of its distinctive characteristics is the use of CSMA/CA at the MAC layer. On the one hand, it allows a fully distributed and uncoordinated access to the wireless channel, with no need for a resource allocation procedure. On the other hand, however, it implies a not negligible resource waste due to frequent collisions as the channel use increases, and arises, as already anticipated, concerns about its reliability under heavy traffic conditions. At the physical layer, IEEE 802.11p is based on orthogonal frequency division multiplexing (OFDM) modulation, with seven non overlapping channels of 10 MHz each; one of these channels is reserved for control purposes and the other six are provided as service channels. Focusing on the PHY layer, modulation and convolutional coding are used, with eight possible combinations of modulation and coding schemes (MCSs), often called Modes. Depending on the adopted MCS, the gross data rate varies between 3 and 27 Mb/s, as summarized in the first columns of Table 5.1.

MCS (Mode)	Modul.	Coding Rate	Data Rate	Tx. Duration	$SINR_{min}$	Tx. Range
1	BPSK	1/2	3.0 Mb/s	573 μs	10 dB	457 m
2	BPSK	3/4	4.5 Mb/s	396 μs	11 dB	420 m
3	QPSK	1/2	6.0 Mb/s	307 μs	13 dB	355 m
4	QPSK	3/4	9.0 Mb/s	218 μs	15 dB	300 m
5	16-QAM	1/2	12.0 Mb/s	173 μs	18 dB	234 m
6	16-QAM	3/4	18.0 Mb/s	129 μs	22 dB	167 m
7	64-QAM	2/3	24.0 Mb/s	107 μs	26 dB	120 m
8	64-QAM	3/4	27.0 Mb/s	99 μs	27 dB	110 m

Table 5.1: IEEE 802.11p modes and corresponding values.

Many techniques are defined for the joint use of control and service channels, including periodic switching between them or parallel use. However, since the control channel is generally reserved to deliver data for safety purposes, using it for sending non critical messages is not appropriated; thus, in this work I assume the use of the control channel only for beacon broadcasting at 10 Hz and I consider one service channel for data transmission.

5.3 LTE

Driven by the already available and almost ubiquitous coverage of cellular systems and by the advances in the direct communications among devices, LTE is becoming a new option for connected vehicles in the recent years. Indeed, from Release 12 the device-to-device functionality and the corresponding channel structure called sidelink were introduced to enable direct communications. Motivated by the increasing interest for the vehicular market, 3GPP has then started working on specific features for V2V. Although enhancements are included in Release 13, V2V was included in Release 14 from September 2016 [113, 114]. The following two are among the main advantages of using LTE for V2V communications. First, the same technology as for cellular communications is used, which implies exploiting the same hardware and most protocols. This aspect represents a significant advantage, especially considering that i) vehicles are already becoming equipped with a cellular interface, ii) the specifications are continuously updated and iii) base stations are already deployed in large numbers. Second, resources are orthogonal, thus allowing higher multiplexing, with a possibly significant increase in reliability and capacity [115]. This advantage comes however at the cost of a higher complexity of devices and protocols. In this case, synchronization is required among the devices and the resource allocation phase becomes one of the most critical aspects, both if controlled by the network (sidelink mode 3) or autonomously performed (sidelink mode 4). In addition to concerns about resource allocation and need for precise synchronization, it is also important to remark

Bandwidth	**RBs per frame**	**RBs for control**	**RBs for random access**	**RBs for shared**
	n_{RB-fr}			$n_{RB-fr}^{(sh)}$
1.4 MHz	120	20	3	97
5 MHz	500	80	12	408
10 MHz	1000	160	24	816
20 MHz	2000	320	60	1620

Table 5.2: Resource blocks of LTE channels

MCS index	**Modul.**	**Coding Rate**	**Data Rate**	n_{RB-b}	$SINR_{min}$	$\hat{r}_{aw}$
0	QPSK	0.131	1.15 Mb/s	114	-2.79 dB	1635 m
1	QPSK	0.169	1.49 Mb/s	88	-1.38 dB	1596 m
2	QPSK	0.207	1.82 Mb/s	72	-0.21 dB	1210 m
3	QPSK	0.266	2.34 Mb/s	56	1.35 dB	1164 m
4	QPSK	0.324	2.85 Mb/s	46	2.67 dB	1119 m
5	QPSK	0.407	3.59 Mb/s	38	4.37 dB	1040 m
6	QPSK	0.484	4.26 Mb/s	32	5.78 dB	984 m
7	QPSK	0.573	5.05 Mb/s	26	7.30 dB	934 m
8	QPSK	0.645	5.68 Mb/s	24	8.49 dB	871 m
9	QPSK	0.731	6.44 Mb/s	22	9.83 dB	803 m
10	QPSK	0.804	7.08 Mb/s	20	10.96 dB	757 m
12	16-QAM	0.465	8.20 Mb/s	16	12.85 dB	700 m
13	16-QAM	0.532	9.37 Mb/s	14	14.80 dB	625 m
15	16-QAM	0.670	11.80 Mb/s	12	18.76 dB	474 m
17	16-QAM	0.744	13.12 Mb/s	10	20.88 dB	424 m
20	16-QAM	1.005	17.71 Mb/s	8	28.22 dB	249 m

Table 5.3: Considered LTE-V2V modulation and corresponding values.

that devices are not yet available on the market and on field experimentations are still being planned. Giving a look to the specifications, unlike IEEE 802.11p, LTE does not provide a single channel bandwidth and different options are possible to deal with different planning needs, as summarized in Table 5.2 [116]. At the MAC and PHY layers LTE-V2V is based on single carrier frequency division multiple access (SC-FDMA) (the same as the uplink towards the eNodeB). Advanced coding techniques and an almost continuous variation of MCS combinations are adopted, which contribute, together with the possible use of portions of the bandwidth, to a higher reliability and range with respect to IEEE 802.11p. As an example, a set of possible combinations is summarized in the first columns of Table 5.3 with reference to packets of 200 bytes and 10 MHz bandwidth (please refer to Appendix I [117] for the details on how these values have been derived).

Feature	Short range RF	Cellular networks	VLC
Today reference	WAVE/IEEE 802.11p	3GPP LTE	IEEE 802.15.7
Frequencies	5.9 GHz	400 MHz - 3.5 GHz	380-800 THz
Use of frequencies	Reserved for ITS	Licensed	Unlicensed
Communication range	Up to 1 km	Ubiquitous	Lower than 100 m
Directionality	Normally none	Normally none	High
Obstacles effect	High impact	Medium impact	Obstructing
Spatial reuse	Limited	Limited	High
Implementation costs	Requires ad hoc devices	Requires ad hoc devices	Uses the available LEDs
V2V support	Yes	Future: D2D mode	Yes
V2I support	RSUs to be deployed	Native	Traffic lights and other light sources

Table 5.4: Visible Light Communication vs. main RF technologies.

5.4 Vehicular visible light networks

This section provides an overview of the VLC technology applied to vehicular networks; after highlighting its peculiarities, the present state of the art is discussed focusing on standardization and real experimentation efforts.

5.4.1 VLC peculiarities

VLC significantly differs from the reference DSRC and LTE technologies in many aspects, including the use of unlicensed and uncongested frequencies, lower coverage and high directivity, and reuse of devices that are already deployed for other scopes. These characteristics are hereafter discussed in details and summarized in Table 5.4.

Unlicensed and uncongested bands.

One of the main advantages of VLC is that it uses an unlicensed and uncongested bandwidth, located between 380 and 800 THz. It is known that DSRC bands around 5.9 GHz have been reserved to the short range use in vehicular networks in most Countries worldwide; however, there are strong concerns and long discussions about what happens when the small number of channels provided by DSRC are used by hundreds of vehicles under congested conditions [118–120]. This issue is also present with reference to LTE, with possible hundreds of vehicles sharing resources of a single cell [121, 122]. In the case of cellular networks, there is also the additional aspect of the participation of a telecom operator, with issues on who would undertake the operating costs.

Short range, high directivity and need for line of sight.

The range of VLC in vehicular scenarios obtained in today experiments is in the order of the tens of meters [110, 123, 124]. These ranges are significantly smaller than those obtainable with DSRC and will never enable the ubiquitous coverage of cellular systems. Compared to RF technologies, VLC propagation is also more sensible to rain and fog, and even the sun position can influence the performance [109]. Furthermore, other aspects make VLC very different from other technologies: the high directivity and low penetration capabilities. These characteristics, on the one hand require that nodes are well aligned and without obstacles in between, but on the other hand imply low interference from neighboring devices and thus lead to high spatial reuse. In addition, these peculiarities also permit high accurate positioning [110] and highly secure communications [111, 112]. An interesting advantage, which is a direct consequence of the high directivity, is also that full-duplex communication with concurrent transmissions in the two directions are easily achieved in VVLN, as shown for example in [109, 125]. The full duplex capability also makes the receiver able to provide an acknowledgment during the transmission, enabling a collision detection mechanism. Differently, full duplex transmissions are still a hard task for researchers in the case of RF [126, 127].

Use of available LEDs as transmitters.

LEDs are already available on new vehicles and they are natural transmitters for VLC. This differs from RF technologies, where optimized antenna systems [128] must be designed and implemented. Concerning the VLC receivers, various options are possible. In fact, whereas photodiodes are the most obvious solution, also LEDs themselves or cameras can be used. The use of LEDs as receivers reduces the necessity of additional components and makes the system more robust against interference from external sources (sun, lampposts) due to a narrower operational bandwidth [125]. Cameras appear instead the best option in terms of achievable throughput, which is significantly increased at the cost of an higher expense [110, 124, 129].

Use of available infrastructure as access network.

VVLN can benefit from a large number of already deployed fixed light sources that are connected or easily connectable to the Internet. Above all traffic lights, that control a significant percentage of city junctions and are oriented in the direction of approaching vehicles. In addition, there are several other light sources that could be involved in VVLN, like variable message panels and road lights. Since some modifications are required to these devices, from this point of view LTE has the advantage of the already existing infrastructure. It is however true that increasing the cellular network capacity requires an expensive deployment of more base stations [130]. In the case of IEEE 802.11p, on the contrary, a new ad hoc infrastructure is required [131].

5.4.2 VLC standardization: IEEE 802.15.7

The increasing interest on the VLC technology has recently led to the development of the IEEE 802.15.7 standard [106, 132], which focuses on PHY and MAC of VLC. Although it is part of IEEE 802.15, dedicated to personal area networks, the specifications explicitly consider vehicles and illuminated roadside devices (such as trafficligths or streetlights) among the addressed applications. The specifications also include detailed procedures for flicker mitigation and dimming support. These two features, that are required while dealing with the LEDs used for illumination purposes, are added to guarantee eye safety and power efficiency [132]. The IEEE 802.15.7 standard enables three different PHY levels, which differ in the adopted modulation and coding scheme. Specifically, PHY I and PHY II support a single light source and work with on off keying (OOK) or variable pulse-position modulation (VPPM) modulations, allowing data rate respectively from 11.67 to 266.6 kb/s and from 1.25 to 96 Mb/s. PHY III is defined for sources with multiple colors (i.e., multiple optical frequencies) that adopt color shift keying (CSK) modulation and allows data rates from 12 to 96 Mb/s. Since the specifications suggest to only use PHY I in outdoor applications, the maximum data rate for vehicular communications is however presently limited to a maximum of 266.6 kb/s.

At the MAC layer four options are foreseen by IEEE 802.15.7: either beacon enabled slotted random access or non-beacon enabled unslotted random access, both with or without CSMA/CA. In VVLNs, non beacon enabled unsolved random access without CSMA/CA seem the preferable solution in most cases. Beacon enabled MAC, in fact, requires a coordinator, thus it can only be imagined when an RSU is involved in the communication; non beacon enabled communications appear to better fulfill the requirements of vehicular networks. At the same time, carrier sensing allows higher throughput and the increasing complexity required for its implementation does not appear to be a problem in the vehicular scenario.

5.4.3 VLC in vehicular scenarios: results from field trials

In the last few years, the growing interest for VLC applied to vehicular networks motivated research groups in USA, Europe, and Asia, to implement VVLN testbeds [109,110,123,124,133–136], as summarized in Table 5.5. The objectives are on the one side to demonstrate the VVLN feasibility and on the other to investigate the achievable performance and push improvements beyond the IEEE 802.15.7 specifications.

Most of measurements are performed in static conditions, either indoor or outdoor [109,133–136]. Very different testbeds in terms of hardware and modulation/coding schemes demonstrated a data rate from 10 to 100 kb/s up to 100 m. For longer distances or larger throughputs, high directivity (through lenses and filters) and multiple LEDs are exploited. In [136], 5 Mb/s are demonstrated using LED fog lights up to 9 m. Three of the cited testbeds adopt commercial LED based traffic lights or car lights [134–136], and all of them use photodiodes at the receiver side. As a general achievement, the sun light was shown not to prevent the use of VLC during daytime, although it reduces the performance more than the street lamps do during night time.

Reference	Transmitter	Receiver	Modul./ Coding	Conditions	Performance
[109] (USA, 2011)	120 white LEDs dissipating 120 mW, 50° half-angle	Photodiode 12° FOV with 4x lens	OOK + Manchester	Static, indoor + outdoor*	100 kb/s @ 100 m
[133] (Italy, 2012)	White LED with lens, 9° half-angle	Photodiode with lens	OOK	Static, indoor	115.2 kb/s @ 31 m
[123] (Portugal, 2012)	240 LEDs	Photodiode	DSSS	Static transmitter, moving receiver	20 kb/s @ 50 m
[134] (France & Romania, 2013)	Commercial LED traffic light or car taillight	Photodiode	OOK + Manchester or Miller	Static, indoor	15 kb/s* @ 20 m (traffic light) or 3 m (taillight)
[135] (Republic of Korea, 2013)	Commercial LED headlamp	Photodiode with lens and color filter	4-VPPM	Static, outdoor	10 kb/s @ 20 m
[136] (Turkey, 2015)	Commercial LED fog lights	Photodiode	4-PAM + Reed-Solomon coding	Static, outdoor	5 Mb/s @ 9 m
[110] (Taiwan & Thailand & USA, 2013)	Scooter LED taillights, 20° half-angle	100 mm^2 photodetector, 90° FOV with no lens	4-VPPM	Two scooters, on the road	10 kb/s @ 10-15 m, 10-40 km/h
[124] (Japan, 2014) (2 testbeds)	32x32 LED array (2x2 LEDs per each bit), 26° half-angle	High speed camera, 1000 fps, 512x1024 pixels, 35 mm focal length	PWM + rate 1/2 turbo coding	Static transmitter, moving receiver	32 kb/s @ 45 m, 30 km/h
	Two red LED transmitters, 40 W optical signal, 20° half-angle	Camera receiver with an optical communication image sensor, 22 (H) x 16 (V) FOV	OOK + Manchester + BCH coding	Two vehicles, on the road	10 Mb/s @ 25 m, 25 km/h

Table 5.5: Field trials. An asterisk is used for the information that was not explicitly provided, thus inferred from the text.

Measurements with a fixed LED based transmitter and a moving receiver were also presented in [110, 123] to reproduce the communication between a traffic light and a vehicle. In both cases, a throughput of few tens of kb/s was obtained with a distance of about 50 m.

Finally, on road measurements of V2V VLC based communications are presented in [110] and [124], focusing on two scooters and two cars, respectively. Whereas 10 kb/s with a distance between 10 and 15 m at 40 km/h are obtained in [110], a significantly larger 10 Mb/s throughput is shown in [124], with a distance up to 25 m at 25 km/h. Such a large throughput was obtained with high directivity and a sophisticated camera as receiver. This is obviously opposed to the aim of low cost, but might be still interesting for the car market.

The camera as a receiver is indeed an option, adopted by both the testbeds presented in [124]. This solution differs from the one used in all other experiments, that use photodiodes. These two possibilities have very different advantages and drawbacks, as already discussed in Section 5.4.1.

In addition to the data rates allowed by the present standard, values of throughput in the order of megabits per second have been thus already demonstrated for VVLNs and higher data rates at longer distances are expected for the future [124, 137–140].

In the following subsections, I discuss the use of VLC in vehicular networks, at first focusing on pure VVLNs and their limitations, and then to the use of VLC in addition to other technologies towards the paradigm of heterogenous vehicular networks.

5.4.4 Pure vehicular visible light networks

The peculiarities of VLC make its use very interesting for VNs; however, the following question arise: what services are possible if it is the only technology on board of vehicles?

A major role for vehicular communications is played by safety applications. Those applications that are based on communications with front and rear vehicles in visibility are indeed perfectly suited to be supported by VLC, thanks to the high reliability and low latency of the communications. Based on the list provided in the final report of an important NHTSA supported project [141], among the most relevant applications enabled by wireless communications to improve safety there are the *emergency electronic brake* and the *forward collision warning*: both of them could be perfectly supported by VLC without the need for other wireless communication technologies. However, due to the need to overtake obstacles, there are a number of applications that are difficultly enabled, or even cannot be enabled, by VLC only, neither through multiple hops. With reference to the NHTSA list, the services of *blind spot warning and lane change warning*, the *do not pass warning*, the *control loss warning*, and the *intersection movement assist* cannot be implemented without the ability to go over the other vehicles and the walls of buildings placed on junctions.

Focusing on non-safety applications, the main drawback of VLC is that it provides a low connectivity degree. To give an idea about this issue, in Figure 5.3 I show the connectivity degree allowed by VLC in different scenarios, as defined hereafter. In particular, two vehicles in a given time instant are said *connected* if there is a path from one to the other, either directly or adopting multiple hops. Considering the separated groups of connected vehicles, I then denote *connectivity degree* the number of vehicles forming the largest one, normalized by the number of vehicles in the scenario. A connectivity degree near to 1 means that most vehicles in the scenario

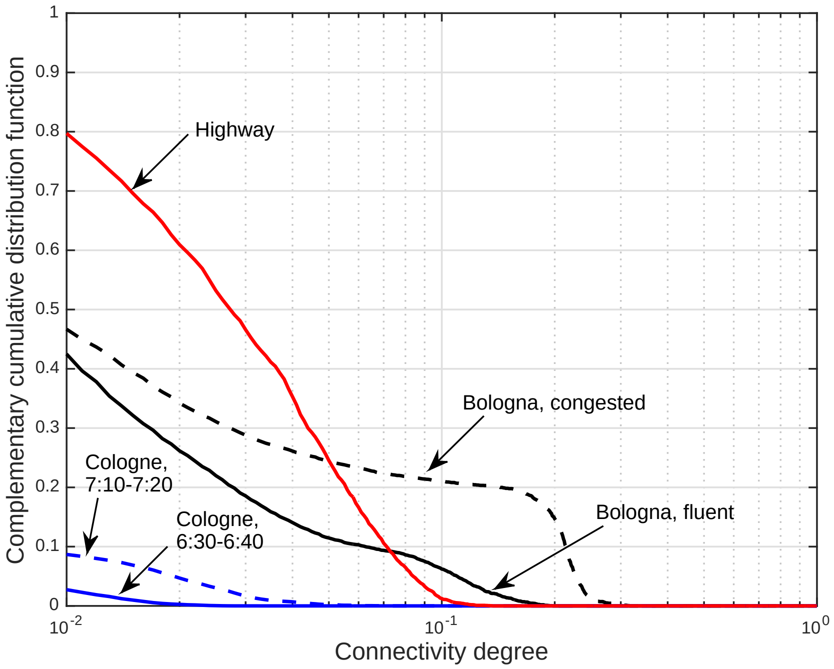


Figure 5.3: Connectivity degree of the VLC technology in different vehicular scenarios.

are connected to each other, whereas a connectivity degree near to 0 means that all vehicles in the scenario are isolated or part of small groups. Figure 5.3, specifically, shows the complementary cumulative distribution function (ccdf) of the connectivity degree that is calculated during the simulations. The scenarios are detailed in Section 5.5.1 and summarized in Table 5.6, whereas the adopted settings are later described and summarized in Table 5.5.7. Focusing, for example, on the Bologna congested scenario, there is nearly 0.2 probability that the largest group of vehicles connected to each other involves at least the 10% of vehicles; such probability falls below 0.015 in the Highway and Cologne scenarios. It is thus clear that the use of pure VVLNs is not sufficient for the implementation of the whole set of safety applications and it cannot provide the full (or at least high) connectivity degree needed for real time or interactive applications. Pure VVLNs applicability is thus confined to some limited safety services and to delay tolerant applications, where an intermittent connectivity is not an issue, such as infotainment content distribution or traffic detection.

5.4.5 VLC as complementary technology

All in all, the limited applicability of pure VVLNs risks to never foster industries to really implement it on the vehicles. Following this observation, it is thus of major relevance to also discuss how VLC can be exploited to improve the scarce resources of the IoV, as an addition to the other technologies that can be applied to implement vehicular services. VVLNs can, in fact, offload part of the RF networks to improve the overall performance and increase the number of implementable services. The unlicensed bandwidth, the reduced deployment cost, and the potential availability of points of access at the road side, are only some of the characteristics that make VLC suitable for this scope.

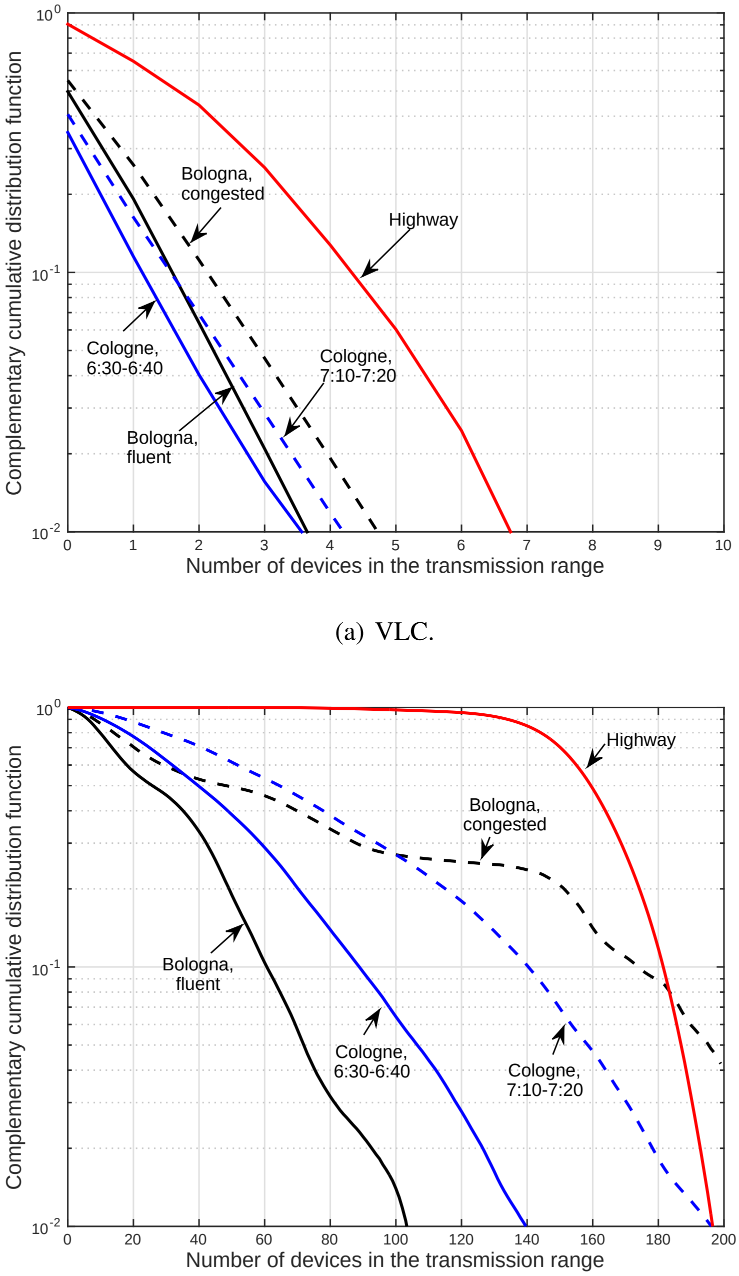
Above the other advantages, let me here remark the spatial reuse allowed by VLC, which makes the full bandwidth being used in almost all links. To give an idea of how many concurrent sources can be present in VLC and to compare it with the case of DSRC, the ccdf of the number of neighbors that are seen by the generic vehicle is shown in Figure 5.4 for both VLC (Figure 5.4(a)) and DSRC (Figure 5.4(b)). The settings detailed in Section 5.7 and summarized in Table 5.5.7 are used. As observable, whereas the number of neighbors with DSRC ranges between tens to hundreds, causing a fragmentation of the available bandwidth, the probability of having more than one neighbor with VLC is less than 0.5 in a highway busy scenario and less than 0.2 in all the others. Hence, even when the available throughput and range of VLC is normally lower than those of RF technologies, still VVLN can provide non negligible additional resources.

Once VLC is applied as a complementary technology (for instance, with respect to DSRC), the main issue is to define the strategy for the use of the joint available resources. To this aim, although several algorithms can be designed, they all lie between the two following (opposite) approaches:

1. VLC is used only in those cases where DSRC is not possible (*DSRC first* approach);

2. VLC is used anytime it is possible in order to maximally offload the DSRC network (*VLC first* approach).

The former approach makes VLC being used only when the other technology cannot be applied, while the latter makes VLC being used anytime it is possible. Which approach is to be preferred clearly depends on the specific conditions, such as the offloaded RF technology, its settings, and the addressed application. For example, if VLC with the settings defined by the IEEE 802.15.7 specifications and DSRC with the settings of IEEE 802.11p are used, the use of *DSRC first* approach causes VLC to be rarely used. This is due to the fact that VLC provides a smaller range and a lower throughput than DSRC. If the *VLC first* approach is instead adopted, VLC can offload part of the traffic from DSRC, thus improving the overall performance. These general considerations, are hereafter explored in a specific example case. It will be shown, through simulations in a realistic urban scenario, that VLC can indeed significantly improve the capacity of the vehicular network.



(a) VLC.

(b) DSRC.

Figure 5.4: Statistical distribution of the number of neighbors in realistic scenarios with VLC and DSRC.

5.5 Simulation Tools and Settings

I consider a urban scenario with all vehicles equipped with both IEEE 802.11p and VLC interfaces for short range communications. In each vehicle, the OBU integrates IEEE 802.11p

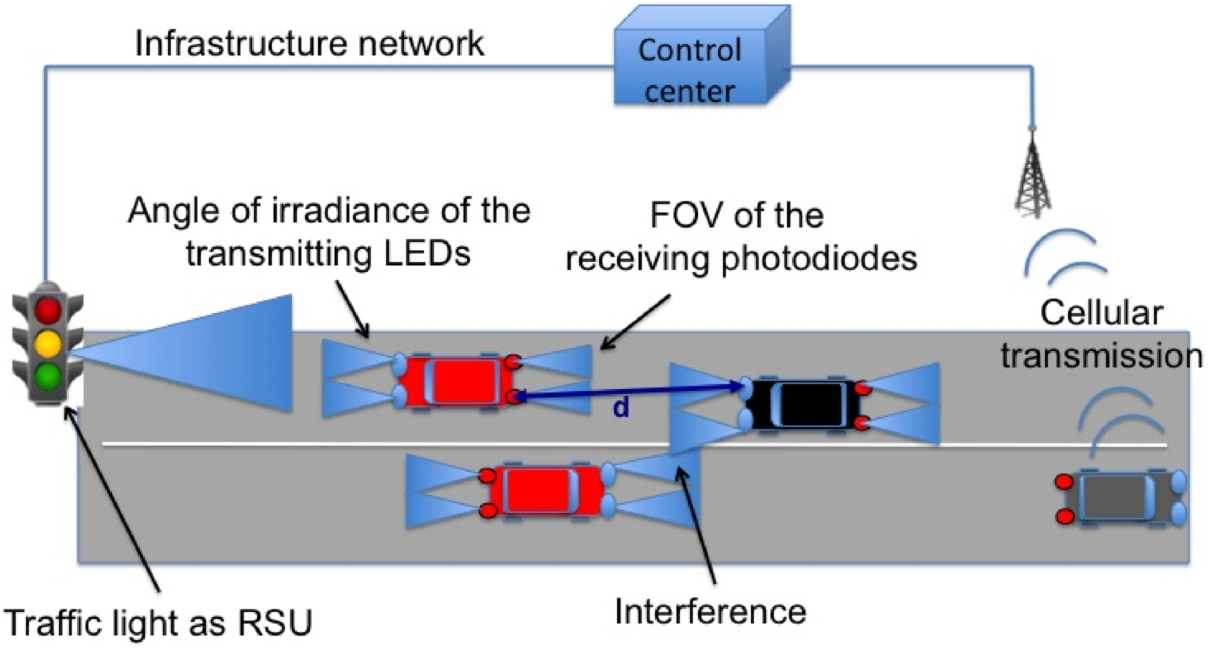


Figure 5.5: System model: vehicles try to exploit VLC for data uploading at the traffic lights, acting as RSU. In case of unavailability, cellular transmission is performed.

and cellular technology, whereas front and rear lights integrate LEDs for transmission and photodiodes for reception through VLC, as represented in Figure 5.5. The OBU acquires several vehicle parameters, referred to as measured data, from sensors and it records them in a queue. When the OBU has recorded some data, it attempts to transmit them through short range communication toward the nearer RSU. RSUs based on IEEE 802.11p are conveniently positioned, whereas RSUs based on VLC correspond to traffic lights (see Figure 5.5). When a maximum number of packets has been accumulated in the transmission queue or after a given time out, data are transmitted through the cellular network to maintain a minimum level of data freshness. Referring to VLC, the transmitter is characterized by a certain angle of irradiance and the receiver by its field of view (FOV): wider angles provide larger service areas, but also lead to performance degradation because of higher probability of receiving undesired light signals. Differently from what commonly considered in the literature (see, e.g., [110]), I assume that head and rear lights are distinct transmitters and receivers. With the objective to transmit as much data as possible through V2V and V2R communications, if an OBU is under coverage of an RSU, its data are directly transmitted through V2R communications. Otherwise, a greedy forwarding (GF) routing algorithm [142] is adopted to find the best route towards an RSU through V2V multiple hops. In particular, the routing algorithm searches for a suitable next relay among the neighbor nodes. The OBU knows the position of the nearest RSU (thanks to a location service, which are out of the scope of the present work) and selects it as the destination; the OBU also knows the position of all its neighbors (thanks to the mechanisms that are described shortly), and considers as possible relays those that are nearer to the destination; the OBU forwards data to the relay which is closest to the destination, if any, and stores the data otherwise.

Based on this, experimental results on the use of VLC and DSRC to offload LTE are obtained in the realistic scenario of Bologna, focusing on the CSVN application [101, 143]. In CSVN, vehicles (hereafter smart vehicles (SVs)) are equipped with an OBU that periodically collects information from various sensors to be delivered to a remote control center. The SVs are all equipped with dual technology wireless systems (VLC and DSRC) and communicate to each other in order to reach, using V2V and V2R, any of the available RSUs. The RSUs then act as gateways towards the control center. The main settings, detailed hereafter, are also summarized in Table 5.5.7.

Please note that this application plays a major role in the IoV, since the periodic generation of measurements that are then sent to a remote control center has been already implemented on millions of vehicles worldwide for insurance purposes and traffic monitoring (currently using cellular networks).

Results are obtained in realistic vehicular scenarios by using the simulation platform for heterogeneous interworking networks (SHINE), which reproduces both IEEE 802.11p and IEEE 802.15.7 from the application layer down to the physical layer [99, 144–146].

5.5.1 Simulation settings: Scenario

The results shown in the paper refer to the five following scenarios:

1. *Bologna downtown, fluent traffic*: a downtown area of the Italian city of Bologna which is 1.8 x 1.6 km^2; the traffic is fluent, with few short queues at the main junctions. There are approximately 455 vehicles on average; the same scenario was used for example in [100, 101];

2. *Bologna downtown, congested traffic*: the same Bologna area of 1.8 x 1.6 km^2, with congested traffic and queues at the main junctions. There are approximately 670 vehicles on average; the same scenario was used for example in [100, 101];

3. *Cologne downtown, 6:30-6:40 a.m.*: a downtown area of the German city of Cologne which is 4.1 x 3.5 km^2. The traffic is fluent and there are approximately 2680 vehicles on average. It is a portion both in time and space of the traffic traces presented in [147]; more details can be found in [143];

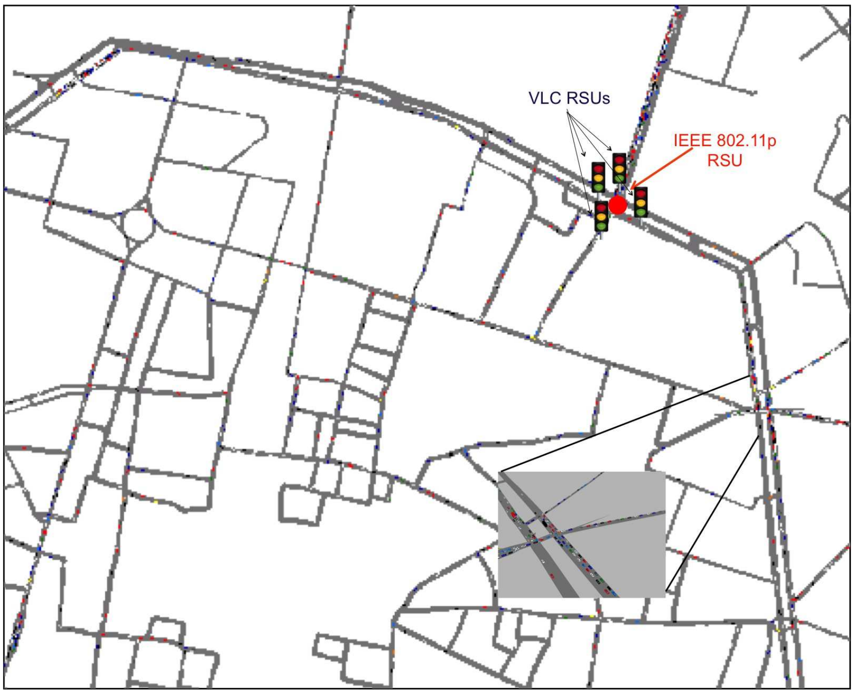


Figure 5.6: Simulated scenario: part of the city center of Bologna (Italy) with one IEEE 802.11p RSU and 4 VLC RSUs represented by traffic lights at a crossroad.

4. *Cologne downtown, 7:10-7:20 a.m.*: the same Cologne area of 4.1 x 3.5 km^2. The traffic is busy and there are approximately 4280 vehicles on average. Also in this case, it is a portion both in time and space of the traffic traces presented in [147] and more details can be found in [143];

5. *Highway, busy*: a 16 km highway segment, with 3 lanes per direction; the traffic is busy, with approximately 1995 vehicles on average.

The Bologna traffic traces are available for download at [144]. The two Bologna scenarios, fluent and congested, are used (Table 5.6). The road-network layout of the scenario is plotted in Fig. 5.13 and consists of a portion of the medium sized Italian city of Bologna of 1.8 x 1.6 km^2. The vehicular traces provide the 2-D position of the SVs, that are all assumed of the same height. The length and width of all vehicles is assumed equal to 4 and 2 meters, respectively.

5.5.2 Simulation settings: Traffic

Both fluent and congested traffic conditions are considered. The former case is characterized by 150 vehicles/km^2 on average, whereas an average density of 230 vehicles/km^2, with car queues at some junctions, characterizes the latter case. In Figure 5.13 a zoomed area of vehicular traffic

Scenario	**Area**	**Average n. of vehicles**
Bologna downtown, fluent [100]	2.88 km^2	455
Bologna downtown, congested [100]		670
Cologne downtown, 6:30-6:40 a.m. [143, 147]	14.35 km^2	2680
Cologne downtown, 7:10-7:20 a.m. [143, 147]		4280
Highway, busy	3+3 lanes, 16 km	1995

Table 5.6: Size and average n. of vehicles if the considered scenarios.

simulated in the congested traffic scenario is reported to provide a visual representation of traffic conditions nearby busy junctions.

5.5.3 Simulation settings: Application

In each SV, the OBU acquires from on-board sensors several vehicle parameters that are periodically packed into $B = 100$ byte packets every T_s seconds, that is, with a data generation rate $\lambda = 1/T_s$ packets/s. Packets are stored in the SV transmitter queue and then attempted to be delivered to any RSU through single or multi-hop communication.

5.5.4 Simulation settings: RSUs

Fixed points of access are placed in the scenario, considering up to 23 crossroads following one of these two cases for each crossroad:

1. One DSRC RSU;

2. Four traffic lights with VLC capability acting as RSU.

The four traffic lights are placed on the four directions of the mostly crowded junction of the scenario. Each traffic light or DSRC fixed point possibly act as RSU to convey packets from vehicles and forwarding them to the remote control center. An example considering only one crossroad is represented in Figure 5.13; the DSRC RSU is placed in the same position as the northern traffic light of these four. The traffic lights considered as VLC RSUs are placed at one side of the road, at a height that does not allow to overcome the top of an approaching vehicle. The crossroads are sorted following the number of vehicles that crossed the junction in a reference time interval (details can be found in [101]).

5.5.5 Simulation settings: Communication technologies and neighbor list update

All SVs are assumed equipped with both a DSRC and a VLC interface, with LEDs used as transmitters and photodiodes as receivers. The neighboring vehicles are continuously tracked thanks to the beaconing mechanism in DSRC [148] (a beacon message is periodically sent by each SV on a control channel, with information that includes the updated position) and to visible light positioning in VLC [149].

5.5.6 Simulation settings: Output Figure

The system performance is evaluated in terms of

- D_R, which is the ratio of packets delivered to the control center through the RSU (i.e., using V2V and V2R),
$$D_R \triangleq \frac{\varphi_{RSU}}{\varphi_{gen}} \tag{5.1}$$
where φ_{gen} is the overall number of packets generated, and φ_{RSU} is the number of packets delivered to the RSUs;

- the average amount of bits per second received by the RSUs, Σ, defined as
$$\Sigma = \frac{\varphi_{\mathrm{RSU}} \cdot B \cdot 8}{T_{\mathrm{sim}}} \tag{5.2}$$
where φ_{RSU} is the number of packets received by the RSUs and T_{sim} is the duration of the simulation.

- L, which is the average delay of delivered packets, in seconds.

The 95% t-based confidence interval is shown for all results. The interval is almost negligible in the majority of the cases.

5.5.7 Simulation settings: PHY and MAC layers

When V2V and V2R communications are carried out by means of DSRC, following [150] and [151] I assume a path loss proportional to the distance raised to the power of 2.2 in line-of-sight (LOS) conditions and I add the effect of buildings and random large-scale fading.

Several measurement campaigns have been carried out in the last decade in order to characterize the DSRC propagation and provide models for VN simulators, such as [150–154]. In this simulator, following [150] and [151], when V2V and V2R communications are carried out by means of DSRC, I refer to the following path loss model: given one source S and its destination D, with d denoting their Euclidean distance, I consider the segment connecting S and D and check the number of buildings that are crossed [150]; I then denote with n_w the number of external walls (i.e., two per building) and with l_b the total length of the segments inside the buildings that are intersected. Then, the path loss is calculated as

$$\mathrm{PL}(d) = \mathrm{PL}_0(1) + 10\mathrm{L_e}\log_{10}(d) + \mathrm{L_w}\cdot n_\mathrm{w} + \mathrm{L_b}\cdot l_\mathrm{b} + X_\sigma \tag{5.3}$$

where $\mathrm{PL}_0(1)$ is the free space path loss at 1 meter distance, $\mathrm{L_e}$ is the path loss exponent assumed equal to 2.2 [150], $\mathrm{L_w}$ is the loss of each external wall of a building assumed equal to 9 dB [150], $\mathrm{L_b}$ is the additional loss inside the buildings assumed equal to 0.4 dB/m [150], X_σ is a lognormal random variable with 0 mean and standard deviation equal to 1.7 [151]. With these values, the average range (when the random contribution is null) is nearly 740 m. With the random contribution, in LOS conditions the range is between 520 and 1050 m with probability 0.96. A threshold model is then assumed for the packet error rate: a transmission between two devices is possible only if the received power P_r is higher than the receiver sensitivity $P_{r_{min}}$; a transmission successfully completes if the average signal to noise and interference ratio (SINR) is higher than a threshold γ_{min}, otherwise an error (or a collision) occurs.

With the considered settings, listed in Table 5.5.7, in 96% of cases the LOS range is between 520 and 1050 m. Sensing and random access procedures, with collisions and retransmissions, are reproduced in details, also including hidden terminals, exposed terminals, and capture effects. The most reliable mode is used, thus the nominal bit rate is 3 Mb/s.

When VLC is adopted, I assume a received power inversely proportional to the distance raised to the power of four [155] and the communication impeded by the presence of any obstacle. In the case of VLC, two front and two rear LED lights are assumed, with integrated photodiodes as receivers; the angle of incidence of the transmitters and the FOV of the receivers are all assumed of 30°.

For VLC, a Lambertian model for the signal propagation is assumed. In fact, although it was shown for example in [136, 156] that the Lambertian model might not completely model the behavior of vehicular lights, this is currently the most adopted model in papers that simulate VVLNs (e.g., [110, 157]). In particular, I assume a received power inversely proportional to the distance raised to the power of four [155]. In addition, a transmission between two devices is possible only if 1) they are in visibility, hence the virtual line connecting them does not cross any obstacle, (i.e., another vehicle or a building), 2) the received power P_r is higher than the receiver sensitivity $P_{r_{min}}$ and 3) the SINR is higher than a threshold γ_min. Specifically, the SINR can be evaluated as [110, 158, 159]

$$\mathrm{SINR} = \frac{\beta^2 P_\mathrm{r}^2}{I + \sigma_\mathrm{shot}^2 + \sigma_\mathrm{th}^2} \tag{5.4}$$

where β is the detector responsivity, I is the interference power, σ_shot^2 is the shot noise variance given by background light sources, such as sunlight and other artificial lights, and σ_{th}^2 is the thermal noise variance, both assumed Gaussian distributed [155]. The received power P_r can be evaluated as

$$P_\text{r} = H(d, \theta, \psi) P_\text{t} \tag{5.5}$$

where P_t is the transmitted power and $H(d, \theta, \psi)$ represents the DC channel gain. Following the generalized Lambertian model, we can write [160]

$$H(d, \theta, \psi) = \begin{cases} \frac{(\text{m}+1)A}{2\pi d^2} \cos^m(\theta) \cos(\psi) & \text{if } \psi < \Psi_\text{C} \\ 0 & \text{otherwise} \end{cases}$$

where A is the physical area of the detector, d is the distance between the transmitter and the receiver, θ is the angle of irradiance, ψ is the angle of incidence, Ψ_C is the half width of the FOV at the receiver, $\phi_{\frac{1}{2}}$ is the half power angle, and m represents the order of the generalized Lambertian radiant intensity

$$\text{m} = -\frac{\ln 2}{\ln(\cos \phi_{\frac{1}{2}})}. \tag{5.6}$$

The interference I is caused by all the transmitting neighbors in visibility (a device which does not transmit, does not cause interference [109]) and can be evaluated as [159]

$$I = \left(\sum_{i=1}^{N_\text{int}} \beta P_{\text{r}i} \right)^2 = \left(\sum_{i=1}^{N_\text{int}} \beta H(d, \theta, \psi) P_{\text{t}i} \right)^2 \tag{5.7}$$

where N_int is the number of interfering neighbors, $P_{\text{r}i}$ is the power received from the i^th interferer, and $P_{\text{t}i}$ is the power transmitted by the i^th interferer. Finally, in this work I assume that i) the maximum distance is fixed to a constant value varying the angle of incidence and that ii) no transmission is possible outside an angle equal to the half-power angle. Hence, denoting with δ_i the portion of time during which the i^th interfering node is transmitting, we can write

$$\text{SINR} = \frac{(\beta H(d, 0, 0) P_\text{t})^2}{\left(\sum_{i=1}^{N_\text{int}} \beta H(d, 0, 0) P_{\text{t}i} \delta_i \right)^2 + \sigma_\text{shot}^2 + \sigma_\text{th}^2}. \tag{5.8}$$

With the considered settings, listed in Table 5.5.7, the LOS range is 50 m. Also in the case of VLC, sensing and random access procedures, with all the consequences, are reproduced in details. Where not differently specified, the highest possible throughput as in the IEEE 802.15.7 specifications is adopted, thus the nominal bit rate is 266.6 kb/s. In did, the highest possible data rate of the IEEE 802.15.7 specifications for VLC and the lowest one of IEEE 802.11p for DSRC were adopted for Figure 5.14 and Figure 5.16 to limit the difference between the two; given the trend of research on these technologies, it is in fact expected that only the VLC data rate will increase significantly, thus a larger difference does not seem realistic.
On both interfaces, retransmissions are performed in case of packet loss up to 7 times.

Param.	Definition	VLC	802.11p
P_t	Transmission power	30 W	0.2 W
β	Detector responsivity	0.54 A/W	-
A	Physical area of the photodiode	1 cm^2	-
ψ_c	FOV of the receiver	30^o	-
m	Order of the generalized Lambertian radiant intensity	20	-
$\gamma_{\min}$	Minimum SNR	11.4 dB	10 dB
d_{max}	LOS range	50 m	520÷1050 m (96% prob.)
R	Nominal data rate	266.6 kb/s (*)	3 Mb/s
B	Packet size	100 bytes	
λ	Packet generation rate	[0.1-10] packets/s	

Table 5.7: Simulation Settings. Asterisks mean that the value is used when not differently specified.

5.5.8 Simulation settings: Routing

Each SV attempts to forward its packets to the nearest RSU adopting the well known greedy forwarded (GF) routing algorithm [101, 161]. With GF, each SV selects as next hop the neighboring SV which maximally reduces the distance from the nearest RSU. More specifically, if the SV is under coverage of an RSU, it performs a direct data transmission to that RSU. Otherwise it considers as possible relays the neighbors that are closer to the destination; the SV then selects as the next hop the relay which is closest to the destination. In the case no other SV is closer to the destination, the data is stored. The GF routing algorithm is firstly performed for each technology separately. If no next hop is available for a given technology, the next hop of the other technology is automatically selected. Otherwise, if a DSRC next hop and a VLC next hop are both available, the adaptive procedure described in the following subsection is performed.

5.6 Congestion-Adaptive VLC-DSRC Selection procedure (CA-VDS)

A simple but effective algorithm named CA-VDS has been designed to manage the joint use of VLC and DSRC. The algorithm exploits the already available capabilities of the receivers and allows to investigate the performance of the two *VLC first* and *DSRC first* opposite approaches (see Section 5.4.5) and solutions in between, by varying a single parameter (the threshold ξ_D, hereafter discussed).

As previously detailed, the position of all DSRC and VLC neighbors are continuously

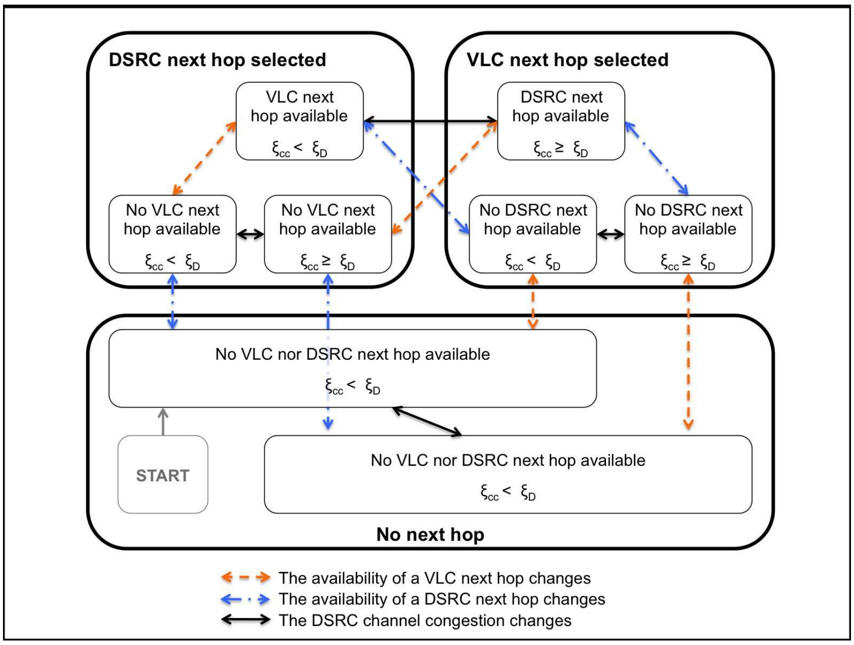

DSRC next hop selected
VLC next hop available
ξcc < ξD
No VLC next hop available
ξcc < ξD
No VLC next hop available
ξcc ≥ ξD
VLC next hop selected
DSRC next hop available
ξcc ≥ ξD
No DSRC next hop available
ξcc < ξD
No DSRC next hop available
ξcc ≥ ξD
No VLC nor DSRC next hop available
ξcc < ξD
START
No VLC nor DSRC next hop available
ξcc < ξD
No next hop
The availability of a VLC next hop changes
The availability of a DSRC next hop changes
The DSRC channel congestion changes

Figure 5.7: State transitions of the technology selection of CA-VDS.

updated; every time a neighbor is available as next hop for both technologies, a selection is performed as follows:

1. In every time interval of duration $T_{\mathrm{cm}} = 0.1$ s, the DSRC channel congestion ξ_{cc} is measured by each SV;

2. DSRC is considered congested and VLC is preferred if $\xi_{\mathrm{cc}} \geqslant \xi_{\mathrm{D}}$, where ξ_{D} is a given threshold. If $\xi_{\mathrm{cc}} < \xi_{\mathrm{D}}$, DSRC is preferred.

CA-VDS can be implemented without an increase of the complexity of the receiver (thus without additional costs). The DSRC channel congestion ξ_{cc} is calculated, in fact, by each SV autonomously and asynchronously thanks to its sensing capabilities, similarly to [162, 163]. Specifically, it is

$$\xi_{\mathrm{cc}} = \frac{t_{\mathrm{busy}}}{t_{\mathrm{busy}} + t_{\mathrm{idle}}} \tag{5.9}$$

where t_{busy} is the time the DSRC medium has been sensed busy and t_{idle} the time it has been sensed idle. From (5.9) it follows that ξ_{cc} goes from 0 (free channel) to 1 (fully used channel). The threshold ξ_{D} defines the DSRC channel congestion level above which VLC is preferred.

As already observed, CA-VDS includes *VLC first* and *DSRC first* as special cases. Please note, in fact, that using $\xi_{\mathrm{D}} = 0$, VLC is always preferred to DSRC irrespective of the channel congestion level (*VLC first*). On the opposite, when $\xi_{\mathrm{D}} = 1$, DSRC is always preferred (*DSRC* *first*). By varying ξ_D from 0 to 1, DSRC has an increasing probability to be selected compared to VLC.

To better clarify the technology selection procedure of CA-VDS, the state transitions performed at each SV are shown in Figure 5.7. Depending on the presence or not of a next hop in each of the two technologies and on the value of the DSRC channel congestion ξ_{cc}, the SV moves among three macro-states that correspond to the selection of a DSRC neighbor as next hop (”DSRC next hop selected”), a VLC neighbor as next hop (”VLC next hop selected”), and no next hop available (”No next hop”). Inside each macro-state, two or three states are possible. For example, a DSRC next hop can be selected either because $\xi_{cc} < \xi_D$ or because no VLC next hop is available; in the latter case, $\xi_{cc} < \xi_D$ and $\xi_{cc} \geqslant \xi_D$ correspond to two different states, since a different behavior follows a variation of the neighbors.

5.7 Experimental results: VLC and DSRC to offload LTE in crowd sensing vehicular networks

Results are shown in this section using:

1. only VLC technology varying the number of available RSUs
2. VLC and DSRC technologies with only one crossroad equipped with 1 DSRC RSU and 4 VLC RSUs
3. VLC and DSRC technologies with only one crossroad equipped with 1 DSRC RSU or 4 VLC RSUs, varying the CA-VDS threshold ξ_D
4. VLC and DSRC technologies with only one crossroad equipped with 1 DSRC RSU or 4 VLC RSUs, varying the amount of data generated by each vehicle
5. VLC and DSRC technologies with only one crossroad equipped with 1 DSRC RSU or 4 VLC RSUs, varying the amount of data generated by each vehicle and the data rate of VLC.

5.7.1 VLC for cellular offloading

The road-network layout of the reference scenario is plotted in Figure 5.8 in which I considered 92 traffic lights at 23 crossroads. In this work, differently to what I sad before (see Table 5.5.7), the value of the distance, FOV and m are variable as underlined in Table 5.7.1. In Fig. 5.9(a), the cumulative distribution function (cdf) of the number of neighbors $N_{\text{neighbors}}$ is plotted varying the transmitter coverage distance d and the receiver FOV. As can be observed, the number of

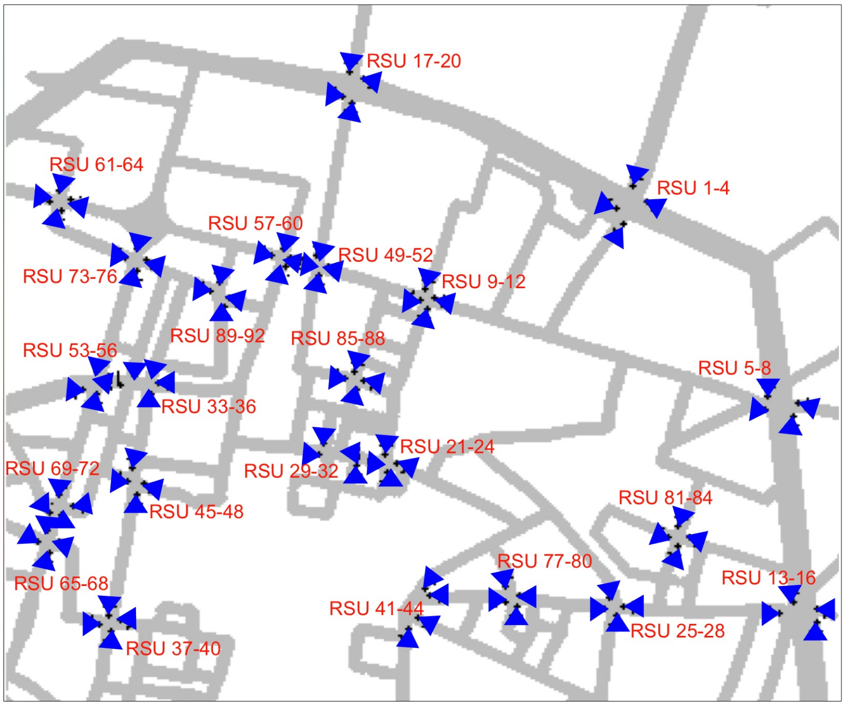


Figure 5.8: Simulated scenario: city of Bologna (Italy) with 92 traffic lights acting as RSUs.

neighbors statistically increases with the distance and the FOV. For example, when $d = 20$ m, the probability to have neither one neighbor is 70% or 80% for FOV of 60° or 30°, respectively. With $d = 100$ m, this value decreases to about 25% or 30% for FOV of 60° or 30°, respectively.

In Fig. 5.9(b), the delivery rate D_{R} is plotted as a function of the packet generation rate λ, for different values of the distance and the FOV. All the 92 traffic lights act as RSU. As expected, D_{R} decreases with higher λ due to the limited capacity of the VLC links. Again as expected, for small values of λ, a higher distance and a higher FOV allow greater values for D_{R}. However, when λ increases, the trend of D_{R} inverts; this is due to a higher number of interferers, which leads to a higher collision probability. Looking at the numbers, I highlight that when $\lambda = 1$ packet/s, D_{R} is in the range [0.7-0.9], hence the system allows important cellular resources to be saved, independently on the coverage distance, the FOV, and the fact that the density of vehicles is not particularly high.

The effect of a variable number of traffic lights acting as RSUs is shown in Fig. 5.9(c). Specifically, Fig. 5.9(c) shows D_R as a function of N_{RSU}, for different values of the distance and the FOV (the traffic light positions are shown in Fig. 5.8). In this case, $\lambda = 0.033$ is assumed. As shown, the improvement obtained by adding VLC capabilities to more traffic lights reduces with the increase of N_{RSU}, almost saturating when nearly 50 RSUs are available. It must also be remarked that the delivery rate does not increase monotonically with a higher N_{RSU} (for

Parameter	Definition	Value
β	Detector resposivity	0.54 A/W
P_{t}	Transmission power	30 W
A	Physical area of the photodiode	1 cm^2
BER*	Target bit error rate (uncoded)	10^{-4}
γ_{min}	Minimum SNR giving BER* in the absence of interference	11.4 dB
d	Distance	Variable (20 m, 50 m, 100 m)
Ψ_{C}	FOV of the receiver	Variable (30°, 60°)
$\Phi_{\frac{1}{2}}$	Half-power angle of the transmitter	Equal to $\Psi_{\mathrm{C}}/2$

m	Order of the generalized Lambertian radiant intensity	m=20 if $\Phi_{\frac{1}{2}}$ =15°
		m=5 if $\Phi_{\frac{1}{2}}$ =30°

Table 5.8: Section 5.7.1 Parameters Settings

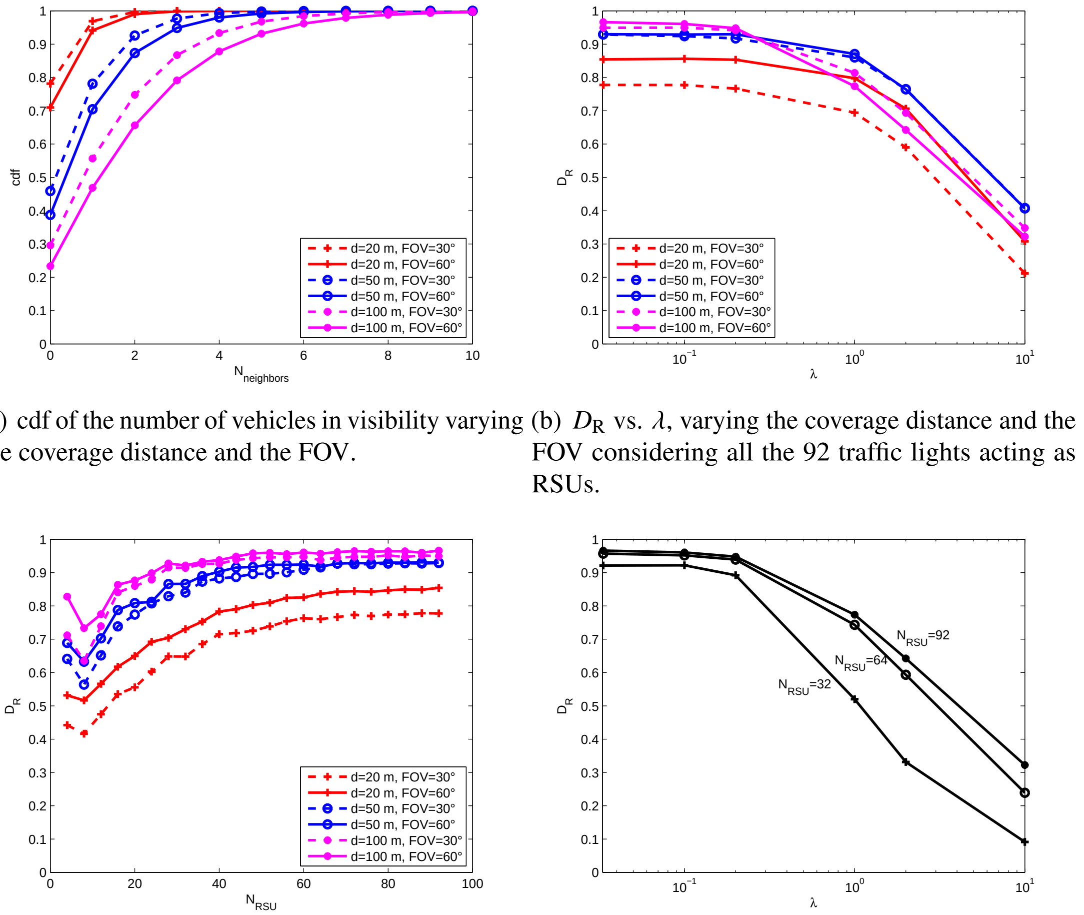
example, D_R is higher with $N_{RSU} = 4$ than with $N_{RSU} = 8$); this is due to the adopted routing algorithm, that suffers from the so called local minima problem, which is strictly related to the exact position of the addressed RSUs [101]. More complex routing protocols would limit this effect, although their investigation is out of the scope of the present work.

In Fig. 5.9(d), the delivery rate D_{R} is plotted as a function of the packet generation rate λ, varying the number of traffic lights acting as RSUs. In this case, d=100 m and FOV=60° are assumed. As expected, D_{R} increases with the number of traffic lights. However, even if there is a great improvement in terms of D_{R} moving from 32 to 64 traffic lights, the gap is lower from 64 to 92, confirming that it is not necessary that all the traffic light of a city are equipped with a VLC interface to provide a remarkable cellular offloading.

5.7.2 Joint use of VLC and DSRC for cellular offloading

In all the figures I compare the performance of:

- DSRC only.
- VLC only.
- DSRC first, as described in Section 5.4.5.
- VLC first, as described in Section 5.4.5.



(a) cdf of the number of vehicles in visibility varying the coverage distance and the FOV.

(b) D_{R} vs. λ, varying the coverage distance and the FOV considering all the 92 traffic lights acting as RSUs.

(c) D_{R} vs. N_{RSU}, varying the coverage distance and the FOV with $\lambda = 0.033$ packet/s.

(d) D_{R} vs. λ varying the number of traffic lights (i.e., the number of RSUs) when d=100 m and FOV=60°.

Figure 5.9: Experimental Results using VLC to offload cellular network.

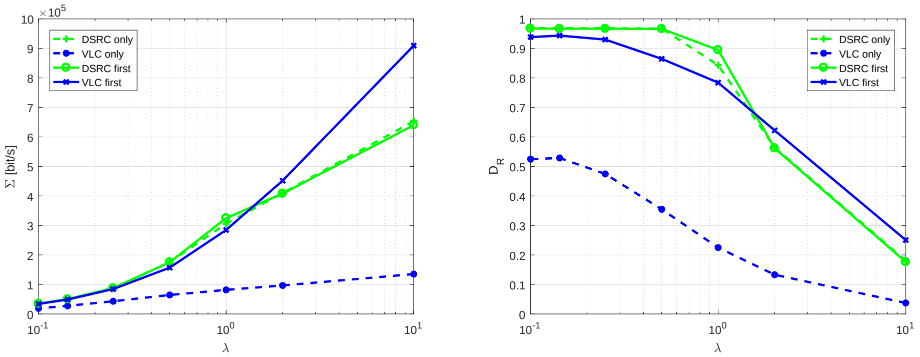
In Figure 5.10 and Figure 5.11, the average amount of bits per second received by the RSUs, Σ, and D_{R} are plotted as a function of the packet generation rate, λ, for fluent and congested traffic conditions, respectively. As expected, in both cases Σ increases and D_{R} decreases with higher generation rates. In addition, it can be highlighted that, when both technologies are used with VLC selected as primary, Σ outperforms all the other cases, for values of λ greater than 1 or 0.5 packets/s in case of fluent (Figure 5.10(a)) or congested (Figure 5.11(a)) traffic conditions, respectively. On the opposite, when VLC only is adopted, Σ reaches the lower values. For example, when λ is equal to 10 packets/s, Σ is 909.4 and 859.3 Kbit/s in fluent (Figure 5.10(a)) and congested (Figure 5.11(a)) traffic conditions, respectively, with VLC as primary technology, whereas it results equal to 135.2 and 142.9 Kbit/s in fluent (Figure 5.10(a)) and congested (Figure 5.11(a)) traffic conditions, respectively, when the sole VLC is adopted.

The results shown in Figure 5.10(a) and Figure 5.11(a) highlight that VLC alone is not as effective as IEEE 802.11p, due to the lower connectivity level it guarantees in the VANET. However, VLC leads to a significant improvement if used with priority in addition to IEEE 802.11p; the use of VLC first, in fact, reduces the congestion in the RF, while the backup over IEEE 802.11p allows to deal with the limited connectivity of VLC.

If IEEE 802.11p is chosen as primary technology, on the contrary, performance is not improved compared to the sole use of IEEE 802.11p itself, and the addition of VLC is ineffective. This is motivated by the large range of transmission of IEEE 802.11p with respect to VLC.

Similar considerations can be drawn in terms of delays, as shown in Figure 5.12, for fluent and congested traffic conditions. As it can be observed, in both figures, L increases with λ; this is due to the increasing number of packets to be delivered and, as a consequence, to the higher number of interferers leading to a higher collision probability. An exception is given by the use of the sole VLC technology: looking, for instance, at Figure 5.12(a), it can be observed that L increases for low values of λ, but then decreases for λ greater than (about) 0.3 packets/s. This is due to the increasing number of collisions: in fact, when λ increases only those vehicles closer to the RSUs (i.e., typically, in visibility) succeed in delivering their packets and do that in a short time, with an important impact on the average delay L (in spite the fact that rate of delivered packets decreases). This effect is even more important in Figure 5.12(b) since referred to congested traffic conditions.

When the sole VLC technology is considered, delay reaches the higher values, especially when the traffic is congested (Figure 5.12(b)), confirming that the sole use of VLC provides the worst performance. When VLC is used as primary technology, L outperforms the other cases when the traffic is congested (Figure 5.12(b)) and λ is greater than 0.8 packets/s, otherwise (i.e., fluent traffic in Figure 5.12(a) and congested traffic in Figure 5.12(b) for $\lambda < 0.8$ packets/s) the adoption of IEEE 802.11p as primary technology or its sole use provides a lower delay.



(a) Σ vs. λ in fluent traffic conditions: comparison of different technologies and of their joint use.

(b) D_{R} vs. λ in fluent traffic conditions: comparison of different technologies and of their joint use.

Figure 5.10: Σ and D_{R} as a function of λ in fluent traffic conditions

I can thus conclude that, in the considered scenario, the joint adoption of IEEE 802.11p and VLC provides better performance in terms of packets delivered to the RSUs with respect to their single use when VLC is chosen as primary technology. As a consequence, this leads to an important percentage of cellular resource saved. Note also that I considered a single crossroad equipped with RSUs; a wider deployment of RSUs will provide further performance improvement. Also in terms of average delay, the adoption of VLC as primary technology provides very good performance, especially when the traffic is congested and the packets generation rate is high.

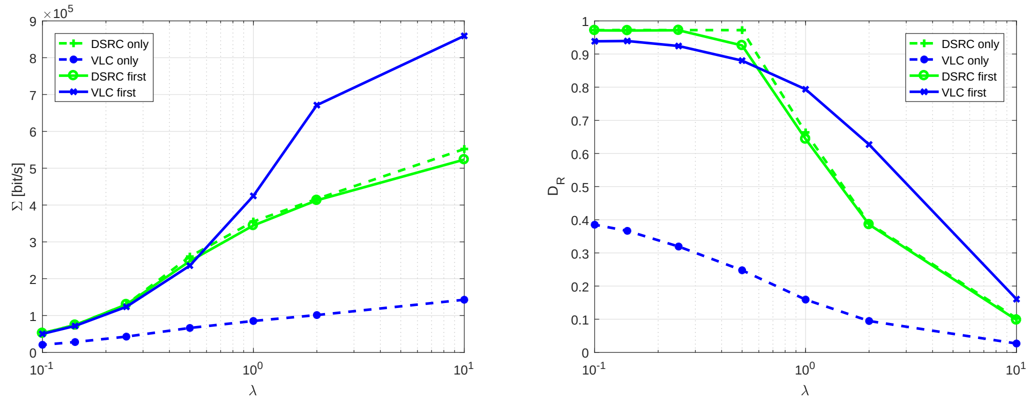
Effect of the threshold ξ_{D}

In this last three Subsections, fixed points of access are placed in the scenario, following one of these two cases:

1. One DSRC RSU;
2. Four traffic lights with VLC capability acting as RSU.

The four traffic lights are placed on the four directions of the mostly crowded junction of the scenario, as represented in Figure 5.13; the DSRC RSU is placed in the same position as the northern traffic light of these four. RSUs are used to convey packets from vehicles and to forward them to a remote control center. The traffic lights considered as VLC RSUs are placed at one side of the road, at a height that does not allow to overcome the top of an approaching vehicle.

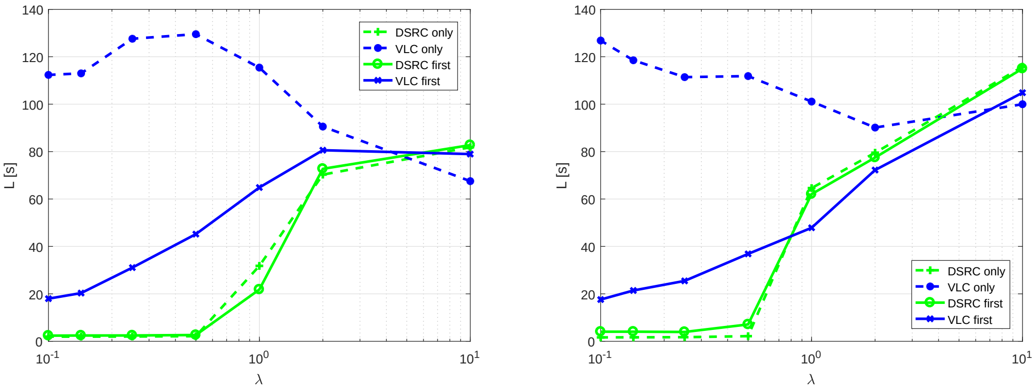
In Figure 5.14, the effect of the threshold ξ_{D} is shown for the two scenarios and both types of RSUs. $\lambda = 2$ packets/s is used. As already remarked, $\xi_{\mathrm{D}} = 0$ means that the *VLC first* strategy is adopted; at the opposite, $\xi_{\mathrm{D}} = 1$ means that the *DSRC first* strategy is adopted.



(a) Σ vs. λ in congested traffic conditions: comparison of different technologies and of their joint use.

(b) D_{R} vs. λ in congested traffic conditions: comparison of different technologies and of their joint use.

Figure 5.11: Σ and D_{R} as a function of λ in congested traffic conditions



(a) L vs. λ in fluent traffic conditions: comparison of different technologies and of their joint use.

(b) L vs. λ in congested traffic conditions: comparison of different technologies and of their joint use.

Figure 5.12: L vs. λ in fluent (a) and congested (b) traffic conditions

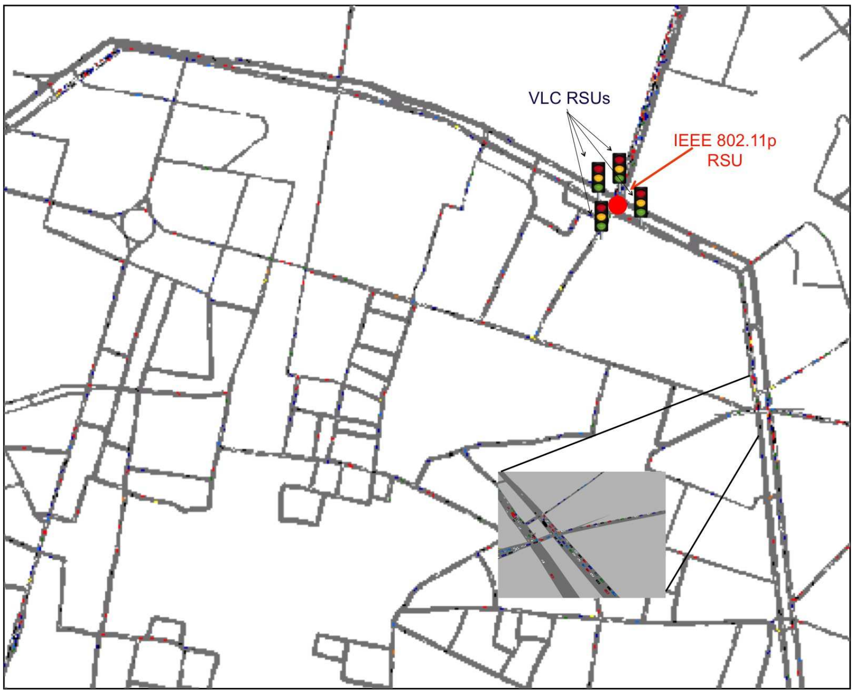


Figure 5.13: Simulated scenario: part of the city center of Bologna (Italy) with one IEEE 802.11p RSU and 4 VLC RSUs represented by traffic lights at a crossroad.

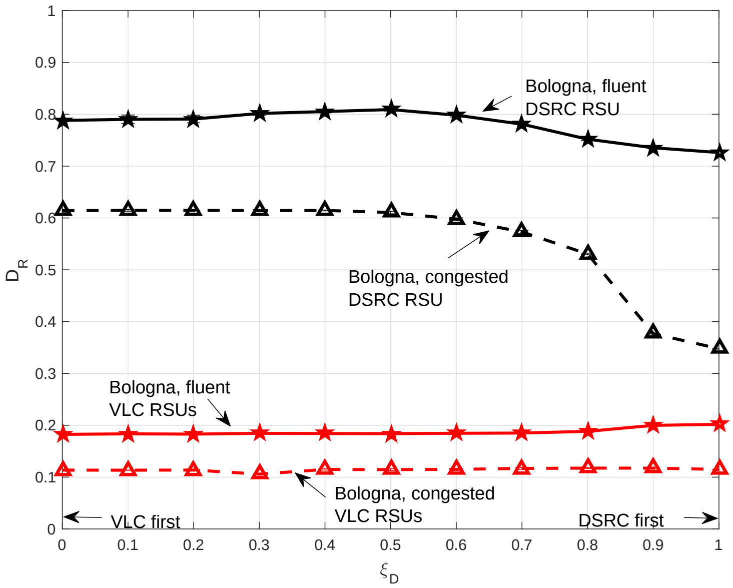


Figure 5.14: Delivery rate vs. DSRC congestion threshold. $\lambda = 2$ packets/s.

As observable in Figure 5.14, when the DSRC RSU is deployed the adoption of a small ξ_D

improves the delivery rate D_R, even significantly. For example, in the case of Bologna congested scenario, D_R grows of more than 75% with ξ_{D} changing from 1 to 0. This effect is remarkable for large values of λ, that is when the data traffic is high and most SVs have something to transmit. The improvement is possible due to the offloading of DSRC in favor of VLC that makes fewer SVs contending for the DSRC medium. In fact, it is shown for example in [164] that an increase of the number of contending nodes reduces the overall capacity of a DSRC network.

In the case of VLC RSUs, the bottleneck is in the bandwidth available at the RSUs themselves, and the value of ξ_{D} is not so relevant. However, it is interesting to note that in the case of Bologna fluent, giving priority to DSRC ($\xi_{\text{D}} = 1$) allows to carry more data in the proximity of the traffic lights, with a small increase in D_R.

Regarding the threshold ξ_{D}, its optimal definition is influenced by several factors, such as the distribution of the vehicles on the road, the propagation medium and the random access mechanism including capture effect, hidden terminals and exposed terminals. However, the results shown in Figure 5.14 suggest that its choice is not critical, since similar performance is achieved following small variations. It can be noted, in any case, that a value lower than 0.5 reduces the DSRC congestions and is thus preferable.

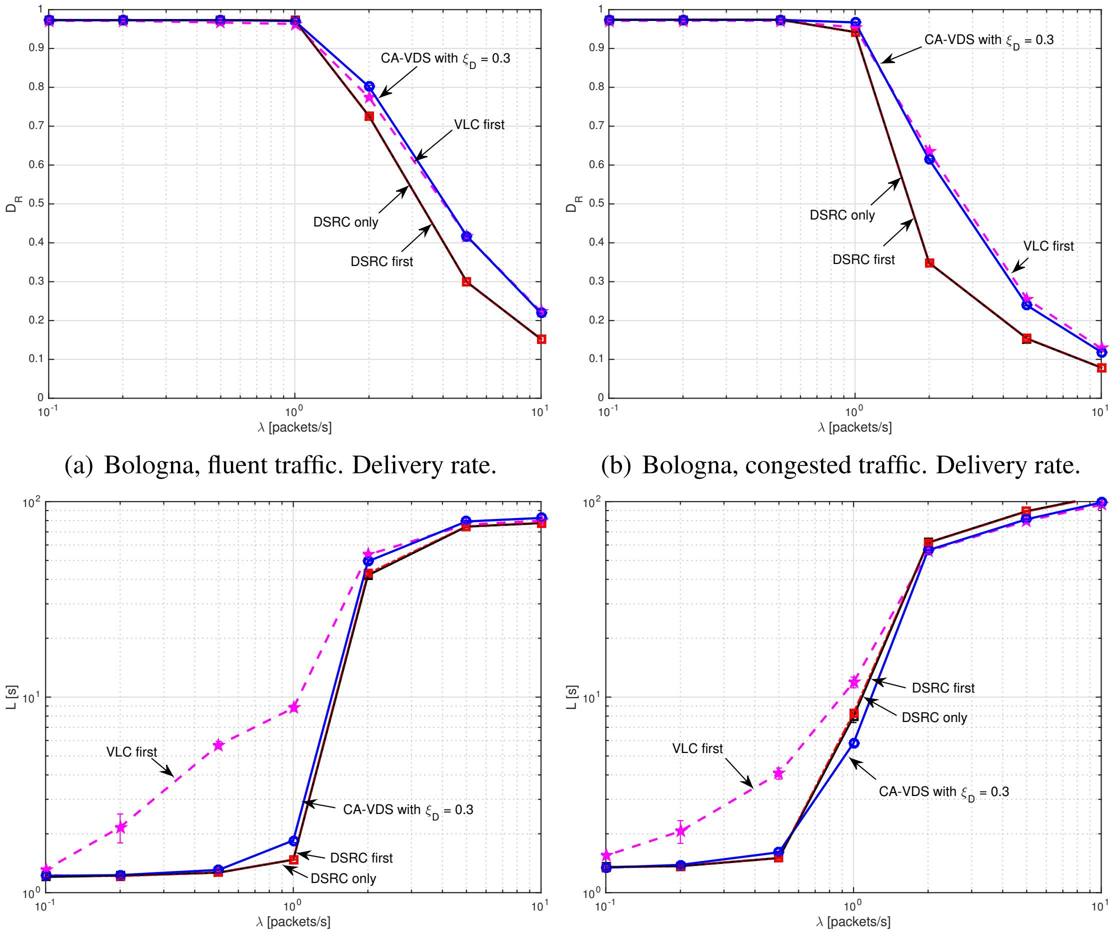
Effect of data traffic load

Results varying λ are then shown in Figure 5.15 for the case of one DSRC RSU and in Figure 5.16 for the case of four VLC RSUs, comparing the performance of:

- *DSRC* or *VLC only* (depending on the RSUs);
- *DSRC first* ($\xi_{\text{D}} = 0$);
- *VLC first* ($\xi_{\text{D}} = 1$);
- *CA-VDS with* $\xi_D = 0.3$.

In particular, assuming one DSRC RSU, the delivery rate D_R and the average delivery delay L are plotted in Figure 5.15 as a function of λ, for fluent (Figure 5.15(a) and Figure 5.15(c)) and congested (Figure 5.15(b) and Figure 5.15(d)) traffic conditions.

Focusing on the delivery rate D_R (Figure 5.15(a) and Figure 5.15(b)), it starts from a value near to 1 (all packets delivered) when the amount of data generated is small ($\lambda \leqslant 1$ packets/s) and then reduces to less then 0.3 when the load is high ($\lambda = 10$ packets/s). As observable, the performance of *DSRC first* is similar to that of *DSRC only*, meaning that, due to the wider coverage provided by DSRC, the addition of VLC is ineffective if DSRC is selected first. When VLC is selected first, for values of λ greater than 1 packets/s D_R is instead higher than both the *DSRC only* and *DSRC first* cases, demonstrating the effectiveness of VLC to increase the



(a) Bologna, fluent traffic. Delivery rate.

(b) Bologna, congested traffic. Delivery rate.

(c) Bologna, fluent traffic. Average delivery delay.

(d) Bologna, congested traffic. Average delivery delay.

Figure 5.15: Delivery rate and average delivery delay varying the packet generation rate with one DSRC RSU.

available resources. The performance of *CA-VDS with* $\xi_D = 0.3$ is similar to that of *VLC first* in all scenarios and for any load.

Concerning the average delivery delay L (Figure 5.15(c) and Figure 5.15(d)), *DSRC only* and *DSRC first* provide smaller values than *VLC first* when the data traffic is reduced (i.e., with $\lambda \leqslant 1$ packets/s). If we focus on Bologna fluent and $\lambda = 1$ packets/s, for example, giving priority to VLC causes an L that is six times the one that follows the priority given to DSRC. DSRC, in fact, allows to reach the destination with fewer hops on average. Remarkably, adopting *CA-VDS with* $\xi_D = 0.3$ the delay is comparable to the cases *DSRC only* and *DSRC first* when data traffic is reduced. Please note that, when the data load increases and the delivery rate decreases, the average delivery delay becomes less relevant. In such case, in fact, the bottleneck is at the RSU receivers and part of the generated packets starve in some queue; as a consequence, if the delivery rate is the same, a higher average delivery delay only means that packets generated far from the RSUs are delivered instead of others generated near to the RSUs.

Figure 5.16 then shows D_R and L as a function of λ when four VLC RSUs are supposed, both for fluent (Figure 5.16(a) and Figure 5.16(c)) and congested (Figure 5.16(b) and 5.16(d)) traffic conditions. As observable, any strategy allowing the use of the heterogeneous VLC and DSRC resources improves D_R dramatically compared to the *VLC only* case. This is due to the lower connectivity level that is guaranteed by VLC in the vehicular network. In several cases, in fact, the SVs do not have a VLC next hop available, and the connectivity is guaranteed only by the DSRC technology. It can also be observed that all the strategies perform similarly in this case; in fact, the bottleneck is represented by the VLC bandwidth of RSUs, which impacts similarly irrespective to the adopted strategy.

Comparing the use of VLC RSUs with the use of DSRC RSU, a smaller D_R is obtained in the former case for the same λ; this is expected due to the smaller data rate available at the VLC RSUs compared to the single DSRC RSU. However, the use of VLC has the great advantage to exploit the traffic lights that are already deployed on intersections; differently, DSRC RSUs require new hardware.

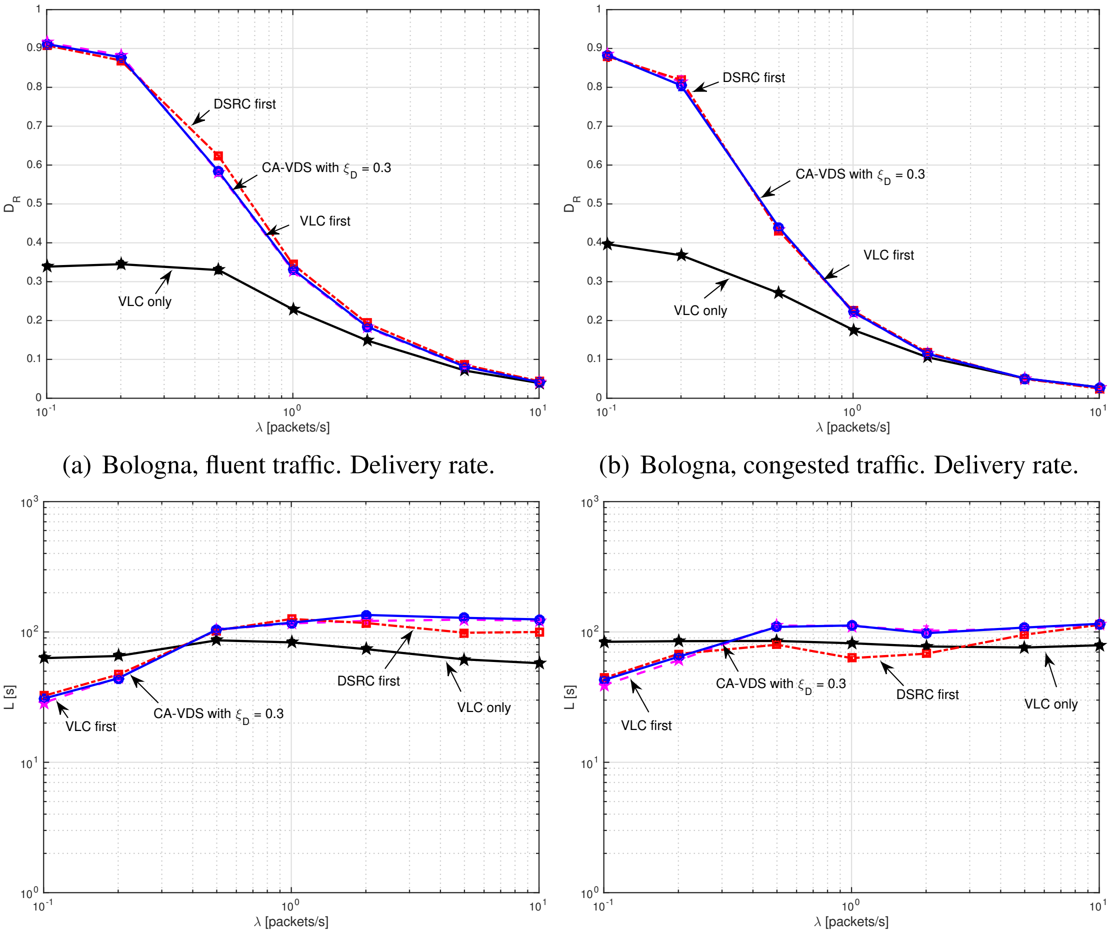
Effect of VLC data rate

In Figure 5.17, the effect of VLC data rate on the delivery rate is investigated. In particular, the previous results are compared with those corresponding to 11.67 kb/s, which is the minimum data rate in the IEEE 802.15.7 specifications, and 10 Mb/s, which is the maximum throughput that has been measured in vehicular field trials [124]. The different data rates are obtained by properly modifying the duration of each transmission.

Results are shown for both the DSRC RSU and VLC RSUs cases, in the Bologna congested scenario. Again, *VLC first*, *CA-VSD with* $\xi_D = 0.3$, *DSRC first*, and either *VLC only* or *DSRC only* (depending on the adopted RSUs) are compared.

Focusing on the DSRC RSU case (Figure 5.17(a)), it can be noted that the VLC throughput does not have a great impact on D_R, and even the use of VLC at 11.67 kb/s in addition to DSRC provides a significant gain compared to *DSRC only*. In this case, in fact, the delivery rate is limited by the RSU capacity, which depends on the DSRC data rate. Although the various links at 11.67 kb/s appear of limited capacity on a first look, the spatial reuse allows almost one link, fully available and free from collisions, per each couple of vehicles.

A slight loss of D_R is only observed if *VLC first* is applied at 11.67 kb/s, when $\lambda \leq 1$. Such a loss is anyway not observed applying *CA-VDS with* $\xi_D = 0.3$. Similarly, a slight improvement is observed if *VLC first* or *CA-VDS with* $\xi_D = 0.3$ are applied at 10 Mb/s, when $\lambda \geq 1$ and $\lambda \leq 5$. As already discussed, *DSRC first* fails to improve the performance compared to *DSRC only*



(a) Bologna, fluent traffic. Delivery rate.

(b) Bologna, congested traffic. Delivery rate.

(c) Bologna, fluent traffic. Average delivery delay.

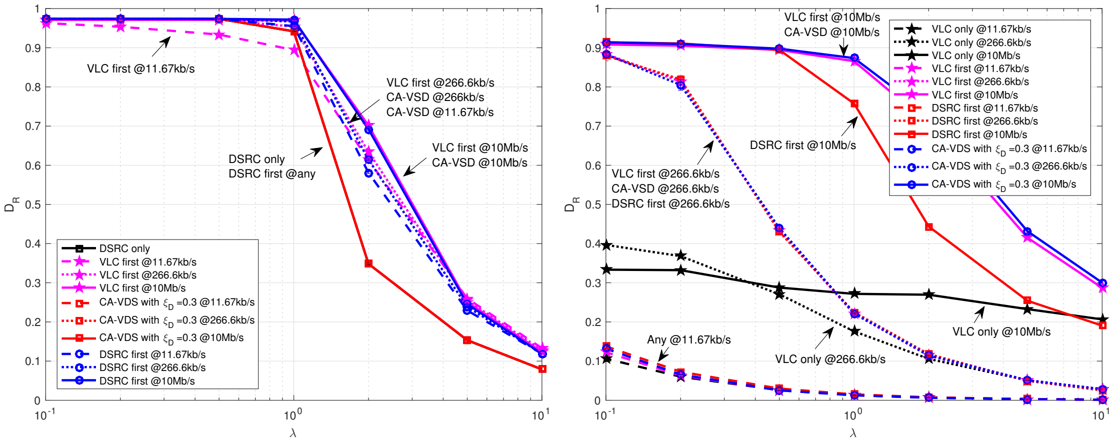
(d) Bologna, congested traffic. Average delivery delay.

Figure 5.16: Delivery rate and average delivery delay varying the packet generation rate with four VLC RSUs.

because DSRC provides higher coverage than VLC; if the VLC link is available towards a neighbor, in fact, the correspondent DSRC link is also available, and always preferred.

Overall, the improvement provided by the addition and use of VLC against *DSRC only* is up to 100% in the case of VLC at 10 Mb/s.

Differently, in the VLC RSUs case (Figure 5.17(b)), the delivery rate is limited by the capacity of the VLC based RSUs. In this case, the D_R curves move to the left or the right with a decrease or an increase of the VLC throughput, respectively. Whereas no significant variation of D_R can be observed comparing *VLC first*, *CA-VDS with* $\xi_D = 0.3$, and *DSRC first* at 11.67 kb/s or 266.6 kb/s, both *VLC first* and *CA-VDS with* $\xi_D = 0.3$ provide a relevant D_R improvement compared to *DSRC first* at 10 Mb/s. In all the cases, the improvement of using both technologies compared to *VLC only* is remarkable.



(a) One DSRC RSU.

(b) Four VLC RSUs.

Figure 5.17: Bologna, congested traffic. Delivery rate varying the packet generation rate, for different data rates of VLC.

In summary, the results shown in Figure 5.17 confirm the effectiveness of the proposed algorithm, as *CA-VSD with* $\xi_D = 0.3$ provides the best D_R in both cases with all VLC data rates.

5.8 Conclusions

I investigated the feasibility and the performance of VLCs in vehicular networks for cellular network offloading. Specifically, after a brief overview of the main characteristics of VLC and its advantages and drawbacks when exploited in dynamic outdoor environments, I investigated the level of connectivity in a urban environment and the feasibility in delivering data for delay tolerant applications without exploiting the cellular network. Numerical results obtained through simulations in a realistic urban scenario showed that even a limited number of traffic lights equipped with VLC allows to offload more than 90% of cellular resources. Furthermore, focusing on the impact of different transmitter and receiver characteristics, it was demonstrated that improving the coverage distance or the FOV does not always lead to a higher performance; the improved coverage does in fact increase the average number of neighbors, with a higher collision probability and consequently a lower data delivery under high network load conditions.

Then I focus my attention on the adoption of VLC as supplementary technology to the RF ones for data exchanging between vehicles and between vehicles and RSUs in vehicular networks. I proposed to exploit this emergent technology in cooperation with DSRC and cellular communications to increase the overall resource availability for the future IoV. I proposed to let the two technologies cooperate to increase the rate of packets delivered through multi hop V2V communications toward an RSU. Example results have been shown focusing on the crowd sensing vehicular network application, considering VLC in addition to DSRC. A cooperative algorithm to adaptively select the technology has been also proposed, with a single parameter allowing to move from VLC always preferred to DSRC to the opposite case. Results have been obtained by means of a realistic simulation tool and show the advantage of the cooperation between the two technologies with respect to their single use. Simulations, performed in realistic urban scenarios with hundred of vehicles, demonstrated the significant improvement obtained by adding VLC to DSRC. The best results were obtained by giving priority to DSRC when its channel is far from congested, and preferring VLC in the other cases.

Future considerations can also take into account the different light intensity of headlights and rear lights. I also believe that VLC will not replace high-speed radio frequency (RF) communications, which allow long range non-line-of-sight links. Future developments may instead consider VLC and other RF communications (such as IEEE 802.11p and LTE for device-to-device) as complementary technologies to be jointly adopted for performance improvements.

Chapter 6

Echo Canceller

6.1 Introduction

Single frequency networks, such as those recently deployed for 4G cellular systems or digital video broadcasting - terrestrial (DVB-T), are characterized by a high spectral efficiency and a reduced coverage planning complexity. Their service coverage can be easily extended by the introduction of proper OCRs, that act as gap fillers. The interest for OCRs has thus significantly increased in recent years, also owing to the growing attention to full duplex radios, that could benefit from OCRs as well.

OCRs suffer, however, from the coupling effect (mainly due to multi-path propagation and energy leakage) between the transmitting and receiving antennas, that operate at the same frequency. This positive feedback could lead to the instability of the repeater, hence the gain of the OCR power amplifier is usually limited to avoid harmful conditions.

The most relevant impairment experienced by on-channel repeaters OCRs is the presence of a coupling-channel between the transmitting and receiving antennas, that generates unwanted echoes. Echoes critically influence the overall system behavior, with harmful effects on the signal quality and, above all, pose a threat on the system integrity. Echo cancellers are usually adopted, therefore, to remove unwanted coupling contributions. Several architectures for digital echo cancellers have been proposed to solve this critical problem [165–178], trying to minimize the possibility of an instable situation and, as a consequence, to increase the gain of the power amplifier and, therefore, the coverage area. In most of the cases, the transmission of a low-power training signal is used to estimate the echoes (that is, the coupling-channel), setting the taps of an echo cancelling digital filter accordingly.

These works, however, evaluate the echo canceller performance through metrics such as mean rejection ratio, echo suppression, or modulation error rate, rather than directly analyzing the probability that the OCR becomes unstable, that is the highest risk factor to be considered in a practical situation (in an unstable condition, the infinite loop generated by the positive feedback represented by the coupling-channel can result in OCR severe damages).

In [179] and [180] I exploited the results of the pioneering paper [181] to analytically evaluate the probability of instability of a digital OCR with echo canceller, considering the coupling-channel estimation errors as the only cause of imperfect echo cancellation. In this work I investigated the OCR stability problems arising from the joint effect of echo estimation errors and the quantization of the echo-cancelling filter taps. The probability of OCR instability is analytically derived and some performance figures are provided varying the quantization level and the channel model in realistic scenarios. In particular, I improved the stability analysis also considering the impact of the quantization error that affects the taps of the echo-cancelling digital filter. In the following, the analytical approach introduced in [179] and [180] is used to evaluate an upper bound on the probability of instability of a digital OCR in the presence of both coupling-channel estimation errors and taps quantization[1]. The impact of taps quantization on the echo canceller performance was studied also in [167], where, however, the coupling-channel estimation errors and the probability of instability were not taken into account.

6.2 System Model and Problem Statement

The digital OCR architecture considered in this chapter is the one introduced in [168], whose lowpass equivalent representation is shown in Fig. 6.1. As can be observed, the received signal $x(t)$, with power P_x and bandwidth B_eq, is analog-to-digital converted and filtered by a digital filter with transfer function $H_\mathrm{R}(f)$, aimed at reducing both the noise and the adjacent channel interference. In the transmitting section, instead, the signal is filtered by a digital filter with transfer function $H_\mathrm{T}(f)$, aimed at removing possible out-of-band emissions, digital-to-analog converted, amplified[2] by the high power amplifier (HPA) and re-transmitted.

Unfortunately, owing to the coupling-channel between the transmitting and receiving antennas, a certain amount of the transmitted power enters the receiving antennas, constituting an interfering positive feedback, the so-called echo, that must be removed. As shown in Fig. 6.1, the echo cancelling unit is mainly constituted by:

- a channel estimator, which derives an estimation $\hat{H}_\mathrm{eq}(f)$ of the transfer function $H_\mathrm{eq}(f)$ that characterizes the equivalent coupling-channel[3] highlighted in the shaded box of Fig.

[1] Unless otherwise specified, in the remainder of this chapter I use the term "quantization" with reference to the taps of the echo-cancelling digital filter.

[2] Non-linear effects possibly introduced by the amplifier are neglected here.

[3] Here I do not consider the degradation introduced by the analog-to-digital converter (ADC) and the digital-to-analog converter (DAC), that can be assumed negligible owing to the strict signal quality requirements. For this reason we can assume that the equivalent discrete-time coupling-channel is linear, with transfer function $H_\mathrm{eq}(f)$.

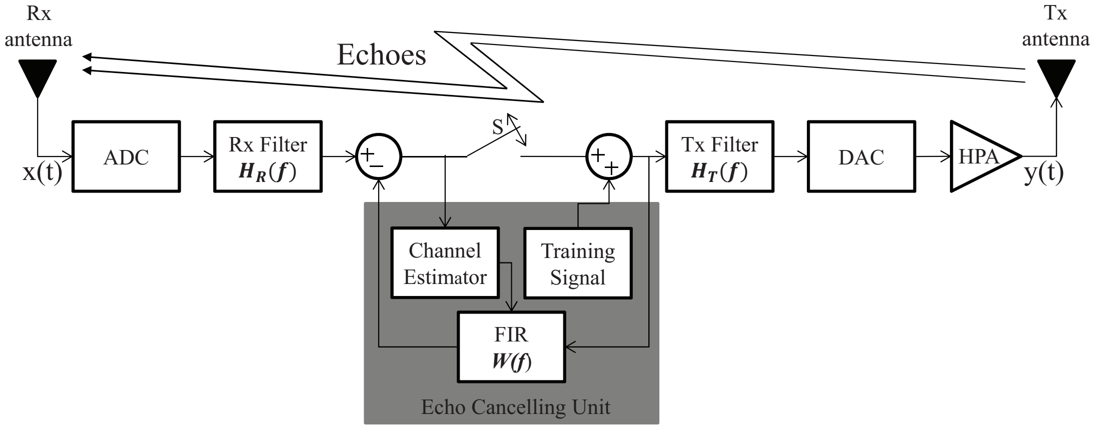


Figure 6.1: General scheme of the OCR with echo canceller.

6.2;

- a finite impulse response (FIR) filter, whose transfer function $W(f)$ is dynamically adjusted on the basis of $\hat{H}_{\text{eq}}(f)$. As shown in Fig. 6.1, the role of the filter is to realize a proper negative feedback in order to cancel the unwanted coupling contribution.

In general, owing to possible estimation errors, $\hat{H}_{\text{eq}}(f) \neq H_{\text{eq}}(f)$, hence the echo cancellation could not be perfect. This phenomenon is further worsened by the quantization of the filter taps, that constitutes an additional cause for the echo cancellation inaccuracy. The joint impact of both impairments will be investigated in the following.

The OCR operates in two different phases: in the first one (denoted *start up mode*), the switch S (see Fig.6.1) is open and the incoming signal is not retransmitted. During this phase the estimation $\hat{H}_{\text{eq}}(f)$ of $H_{\text{eq}}(f)$ is carried out through the insertion of a locally generated training signal (e.g., a pseudo-noise sequence or a train of sounding pulses) with power P_{s} and duration MT_{s}, where M is the length of the training sequence and T_{s} is the sampling frequency.

Once the equivalent coupling-channel has been estimated and the taps of the echo cancelling filter have been accordingly derived, the switch S is closed and the OCR starts transmitting the signal to be repeated (*steady state*). While operating in the steady-state mode, the OCR may keep tracking the possible channel variations by continuously transmitting the training signal superimposed to the useful signal or implementing the least mean square (LMS) strategy.

Since the local training signal represents an impairment for the final user, the ratio $P_{\text{s}}/P_{\text{x}}$ must be kept properly low. From the perspective of the channel estimator, the overall estimation SNR is

$$\gamma_{\text{e}} = M \frac{P_{\text{s}}}{N_0 B_{\text{eq}} + P_{\text{x}}} , \tag{6.1}$$

where N_0 is the thermal noise power spectral density (PSD).

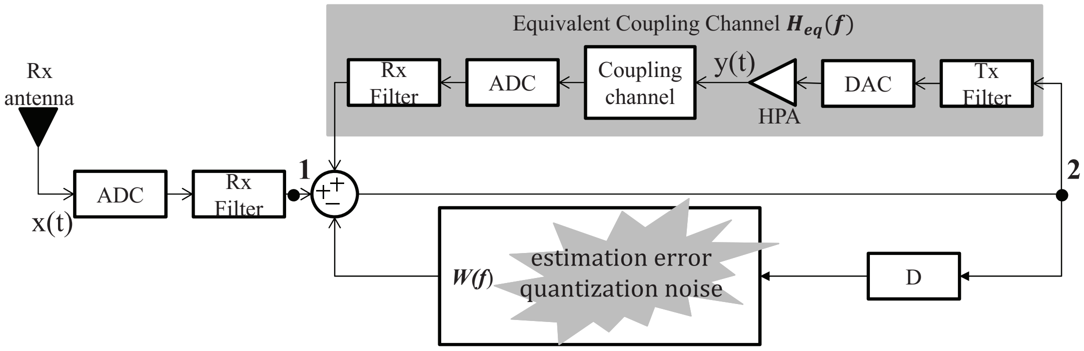


Figure 6.2: Equivalent coupling-channel.

Table 6.1: Filters specifications

Specifications	Implemented filter	Approximated ideal filter
pass-band bandwidth	7.61 MHz	7.61 MHz
pass-band ripple	0.2 dB	0 dB
out of band attenuation	65 dB	∞ dB
group delay	3.1 μs	6.3 μs

6.2.1 The Equivalent Coupling Channel

To characterize the equivalent coupling channel to be estimated, highlighted in the shaded box of Fig.6.2, I introduced, first of all, the impulse response of the cascade constituted by the transmitting and receiving filters:

$$h(t) = \mathcal{F}^{-1}\left\{H(f)\right\} \tag{6.2}$$

where

$$H(f) \triangleq T_{\mathrm{s}} H_{\mathrm{R}}(f) H_{\mathrm{T}}(f) \mathrm{rect}(f T_{\mathrm{s}}) \; . \tag{6.3}$$

In table 6.3 I report both the specifications of the filter implemented[4] in the OCR prototype described in [168] and its approximation as an ideal filter assumed in [179]. As far as the “proper” coupling channel is concerned, I considered the four channel models adopted in [167], that is, Rural Area (RA6), Typical Urban (TU12), Bad Urban (BA12) and Hilly Terrain (HT12).

From the previous definitions and assumptions it follows that the equivalent discrete-time coupling-channel impulse response can be written as

$$h_{\mathrm{eq}}[n] = \sqrt{G} \sum_{l=0}^{L-1} h_l h(nT_{\mathrm{s}} - \tau_l), \tag{6.4}$$

[4] The filter design criteria are detailed in [182].

where G is the HPA gain, L is the total number of paths, and τ_l and h_l are the delay and the complex gain of the l-th path, respectively. The relative (with respect to the first delay τ_0) delays $\tau_l - \tau_0$ are reported in Table 6.2. For the sake of conciseness, here I do not report the h_l values, that are not relevant for the following analysis.

Let denote with k_m and k_M the beginning and the end of impulse response $h_\mathrm{eq}[k]$ of the equivalent discrete-time coupling channel, both normalized with respect to T_s. By defining k_m and k_M such that $h_\mathrm{eq}[k] = 0$ for $k < k_\mathrm{m}$ and $k > k_\mathrm{M}$, it can be easily shown that

$$k_\mathrm{m} = \left\lfloor \frac{\tau_\mathrm{g} - \frac{\tau_\mathrm{IR}}{2} + \tau_0}{T_\mathrm{s}} \right\rfloor \tag{6.5}$$

$$k_\mathrm{M} = \left\lceil \frac{\tau_\mathrm{g} + \frac{\tau_\mathrm{IR}}{2} + \tau_{L-1}}{T_\mathrm{s}} \right\rceil \tag{6.6}$$

where τ_g is the group delay of the filtering cascade (6.3) and τ_IR is the duration of its impulse response (6.2).

In the numerical results Section I considered both a really implemented cascade of FIR filters, for which[5] $\tau_\mathrm{g} - \frac{\tau_\mathrm{IR}}{2} = 0$, and an ILP filter cascade with bandwidth B_eq, whose the sinc-type (hence infinite) impulse response is assumed to be extinguished after N_lobes per-side. In this case $\tau_\mathrm{IR} = \frac{2(N_\mathrm{lobes}+1)}{B_\mathrm{eq}}$. In the following this filter will be denoted *truncated ILP filter*.

6.2.2 The Echo Cancelling Filter

Let introduce the *cancelling window*, that is the time interval in which the estimated impulse response is considered in order to derive the taps of the echo-cancelling filter. Denote with D its starting instant and with P its duration, both normalized with respect to T_s. The duration τ_g is the group delay of the filtering cascade, that entails that $2\tau_\mathrm{g}$ is the duration of the filtering cascade impulse response. By comparing the value in tables 6.3 and 6.2, it can be noticed that the duration of filters impulse response ($2\tau_\mathrm{g}$) and the delays introduced by the channel are comparable, hence the filter effect cannot be neglected in the evaluation of the cancelling window position (differently than what generally done in the literature). The cancelling window fits the estimated coupling channel impulse response (see Fig. 6.3) if

$$D = \hat{k}_\mathrm{m} \tag{6.7}$$

$$P = \hat{k}_\mathrm{M} - \hat{k}_\mathrm{m} + 1 \tag{6.8}$$

where $\hat{k}_\mathrm{m}$ and $\hat{k}_\mathrm{M}$ are the estimated values of k_m and k_M, respectively. The filter taps are finally derived by sampling the estimated impulse response $\hat{h}_\mathrm{eq}[k]$ within the cancelling window[6] and

[5] For digital symmetrical FIR filters it is $\tau_\mathrm{IR} = 2\tau_\mathrm{g}$.

[6] Of course a delay D is introduced on the signal at the input port of the filter. The value of D is derived by the channel estimation unit.

Table 6.2: Channel delays profiles

Rural Area (RA6)	Typical Urban (TU12)	Bad Urban (BU12)	Hilly Terrain (HT12)
$[ns]$	$[ns]$	$[ns]$	$[ns]$
0	0	0	0
100	200	200	200
200	400	400	400
300	600	800	600
400	800	1600	800
500	1200	2200	2000
	1400	3200	2400
	1800	5000	15000
	2400	60000	15200
	3000	7200	15800
	3200	8200	17200
	5000	10000	20000

quantizing with a granularity q that depends on the number n_b of quantization bits:

$$w_k = \hat{h}_\mathrm{eq}[k + D] + \epsilon_{\mathrm{q}_k} \, , \forall k \in [0, P - 1] \, , \tag{6.9}$$

where the quantization error ϵ_{q_k} is assumed to be a zero-mean RV with variance

$$\mathbf{E}\left\{|\epsilon_{\mathrm{q}_k}|^2\right\} = \frac{q^2}{12} \, . \tag{6.10}$$

By considering, without loss of generality, $\hat{h}_\mathrm{eq}[k]$ within the range $[-1,\ 1]$, it results $q = 2^{-n_\mathrm{b}+1}$.

As far as the estimation error is concerned, in [179] it is shown that the discrete-time equivalent coupling channel estimation results in

$$\hat{h}_\mathrm{eq}[k] = h_\mathrm{eq}[k] + \nu[k] \, , \tag{6.11}$$

where $\nu[k]$ is a zero-mean RV with variance

$$\mathbf{E}\left\{|\nu[k]|^2\right\} = \frac{1}{\gamma_\mathrm{e}} \, . \tag{6.12}$$

While conditions (6.7), (6.8) can be easily fulfilled by adaptively varying the cancelling window position (once the physical coupling channel has been estimated), condition can be fulfilled only approximatively, owing to the joint effect of estimation and quantization errors.

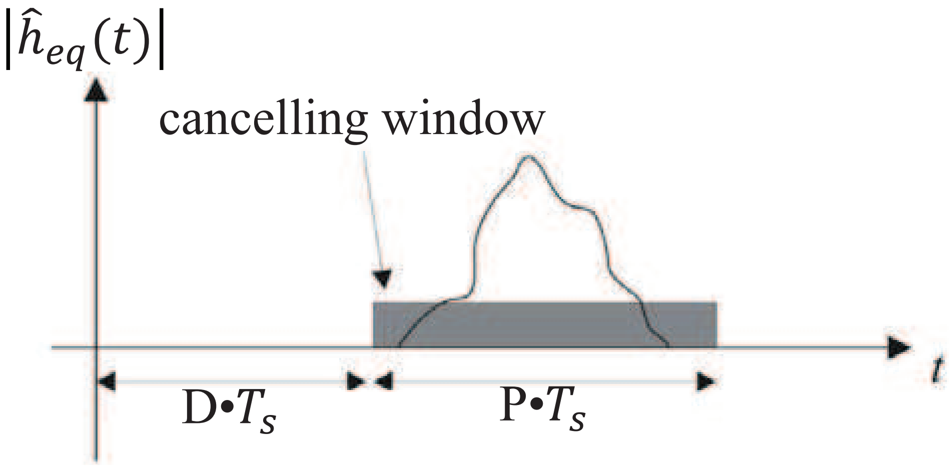


Figure 6.3: Cancelling window.

6.2.3 Coupling Channel Estimation and Quantization

In [179] it is shown that the discrete-time equivalent coupling channel estimation results in

$$\hat{h}_{\mathrm{eq}}[k] = h_{\mathrm{eq}}[k] + \nu[k]\,, \tag{6.13}$$

where $\nu[k]$ is a zero-mean RV with variance

$$\mathbf{E}\left\{|\nu[k]|^2\right\} = \frac{1}{\gamma_{\mathrm{e}}}\,. \tag{6.14}$$

Introducing the effect of quantization, with granularity q, on the taps w_k of the echo canceller's filter leads to the following expression:

$$w_k = \hat{h}_{\mathrm{eq}}[k + D] + \epsilon_{\mathrm{q}_k} \tag{6.15}$$

where ϵ_{q_k} is assumed to be a zero-mean RV with variance

$$\mathbf{E}\left\{|\epsilon_{\mathrm{q}}[k]|^2\right\} = \frac{q^2}{12}\,. \tag{6.16}$$

By considering $h_{\mathrm{eq}}[k]$ within the range $[-1,\ 1]$, it results $q = 2^{-n_{\mathrm{b}}+1}$, with n_{b} denoting the number of quantization bits.

6.2.4 Overall Transfer Function

With reference to Fig.6.2, the transfer function of the system in between the input port 1 and the output port 2 can be expressed as

$$H_{\mathrm{OCR}}(z) = \frac{1}{1 - \left[H_{\mathrm{eq}}^{(\mathrm{D})}(z) - W(z)\right] z^{-D}} \tag{6.17}$$

where $H_{\mathrm{eq}}^{(\mathrm{D})}(z)$ and $W(z)$ are the Z-transform of $\{h_{\mathrm{eq}}[k+D]\}$ and $\{w_k\}$, respectively. By defining the error $\epsilon_{\mathrm{T}}[k] \triangleq w_k - h_{\mathrm{eq}}[k+D]$, it follows from (6.15), (6.16), (6.13), and (6.14), that

$$\mathbf{E}\left\{|\epsilon_{\mathrm{T}}[k]|^2\right\} = \gamma_{\mathrm{T}} \triangleq \left(\frac{1}{\gamma_{\mathrm{e}}} + \frac{2^{-2n_{\mathrm{b}}}}{3}\right)^{-1} . \tag{6.18}$$

As can be observed, γ_{T} depends on both the estimation SNR γ_{e} and the number of quantization bits n_{b}. Thus, it is the key parameter for the instability analysis outlined in the following Section.

6.2.5 Upper Bound on the Probability of Instability

By supposing that the coupling channel impulse response estimation is accurate enough to have $\hat{k}_{\mathrm{m}} = k_{\mathrm{m}}$, $\hat{k}_{\mathrm{M}} = k_{\mathrm{M}}$, equation (6.17) can be worked out, yelding

$$H_{\mathrm{OCR}}(z) = \frac{z^{k_{\mathrm{M}}}}{z^{k_{\mathrm{M}}} + \sum_{k=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} \epsilon_{\mathrm{T}}[k] z^{k_{\mathrm{M}}-k_{\mathrm{m}}-k}} . \tag{6.19}$$

By following a procedure similar to the one proposed in [179], based on the study of zeroes distribution [181], it is possible, thanks to (6.18), to evaluate an upper bound P_{u} on the probability that the OCR is unstable:

$$P_{\mathrm{u}} = k_{\mathrm{M}} - \int_0^1 \frac{2}{r} \left[\frac{\Lambda(r) + \gamma_{\mathrm{T}}\Psi(r)}{\sum_{l=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} r^{2l}}\right] \exp\left[-\gamma_{\mathrm{T}} \frac{r^{2k_{\mathrm{M}}}}{\sum_{l=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} r^{2l}}\right] dr \tag{6.20}$$

where

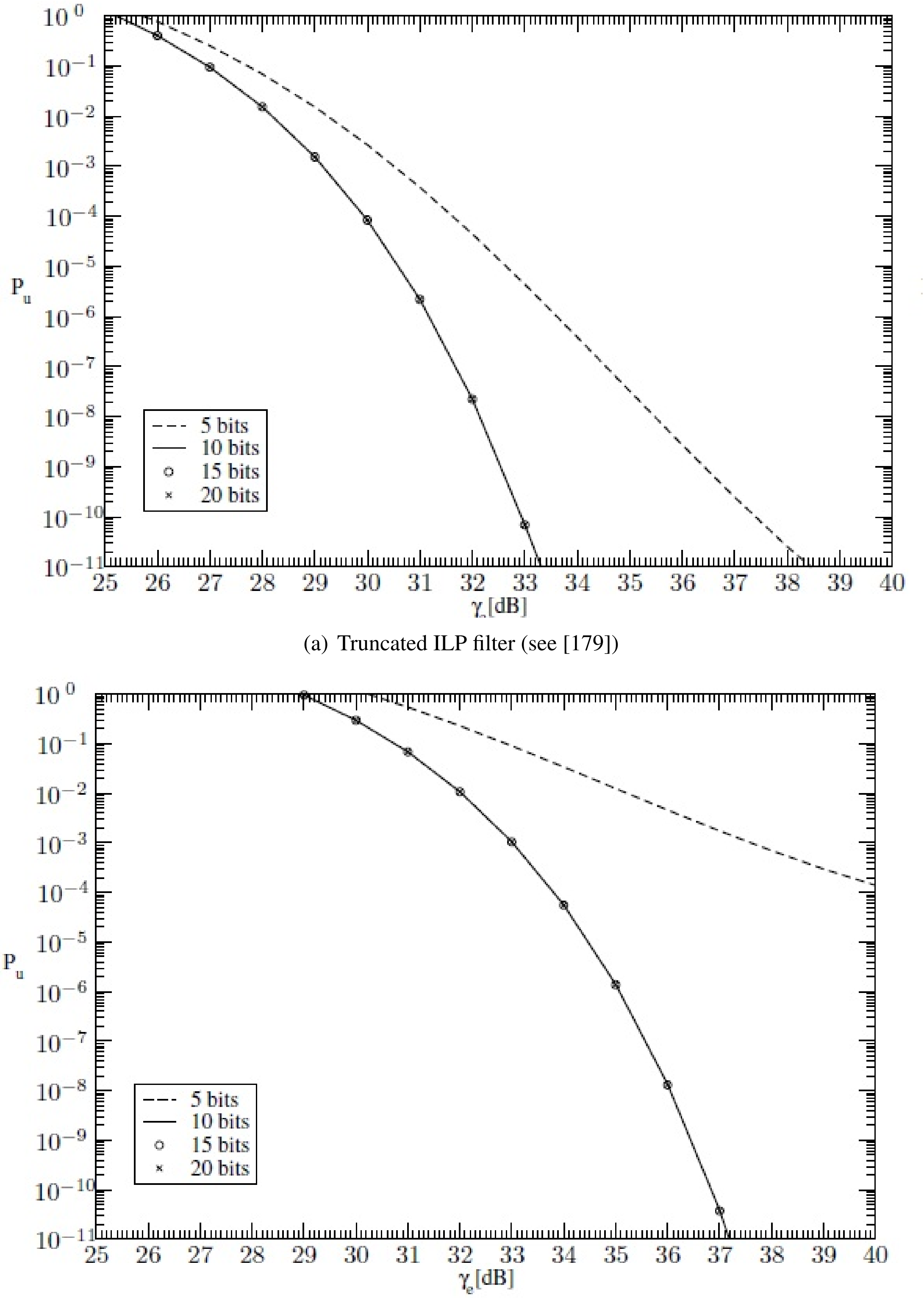
$$\Lambda(r) \triangleq \sum_{l=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} \left| l - \frac{\sum_{k=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} k r^{2k}}{\sum_{k=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} r^{2k}} \right|^2 r^{2l} \tag{6.21}$$

$$\Psi(r) \triangleq \left| k_{\mathrm{M}} - \frac{\sum_{k=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} k r^{2k}}{\sum_{k=0}^{k_{\mathrm{M}}-k_{\mathrm{m}}} r^{2k}} \right|^2 r^{2k_{\mathrm{M}}} . \tag{6.22}$$

6.3 Numerical Results

Numerical results have been derived assuming $B_{\mathrm{eq}} = 7.61\ MHz$, $T_{\mathrm{s}} = 80\ ns$, and $\frac{P_{\mathrm{s}}}{N_0 B_{\mathrm{eq}} + P_{\mathrm{x}}} = 0\ dB$. These choice originate from the echo canceller prototype I implemented following the criteria outlined in [168]. In Table 6.3 I reported both the specifications of the filter cascade $H(f)$ (see (6.3)) implemented in our OCR prototype[7] and of the truncated ILP filter cascade assumed in [179]. The latter has been considered for a full comparison with the results presented

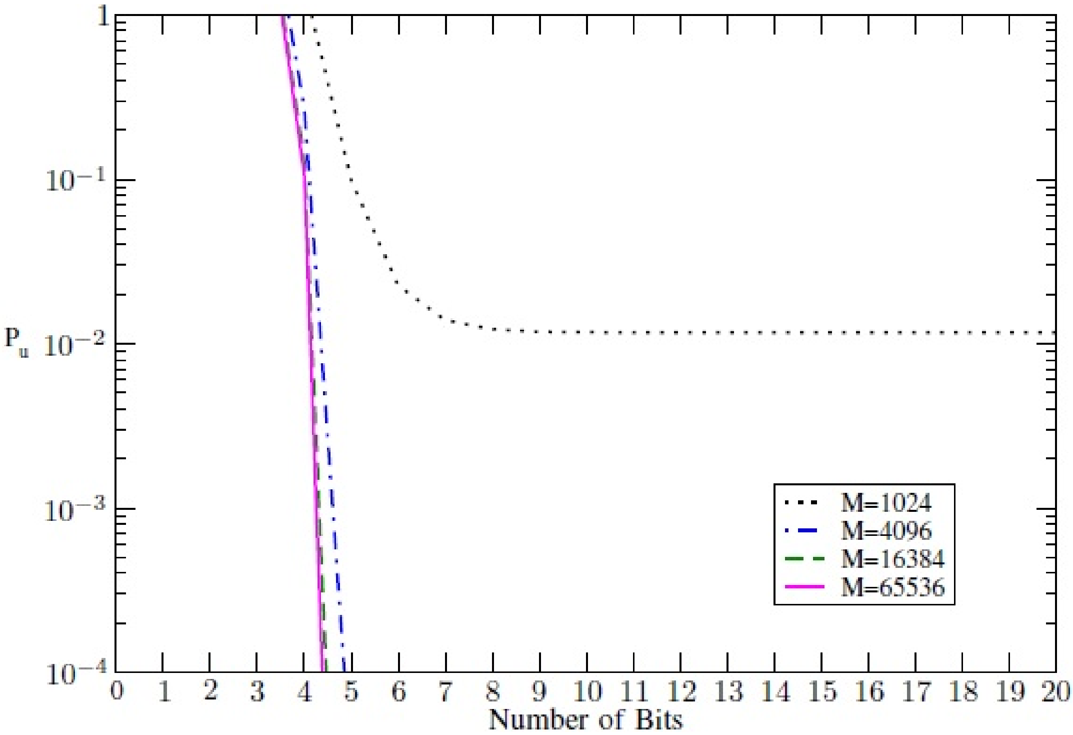
[7] The filter design criteria are detailed in [182].



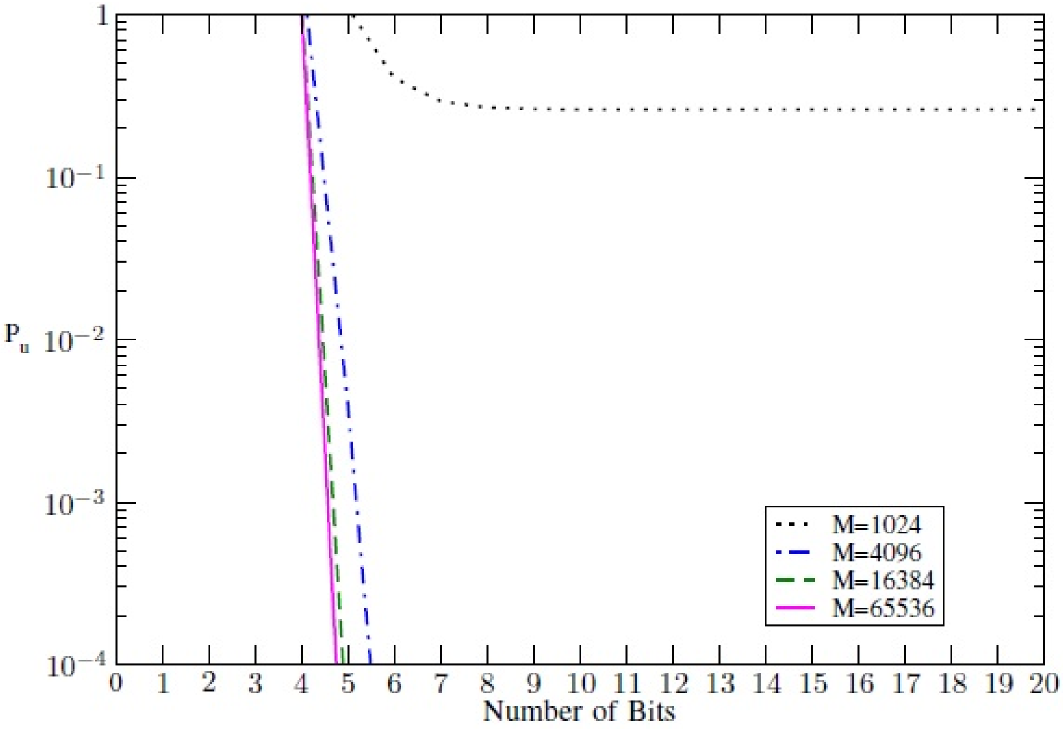
(a) Truncated ILP filter (see [179])

(b) Implemented filter

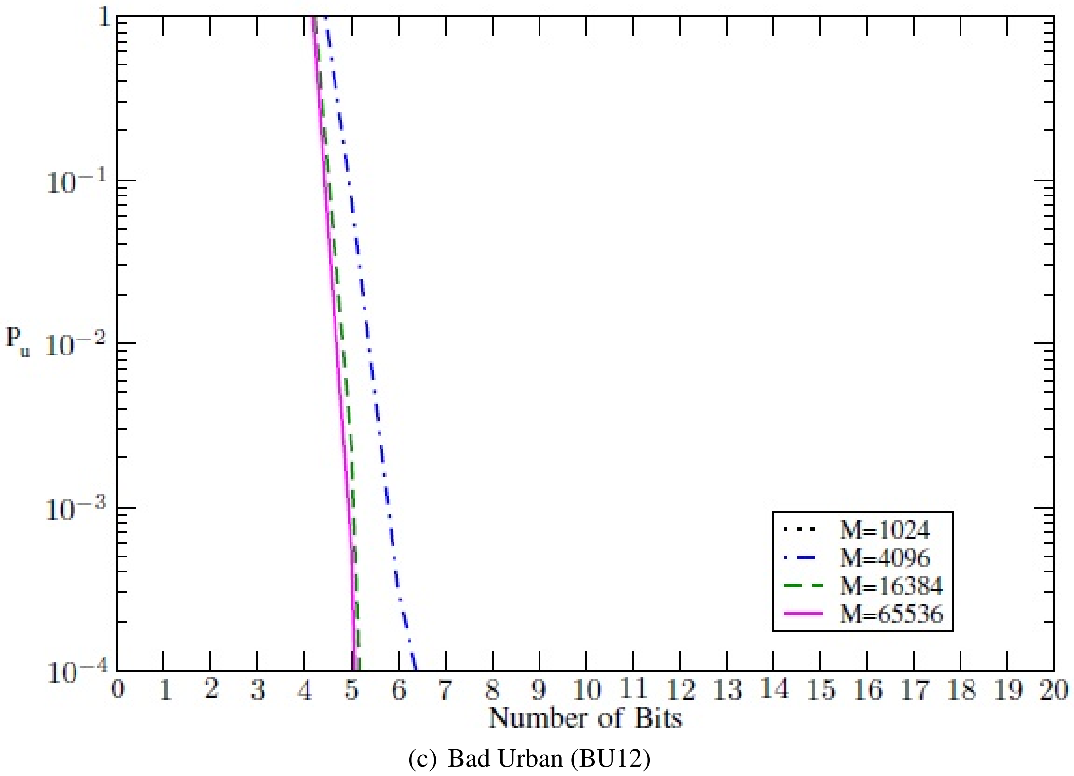
Figure 6.4: Probability of Instability as a function of the estimation SNR for different numbers of quantization bits. TU12 channel



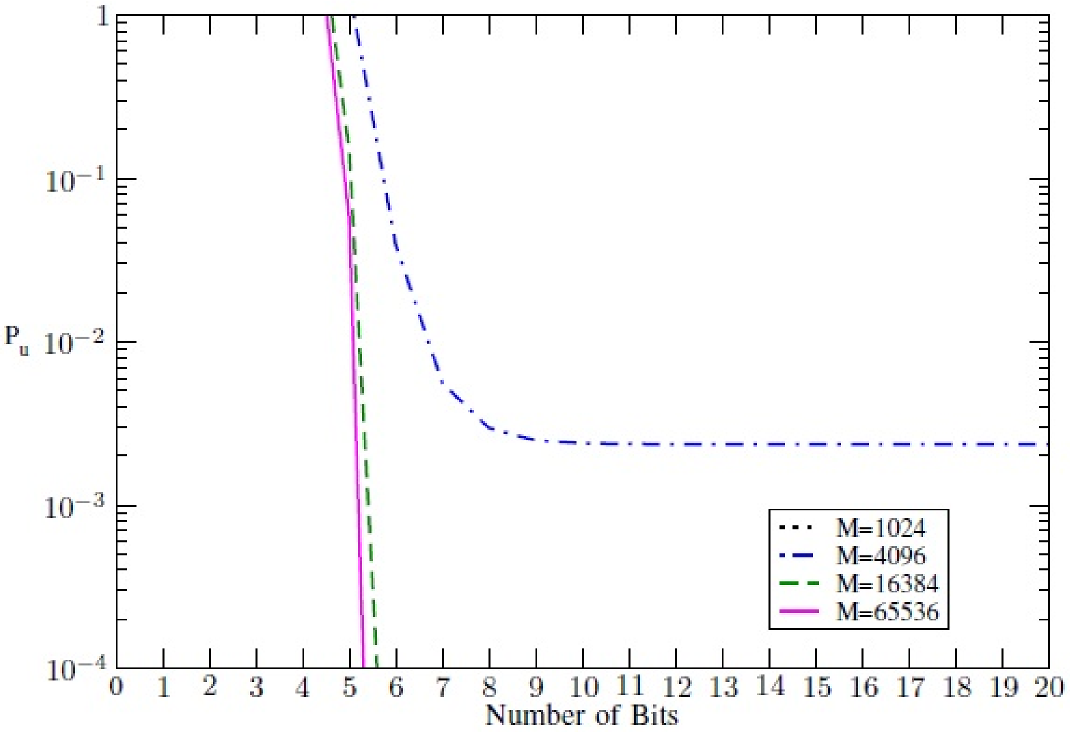
(a) Rural Area (RA6)



(b) Typical Urban (TU12)



(c) Bad Urban (BU12)



(d) Hilly Terrain (HT12)

Figure 6.5: Probability of Instability as a function of the number of quantization bits for different length of training sequences. Implemented filter.

Table 6.3: Filters specifications

Specifications	Implemented filter	Truncated ILP filter [179]
pass-band bandwidth	7.61 MHz	7.61 MHz
pass-band ripple	0.2 dB	0 dB
out-of-band attenuation	65 dB	∞ dB
τ_{g}	4.47 μs	6.37 μs
τ_{IR}	8.94 μs	0.52 μs

in [179], that do not take into account the quantization effect. The corresponding values for τ_{g} and τ_{IR} are reported in Table 6.3.

The impact on P_{u} of both the quantization and the estimation accuracy is shown in figures 6.4(a) and 6.4(b), where P_{u} is plotted as a function of the estimation SNR γ_{e} based on different significant values of quantization bits n_{b}, for the truncated ILP filter considered in [179] and for the really implemented filter described in [168], respectively. I assumed a urban environment (TU12) and the different curves refer to different values of n_{b}.
It can be noticed that, as expected, P_{u} tends to zero for increasing values of γ_{e} for all the considered n_{b}: better coupling-channel estimations result in more robust operating conditions for the OCR. In figures 6.4(a) and 6.4(b) we can also appreciate the impact of the quantization, that is surely negligible when n_b is larger than 10. Let us observe, moreover, that all curves in Fig. 6.4(a) for $n_{\mathrm{b}} \geq 10$ coincide with the curve corresponding to $L = 12$ reported in [179], where the same repeater filtering and channel model are considered, and the quantization effect is not considered. A comparison between figures 6.4(a) and 6.4(b) also shows the effect of actual filtering, that is usually neglected by the literature on this topic.

The impact of the channel (rural area, urban area, bad urban and hilly terrain) is investigated in figures 6.5(a), 6.5(b), 6.5(c), and 6.5(d), that show P_{u} as a function of the number of quantization bits for different length of training sequences. As can be observed in Fig. 6.5(a), that refers to rural areas, provided that the training sequence is long enough (e.g., $M \geq 4096$), reasonable values ($< 10^{-4}$) of P_{u} can be achieved with $n_b \geq 5$. Similar results can be obtained for the other channel models considered, as shown in figures 6.5(b), 6.5(c), and 6.5(d). They are summarized in Table 6.4.

6.4 Conclusions

In this work I showed the impact of both the non perfect echo estimation and the echo canceller quantization on the stability of digital OCRs that estimates the coupling-channel by means of

Table 6.4: Minimum number of quantization bits for $P_{\mathrm{u}} \leq 10^{-4}$

Length of training sequence	Rural Area (RA6)	Typical Urban (TU12)	Bad Urban (BU12)	Hilly Terrain (HT12)
M=1024	-	-	-	-
M=4096	5	6	7	-
M=16384	5	5	6	6
M=65536	5	5	6	6

training sequences. In particular, I derived an analytical expression for the upper bound of the probability of instability as a function of estimation SNR and number of quantization bits. Numerical results were derived for realistic filtering and common models for the coupling-channel.

The results show that, provided that the training sequence is long enough, a limited amount of bits ($5-7$ depending on the coupling-channel model) for the quantization of the canceller taps is sufficient. Any further increase has a totally negligible impact on the stability of the OCR. On the contrary, an increase of the estimation SNR always leads to a decreasing probability of instability. It means that, while the number of quantization bits adopted in the ADC and the DAC as well as the number of bits used to quantize the taps of the receive and transmit filters must be large enough (~ 16) to fulfill the strict requirements on the signal quality, the number of quantization bits of the echo canceller taps can be much lower, since the stability characteristics of the OCR mainly depends on the filtering, the coupling-channel between the antennas and the estimation errors, as analyzed in [180].

Conclusions

In conclusion, during my Ph.D I focused my research on heterogeneous networks. In particular, Wireless Sensor Networks and Vehicular Networks have been investigated.

Concerning irregular sampling, an optimal linear space-invariant interpolator has been derived. I analyzed the estimation of a finite-energy signal from its samples affected by measurement errors and scattered in $\mathbb{R}^d$ according to a homogeneous poisson point process. The effect of the distortion due to measurement errors on the normalized estimation mean square error is equivalent to that of a reduction of samples intensity, which can be compensated for by increasing the number of nodes inside the sampling area. An increasing number of sensors leads to a decreasing normalized estimation mean square error in spite of the corresponding increasing measurement error for each sensor, if a constraint in the overall estimation energy is imposed. A simple but significant expression for the normalized estimation mean square error as a function of the estimation time and the capacity-per-volume has been derived, when the energy constraint is imposed on each sensor due to the battery lifetime limitation.

I also introduced and investigated a novel information diffusion strategy, namely TAS. The TAS algorithm has been designed to efficiently exploit the peculiarities of the distributed evaluation of confidence regions via SPS. The performance of TAS algorithm was compared with other information diffusion algorithms on structured and unstructured topologies. Simulation results provide a characterization of the trade-off for the achievable average confidence region volume as a function of the required amount of data that each node should transmit on average. In particular, the proposed TAS algorithm outperforms the FL in unstructured topologies.

Another novel algorithm, named Finite Time Consensus with Memomry (FTCM), was introduced for the distributed evaluation of the average consensus, which exploits the node memory to facilitate the consensus evaluation. I also derived an adaptation to the distributed average consensus problem of the TAS algorithm for the distributed confidence region evaluation. The performances of the two algorithms, in terms of efficiency in the usage of network resources and convergence speed, have been compared with those of classic or recently introduced algorithms, such as Metropolis Consensus, Finite Time Consensus, Flooding and Network Coding. The outcomes of performance investigations, carried out considering different topologies, show that FTCM is very well behaving when operated on unstructured random network topologies, whereas TAS outperforms its competitors when structured networks are considered, either tree or clustered networks.

Then, an introduction and evaluation of the performance of a new Distributed Faulty Node Detection algorithm in Delay Tolerant Networks was derived. In this work, I investigated the impact of Byzantine attacks on the performance of a distributed faulty node detection algorithm in the context of delay tolerant networks. The affect of Byzantine attack on the equilibrium is analyzed theoretically, which is helpful to adjust the algorithm parameters in order to ensure the robustness of the DFD algorithm. Both ideal movement model and real databases have been considered in the simulations to achieve the results.

During my Ph.D studies, I have also investigated the feasibility and the performance of VLCs in vehicular networks (VNs) for cellular network offloading. Specifically, I have studied the level of connectivity in a urban environment and the feasibility in delivering data for delay tolerant applications without exploiting the cellular network. Numerical results have been obtained through simulations in a realistic urban scenario, showing that even a limited number of traffic lights equipped with VLC allows to offload more than 90% of cellular resources. Furthermore, focusing on the impact of different transmitter and receiver characteristics, it was demonstrated that improving the coverage distance or the FOV does not always lead to a higher performance; the improved coverage does in fact increase the average number of neighbors, with an higher collision probability and consequently a lower data delivery under high network load conditions.

Still in the framework of VNs, I focused my attention on the adoption of VLC as supplementary technology to the RF ones for data exchanging between vehicles and between vehicles and RSUs in vehicular networks. I proposed to let the two technologies cooperate to increase the rate of packets delivered through multi hop V2V communications toward an RSU. A cooperative algorithm to adaptively select the technology has been also proposed, with a single parameter allowing to move from VLC always preferred to DSRC to the opposite case. Simulations, performed in realistic urban scenarios with hundred of vehicles, demonstrated the significant improvement obtained by adding VLC to DSRC. The best results were obtained by giving priority to DSRC when its channel is far from being congested, and preferring VLC in the other cases.

I also studied the impact of echo canceller taps quantization on signal repeaters stability. In particular, I showed the impact of both non perfect echo estimation and quantization on the stability of digital on-channel repeater that estimates the coupling-channel by means of training sequences. Results show that, provided that the training sequence is long enough, a limited amount of bits for the quantization of the canceller taps is sufficient. Any further increase has a totally negligible impact on the stability of the on-channel repeater. On the contrary, an increase of the estimated Signal to Noise Ratio always leads to a decreased probability of instability. Therefore, while the number of quantization bits adopted in the ADC and the DAC and the number of bits used to quantize the taps of the receive and transmit filters must be large enough to fulfill the strict requirements on the signal quality, the number of quantization bits of the echo canceller taps can be much lower, since the stability characteristics of the on-channel repeater mainly depends on the filtering, the coupling-channel between the antennas and the estimation errors.

Bibliography

[1] F. Zabini, A. Calisti, D. Dardari, and A. Conti, "Random sampling via sensor networks: Estimation accuracy vs. energy consumption," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 130–134.

[2] Alex Calisti, Davide Dardari, Gianni Pasolini, Michel Kieffer, and Francesca Bassi, "Experimental characterization of information diffusion algorithms for confidence region evaluation over wireless sensor networks," Jncw 2015, University of Bologna and Supelec, Barcelona, Spain, Oct. 2015.

[3] A. Calisti, D. Dardari, G. Pasolini, M. Kieffer, and F. Bassi, "Information diffusion algorithms over wsns for non-asymptotic confidence region evaluation," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[4] V. Zambianchi, F. Bassi, A. Calisti, D. Dardari, M. Kieffer, and G. Pasolini, "Distributed non-asymptotic confidence region computation over sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. PP, no. 99, pp. 1–16, 2017.

[5] W. Li, F. Bassi, M. Kieffer, A. Calisti, G. Pasolini, and D. Dardari, "Distributed faulty node detection in dtns in presence of byzantine attack," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[6] A. Bazzi, B.M. Masini, A. Zanella, and A. Calisti, "Visible light communications in vehicular networks for cellular offloading," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, June, pp. 1416–1421.

[7] Alessandro Bazzi, Barbara M. Masini, Alberto Zanella, and Alex Calisti, "Visible light communications as a complementary technology for the internet of vehicles," *Computer Communications*, vol. 93, no. Supplement C, pp. 39 – 51, 2016, Multi-radio, Multi-technology, Multi-system Vehicular Communications.

[8] F. Zabini, A. Calisti, and G. Pasolini, "Impact of echo canceller taps quantization on repeater stability," in *2015 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, May 2015, pp. 186–190.

[9] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, *Wireless Sensor and Actuator Networks: technologies, analysis and design*, Elsevier, 2008.

[10] H. Landau, "Necessary density conditions for sampling and interpolation of certain entire functions," *Acta Math.*, vol. 117, pp. 37–52, 1967.

[11] H. Landau, "Sampling data transmission, and the nyquist rate," *Proc. IEEE*, vol. 55, no. 10, pp. 1701–1706, 1967.

[12] Farock A. Marvasti, *Non Uniform Sampling: Theory and Practice*, Springer, June 2001.

[13] Farokh A. Marvasti, "Signal recovery from nonuniform samples and spectral analysis on random nonuniform samples," Tokyo, Japan, 1986, pp. 1649–1652.

[14] D. Dardari, A. Conti, C. Buratti, and R. Verdone, "Mathematical evaluation of environmental monitoring estimation error through energy-efficient wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 7, pp. 790–802, jul 2007.

[15] Animesh Kumar, Prakash Ishwar, and Kannan Ramchandran, "High-resolution distributed sampling of bandlimited fields with low-precision sensors," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 476–492, Jan. 2011.

[16] Javier Matamoros, Flavio Fabbri, Carles Anton-Haro, and Davide Dardari, "On the estimation of randomly sampled 2D spatial fields under bandwidth constraints," *IEEE Trans. Wireless Commun.*, vol. 10, no. 12, pp. 4184–4192, Dec. 2011.

[17] Flavio Zabini and Andrea Conti, "Process estimation from randomly deployed wireless sensors with position uncertainty," in *Globecom*, Houston, USA, Dec. 2011.

[18] GÃ¼nter Reise, Gerald Matz, and Karlheinz Grochenig, "Distributed field reconstruction in wireless sensor networks based on hybrid shift-invariant spaces," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5426 – 5439, Oct. 2012.

[19] Ning Sun and Jingxian Wu, "Optimum sampling in spatial-temporally correlated wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, 2013, doi:10.1186/1687-1499-2013-5, 18 pages.

[20] Moe Z. Win, P. C. Pinto, and L. A. Shepp, "A mathematical theory of network interference and its applications," *Proc. IEEE*, vol. 97, no. 2, pp. 205–230, Feb. 2009.

[21] E. Salbaroli and A. Zanella, "Interference analysis in a Poisson field of nodes of finite area," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 1776–1783, May 2009.

[22] M. Haenggi, J. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti, "Stochastic geometry and random graphs for the analysis and design of wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 7, pp. 1776–1783, Sept. 2009.

[23] A. Rabbachin, Andrea Conti, and Moe Z. Win, "Wireless network intrinsic secrecy," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 56–69, Feb 2015.

[24] O. A. Z. Leneman, "Random sampling of random processes: Optimum linear interpolation," *J. of Franklin Inst.*, vol. 4, no. 281, pp. 302–314, 1966.

[25] O. A. Z. Leneman, "Random sampling of random processes: Impulse processes," *Inf. and Control*, , no. 9, pp. 347–363, 1966.

[26] O. A. Z. Leneman and J. B. Lewis, "Random sampling of random processes, mean-square comparison of various interpolators," *IEEE Trans. Autom. Control*, vol. AC, no. 11, pp. 396–403, 1966.

[27] Alessandro Nordio, Carla Fabiana Chiasserini, and Emanuele Viterbo, "Performance of linear field reconstruction techniques with noise and uncertain sensor locations," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3535,3547, Aug. 2008.

[28] B.C. Csáji, Marco C. Campi, and Erik Weyer, "Non-asymptotic confidence regions for the least-squares estimate," in *Proc. IFAC SYSID, Brussels, Belgium*, Jun. 2012, pp. 227–232.

[29] V. Zambianchi, M. Kieffer, F. Bassi, Gianni Pasolini, and Davide Dardari, "Distributed sps algorithms for non-asymptotic confidence region evaluation," in *Networks and Communications (EuCNC), 2014 European Conference on*, Jun. 2014, pp. 1–5.

[30] Elias M. Stein and Guido Weiss, *Introduction To Fourier Analysis on Euclidean Spaces*, Princeton University Press, 1971.

[31] John F.C. Kingman, *Poisson Processes*, Oxford University Press, 1st edition, 1993.

[32] Athanasios Papoulis and S. Unnikrishna Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 2002.

[33] H. Robbins, “An extension of wiener filter theory to partly sampled systems,” *IRE Transactions on Circuit Theory*, vol. 6, no. 4, pp. 362–370, December 1959.

[34] H. A. Khan and L. F. Chaparro, “Non-stationary wiener filtering based on evolutionary spectral theory,” in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Apr 1997, vol. 5, pp. 3677–3680 vol.5.

[35] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.

[36] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson, “Wireless sensor networks for habitat monitoring,” in *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, New York, USA, September 2002, pp. 88–97.

[37] T.Q.S. Quek, D. Dardari, and M.Z. Win, “Energy efficiency of dense wireless sensor networks: to cooperate or not to cooperate,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 2, pp. 459–470, February 2007.

[38] S. M. Kay, *Fundamentals of Statistical Processing: Estimation Theory*, vol. 1, Prentice-Hall Signal Processing Series, 1993.

[39] I. D. Schizas, G. Mateos, and G. B. Giannakis, “Distributed LMS for consensus-based in-network adaptive processing,” *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365–2382, June 2009.

[40] G. Mateos, I. D. Schizas, and G. B. Giannakis, “Distributed recursive least-squares for consensus-based in-network adaptive estimation,” *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4583–4588, November 2009.

[41] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, January 2007.

[42] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *Proc. of the 46th IEEE Conference on Decision and Control*, New Orleans, USA, December 2007, pp. 5492–5498.

[43] R. Olfati-Saber, “Kalman-consensus filter: Optimality, stability, and performance,” in *Proc. of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 2009, pp. 7036–7042.

[44] B. Yang and J. Scheuing, “Cramér-Rao bound and optimum sensor array for source localization from time differences of arrival,” in *Proc. of IEEE Acoustics, Speech, and Signal Processing, ICASSP*, Philadelphia, PA, USA, March 2005, pp. 961–964.

[45] Xiaohong Sheng and Yu-Hen Hu, “Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks,” *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 44–53, Jan 2005.

[46] N. Patwari, J.N. Ash, S. Kyperountas, AO. Hero, R.L. Moses, and N.S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, July 2005.

[47] D. Jourdan, D. Dardari, and M.Z. Win, “Position error bound for UWB localization in dense cluttered environments,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 2, pp. 613–628, April 2008.

[48] M. C. Campi and E. Weyer, “Guaranteed non-asymptotic confidence regions in system identification,” *Automatica*, vol. 41, no. 10, pp. 1751–1764, October 2005.

[49] M. C. Campi, S. Ko, and E. Weyer, “Non-asymptotic confidence regions for model parameters in the presence of unmodelled dynamics,” *Automatica*, vol. 45, no. 10, pp. 2175–2186, October 2009.

[50] B. C. Csáji, M. C. Campi, and E. Weyer, “Signed-Perturbed Sums: A new system identification approach for constructing exact non-asymptotic confidence regions in linearregression models,” *IEEE Trans. on Signal Processing*, vol. 63, no. 1, pp. 169–181, January 2015.

[51] S. Kolumban, I. Vajk, and J Schoukens, “Perturbed datasets methods for hypothesis testing and structure of corresponding confidence sets,” *Automatica*, vol. 51, pp. 326–331, January 2015.

[52] M. Kieffer and E. Walter, “Guaranteed characterization of exact non-asymptotic confidence regions as defined by LSCR and SPS,” *Automatica*, vol. 50, no. 2, pp. 507–512, February 2014.

[53] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” in *Proc. of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, WA, USA, August 1999, MobiCom ’99, pp. 174–185, ACM.

[54] C. Perkins and E. Royer, “Ad hoc on-demand distance vector (aodv) routing,” 2003, RFC 3561.

[55] E. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*, Springer-Verlag, London, 1997.

[56] S.-Y.R. Li, R.W. Yeung, and Ning Cai, “Linear network coding,” *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[57] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct 2003.

[58] Lin Xiao and Stephen Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[59] L. Xiao, S. Boyd, and S. Lall, “A scheme for robust distributed sensor fusion based on average consensus,” in *Proc. of the Fourth International Symposium on Information Processing in Sensor Networks, IPSN 2005*, Los Angeles, USA, April 2005, pp. 63–70.

[60] Jin-Jun Xiao, A. Ribeiro, Zhi-Quan Luo, and G.B. Giannakis, “Distributed compression-estimation using wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 23, no. 4, pp. 27–41, 2006.

[61] L. Xiao, S. Boyd, and S. J. Kim, “Distributed average consensus with least-mean-square deviation,” *Journal of Parallel and Distributing Computing*, vol. 67, no. 1, pp. 34–46, 2007.

[62] Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra, “An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks,” *Wireless Communications and Mobile Computing*, vol. 7, no. 7, pp. 863–875, 2007.

[63] S.M. Rump, “INTLAB - INTerval LABoratory,” in *Developments in Reliable Computing*, Tibor Csendes, Ed., pp. 77–104. Kluwer Academic Publishers, Dordrecht, 1999.

[64] P. Gupta and P.R. Kumar, “The capacity of wireless networks,” *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar 2000.

[65] EMBIT, “EMB-Z2530PA,” Apr. 2016.

[66] “Ieee standard for low-rate wireless networks,” *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pp. 1–709, April 2016.

[67] R. Verdone, D. Dardari, G. Mazzini, and A. Conti, ," in *Wireless Sensor and Actuator Networks: technologies, analysis and design*. 2008, Elsevier Ltd, London.

[68] S. Kar and J. M. F. Moura, "Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 99–109, May 2013.

[69] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355–369, Jan 2009.

[70] A. Sandryhaila, S. Kar, and J. M. F. Moura, "Finite-time distributed consensus through graph filters," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1080–1084.

[71] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, Sept 2005, pp. 264–267.

[72] S. Y. R. Li, R. W. Yeung, and Ning Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[73] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, Oct 2003.

[74] Maurice J Khabbaz, Chadi M Assi, and Wissam F Fawaz, "Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.

[75] Paulo Rogério Pereira, Augusto Casaca, Joel JPC Rodrigues, Vasco NGJ Soares, Joan Triay, and Cristina Cervelló-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1166–1182, 2012.

[76] Wei Peng, Feng Li, Xukai Zou, and Jie Wu, "Behavioral malware detection in delay tolerant networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 53–63, 2014.

[77] Haojin Zhu, Suguo Du, Zhaoyu Gao, Mianxiong Dong, and Zhenfu Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," *IEEE Trans. on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 22–32, 2014.

[78] Erman Ayday and Faramarz Fekri, "An iterative algorithm for trust management and adversary detection for delay-tolerant networks," *IEEE Trans. on Mobile Computing*, vol. 11, no. 9, pp. 1514–1531, 2012.

[79] Yang Zhang, Nirvana Meratnia, and Paul Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

[80] Arunanshu Mahapatro and Pabitra Mohan Khilar, "Fault diagnosis in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2000–2026, 2013.

[81] Wenjie Li, Francesca Bassi, Davide Dardari, Michel Kieffer, and Gianni Pasolini", "Defective sensor identification for WSNs involving generic local outlier detection tests," *IEEE Trans. on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 29–48, 2016.

[82] Wenjie Li, Laura Galluccio, Michel Kieffer, and Francesca Bassi, "Distributed faulty node detection in DTNs," in *Proc. International Conference on Computer Communication and Networks*, 2016.

[83] Hongzi Zhu, Luoyi Fu, Guangtao Xue, Yanmin Zhu, Minglu Li, and L.M. Ni, "Recognizing exponential inter-contact time in vanets," in *Proc. INFOCOM*, March 2010, pp. 1–5.

[84] James Scott, Richard Gass, Jon Crowcroft, Pan Hui, Christophe Diot, and Augustin Chaintreau, "CRAWDAD dataset cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.org/cambridge/haggle/20090529, May 2009.

[85] World Health Organization, "Global status report on road safety," Tech. Rep., WHO, 2013.

[86] U.S. Department of Transportation, "Vehicle safety communications project task 3 final report - identify intelligent vehicle safety applications enabled by dsrc," Tech. Rep., National Highway Traffic Safety Administration, 2005.

[87] "NHSTA web page," Accessed on March 2016.

[88] "Standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks-specific requirements part 11 - amendment 6: Wireless access in vehicular environment," 2010.

[89] "IEEE trial-use standard for wireless access in vehicular environments (WAVE) - multi-channel operation," 2006.

[90] "ETSI-CEN cooperative intelligent transport systems standards, release 1 for early procurement," Accessed on March 2016.

[91] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *Communications Magazine, IEEE*, vol. 51, no. 5, pp. 148–157, May 2013.

[92] A. Bazzi, B. M. Masini, and A. Zanella, "Performance analysis of V2V beaconing using LTE in direct mode with full duplex radios," *IEEE Wireless Communications Letters*, vol. 4, no. 6, pp. 685–688, Dec 2015.

[93] Mario Gerla and Leonard Kleinrock, "Vehicular networks and the future of the mobile internet," *Computer Networks*, vol. 55, no. 2, pp. 457 – 469, 2011, Wireless for the Future Internet.

[94] C. Sommer, O.K. Tonguz, and F. Dressler, "Traffic information systems: Efficient message dissemination via adaptive beaconing," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 173–179, 2011.

[95] Sherali Zeadally, Ray Hunt, Yuh-Shyan Chen, Angela Irwin, and Aamir Hassan, "Vehicular ad hoc networks (VANETs): status, results, and challenges," *Telecommunication Systems*, vol. 50, no. 4, pp. 217–241, 2012.

[96] M. Amadeo, C. Campolo, and A. Molinaro, "Enhancing IEEE 802.11p/WAVE to provide infotainment applications in VANETs," *Ad Hoc Networks*, vol. 10, no. 2, pp. 253–269, 2012.

[97] Saif Al-Sultan, Moath M. Al-Doori, Ali H. Al-Bayatti, and Hussien Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of Network and Computer Applications*, vol. 37, pp. 380 – 392, 2014.

[98] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 166–172, Dec. 2014.

[99] A. Bazzi, B.M. Masini, and O. Andrisano, "On the frequent acquisition of small data through RACH in UMTS for ITS applications," *Vehicular Technology, IEEE Transactions on*, vol. 60, no. 7, pp. 2914–2926, Sep. 2011.

[100] A. Bazzi, B.M. Masini, and G. Pasolini, "V2V and V2R for cellular resources saving in vehicular applications," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, Apr., pp. 3199–3203.

[101] Alessandro Bazzi, Barbara M. Masini, Alberto Zanella, and Gianni Pasolini, "IEEE 802.11p for cellular offloading in vehicular sensor networks," *Computer Communications*, vol. 60, pp. 97 – 108, 2015.

[102] G. Corbellini, K. Aksit, S. Schmid, S. Mangold, and T. Gross, "Connecting networks of toys and smartphones with visible light communication," *Communications Magazine, IEEE*, vol. 52, no. 7, pp. 72–78, Jul. 2014.

[103] A. Devi and S. Prince, "Design and implementation of a radio on visible light system for indoor communication," in *Communications and Signal Processing (ICCSP), 2014 International Conference on*, Apr. 2014, pp. 132–136.

[104] N.O. Tippenhauer, D. Giustiniano, and S. Mangold, "Toys communicating with leds: Enabling toy cars interaction," in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, Jan 2012, pp. 48–49.

[105] D. Giustiniano, N.O. Tippenhauer, and S. Mangold, "Low-complexity visible light networking with led-to-led communication," in *Wireless Days (WD), 2012 IFIP*, Nov 2012, pp. 1–8.

[106] "Standard for local and metropolitan area networks-part 15.7: Short-range wireless optical communication using visible light," .

[107] G. Cossu, R. Corsini, A.M. Khalid, S. Balestrino, A. Coppelli, A. Caiti, and E. Ciaramella, "Experimental demonstration of high speed underwater visible light communications," in *Optical Wireless Communications (IWOW), 2013 2nd International Workshop on*, Oct, pp. 11–15.

[108] G. Cossu, M. Presi, R. Corsini, P. Choudhury, A.M. Khalid, and E. Ciaramella, "A visible light localization aided optical wireless system," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, Dec, pp. 802–807.

[109] Cen B. Liu, Bahareh Sadeghi, and Edward W. Knightly, "Enabling vehicular visible light communication (V2LC) networks," in *Proceedings of the Eighth ACM International Workshop on Vehicular Inter-networking*, New York, NY, USA, VANET 2011, pp. 41–50, ACM.

[110] Shun-Hsiang Yu, O. Shih, Hsin-Mu Tsai, N. Wisitpongphan, and R. Roberts, "Smart automotive lighting for vehicle safety," *Communications Magazine, IEEE*, vol. 51, no. 12, pp. 50–59, December 2013.

[111] A. Mostafa and L. Lampe, "Pattern synthesis of massive LED arrays for secure visible light communication links," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*, June, pp. 1350–1355.

[112] A. Mostafa and L. Lampe, "Physical-layer security for MISO visible light communication channels," *Selected Areas in Communications, IEEE Journal on*, vol. 33, no. 9, pp. 1806–1818, Sept 2015.

[113] Qualcomm, "V2x communication in 3gpp," 2014.

[114] "3gpp: Initial cellular v2x standard completed," Sep. 2016.

[115] H. Seo, K. D. Lee, S. Yasukawa, Y. Peng, and P. Sartori, "Lte evolution for vehicle-to-everything services," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 22–28, June 2016.

[116] L. C. Tung and M. Gerla, "Lte resource scheduling for vehicular safety applications," in *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, March 2013, pp. 116–118.

[117] A. Bazzi, B. M. Masini, A. Zanella, and I. Thibault, "On the performance of ieee 802.11p and lte-v2v for the cooperative awareness of connected vehicles," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10419–10432, Nov 2017.

[118] M.A. Benatia, L. Khoukhi, M. Esseghir, and L.M. Boulahia, "A markov chain based model for congestion control in VANETs," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, March, pp. 1021–1026.

[119] Quincy Tse, Weisheng Si, and J. Taheri, "Estimating contention of IEEE 802.11 broadcasts based on inter-frame idle slots," in *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*, Oct, pp. 120–127.

[120] Lin Cheng and Han Wu, "A study on rush hour vehicular communication blockage," in *Antennas and Propagation (APSURSI), 2011 IEEE International Symposium on*, July, pp. 829–831.

[121] A. Vizzarri, “Analysis of VoIP over LTE end-to-end performances in congested scenarios,” in *Artificial Intelligence, Modelling and Simulation (AIMS), 2014 2nd International Conference on*, Nov, pp. 339–343.

[122] J.C. Medina and R.F. Benekohal, “Vehicle detection design (coverage and accuracy) and the performance of congested traffic networks,” in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, Sept, pp. 179–184.

[123] N. Kumar, N. Lourenco, D. Terra, L.N. Alves, and Rui L. Aguiar, “Visible light communications in intelligent transportation systems,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, June, pp. 748–753.

[124] T. Yamazato, I. Takai, H. Okada, T. Fujii, T. Yendo, S. Arai, M. Andoh, T. Harada, K. Yasutomi, K. Kagawa, and S. Kawahito, “Image-sensor-based visible light communication for automotive applications,” *Communications Magazine, IEEE*, vol. 52, no. 7, pp. 88–97, July 2014.

[125] Qing Wang and Domenico Giustiniano, “Communication networks of visible light emitting diodes with intra-frame bidirectional transmission,” in *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, New York, NY, USA, CoNEXT ’14, pp. 21–28, ACM.

[126] D. Kim, H. Lee, and D. Hong, “A survey of in-band full-duplex transmission: From the perspective of PHY and MAC layers,” *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–1, 2015.

[127] Dinesh Bharadia, Emily McMilin, and Sachin Katti, “Full duplex radios,” in *Proceedings of the ACM SIGCOMM 2013 Conference*, New York, NY, USA, pp. 375–386, ACM.

[128] L. Ekiz, A. Thiel, O. Klemp, and C.F. Mecklenbrauker, “MIMO performance evaluation of automotive qualified LTE antennas,” in *7th European Conference on Antennas and Propagation, EuCAP 2013*, pp. 1412–1416.

[129] Peng Ji, Hsin-Mu Tsai, Chao Wang, and Fuqiang Liu, “Vehicular visible light communications with LED taillight and rolling shutter camera,” in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*, May, pp. 1–6.

[130] C. Ide, B. Niehoefer, T. Knaup, D. Weber, C. Wietfeld, L. Habel, and M. Schreckenberg, “Efficient floating car data transmission via LTE for travel time estimation of vehicles,” in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, Sept, pp. 1–5.

[131] Yingsi Liang, Hui Liu, and D. Rajan, "Optimal placement and configuration of roadside units in vehicular networks," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, May, pp. 1–6.

[132] S. Rajagopal, R.D. Roberts, and Sang-Kyu Lim, "Ieee 802.15.7 visible light communication: modulation schemes and dimming support," *Communications Magazine, IEEE*, vol. 50, no. 3, pp. 72–82, March 2012.

[133] R. Corsini, R. Pelliccia, G. Cossu, A.M. Khalid, M. Ghibaudi, M. Petracca, P. Pagano, and E. Ciaramella, "Free space optical communication in the visible bandwidth for V2V safety critical protocols," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*, Aug, pp. 1097–1102.

[134] A.-M. Cailean, B. Cagneau, L. Chassagne, S. Topsu, Y. Alayli, and M. Dimian, "Visible light communications cooperative architecture for the intelligent transportation system," in *Communications and Vehicular Technology in the Benelux (SCVT), 2013 IEEE 20th Symposium on*, Nov, pp. 1–5.

[135] Jong-Ho Yoo, Rimhwan Lee, Jun-Kyu Oh, Hyun-Wook Seo, Ju-Young Kim, Hyeon-Cheol Kim, and Sung-Yoon Jung, "Demonstration of vehicular visible light communication based on LED headlamp," in *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*, July, pp. 465–467.

[136] B. Turan, S. Ucar, S. C. Ergen, and O. Ozkasap, "Dual channel visible light communications for enhanced vehicular connectivity," in *Vehicular Networking Conference (VNC), 2015 IEEE*, Dec, pp. 84–87.

[137] L. Grobe, A. Paraskevopoulos, J. Hilt, D. Schulz, F. Lassak, F. Hartlieb, C. Kottke, V. Jungnickel, and K.-D. Langer, "High-speed visible light communication systems," *Communications Magazine, IEEE*, vol. 51, no. 12, pp. 60–66, December 2013.

[138] P. H. Pathak, X. Feng, P. Hu, and P. Mohapatra, "Visible light communication, networking, and sensing: A survey, potential and challenges," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2047–2077, Fourthquarter 2015.

[139] Ozgur Ergul, Ergin Dinc, and Ozgur B. Akan, "Communicate to illuminate: State-of-the-art and research challenges for visible light communications," *Physical Communication*, vol. 17, pp. 72 – 85, 2015.

[140] Maytee Zambrano Carlos Medina and Kiara Navarro, "Led based visible light communication: Technology, applications and challenges - a survey," *International Journal of Advances in Engineering & Technology (IJAET)*, vol. 8, no. 4, pp. 482–495, August 2015.

[141] "DOT HS 811 492A, vehicle safety communications - applications (VSC-A), final report," Tech. Rep., Crash Avoidance Metrics Partnership, Sept. 2011.

[142] James Bernsen and D. Manivannan, "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification," *Pervasive and Mobile Computing*, vol. 5, no. 1, pp. 1 – 18, 2009.

[143] Alessandro Bazzi and Alberto Zanella, "Position based routing in crowd sensing vehicular networks," *Ad Hoc Networks*, vol. 36, Part 2, pp. 409 – 424, 2016, Vehicular Networking for Mobile Crowd Sensing.

[144] "SHINE web page," Accessed on March 2016.

[145] A. Bazzi, G. Pasolini, and C. Gambetti, "SHINE: Simulation platform for heterogeneous interworking networks," in *Communications, ICC 2006. IEEE International Conference on*, 2006, vol. 12, pp. 5534–5539.

[146] C. Sommer and F. Dressler, "Progressing toward realistic mobility models in vanet simulations," *Communications Magazine, IEEE*, vol. 46, no. 11, pp. 132–137, November 2008.

[147] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *Mobile Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2013.

[148] "SAE J2735: Dedicated short range communications (DSRC) message set dictionary," 2015.

[149] R. Roberts, P. Gopalakrishnan, and S. Rathi, "Visible light positioning: Automotive use case," in *Vehicular Networking Conference (VNC), 2010 IEEE*, Dec 2010, pp. 309–314.

[150] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of IEEE 802.11p radio shadowing in urban environments," in *Wireless On-Demand Network Systems and Services (WONS), 2011 Eighth International Conference on*, Jan 2011, pp. 84–90.

[151] J. Karedal, N. Czink, A. Paier, F. Tufvesson, and A. F. Molisch, “Path loss modeling for vehicle-to-vehicle communications,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 1, pp. 323–328, Jan 2011.

[152] C. F. Mecklenbrauker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernado, T. Zemen, O. Klemp, and N. Czink, “Vehicular channel characterization and its implications for wireless system design and performance,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1189–1212, July 2011.

[153] Lin Cheng, B. E. Henty, Fan Bai, and D. D. Stancil, “Highway and rural propagation channel modeling for vehicle-to-vehicle communications at 5.9 GHz,” in *Antennas and Propagation Society International Symposium, AP-S 2008. IEEE*, July, pp. 1–4.

[154] Lin Cheng, B.E. Henty, D.D. Stancil, Fan Bai, and P. Mudalige, “Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 GHz dedicated short range communication (DSRC) frequency band,” *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1501 –1516, Oct. 2007.

[155] J.M. Kahn and J.R. Barry, “Wireless infrared communications,” *Proceedings of the IEEE*, vol. 85, no. 2, pp. 265–298, Feb 1997.

[156] W. Viriyasitavat, Shun-Hsiang Yu, and Hsin-Mu Tsai, “Short paper: Channel model for visible light communications using off-the-shelf scooter taillight,” in *Vehicular Networking Conference (VNC), 2013 IEEE*, Dec 2013, pp. 170–173.

[157] M. Y. Abualhoul, M. Marouf, O. Shagdar, and F. Nashashibi, “Platooning control using visible light communications: A feasibility study,” in *Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on*, Oct 2013, pp. 1535–1540.

[158] Pengfei Luo, Z. Ghassemlooy, Hoa Le Minh, E. Bentley, A. Burton, and Xuan Tang, “Fundamental analysis of a car to car visible light communication system,” in *Communication Systems, Networks Digital Signal Processing (CSNDSP), 2014 9th International Symposium on*, Jul. 2014, pp. 1011–1016.

[159] M. Rahaim and T.D.C. Little, “SINR analysis and cell zooming with constant illumination for indoor VLC networks,” in *Optical Wireless Communications (IWOW), 2013 2nd International Workshop on*, Oct. 2013, pp. 20–24.

[160] N. Kumar, D. Terra, N. Lourenco, L.N. Alves, and Rui L. Aguiar, “Visible light communication for intelligent transportation in road safety applications,” in *Wireless Communi-*

cations and Mobile Computing Conference (IWCMC), 2011 7th International, Jul. 2011, pp. 1513–1518.

[161] M. Mauve, J. Widmer, and H. Hartenstein, “A survey on position-based routing in mobile ad hoc networks,” *Network, IEEE*, vol. 15, no. 6, pp. 30–39, Nov 2001.

[162] J. Gu, D. Zhang, “A new measurement-based admission control method for IEEE 802.11 wireless local area networks,” in *IEEE Personal, Indoor and Mobile Radio Communications (PIMRC), 2003.*, Beijing, China, September, pp. 2009 – 2013 vol.3, IEEE.

[163] A. Bazzi, “WLAN hot spots to increase UMTS capacity,” in *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, Sept 2010, pp. 2488–2493.

[164] C. Han, M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen, “Analytical study of the IEEE 802.11p MAC sublayer in vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 873–886, June 2012.

[165] H. Hamazumi, K. Imamura, K. Shibuya, and M. Sasaki, “A study of a loop interference canceller for the relay stations in an sfn for digital terrestrial broadcasting,” in *in Proc. of IEEE Global Telecomm. Conf. (Globecom)*, San Francisco, CA, Nov. 2000, vol. 1, pp. 167–171.

[166] W.K. Kim, Y.T. Lee, S.I. Park, H.M. Eum, J.H. Seo, and H.M. Kim, “Equalization digital on-channel repeater in the single frequency networks,” *IEEE Trans. Broadcast.*, vol. 52, no. 2, pp. 137–146, June 2006.

[167] Karim Medhat Nasr, John P. Cosmas, Maurice Bard, and Jeff Gledhill, “Performance of an echo canceller and channel estimator for onchannel repeaters in dvb-t/h networks,” *IEEE Trans. Broadcast.*, vol. 53, no. 3, pp. 609–618, Sept. 2007.

[168] Giovanni Chiurco, Matteo Mazzotti, Flavio Zabini, Davide Dardari, and Oreste Andrisano, “FPGA design and performance evaluation of a pulse-based echo canceller for dvb-t/h,” *IEEE Trans. Broadcast.*, vol. 58, no. 4, pp. 660–668, Dec. 2012.

[169] S.I. Park, Y.-J. Lee, H.M. Kim, and H.-N. Kim, “Equalisation digital on-channel repeater with a feedback interference canceller for the advanced television systems committee terrestrial digital television system,” *Communications, IET*, vol. 7, no. 16, pp. 1769 – 1776, Nov. 2013.

[170] Yong-Tae Lee, Sung Ik Park, Ho min Eum, Jae Hyun Seo, Heung Mook Kim, Seung Won Kim, and Jong Soo Seo, "A design of equalization digital on-channel repeater for single frequency network atsc system," *IEEE Trans. Broadcast.*, vol. 53, no. 1, pp. 23–37, Mar. 2007.

[171] Suk Chan Kim, Dong Chan Park, Sung Ik Park, and Yong Tae Lee, "Loopback signal cancellation scheme in the equalization digital on-channel repeater," in *Consumer Electronics, 2007. ICCE 2007. Digest of Technical Papers. International Conference on*, Las Vegas, NV, Jan. 2007, pp. 1–2.

[172] Sung Ik Park, So Ra Park, Homin Eum, Jae-Young Lee, Yong-Tae Lee, and Heung Mook Kim, "Equalization on-channel repeater for terrestrial digital multimedia broadcasting system," *IEEE Trans. Broadcast.*, vol. 54, no. 4, pp. 752–760, Dec. 2008.

[173] Young Ho Oh, Se Young Kim, and Dae Jin Kim, "An equalization technique to combat strong pre-echoes in sfn with docr," in *Consumer Electronics, 2008. ICCE 2008. Digest of Technical Papers. International Conference on*, Las Vegas, NV, Jan. 2008, pp. 1–2.

[174] Sung Ik Park, Homin Eum, So Ra Park, Geon Kim, Yong Tae Lee, Geon Kim, and Wangrok Oh, "Equalization ocr with feedback interference canceller for t-dmb system," in *Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on*, Bilbao, May 2009, pp. 1–5.

[175] Jin-Yong Choi, Min-Sung Hur, Young-Woo Suh, Jong-Seob Baek, Yong-Tae Lee, and Jong-Soo Seo, "Interference cancellation techniques for digital on-channel repeaters in t-dmb system," *IEEE Trans. Broadcast.*, vol. 57, no. 1, pp. 46–56, Mar. 2011.

[176] Sung Ik Park, Homin Eum, Heung Mook Kim, Young-Jun Lee, Soo In Lee, and H.-N. Kim, "Equalization digital on-channel repeater with a feedback interference canceller for the atsc terrestrial dtv system," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium on*, Nuremberg, June 2011, pp. 1–4.

[177] Xiuyan Feng, Lu Si, and Zhibin Zeng, "Research and improvement of feedback cancellation for cmmb on-channel repeater," in *Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on*, Yichang, Apr. 2012, pp. 3406 – 3410.

[178] Sungho Jeon, Zungkon Yim, Hyeun-Pil Jin, Jeong-Min Choi, and Jong-Soo Seo, "Performance of maximal ratio combining digital on-channel repeater for terrestrial dmb/dab," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2013 IEEE International Symposium on*, London, June 2013, pp. 1–5.

[179] F. Zabini, M. Mazzotti, D. Dardari, and O. Andrisano, “An upper bound on the probability of instability of a dvb-t/h repeater with a digital echo canceller,” in *in Proc. of IEEE Global Telecomm. Conf. (Globecom)*, Miami, FL, Dec. 2010, pp. 1–5.

[180] Flavio Zabini, Matteo Mazzotti, Giovanni Chiurco, Davide Dardari, and Oreste Andrisano, “Performance and stability analysis of echo cancellers based on training sequences,” *IEEE Trans. Broadcast.*, vol. 60, no. 3, pp. 437–451, Sept. 2012.

[181] R. Schober and W.H. Gerstacker, “The zeros of random polynomials: further results and applications,” *IEEE Transactions on Communications*, vol. 50, no. 6, pp. 892–896, Jun. 2002.

[182] G. Pasolini and R. Soloperto, “Multistage decimators with minimum group delay,” in *IEEE International Conference on Communications (ICC)*, May 2010, pp. 1–6.

Ringraziamenti

"Niente nella vita va temuto, deve essere solamente compreso. Ora è tempo di comprendere di più, così possiamo temere di meno." [Marie Curie]

Sicuramente vi starete chiedendo perchè io sia voluto entrare nel mondo della ricerca... Beh, a questa domanda vorrei rispondere con le parole di Sir Isaac Newton, il quale disse "non so come il mondo potrà giudicarmi ma a me sembra soltanto di essere un bambino che gioca sulla spiaggia, e di essermi divertito a trovare ogni tanto un sasso o una conchiglia più bella del solito, mentre l'oceano della verità giaceva inesplorato davanti a me". Perchè tutto va compreso, e con il giusto livello di curiosità.

"La curiosità non è peccato, Harry, ma dovresti esercitare cautela." [Harry Potter]

Un ignaro lettore potrebbe pensare che, dopo aver scritto una tesi, occuparsi dei ringraziamenti sia un gioco da ragazzi. Ebbene, non è così. Le persone che prenderanno in mano questo tomo infatti, quasi sicuramente finiranno per leggere queste righe. Quindi, ora che ne ho la possibilità, lasciatemi scrivere liberamente; spero di annoiarvi, cosi da non farvi arrivare fino a questo punto!

Nella scrittura dei ringraziamenti non può mancare una persona speciale e misteriosa alla quale dedicare il ringraziamento più importante. Quella persona che entra nella tua vita in punta di piedi per poi finire, senza accorgersene, ad essere il pilastro fondamentale del tuo cuore. La ringrazio per il continuo supporto, la sua semplicità e la leggerezza con la quale mi ha fatto stare bene in ogni momento, la voglia di vivere e condividere che mi ha donato e per il suo sorriso che rappresenta la felicità.

Ringrazio tutta la mia famiglia e, in particolar modo i miei genitori. A loro va tutta la mia stima, il mio rispetto e la mia riconoscenza. Mi auguro che tutti i loro sacrifici siano in questo modo, almeno in parte, ripagati. Un ringraziamento speciale va a mia sorella ed al mio nipotino

Lukas che 10 anni fa mi ha fatto riscoprire quanto la vita sia preziosa attraverso lo stupore dei suoi occhi curiosi.

Ringrazio altresì tutto il gruppo con la quale ho lavorato (WiLab) coordinato dal Prof. Ing. Oreste Andrisano il quale ringrazio particolarmente. Ringrazio Barbara, Cristina, Lorenza, Alberto, Alessandro, Davide, Flavio, Gianni, Marco e Mirko, perchè saremo tutti ingegneri ma quando un ringraziamento arriva dal cuore non ha bisogno di titoli; li ringrazio per la disponibilità, gli insegnamenti ed il supporto dimostratomi durante questi anni di lavoro. In particolar modo ringrazio il mio supervisore Ing. Gianni Pasolini ed i correlatori Ing. Barbara Mavì Masini e Prof. Ing. Davide Dardari per tutto ciò che hanno fatto per me in questi 3 anni.

Ringrazio tutti i giovani ricercatori che hanno iniziato o condiviso questa esperienza con me, i professori e ricercatori di ingegneria di Bologna e Cesena e tutti i colleghi incontrati durante questa esperienza. Tra di essi, vorrei ringraziare in particolar modo Antonella Molinaro con i suoi ottimi consigli in ambito veicolare ed i professori Michele Rossi e Gaudenzio Meneghesso per la bellissima parentesi ed esperienza in Bressanone.

Ringrazio i ricercatori conosciuti durante la mia esperienza alla Centrale Supelec a Parigi, dal mio corrispondente Michel Kieffer ai giovani Ph.D, senza tralasciare Wenjie Li, Francesca Bassi, Marco Di Renzo e Giuseppe Valenzise.

Un grazie a tutti i miei amici che sono sempre stati presenti e mi hanno sopportato e apprezzato per come sono, con i miei pregi e soprattutto i miei difetti; condividendo le mie passioni, rispettando i miei silenzi e sopportando le mie battute. Degli Amici sinceri e leali che mi hanno fatto riscoprire la bellezza di avere delle persone su cui poter contare.