Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Ingegneria elettronica, telecomunicazioni e tecnologie dell'informazione

Ciclo XXX

Settore Concorsuale: 09/F2

Settore Scientifico Disciplinare: ING-INF/03

INTERNET OF THINGS AND HUMANS

Presentata da: Colian Giannini

Coordinatore Dottorato

Alessandro Vanelli Coralli

Supervisore

Roberto Verdone

Co-Supervisore

Chiara Buratti

Esame finale anno 2018

To my grandma Ada,
my sister Greta,
my mother Iva,
my aunt Pia,
and Gio.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We live in an era where the never ending demand for capacity and the need for ubiquitous radio coverage requires particular attention in the way we design radio networks. With the incoming paradigms, like industry 4.0, machine to machine communication and (above all) Internet of Things, cellular networks will be overburden even more. It is widely acknowledged that conventional current (4G) and near-future (5G) macrocell architecture will not be able to support such traffic increase, even after the allocation of additional spectrum. To give an example, forecasts from CISCO [1] show that streaming of video on-demand files is the main reason for the dramatic growth of data traffic over cellular networks to such an extent that an increase of two orders of magnitude compared to current volume is expected in the next 5 years.

Moreover, space-time and content heterogeneity of the data traffic caused by the pervasive spread of smartphones and of content providers should be exploited to improve network performance. However, current networks performance are deteriorated by this heterogeneity. Indeed, with the current networks, peaks of traffic (e.g. concert, football match, disaster scenario) are seen as a problem and to deal with it the solution is simply to put some kind of additional cells (macro or micro) increasing the cell density with the consequent increase in capacity per square meter. Unfortunately, this can be done only when it is known where, when and how big the peak of traffic is, in other words when the peak of traffic is predictable. However, since this is not always the case, we need the network to be flexible in order to adapt to sudden changes in the traffic demand.

In order to tackle these problems, one of the most promising approach to deliver the system area spectral efficiency consists of shrinking the cell size and essentially bringing the content closer to the users. To do so, it is needed a deployment of small base stations (BSs) able to achieve through local communication an high-density spatial reuse of radio resources. Such pico- and femto-cell networks, which are usually combined with macrocells into a heterogeneous network, have been receiving a lot of attention in the recent literature, (e.g., see [2] and references therein). A drawback of this approach, though, is the urgency of high-speed backhaul to connect the small cells between each other and to the core network [2]. In particular, current research trends consider scenarios where the density of small cell access points will be comparable to the user density [3],[4]. In such a situation, deploying high-speed fiber backhaul will be too expensive.

Moreover, regardless whatever deployment of macro, micro and femto cells, we live in an era in which connectivity is a commodity given as always granted. A lot of technologies and services heavily rely on stable networks or at the very least on an internet connection. Nonetheless, whenever also a basic connectivity fails because of a disaster (e.g. floods, earthquakes, hurricanes) or because of a terrorist attack or in war zone, not even a basic form of radio communication can be provided. This latter problem could be reduced if it was possible to have a flexible network adapting to the environment "on the go".

The problems underlined till now, can be summarized with two keywords: *lack of capacity* and *lack of flexibility*. In this dissertation, I will describe several works I did during my Ph.D. aiming to solve or at least alleviate the aforementioned problem. My work unfolds starting from a couple of intuitions. On one hand, traffic demand is not just a stack of data to be processed, transmitted and answered to, but the kind of data producing the traffic demand matters. As an example, what we call traffic demand is composed of data with very heterogenous characteristics and requirements: video streaming traffic needs to be served within few seconds and to do so we can exploit the known popularity of the content of the video itself by pre-caching it close to the end user, while environmental data collected by sensors for monitoring purposes have relaxed constraints in terms of delay that allow to use delay tolerant approaches. Thus, it would be smart to find a way to treat different kinds of traffic in the proper way. This facet of my work is treated under different points of view in chapter I and in chapter IV either.

On the other hand, in current networks, the users are seen as "passive" users, they have no role in the network except for being source and/or destination of a traffic stream. Actually, there are several reasons to envision that users (people, cars, buses) could be exploited as "active" users participating to the network itself fostering its performance. For instance, it turned out that people moving around carrying a smartphone in their pocket, do not move randomly, but they follow mobility patterns studied in the recent past that are to some extent predictable. It would be useful to use people as "data mule" to physically carry data around with them toward a given destination or toward another user that might be a better candidate to forward the data to the final destination. Obviously this case scenario requires some time for the data to be carried to its destination, however as aforementioned, several types of data traffic exist. This kind of considerations are accounted in the so called Delay Tolerant Networks (DTN), to which chapter I of this dissertation is devoted.

In figure 1.1 it is shown a concept map to describe the topics treated in this dissertation.

In particular, the Targets to achieve are *Capacity* and *Flexibility* while the tool used to achieve them are the exploitation of the *Kind of Traffic* and of *Active Users*. The topics I will treat belong to two domains, namely Cellular Networks and Delay Tolerant Networks.

This dissertation is divided in four main parts related to: Delay Tolerant Networks (part I), Unmanned Aerial Base Station (UAB) aided Networks (part II), Device to Device Communications(D2D) (part III) and Femto-caching (part IV).

In particular, The part related to the DTN addresses an experimental character-

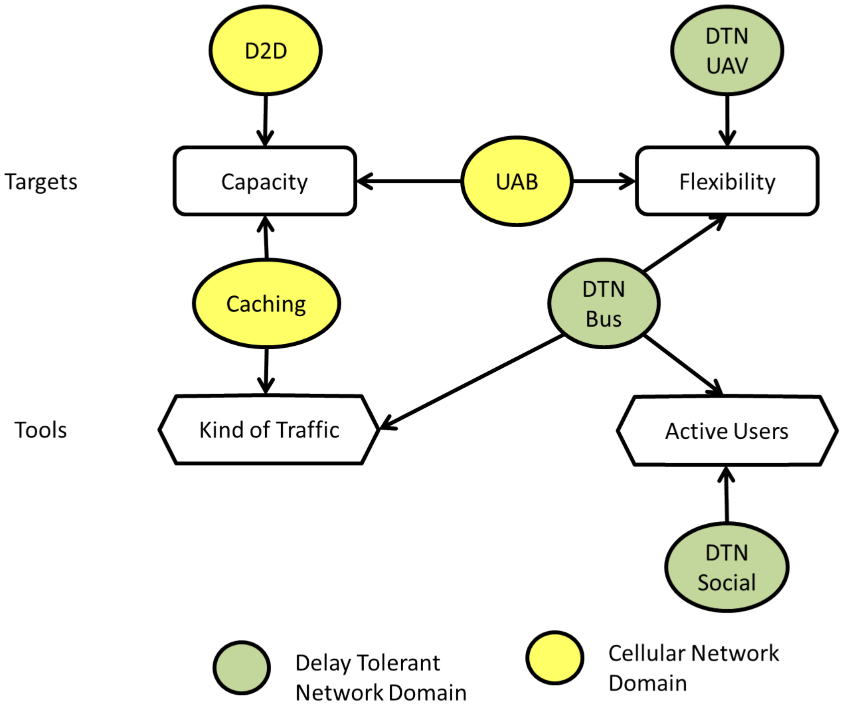

Figure 1.1: Concept Map

ization of human mobility and sociality, the design of a communication protocol exploiting bus mobility, and the study of a DTN operated by an Unmanned Aerial Vehicle.
In the second part we build a theoretical framework based on optimization theory to face the problem of a UAB aided network. More precisely, the aim is to optimize the route of an UAB in order to serve cellular users that could not be satisfied by the Base Station because of the channel condition or because of traffic congestion.
The third part, related to D2D communications, shows an Integer Linear Program for mode selection together with an analysis of the impact of interference both in uplink and downlink.
Finally the last part faces the complex problem of Femto-caching by a summary of the most important literature on the topic. Then a new dynamic and distributed policy is presented with the relative performance analysis. For this last part only preliminary results will be presented.

This dissertation is based on works published in some conferences, proceedings or submitted to journals:

- Paper [5] was presented in a Special Session of the conference EuCNC 2015 in Paris and it is present in the proceedings of the conference.

- Paper [6] was presented at the International Symposium on Wireless Communication Systems (ISWCS) 2015 in Brussels

- Paper [7] was presented at the International Symposium on Wireless Communication Systems (ISWCS) 2016 in Poznan
- Paper [8] was presented at the AEIT International Annual Conference 2016 in Capri
- Paper [9] was presented at the workshop Soft5G in the contest of the International Teletraffic Congress (ITC 29) in Genova
- A submission of paper [10] is forthcoming and authors believe it worth a submission to journal.

Part I

Delay Tolerant Networks

Chapter 2

Introduction

This part of the dissertation regards papers [5], [6], [7] and [8]. Delay Tolerant Networks, Opportunistic Networks or Pocket Switched Networks are three commonly used names to describe the same networking paradigm. These terms, describe networks whose goal is to enable communications in disconnected environments, where the absence of end-to-end paths between sender and receiver impairs the communication [11]. In this networks data are delivered, from a source to a destination, based on pair-wise contact opportunities, exploiting a multi-hop based communication, where intermediate nodes act as relays in a "store-carry and forward" fashion. Mobility of nodes is the key enabler.

From this definition the aspects of delay and opportunism inherent to this kind of networks are revealed. The term Pocket Switched Network is used when the focus have to be put on the fact that the nodes of the considered network are mobile phones carried around by people that usually keep the mobile phone in their pocket. Other important facts underlined by this latter term are the importance of human mobility characterization and the absence of dedicated infrastructure. In the following, these networks are referred to as Delay Tolerant Networks (DTN).

A couple of examples to motivate the use DTN are reported hereafter. Both the examples consider situations where there are data to be transmitted with relaxed constraints in terms of delay. The first notable example is given by the Songdo International Business District, a smart city (South Korea) built from scratch on 600 hectares, where over 20.000 residential units are occupied or under construction. In Songdo everything (streets, building, electric system, water pipes...) is provided with sensors that monitor all the aspects of the daily life 24 hours per day. In 2020 it is foreseen there will be 65.000 people living in Songdo, while another 300.000 people will commute there daily. As a result a massive amount of data related to traffic, garbage, climate, energy consumption, water consumption, will be generated. All these data will be observed and analyzed by a central monitoring hub. A possible solution to help the network sustaining such a big data traffic and to reduce infrastructure and telecommunication related costs, is provided by Delay Tolerant Networks (DTN).

Similarly, even in more humble scenarios, for smart building application, sensors placed in buildings have to be read for monitoring purposes. The gathered data must be sent to utility providers in order to react by offering new services to users or by reducing resources consumption. For instance, in Italy the authority imposes the use of Wireless Metering Bus (W-MBUS) at 169 MHz, as standard for the communication between meters and data concentrators. Unfortunately, it is foreseen that this standard will not provide enough throughput as the data traffic increases. Therefore, a different way to deliver the sensed data to the central unit should be considered.

To give a look outside the research world, a couple of real applications of DTN-like networks are present in the market. The first, is for wild life or sea life tracking, where animals are provided with devices recording contacts between each other to understand interactions between animals. An example of that is ZebraNet.

Another application is FireChat by OpenGarden. OpenGarden is a company proposing peer-to-peer mesh networking solutions, which released the mobile application FireChat in 2014. Firechat uses the concept of wireless mesh networking to enable smartphones to connect with each other via Bluetooth or Wi-Fi, without the need of an Internet connection, in order to exchange messages. In particular, this application became famous during the Hong Kong protest in 2014 when the government precluded the use of internet to censor protesters. In that occasion, FireChat experienced more than 500.000 downloads in two weeks along with 10.2 million chat session and 1.6 million chatrooms. Other scenarios where FireChat was successfully exploited are:

- Natural disasters: flood in Kashmir (April 2015) and Chenai (October 2015), volcano eruption in Ecuador (August 2015), hurricane in Mexico (October 2015);
- Massive events: pro-democracy protests in Taiwan (April 2014) and Hong Kong (September 2014), visit of the pope in the Philippines (January 2015);
- Historical elections: Venezuela (December 2015) and the Republic of Congo (March 2016);
- Large festivals in India, Canada and the US.

Even if this fact checking seems outside a Ph.D. dissertation, looking at the situations and at the average life condition of people in some of the mentioned places, it can be perceived how the use of this technology can be part of important positive changes having an effective impact on people life, and this should always be the aim of technology.

In this chapter I will describe my work in the field of DTN. It mainly faces two aspects: People mobility-sociality characterization through experimentation and DTN protocol design using a simulative approach.

Chapter 3

Epidemic Information Dissemination in Opportunistic Scenarios: a Realistic Model Obtained from Experimental Traces

Opportunistic networks consist of nodes moving around and occasionally coming into each other proximity. During the limited proximity time nodes can exchange data. This can result in a data dissemination process which is usually governed by a replication based mechanism similar to the epidemic information spreading. Epidemic models have been proposed in the literature [12]–[13] in order to predict performance in data dissemination in opportunistic scenarios. However these models usually rely on the use of Ordinary Differential Equations (ODEs) obtained as limits of the Markov models to illustrate nodes' interactions and inter-contacts. In order to be treatable, inter-contact time distributions among nodes' are usually assumed to be exponential and modeled by using Markov chains. However, it has been shown in the literature [14]–[15] that nodes' inter-contacts are power-law distributed with exponential tails. Accordingly, in this paper we propose a theoretical framework to model epidemic information dissemination while coping with realistic inter-contact time distributions.

Numerical results are obtained by considering realistic inter-contact traces collected at the last EUCNC 2014 event (a scientific conference organised in Bologna gathering 550 participants) and integrating them in the analytic framework. Comparison between realistic traces used for identifying the inter-contact rate among nodes and theoretical models assess the validity of the framework in terms of realistic description.

3.1 Inter-Contact Time Distribution and Markov Model

It has been deeply discussed in the literature that inter-contact time distributions among nodes exhibit a dichotomy and are power-law distributed with an exponential decay [14]–[15]. Accordingly in this paper, taking inspiration from the work of [16], we create an equivalent Markov chain to model the inter-contact time distribution between pairs of users when taking into account real traces. More specifically we consider N different exponential distributions which fit the power law intercontact time distribution for different time scales. The value of the inter-contact rate λ_i is quantized into finite discrete levels. Transitions between levels are assumed to occur with exponential transition rates which depend on the current state. Accordingly we are approximating the inter-contact rate as a continuous time Markov process. The state space consists of the set of quantized λ_i levels up to the maximum of N. The number of states and the transition rates are tuned to fit the real measured data. Based on the real measured data, there are infinite possible choices of the Markov model which can fit the model. The choice of the specific model has been done based on the will to preserve both simplicity in the model as well as generality. Accordingly, we have considered a complete Markov model, as the one shown in Figure 3.1 where transitions are possible between any pair of states i and j, provided that the transition rate λ_{ij} from state i to j is such that $\lambda_{ij} = \lambda_i \; \forall j \neq i$.

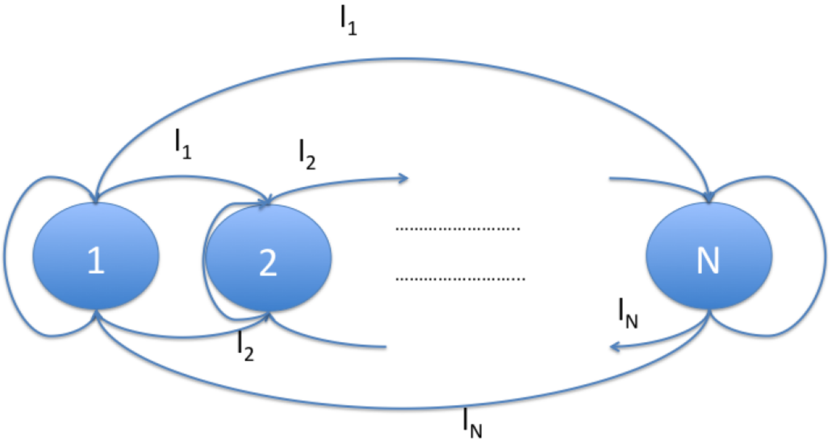

Figure 3.1: Markov model of the inter-contact rate process.

By solving the standard System used to model continuous Markov Chains

$$\begin{cases} \mathbf{\Pi} \cdot \mathbf{Q} = 0 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases} \tag{3.1}$$

where $\mathbf{Q}$ is the matrix of the transition rates and $\mathbf{\Pi}$ is the array of the stationary state probabilities π_i, it is possible to completely characterize the Markov chain of the inter-contact rates.

3.2 Epidemic Information Dissemination Model

We consider an opportunistic information dissemination model where users move around and occasionally come into each other proximity. As nodes come into contact, they exchange data and, using this epidemic dissemination approach, the information is delivered at destinations. Without loss of generality, we assume that when two nodes meet, the transmission opportunity allows to transfer only one data packet per flow. In our model we have N_N network nodes and a source node. We assume that a multicast dissemination is ongoing where D destination nodes have to receive the data packet. In order to avoid network overloading, we assume that a recovery is performed where a node, once delivered the packet to the destination, can delete the copy from its buffer to save storage space and prevent further infection. At the same time, the node saves information on the packet just delivered and consequently does not relay again the same packet in case it receives it again.

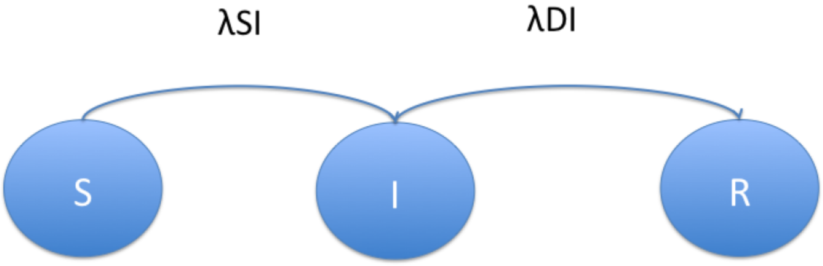

Figure 3.2: Markov model of the infection process.

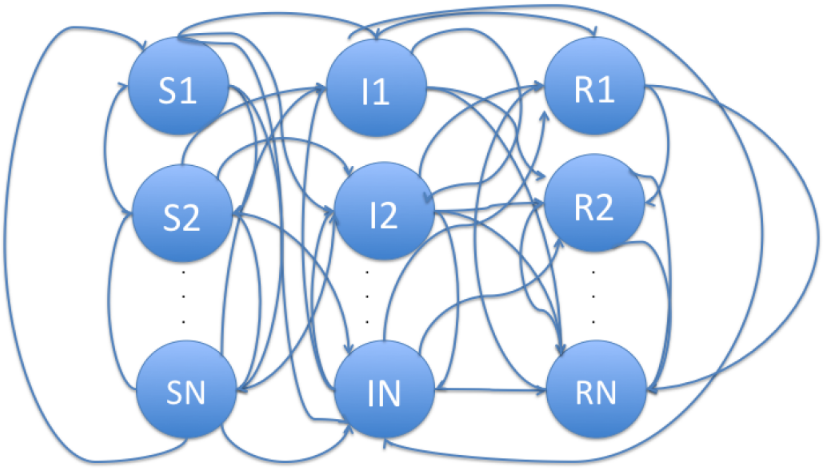

Figure 3.3: Modified Markov model of the infection process.

Similarly to what was proposed in [17], we denote as $I(t)$ the number of nodes which possess a data packet at time t; moreover, we denote as $S(t)$ the number of susceptible nodes at time t, i.e. those nodes that are able to accept a packet since they did not yet receive it. Moreover we denote as $R(t)$ the number of nodes that have recovered at time t. Similarly to [17], a Markov chain model of an infectious process with susceptible, infected, and recovered states can be provided as shown in Figure 3.2.
The model is embedded with the underlying Markov chain of the inter-contact rate process discussed above.Accordingly the Markov model of the infection process reduces to the one shown in Figure 3.3. By considering the transitions from state $S_i(t)i \in \{1, 2, ..., N\}$ to state $I_j(t)j \in \{1, 2, ..., N\}$ and $R_k(t)k \in \{1, 2, ..., N\}$ it is possible to derive a system of ODEs as a fluid limit of the above Markov model. Hence it can be written

$$\begin{cases} \dfrac{\mathrm{d}S_i}{\mathrm{d}t} = -\lambda_{ii}S_i(t)I_i(t) - \sum\limits_{j\neq i} S_i(t)I_j(t)\lambda_{ij} + \sum\limits_{j\neq i} S_j(t)\lambda_{ji} - \sum\limits_{j\neq i} S_i(t)\lambda_{ij} \\ \dfrac{\mathrm{d}I_i}{\mathrm{d}t} = \lambda_{ii}S_i(t)I_i(t) - \lambda_{ii}I_i(t)D + \sum\limits_{j\neq i} S_j(t)I_i(t)\lambda_{ji} - \sum\limits_{j\neq i} I_j(t)\lambda_{ji} - \sum\limits_{j\neq i} I_i(t)\lambda_{ij} \\ \dfrac{\mathrm{d}R_i}{\mathrm{d}t} = \lambda_{ii}I_i(t)D + \sum\limits_{j\neq i} DI_j(t)\lambda_{ji} \end{cases} \tag{3.2}$$

where λ_{ij} are the elements of the matrix $\mathbf{Q}$ for the inter-contact rate process and $I_i(t)$, $R_i(t)$ and $S_i(t)$ are the number of infected nodes, recovered ones and susceptible nodes in state i, respectively. In the following we denote as $I(t)$ the overall number of infected nodes, i.e. $I(t) = \sum_i I_i(t)$ and as $R(t)$ the overall number of recovered nodes, i.e. $R(t) = \sum_i R_i(t)$. In the above system of ODEs, the variation in the number of infected nodes is positively impacted by the rate at which susceptible nodes in the various states of the underlying Markov chain transit to the state $I_i(t)$, and negatively impacted by recovery of nodes as soon as they meet the destinations D or from transitions out of the state $I_i(t)$. Similarly, for the number of recovered nodes, the variation is positively impacted by nodes in states $I_h(t)$ with $h \in \{1, 2, ..., N\}$ which transit to state $R_i(t)$ and negatively impacted by nodes which transit from state $R_i(t)$ to the other recovered states. Concerning the number of susceptible nodes in state i, $S_i(t)$, this value increases when transitions are executed from other susceptible states to i and decreases when transitions are executed to state $I_j(t)$ with $j \in \{1, 2, ..., N\}$. By solving this ODEs system it is possible to study the variation in the number of recovered and infected nodes in the network.

3.3 Numerical Evaluation

In this section we will investigate how the information dissemination procedure works when considering the infection process enhanced with the underlying Markov chain of the inter-contact rate process obtained by real traces. We will also compare the results obtained by using the theoretical framework upon assuming that intercontact rate among nodes is exponentially distributed. To this purpose this section includes details for the characterization of the inter-contact time in realistic scenarios and the results of the numerical evaluation.

Realistic Inter-Contact Time Characterization
In order to realistically characterize the nodes inter-contact rate process we carried out an experiment. The aim of the experiment was to characterize the behavior of the attendees to a conference. The experiment was carried out during the last EuCNC 2014 conference held in Bologna. To collect the data, DataSens was exploited. This is one of the facilities offered by EuWIn[1]; more specifically it is a platform consisting of 100 IEEE 802.15.4 compliant devices. The devices are TI CC2530 with a receiver sensitivity of -102 dBm. During the conference, a number of nodes between 35 and 45 were given to attendees, who carried them in their pocket/bag from 9 a.m. until 2 p.m.. The experiment was repeated for three consecutive days. All devices transmitted a beacon every minute, with a transmit power set to 0 dBm. Each device, upon receiving a beacon from another node, records the following data:

- *Identifier* (ID) of the transmitting node.
- *Timestamp*, that is the instant when the beacon is received; the resolution of the timestamp was one minute and the time is calculated from when the device is switched on.
- *Received Signal Strength Indication* (RSSI): the power with which the beacon is received.

In order to elaborate the data and obtain the model of the inter-contact rate, a post processing phase was needed. The aim of the post processing operations is twofold: i) to find the distribution of the Inter Contact Time (ICT) and ii) to find an approximation of this distribution as a weighted sum, with weights π_i, of N (negative) exponential distributions, with parameter λ_i, (WSE) :

$$WSE_n(x) = \sum_{i=1}^{n} \pi_i \, \lambda_i \, e^{-\lambda_i \, x} \tag{3.3}$$

subject to the condition:

$$WSE_n(x) = \sum_{i=1}^{n} \pi_i = 1. \tag{3.4}$$

The first step of the post processing phase is to impose a threshold on the RSSI to avoid to consider encounters between people far from each other. To this aim we set the RSSI threshold for each node to the average RSSI evaluated from its log. Only encounters registered with an RSSI value higher than the RSSI threshold are kept. This per-node decision is motivated by the fact that every node works in different conditions (e.g., the node could be in the pocket or in the bag) and so it requires an adhoc setting decision rule on the RSSI threshold. In order to evaluate the inter-contact rate distribution we considered the three conference days and we observed that in all the cases the distribution

[1] European Laboratory of Wireless Communications for the Future Internet built in the context of the FP7 Network of Excellence Newcom# - See www.euwin.org

function could be approximated by a Generalized Pareto Distribution (GPD), i.e.:

$$GPD(x|k,\sigma,\theta) = \left(\frac{1}{\sigma}\right)\left(1+\frac{k\,(x-\theta)}{\sigma}\right)^{-1-\frac{1}{k}} \tag{3.5}$$
$$for\ k > 0\ and\ \theta < x$$

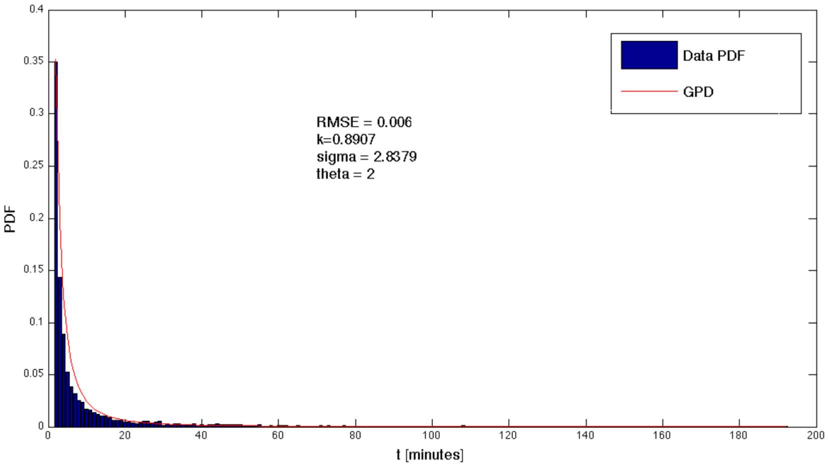

Figure 3.4: ICT distribution function.

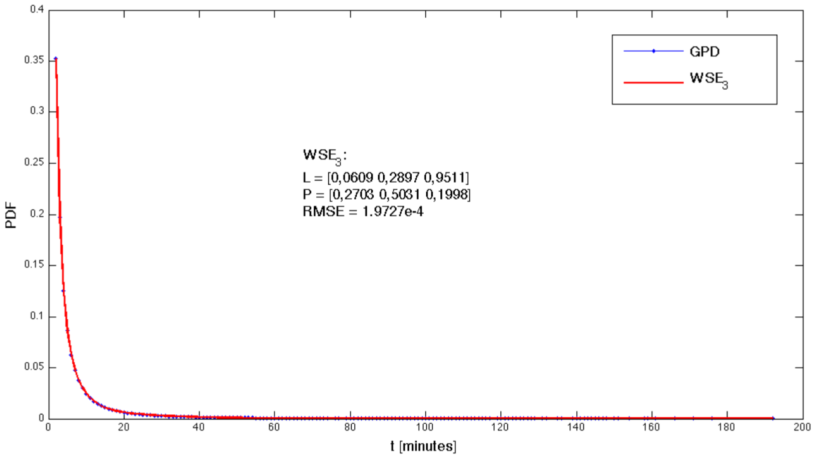

Figure 3.5: Comparison between GPD and WSE_3.

Accordingly, we found an approximation of $GPD(x|k,\sigma,\theta)$, in the form reported in Eqs. 3.3 and 3.4 such that the Root Mean Squared Error (RMSE) is in the order of 10^{-2}. To this purpose we derived the parameters π_i and λ_i for a function of type $WSE3$ since we experimentally observed that 3 exponential distributions are sufficient to model the ICT process[2]. In Figure 3.4 we show the

[2] Observe that the choice to use k exponential distributions is driven by a tradeoff between complexity abd fitting accuracy

distribution function for the inter contact time when considering experimental measurements and in Figure 3.5 we compare the fitting obtained and show the array of the values $L = [\lambda_1 \lambda_2 \lambda_3]$ and $\pi = [\pi_1 \pi_2 \pi_3]$.

Numerical Results

Based on the use of the above fitting of real traces, we were able to obtain

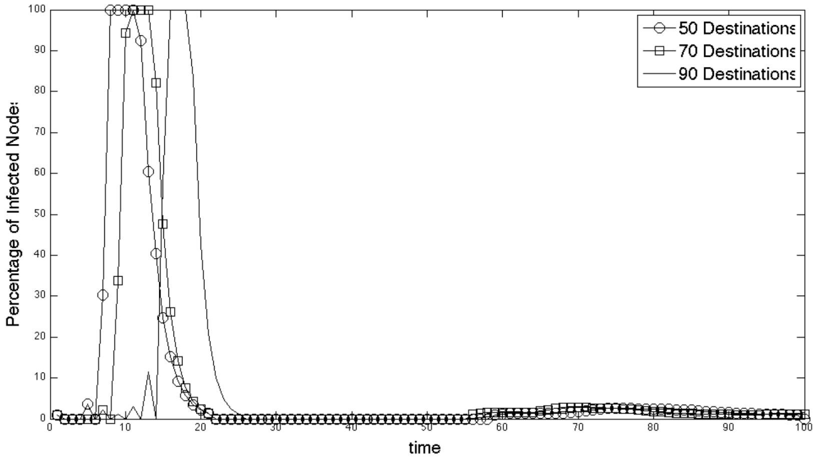

Figure 3.6: Percentage of infected nodes as a function of time with $N_N = 100$ and D destinations.

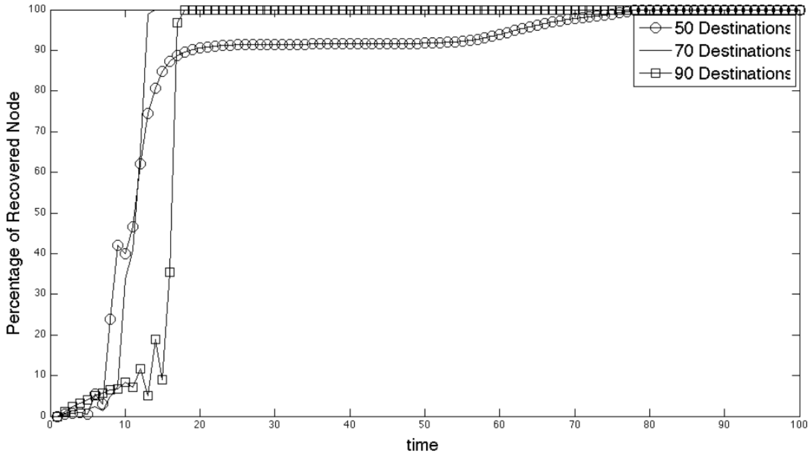

Figure 3.7: Percentage of recovered nodes as a function of time with $N_N = 100$ and D destinations.

the matrix $\mathbf{Q}$ and the array Π and use them in the model described in eqs.3.2. Accordingly in Figures 3.6 and 3.7 we show the evolution of the percentage of infected and the recovered nodes for different values of the number of destinations. Observe that upon increasing the number of destinations to be reached, this results as expected in a slight delay in the time needed to infect 100% of the nodes. However, the time delay is quite limited although increasing the number of destinations of almost 80% (from the case of 50 Destinations to 90 Destinations). Also we note that after 100% of nodes are infected, then a parallel increase in the number of recovered nodes is met till all nodes be ome recovered and the infection stops. It is obvious that a higher number of destinations implies a faster recovery process.

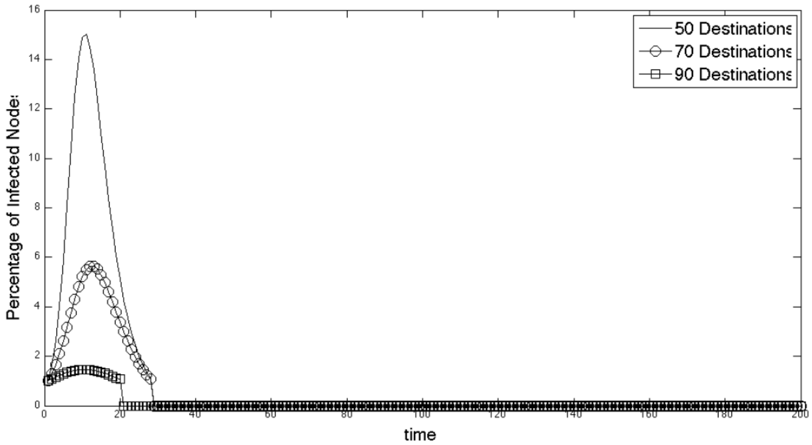

Figure 3.8: Number of infected nodes with the assumption of exponential ICT distribution with $N_N = 100$ and D destinations.

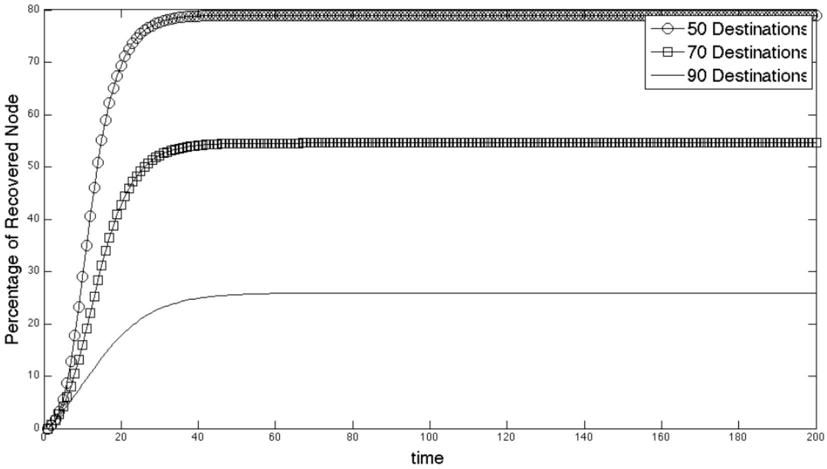

Figure 3.9: Number of recovered nodes with the assumption of exponential ICT distribution with $N_N = 100$ and D destinations.

In Figures 3.8 and 3.9 we report the curves obtained for the number of infected nodes and the recovered nodes in case an exponential distribution of inter-contact times is considered with an average rate $\lambda = 0.008$. Observe that, the use of an exponential distribution implies a high degree of approximation in the performance of the dissemination process since less recovered nodes are obtained using the purely exponential modeling with a consequence that the dissemination process is in this case less aggressive compared to the realistic case. Indeed, in the realistic scenario, the dissemination procedure results too aggressive and almost 100% of nodes are recovered after less than 15 time units. This implies that the recovery is too prompt and nodes are recovered before they can spread the information. So the exponential modeling results too optimistic as compared to the realistic case. This was expected by considering indeed the specific features of the power law distribution.

3.4 Conclusions

In this work we have provided an analytical model of epidemic information dissemination in opportunistic networks which consists of an underlying Markov chain obtained from realistic inter-contact time distributions. To this purpose we have considered a fluid Markov model which relies on an underlying Markov chain which models realistic inter-contact times among nodes by including the well known power-law with exponential tail behavior. Numerical results assess the effectiveness and realistic features which are caught by the model.

Chapter 4

Delay Tolerant Networking for Smart Cities

4.1 Delay Tolerant Networking for Smart Cities: Exploiting Bus Mobility

The aim of the work described in this chapter is to design and validate through simulations a routing protocol based on the DTN paradigm, called Sink and Delay Aware Bus (S&DA-Bus). The peculiarities and novelties of S&DA-Bus are:

- Exploiting the ubiquity of the public transportation typical of a urban scenario in order to deliver data generated by buildings to the final central unit (sink);
- Defining a centrality metric that takes into account the "social role" of the sink (Sink Aware);
- Considering the Inter-Contact-Time between buses and sink to estimate the time that will elapse before the next bus-sink contact (Delay Aware).

The rest of this section is organized as follows: in Section 4.1.1 related works are presented; Section 4.1.4 focuses on the description of the reference scenario; Section 4.1.3 describes the proposed S&DA-Bus protocol; parameters setting and simulation specifics are given in Section 4.1.4, in the same section are reported the numerical results. Closing remarks are provided in the last section.

4.1.1 Related Work

Several works have been done in the past regarding routing protocols for DTN. First Contact [18] routing is a simple routing protocol where a node having a packet forwards it to the first node in the radio range. Once a node has forwarded a packet it can not receive the same packet anymore to prevent loops. Epidemic Routing [19] is similar to First Contact but after a transmission a copy of the packet is kept in the message buffer, therefore, multiple copy of the same packet are generated, such as every possible route from source to destination is explored.

Spray and Wait Routing protocol [20] tries to exploit the advantages of Epidemic routing, but limiting the network load. The source is allowed to generate at most L replicas of the same packet, then, the packet spreading process is split into two phases: Spray and Wait. During the Spray phase nodes spray and forward packets according to a well defined logic. During the Wait phase nodes stand by for direct delivery to the final destination.

The routing protocols proposed in [18, 19, 20] share the absence of a centrality metric used for the selection of "the best next relay".

Other more complex routing protocols like PROPHET [21] or MaxProp [22], try to predict the best route from source to destination based on contact history and on centrality metrics derived from it. Even if these protocols can be applied to our scenario they do not exploit characteristics that are peculiar of the scenario and the application under study in this work.

An example of DTN applied to a smart city is given in [23], where the network is used to spread information about the state of trash bins to inform the service personnel. In particular, in [23] trash bins are equipped with wireless sensors that transmit information about their state to garbage trucks, which are able to connect to the Internet. The information obtained through Internet by all the trucks are used to optimize the routes, redirecting garbage trucks in the city to minimize costs. Another example is provided in [24], where the public transportation in the city of Aachen is studied. The obtained results are used to simulate an Epidemic-like routing protocol.

To the best of our knowledge, the work in [25] focuses on a scenario very similar to that of this paper, but it considers the transport system of a city as an autonomous system where information are generated by and delivered to nodes belonging to the system itself. In our case the transport system is a sort of mobile backbone integrated in the smart city.

On the other hand, this work is inspired by [26], considering an application similar to that of this work, but a different scenario. Moreover, in [26] the mobility is created according to the Heterogeneous Human Walk mobility model which "*directly constructs synthetic overlapping communities rather than detecting them from input or generated social graphs*", enhancing the performances of the proposed protocol (OBSEA) that is designed to exploit the existing communities. In addition, [26] mostly focuses on social behaviour of people and on the problem of "selfish nodes" that cooperate in the DTN trying to optimize just their own profit (e.g., transmitting just specific packets to save battery), thus not following the forwarding logic imposed by the communication strategy.

4.1.2 Reference Scenario

The reference scenario consists of a smart city, where buildings generate data to be transmitted to a sink, called hereafter District Concentrator (DC). Data sensed inside a smart building are collected by several sensor networks deployed in every unit (e.g., flats, offices) of the building and they are gathered by the Building Concentrator (BC), which is in charge of the transmission toward the DC. Data generated by BCs are useful mostly for monitoring of environmental variables, like temperature or energy consumption, therefore they are expected to be small in size (some bytes) and to have relaxed requirements in terms of latency. These characteristics of the data to transmit are such to justify the use

of a DTN.
Figure 4.1 shows our reference scenario: people, vehicles and buses are nodes of the network and they act as relays. This is done in order to exploit the urban mobility such nodes have by nature. For example, it could be useful to exploit as a relay a person that works in an office close to the DC, or a bus that collects data from all passengers in order to deliver them to the DC when it passes by. Special focus of the S&DA-Bus protocol is on the city buses, because of their predictable movements based on a specific travel schedule. Further details on the mobility models used are given in Section 4.1.4.

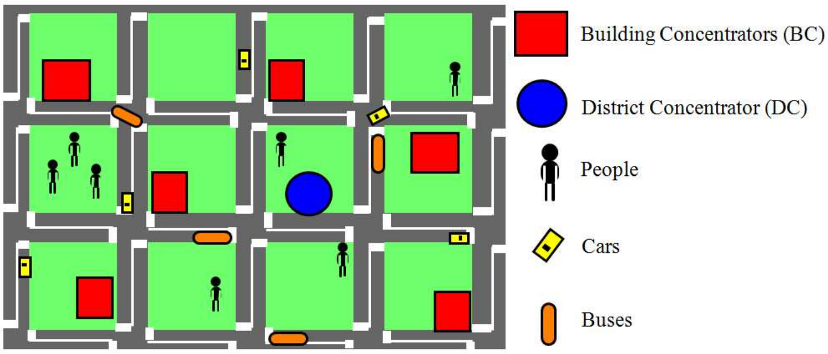

Figure 4.1: Reference Scenario.

4.1.3 S&DA-Bus Proposed Protocol

Considering the specific scenario described above, the aim of S&DA-Bus routing protocol is to select the best node, being a person, a vehicle or a bus, to be used as relay to deliver data to the DC.

Sink Aware Centrality

We introduce a centrality metric called Sink-Aware Centrality (SAC) proposed in OBSEA [26]. To give a proper definition of SAC we need to introduce:

- Inter Contact Time
- Contact Graph
- K-Clique Community
- Local Centrality

The Inter Contact Time (ICT) is related to the contact frequency between two nodes. In particular, given two nodes, the ICT between them is the time elapsing between two consecutive encounters. Since in a DTN forwarding of packet happens upon contact, ICT is an important characteristic to account for.
The Contact Graph is a graph $G(V, E)$ where vertices represent nodes of the network and an edge between two vertices exists if the ICT is below a fixed threshold, denoted as ICT_{th}. In particular the ICT between two nodes is updated every time the two nodes start a new contact; whenever the ICT goes

above ICT_{th} if an edge exists it is deleted. In this sense the Contact Graph stores the recent contacts history of the DTN's nodes moving in the area. Note also that the contact graph is built in a centralized manner.
The communities are extracted from the contact graph just described. Communities are defined as K-Clique Communities [27]. A K-Clique Community is the union of all K-Cliques that can be reached from each other through a series of adjacent K-Cliques[1][27]. We choose this definition of community because it admits overlapping communities, thus, a node can belong simultaneously to multiple communities. The procedure used to get the communities out of the Contact Graph is called Clique Percolation Method (CPM) algorithm [27]. In order to run the CPM algorithm, the knowledge of the whole graph is needed, this fact explains the choice of generating the Contact Graph in a centralized manner.

The local centrality for a node x in the i-th community is denoted as h_x^i and it is defined as the number of edges node x has in community i. Thus, given the set $\mathbb{C}$ of all the communities in the Contact Graph, to each node is associated a social profile $(\mathbb{C}_x, \mathbf{h}_x)$, where $\mathbb{C}_x \subseteq \mathbb{C}$ is the vector of communities to which node x belongs to and $\mathbf{h}_x$ is the vector of h_x^i for each community $\mathcal{C}_x^i \in \mathbb{C}_x$.

Finally, the SAC, for node x, denoted as H_x, is defined as:

$$H_x = \sum_{\mathcal{C}_x^i \in \mathbb{C}_{sink} \cap \mathbb{C}_x} h_x^i \tag{4.1}$$

where $\mathbb{C}_{sink}$ is the set of all the communities to which the DC, our sink, belongs to. The SAC H_x for the node x is the sum of those h_x^i such that the i-th community contains both x and the DC.

Routing algorithm: S&DA-Bus

When nodes x and y meet, they calculate the weight $w_{x,y}$ of the link: if $w_{x,y} > 0$ x forwards its packet to y. The weight $w_{x,y}$ at the instant t, is given by:

$$\begin{aligned} w_{x,y}(t) = &(Q_x - Q_y) + \\ &+ \alpha\Big(H_y\big(1 + \beta f(\hat{t}_{y-DC})\big) - H_x\big(1 + \beta f(\hat{t}_{x-DC})\big)\Big) \end{aligned} \tag{4.2}$$

where:

- Q_x and Q_y are the number of packets in the buffers of node x and node y respectively,
- α and β are two coefficients (see below),
- $\hat{t}_{x-DC}$ is the time elapsed since the last x-DC contact, the same old for node y,
- $f(t)$ is a generic non-decreasing function such that $max(f(t)) = 1 \ \forall \ t \geq t_{th}$.

[1] Definitions:

- K-Clique: complete subgraphs of size K.
- Adjacent K-Cliques: K-Cliques with K - 1 shared nodes.

$w_{x,y}(t)$, given by Eq. (4.2), is the sum of two contributions: the former is related to the queue of nodes, the latter is related to the social properties of nodes by means of the SAC: H_x and H_y.
The role of α is to weight the relevance of the social contribution; in particular if α is zero, the protocol becomes a pure backpressure protocol. In this case, indeed, node x will forward a packets to node y only if the buffer of x contains more packets than the buffer of y. Therefore, in this case, the aim is to distribute packets among nodes in order to prevent packet losses due to the memory overload of a node. By increasing α a node is prioritized as a next relay not only based on memory consideration but also based on its SAC. Considering a node y and setting $\beta > 0$, the more $\hat{t}_{y-DC}$ increases, the higher is $f(\hat{t}_{y-DC})$ and, in its turn, it is more probable for $w_{x,y}$ to be positive, making y a suitable relay.
Finally, the threshold t_{th} of the function $f(t)$ is set to the average ICT among all buses and the DC (around 40 minutes in our case). Equation (4.3) is an example of $f(t)$ and Figure 4.2 is its graphical representation for the case $t_{th} = 40$.

$$f(t) = \begin{cases} \dfrac{t}{t_{th}} & \text{if } 0 \leq t \leq t_{th} \\ 1 & \text{if } t > t_{th} \end{cases} . \tag{4.3}$$

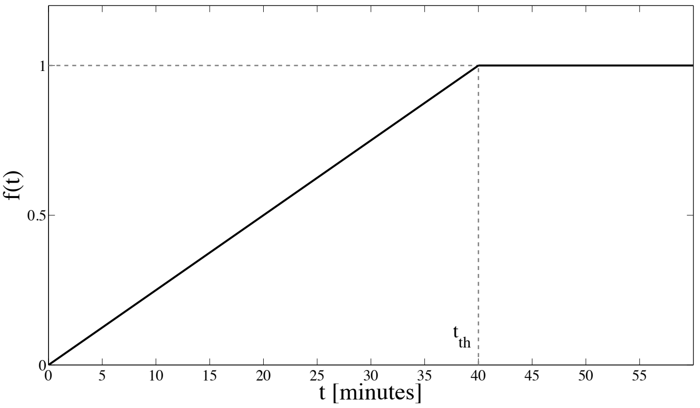

Figure 4.2: Example of $f(t)$ for $t_{th} = 40$.

The aim of S&DA-Bus is to take advantage of the pattern of buses' movement exploiting its periodicity. According to this consideration we shaped $w_{x,y}$. As a consequence, the effect obtained is twofold: i) buses tend to be preferred as relays ii) the preferred relay among buses is the one which met the DC longer ago (the one with higher $\hat{t}_{bus-DC}$). This allows to choose as next relay a bus which is moving in the direction of the DC, rather than one that has just met the DC, even if they had almost the same social profile. Notice also that if a bus never encounters the DC in its route, $\hat{t}_{bus-DC} = 0$ at every instant, and the bus will be considered as a common node.
Note that in the case of OBSEA, the weight $w_{x,y}$ is obtained by setting $\beta = 0$ [26].

Single-Copy and Multi-Copy Forwarding

Two forwarding modes have been considered: Single-Copy (S&DA-Bus-SC) and Multi-Copy (S&DA-Bus-MC). The first mode is, as suggested by its name, the case where each message in the network is unique. Each time a node forwards a packet according to the logic described in 4.1.3, it deletes such packet from its own queue.

The second mode allows to have more copies of the same message in the network. In particular, each node in the network forwards at most n_{copy} replicas of the same packet. We underline that also in the Multi Copy mode, the forwarding decision for each replica is taken according to Eq. (4.2).
Obviously, the introduction of the Multi-Copy mode will increase the network overhead, though increasing the number of delivered packets and reducing the average delay (this tradeoff is shown in Section 4.1.4). However, as it will be shown in section 4.1.4, it is possible to limit the network overhead, while increasing the number of packets delivered. For this purpose, we introduced a Replication Window (RW), which is a packet property defined as a timer started at packet creation. When the RW of a packet expires, the packet can not be replicated anymore, even if less than n_{copy} replicas were forwarded. This mechanism prevents packets to be replicated when they have become old. In addition, it can happen that a packet with such a long lifetime have already been delivered to the DC, making further replications useless.

4.1.4 Simulation and Numerical Results

The numerical results shown in this section, are obtained through the java simulator Opportunistic Network Environment (ONE) simulator [28].

Simulation Settings

The simulator reproduces the city center of Bologna (about 6 km^2) (see Figure 4.3), where we placed 485 nodes, divided in:

- 1 DC: fixed node placed in the center of the city, where most of the buses pass;
- 50 BCs: fixed nodes randomly placed;
- 34 buses (2 per route): mobile nodes that follow predefined routes, some of them are circular some other are "ping-pong" routes (reflecting the real buses routes in Bologna);
- 400 people: mobile nodes that follows the Working Day Movement model [29].

The node transmission range is set to 25 m, with the exception of the links people-buses that are set to 5 m, this follows the assumption that people communicate with a bus just when they are passengers (Figure 4.4).

Every building generates a packet of 1500 bytes every 15 minutes (for the whole simulation's duration: 24 hr.), with a total of 4717 packets generated during the simulation time. Note that the total number of packets generated

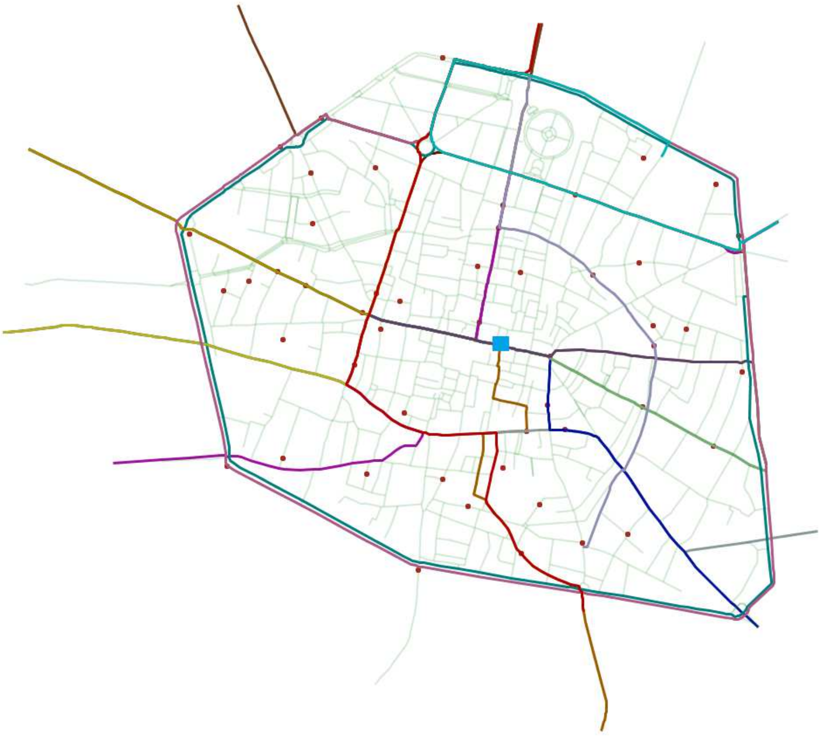

Figure 4.3: Reference Scenario: the city center of Bologna. Circles are BCs and the rectangle in the map center is the DC. Thick lines represent buses routes.

is not $24 \times 4 \times 50 = 4800$ because every building generates the first packet in an instant randomly chosen in the first hour. Every packet has the DC as its destination. The communities considered are K-Clique Communities, with $K = 4$. The Contact Graph is created imposing $ICT_{th} = 5$ hr (defined in 4.1.3). Finally for S&DA-Bus-MC, n_{copy} is set to 5.

Performance Metrics

From now on we will refer to the number of packets received at DC as Delivered Packets and to the total number of packets generated by buildings as Generated Packets. The performance were evaluated according to the following metrics:

- Delivery Probability: Delivered Packets / Generated Packets;
- Average Delivery Delay: delivery delay (Delivery Time - Generation Time) averaged over all Delivered Packets;
- Overhead: Ratio between the overall number of packets transmitted in the network and the number of Delivered Packets. We can see this metric as an indication of the network energy consumption since it tells about the number of forwardings needed to deliver one packet to the DC. Overhead is the price that Multi Copy schemes (e.g. S&DA-Bus-MC, Epidemic) have to pay to have better performance than Single Copy schemes.

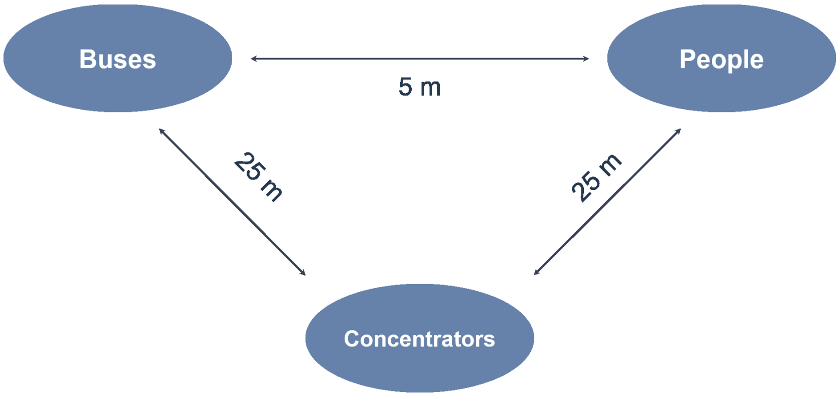

Figure 4.4: Radio interfaces transmission range.

The protocols used as benchmarks for comparison are:

- Epidemic with Oracle [2]: it is an upper limit on Delivery Probability and Overhead, while it is a lower bound in terms of Delivery Delay.
- OBSEA, presented in [26].

This benchmark are compared with: S&DA-Bus Single-Copy (SC) and S&DA-Bus Multi-Copy (MC). Before showing the relevant results, we will motivate the choice of parameters used to obtained the final results.

Parameters Setting

Either in S&DA-Bus-SC and in S&DA-Bus-MC we fixed $\beta = 5$, obtained through optimization. As Table 4.1 shows, as β grows, every metric improves till $\beta = 5$, then a plateau is reached and performance do not significantly increase further. Notice that even if Table 4.1 is relative to S&DA-Bus-SC, the same trend holds also for S&DA-Bus-MC.

β	Delivery Probability	AVG Delay [min]
0	0.538	195
0.5	0.607	174
1	0.609	162
5	0.613	158
9	0.613	157
29	0.613	157
49	0.614	157
99	0.614	157

Table 4.1: Performance of S&DA-Bus-SC by varying β.

To choose an appropriate value for α, we tested OBSEA on our scenario finding the results reported in Table 4.2. We see that $\alpha = 10$ gives the best results in terms of Delivery Probability at the expense of an increase of the

[2] Once a packet is delivered all its copies in the network are immediately deleted. Obviously this is unrealistic, but it well fits our purposes while all the considerations remain valid.

Average Delay of about 10 minutes only.
Moreover we verified that, by setting $\alpha = 10$, the contribute related to the queue in Equation (4.2) can be almost neglected in the majority of the cases. This is the desired effect since we are interested in the social aspect of the protocol while we do not focus on the back pressure behaviour. For the same reason the buffer size was set to infinite.

α	Delivery Probability	AVG Delay [min]
0	0.414	280
0.5	0.497	190
10	0.543	201

Table 4.2: Performance of OBSEA by varying α.

Table 4.3 shows the performance of S&DA-Bus-MC by varying RW. The table shows that an increase in RW causes a worsening in performance. This counter intuitive fact, is explained by the First-In-First-Out policy used for the queue management. Indeed, once a packet is inserted in the queue, it has to wait the transmission of all the packet and their replicas, that were already queued. For this reason we set RW = 4 hr.

RW[hr]	Delivery Probability	AVG Delay [min]	Overhead
4	0.68	137	63.44
5	0.67	138	73.75
24	0.59	145	105.3

Table 4.3: Performance of S&DA-Bus-MC by varying RW.

Numerical Results

Figures 4.5, 4.6 and 4.7 show the comparison among the cited routing protocols. As expected, the Epidemic routing protocol has the best performance in terms of Delivery Probability and Average Delay (except for the MC case), while it is the worst in terms of Overhead. In principle the Epidemic routing protocol should be the lower bound in terms of delay, however we are considering an Average Delay evaluated by averaging the delay over the total number of Delivered Packets. In all cases a confidence interval of 95% is considered. Such interval is shown in Figure 6 and, as can be noted, it is tight to the mean value in all cases (intervals duration is around 15 and 20 minutes). The reason why the Average Delay in the Epidemic case is higher than in the S&DA-Bus-MC case is the following. From simulations' results we verified that those packets that are delivered with a huge delay in the Epidemic case, are the same packets that the other routing protocols are not able to deliver. It follows that the delay in the delivery of those critical packets affects the Average Delay of the Epidemic routing protocol, while it is not accounted for in the other cases and in particular in S&DA-Bus-MC.
From Figure 4.5 it is possible to see that S&DA-Bus-SC improves the Delivery Probability of the simple OBSEA of around 10%, while reducing the Average Delay and Overhead with respect to OBSEA. This effect is obtained by accounting for the periodic movement of buses. Passing from S&DA-Bus-SC to

S&DA-Bus-MC we have an improvement in Delivery Probability and Average Delay, but this is obtained at the expense of an Overhead 11 times higher in the Multi Copy case.

Finally, we recall that the comparison with Epidemic routing was intended to have an upper bound in terms of Delivery Probability, which is the case.

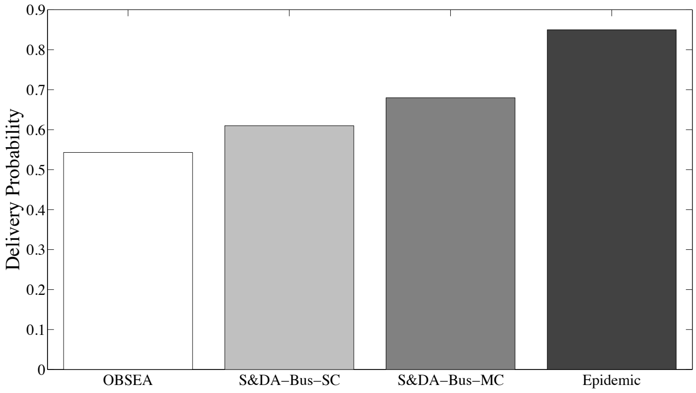

Figure 4.5: Delivery Probability: comparison.

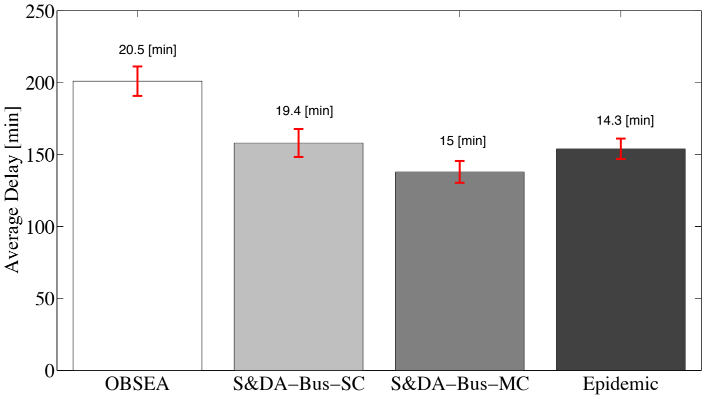

Figure 4.6: Average Delay: comparison and 95% confidence intervals. The number on top of each bar is the width of the confidence interval.

4.1.5 Conclusions

I have proposed S&DA-Bus, a routing protocol applying DTN paradigm to smart city environments. The role of buses is exploited to somehow reduce the uncertainty typical of DTN, by creating a backbone composed of buses acting as relays. The benefits of this approach were shown through software simulation, demonstrating a clear improvement with respect to identified benchmark protocols.

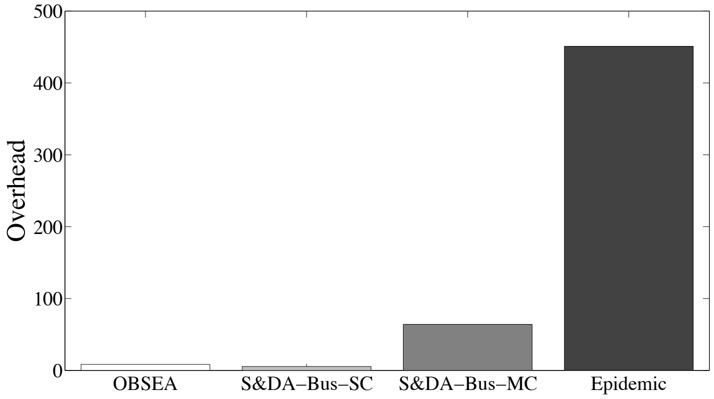

Figure 4.7: Overhead: comparison.

4.2 Delay Tolerant Networking for Smart City Through Drones

Unmanned Aerial Vehicles (UAVs) were used commercially for the first time in Japan at the beginning of the 1980s, when unmanned helicopters proved to be an efficient way of supplementing piloted helicopters to spray pesticides on rice fields. Progress has surged forward in technological capabilities, regulations and investment support, providing many new possible applications, particularly in agriculture, infrastructure, security, transport, media and entertainment, telecommunications, mining and insurance[3]. In the field of telecommunications, drones can help companies to address challenges, such as further development in order to cover white spots. In particular, drones can become part of the infrastructure, by playing a role in gathering data from traffic sources located on the ground.

In this work we study the impact of using a drone as an additional relay in the DTN. The flight duration is limited, therefore the number of buildings from which the drone can gather data during a flight is limited and depends on buildings location and on the reception range of the drone. Performance, in terms of average delay and delivery probability, are shown by changing the instant in which the flight is performed; the latter, in fact, affects the number of isolated nodes and their location in the area, that is performance. Results show that there exists an optimum starting time for the flight maximising the delivery probability.

4.2.1 Related Work

Several works have been done in the past regarding routing protocols for DTN. First Contact [18] routing is a simple routing protocol where a node having a packet forwards it to the first node in the radio range. Once a node has forwarded a packet it can not receive the same packet anymore to prevent loops. Epidemic Routing [19] is similar to First Contact but after a transmission a copy

[3] http://www.pwc.pl/en/drone-powered-solutions.html.

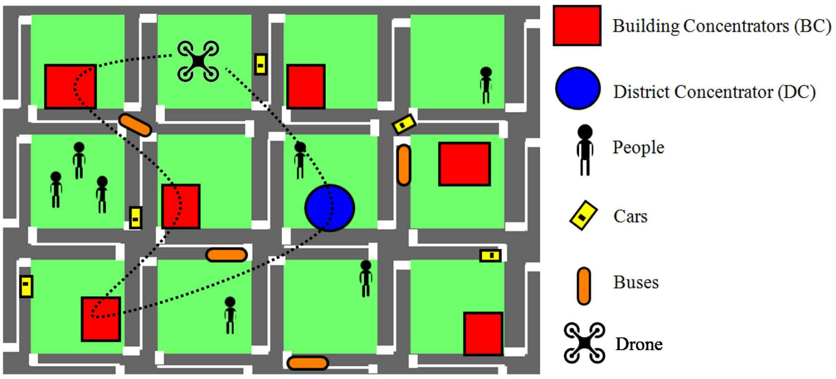

Figure 4.8: Reference Scenario.

of the packet is kept in the message buffer, therefore, multiple copy of the same packet are generated, such as every possible route from source to destination is explored.
Spray and Wait Routing protocol (S&W) [20] tries to exploit the advantages of Epidemic routing, but limiting the network load. The source is allowed to generate at most L replicas of the same packet, then, the packet spreading process is split into two phases: Spray and Wait. During the Spray phase nodes spray and forward packets according to a well defined logic. During the Wait phase nodes stand by for direct delivery to the final destination.
An example of DTN applied to a smart city is given in [23], where the network is used to spread information about the state of trash bins to inform the service personnel. In particular, in [23] trash bins are equipped with wireless sensors that transmit information about their state to garbage trucks, which are able to connect to the Internet. The information obtained through Internet by all the trucks are used to optimize the routes, redirecting garbage trucks in the city to minimize costs. Another example is provided in [24], where the public transportation in the city of Aachen is studied. The obtained results are used to simulate an Epidemic-like routing protocol.

4.2.2 Reference Scenario

The reference scenario consists of a smart city, where buildings generate data to be transmitted to a sink, called hereafter District Concentrator (DC). Data sensed inside a smart building are collected by several sensor networks deployed in every unit (e.g., flats, offices) of the building and they are gathered by the Building Concentrator (BC), which is in charge of the transmission toward the DC. Data generated by BCs are useful mostly for monitoring of environmental variables, like temperature or energy consumption, therefore they are expected to be small in size and to have relaxed requirements in terms of latency. These characteristics of the data to transmit are such to justify the use of a DTN.
Figure 4.8 shows our reference scenario: people, vehicles, buses are nodes of the network and they act as relays. This is done in order to exploit the urban mobility such nodes have by nature. For example, it could be useful to exploit as a relay a person that works in an office close to the DC, or a bus that collects

data from all passengers in order to deliver them to the DC when it passes by. In addition, differently from usual DTN, we consider a drone as further node, able to define a route to fly in order to get packets from those buildings that are isolated, that is not being served by common DTN nodes.

4.2.3 Air Interfaces and Routing Protocol

In our scenario, the drone, the DC, the BCs and all objects moving in the city are equipped with short-range air interfaces. We assume the drone can gather data only from buildings (people/bus/car-drone communication is not considered in this paper), that are at a distance lower than 50 m, and it can start delivering the gathered data to the DC, again when it is at a distance lower than 50 m. While for the communication among objects moving on the ground (people, cars, buses) and the BC or DC we assume a 10 meters transmission range.

We also assume a fixed packet transmission time equal to 0.1 s. Therefore, when two relays enter in contact they can exchange a number of packets which depends on the duration of the contact.

All nodes in the scenario use as routing protocol the binary version of S&W [20]. Accordingly, each source, during the Spray phase, can spread around at most L_0 copies of a message. In particular, at the $i - th$ contact, a source forwards $L_i/2$ copies of the message to the node it enters in contact with. Thus, the $i - th$ node encountered by the source will have $L_i/2$ copies that it can further spread around using the same logic. A node stops spreading a message when just one copy of the message is left in its buffer.

4.2.4 Drone Flight

In order to study the effect of the drone in the DTN under consideration, it is important to define when and where the drone should fly. We selected the DC as base location for the drone, thus we assume that the drone and the DC share the same information about delivered packets. Moreover it makes sense to assume the knowledge of all the BCs involved in the DTN together with their coordinates since they are in fixed positions. With these information the drone is able to determine the isolated buildings: a building is considered isolated if no packet was received at DC from that building at the moment of the observation. Obviously, the set of isolated buildings changes over time becoming smaller and smaller. In principle, the drone should start its flight passing by all the isolated buildings and entering in communication range with each of them. Moreover, the drone should stay in contact with isolated buildings enough time to receive all their packets in order to deliver them at the DC. However, as stated above, the drone has a limited battery-life. We assume our drone can fly for one or two hours at a speed of 5 m/s. Because of this constraint the drone has to select a subset of isolated buildings to serve.

The choice of the isolated buildings to serve is simply done according to a distance-based approach. In particular, the drone starts its flight from the DC location and then we apply a modified version of the closest neighbour strategy to identify at each step the next building to flight toward, considering that the drone can move to the next isolated building only if it has enough battery to go back to the DC after data collection at that building.

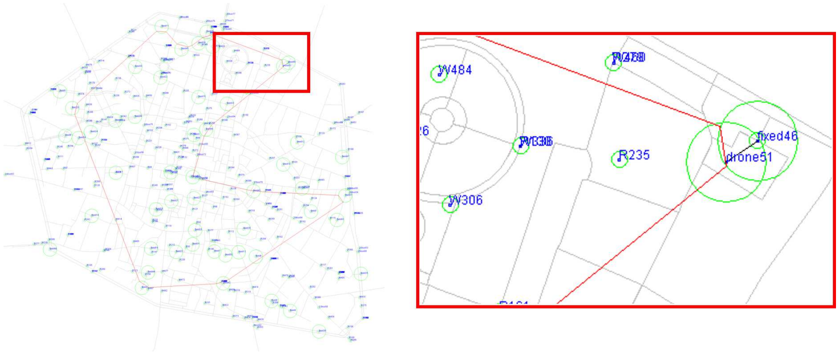

Figure 4.9: Example of drone trajectory: it cuts the range of a building in order to stay in contact enough to receive all the packets while moving at constant speed.

Moreover, the drone trajectory is calculated such that the drone can move at constant speed while receiving all the packets from all the building it passes by. In particular, Figure 4.9 shows that in our version of closest neighbour search, we do not simply select the next building to serve but, when possible, the drone flies over a building cutting its radio range, while assuring a radio contact long enough to allow the transmission of all the packets from the BC to drone. This consideration allows to make the drone trajectory shorter saving energy.

Another problem is related to the selection of the instant in which the drone should start its flight. Intuitively, the sooner the drone starts, the larger is the set of isolated buildings it can serve. From one hand, an early departure of the drone, can reduce the average delivery delay, on the other hand the buildings served by the drone could be some of those buildings that eventually will be served by other moving objects even without the help of the drone. Consequently, flying too early could bring the drone to serve buildings that would be served by other relays later, resulting in a waste of the drone service. The opposite case is when the drone starts flying close to the end of the simulation, in this case, the delivery probability is expected to be increased while there is no advantage in terms of average delivery delay.

We performed simulations starting the drone flight at different time instants from the beginning of the simulation.

4.2.5 Simulation and Numerical Results

The numerical results shown in this section, are obtained through the integration of the java simulator Opportunistic Network Environment (ONE) simulator [28].

Simulation Settings

The simulation Settings are the same described in Section 4.1.4, that are reported here for the reader convenience.

The simulator reproduces the city center of Bologna (6 km^2 about) (see Figure 4.3), where we placed 486 nodes, divided in:

- 1 DC: fixed node placed in the center of the city, where most of the buses pass through;
- 50 BCs: fixed nodes randomly placed;
- 34 buses (2 per route): mobile nodes that follow predefined routes, some of them are circular some other are “ping-pong” routes (reflecting the real buses routes in Bologna);
- 400 people: mobile nodes that follows the Working Day Movement model [29].
- 1 Drone: mobile flying node following a predefined route not constrained by streets (differently from all the other nodes).

Every building generates 100 packets at the beginning of the simulation. Every packet has the DC as its destination.

Numerical Results

From now on we will refer to the number of packets received at DC as Delivered Packets and to the total number of packets generated by buildings as Generated Packets. The performance were evaluated according to the following metrics:

- Delivery Probability: Delivered Packets / Generated Packets;
- Average Delivery Delay: delivery delay (Delivery Time - Generation Time) averaged over all Delivered Packets;
- Average number of Hops: Average number of hops needed by packets to be delivered;
- Overhead: Ratio between the overall number of packets transmitted in the network and the number of Delivered Packets. We can see this metric as an indication of the network energy consumption since it tells about the number of forwardings needed to deliver one packet to the DC.

In Figures 4.11, 4.12, 4.13, 4.14 the delivery probability, the average delivery delay, the average overhead and the average number of hops are shown, respectively, in the cases without drone or with the drone when the flight duration is one or two hours and for drone departure at 4, 8, 16 or 22 hours from the start of the simulation.

The first information given by all the figures is that the use of the drone improves all the performance under consideration. In particular, in Figure 4.11 the best result is obtained when the drone departure is after 8 hours. This is explained by the fact that after 8 hours the set of isolated buildings is smaller and the isolated buildings tend to be close to the border of the map (far from the DC placed at the center of the city) forcing the drone to a wide trajectory (see Figure 4.10), thus due to the battery life the drone can gather data from the few buildings it can reach. On the other hand, when departure time is set to 4 hours, the number of isolated buildings is higher and the drone is able to gather data from more buildings, however it is more probable that the data delivered by the drone are the same data that would be delivered even without it by the other objects.

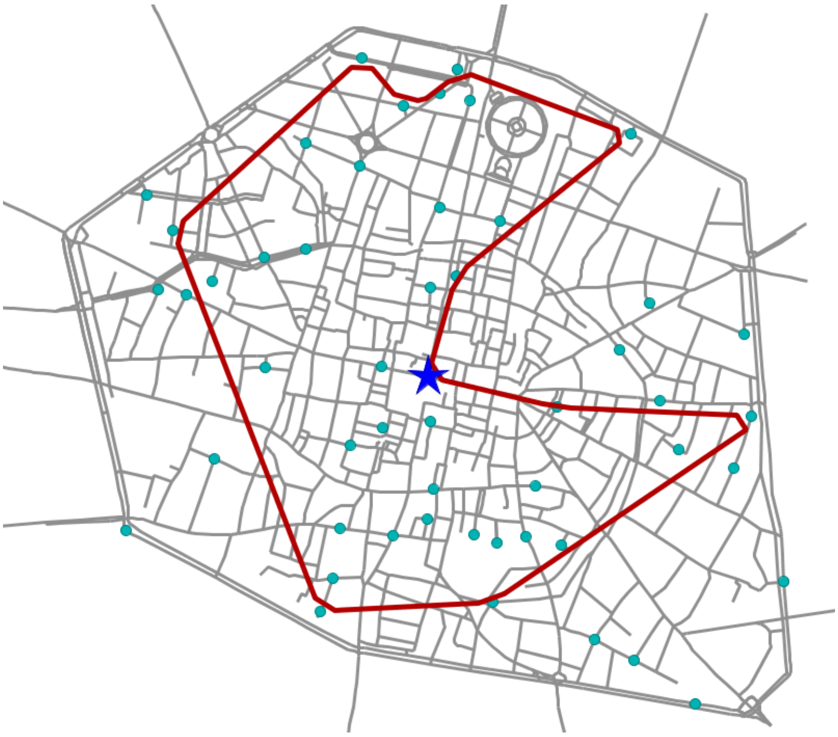

Figure 4.10: Trajectory of the drone if the departure time is 22 hrs and fly time 2 hrs. The star represent the DC, circles are BCs.

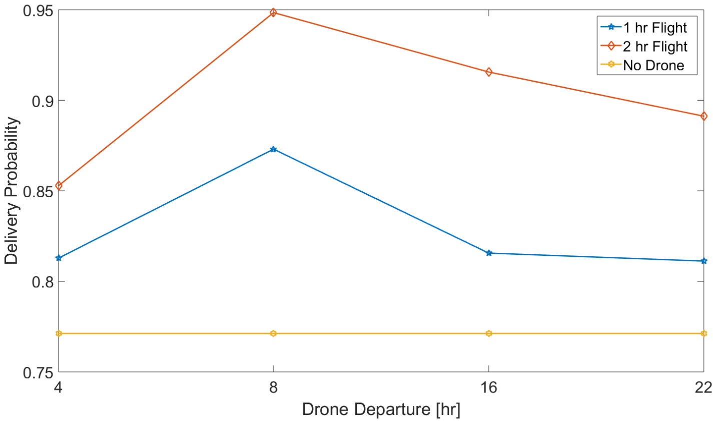

Figure 4.11: Delivery probability for drone departure at 4, 8, 16 or 22 hours from the beginning of the simulation.

Figure 4.12 shows that the sooner the drone departs, the lower is the average delivery delay. Notice that once the drone starts its flight, in one case it takes one hour to deliver the gathered packets, while in the other it takes two hours, but a two hours flight allows the drone to deliver more packets with a consequent drop of the average delay. This fact explains the crossing between the curves. Indeed, with a two hours flight at the beginning of the simulation, the number of packets delivered soon is enough to compensate for the two hours needed for the delivery. After the crossing, there are a lot of packets already delivered with a delay which increases the average such that the packets delivered through the drone need to be delivered quickly in order to lower the average, thus a delivery of these packets with a one hour flight keep the delay lower. The average delay when drone departure is higher than 16 hours increases above the case without drone, because all the packets delivered by the drone have an high delay, nonetheless they could not be delivered without the drone; consequently, the delivery probability with the drone is improved.

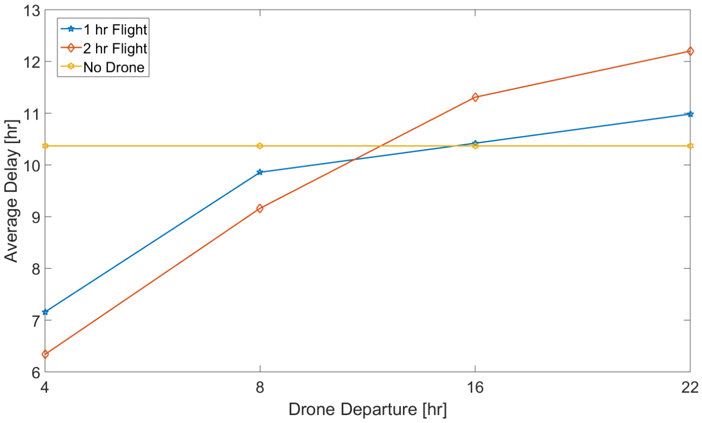

Figure 4.12: Average Delay for drone departure at 4, 8, 16 or 22 hours from the beginning of the simulation.

Figures 4.13 and 4.14 show a clear improvement also in terms of average overhead and average number of hops. The reason is that once the drone gets the packets it delivers them directly to the DC without further transmissions, hence more packets the drone delivers, less copies are created. Moreover each packet delivered by the drone is delivered in 2 hops.

4.2.6 Conclusions

In this work we consider a delay tolerant network for smart city applications, where buildings have to deliver a set of data to a final destination in the city, denoted as district concentrator. The proposed solution exploits the flight of a drone over the city with the aim of gathering the data generated by those buildings that are isolated. An optimum instant for starting the drone flight, maximising the delivery probability is found: the optimum is reached when a tradeoff between the number of isolated nodes in the area and the sparsity of their distribution in the city is found. In particular, in the best case it is possible to increase the delivery probability of the DTN of about 20% while reducing the average delay w.r.t the case when the drone is not used.

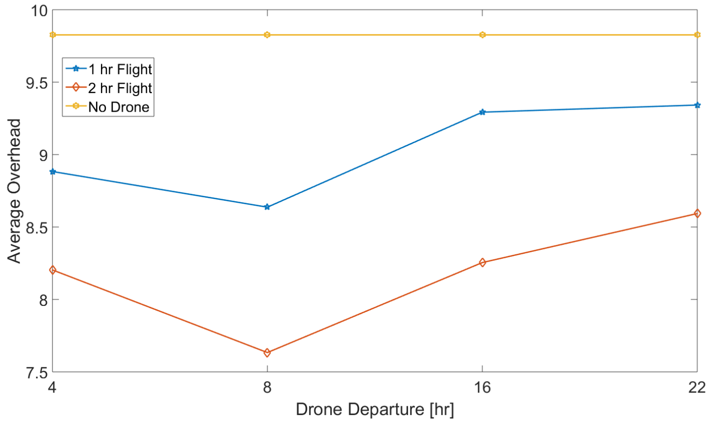

Figure 4.13: Average Overhead for drone departure at 4, 8, 16 or 22 hours from the beginning of the simulation.

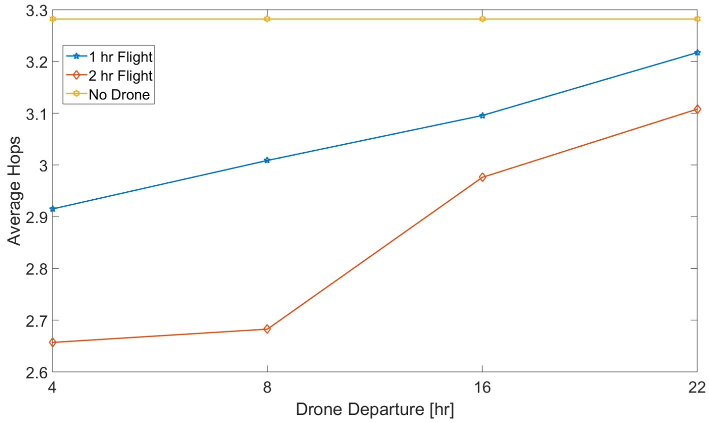

Figure 4.14: Average hops for drone departure at 4, 8, 16 or 22 hours from the beginning of the simulation.

Part II

Unmanned Aerial Base Stations

Chapter 5

Unmanned Aerial Vehicles: Route Optimization for crowded networks

This part of the dissertation regards papers [7] and [10]

5.1 Introduction

The next generation of cellular network is going to be standardized in the next years to come. Differently from the switch from 3G to 4G that actually was an evolution with an improvement of performance, the claim is for 5G to be a revolutionary generation. Hence, the challenges for 5G networks are countless. In fact, the increasing demand for new services and applications required by cellular users and industries make the market mature enough to accept disruptive innovations such as vehicular communications, industry 4.0 and Internet of Things to mention some. As a consequence, forecasts from [1] reports an increase of connected devices from 17.1 billions in 2016 to 27.1 billions in 2021. This forecast implies a much higher density of devices to be managed by the network. Intuitively, the increase of density will produce a larger standard deviation in the traffic generation process; in turn, a network deployment based on consideration about average or peak traffic predictions will bring to highly sub-optimal results. Therefore, it is widely accepted that 5G networks need to be as much adaptive as possible to the kind and the entity of traffic generated in space and time.

In order to reach such a degree of adaptability, future radio networks are expected to be characterized by flexible nodes able to autonomously react to spatial/temporal variations of traffic demand. The fundamental unavoidable problem even after the release of 5G standard remains: infrastructure equipment placed in static location and its density in the considered area put a limit on the offered traffic reachable.

Recent studies on Unmanned Aerial Vehicles (UAVs, a.k.a. drones) and on their use as Unmanned Aerial Base Stations (UABSs), might be a viable alternative to classic static Terrestrial Base Stations (TBSs) [30]. The advantages in using

UAVs are manyfold:

- The UAVs can fly over the terrestrial plane serving users **WHERE** TBSs cannot reach offering coverage and capacity,
- With a proper trajectory planning in agreement with TBSs the UAV can satisfy traffic demand **WHEN** needed,
- Line of Sight (LoS) conditions are easy to achieve either in the front hand toward the users or in the wireless back-haul,
- the wireless back-haul links can use millimeter waves or Visible Light Communications or whatever other wireless technology.

Taking inspiration from the work [31] by S.Mignardi et al., the focus of this work is on optimal trajectory planning for UABS with limited battery capacity. In particular, the interest is on the provision of high throughput video services to mobile users in urban environment.
Thanks to the collaboration with prof. V. Cacchiani and the group of operational research of the University of Bologna, in this work we build a generic framework enabling the description of the UABS Trajectory Planning Problem (UT2P) through an Integer Linear Program (ILP) resulting in an optimal trajectory which maximizes a given objective function. Moreover, we propose an heuristic algorithms giving sub-optimal solutions that, however, requires much less computational time. One of the most important aspect of this work is the framework developed to deal with a traffic demand varying both in space and in time.
In section 5.2, we give an overview of the literature about UAV-aided networks. Section 5.3 describes the scenario we account for, with an introduction to the terminology used in the rest of the paper and an insight on the traffic and channel models considered. How to model the problem of UAB route optimization is reported in section 5.4. In the same section the ILP to find the optimal UABS trajectory is proposed together with all the parameters needed for its definition. In section 5.5 we describe an heuristic algorithm to be used in place of the optimal one given the computational burden it implies. Finally, comparisons between the heuristic algorithm and the optimal one, together with other results will be discussed in section 5.6.

5.2 Literature Overview

The first results about the use of UAV-aided networks were related to link-level considerations, where the main focus were on the characterization of the path loss, and on its impact on the air-to-ground channel. For instance, in [32] and [33], it is studied the effect of the user-UAV angle w.r.t. the ground plane with the consequent drone height and it was shown that these parameters have a non negligible impact on the mean path loss and on the shadowing formulation.
Further works on UABs to find an acceptable trade-off between coverage, capacity and connectivity were proposed in [34] and [35]. In particular, the former considers a public safety application showing capacity and coverage related metrics evaluated by means of Monte Carlo simulations.
The objective of drone altitude optimization for downlink coverage was studied

in [36] where through circle pack-theory, multiple UABs are coordinated to use power control in order to limit interference. The topic of the maximization of downlink coverage and of optimal altitude, was also studied in [37] but with the focus on energy consumption. The authors also face the possibility of using multiple UAVs is discussed along with an examination of the effect of the distance between them in terms of mutual interference. Energy consumption was also addressed in [38], by exploiting scheduling and game theory to coordinate UABs.

In [39] a 3D-scenario is considered where multiple UAVs act as relays in a Device to Device-like communication. In the paper the authors optimize the length of the links user-UAV-BS through particle swarm optimization.

UAVs as relays were studied also in [40] where UAVs follow data traffic density distribution in the area and in [41] where a UAV is used as a mule in a delay tolerant network for smart cities application.

The paper [42] is of particular interest for our work, since it proposed a method to reconstruct a radio map of the considered environment through UAVs. Hence, through the proposed approach, it is possible to let an UAV fly offline in order to reconstruct and store in its memory the radio map. After that, for the purpose of the application we are proposing in this paper, it would be possible for the UAB to know the channel UAB-user for any user in the considered area, provided that the user position is known.

Finally, in [31] it is considered an ultra-dense cellular scenario where the TBSs are not able to serve all the users because of lack of coverage or of radio resources. Then, unsatisfied users have to be served by a UAB, however these users have to be satisfied quickly enough in order to meet their requirement in terms of downlink throughput.

In this paper, we take inspiration from [31] as starting point. However, differently from all the other papers, we developed a generic framework that allows the description of the considered scenario through the definition of an ILP that could be solved returning an optimal solution to satisfy as many users as possible.

5.3 Scenario

Reference Scenario

The scenario we consider is a cellular scenario where TBSs and mobile users are distributed. An UAB is present in the considered area and it is placed in its Home-Base (HB), that typically coincides with a TBS. The role of the HB is to allow the recharge of the UAB battery; thus, before expiration of the battery, the UAB must come back to the HB[1].

Among all the users, some are defined as unsatisfied. Unsatisfied users (UUs) are those users that cannot be served by TBSs in the traditional ways because of bad channel quality (either in terms of low Signal to Noise Ratio -SNR- or low Signal to Interference Ratio -SIR-) or limited amount of radio resources available.

[1] Note that in practice the time needed for a complete recharge is comparable to the battery life-time. Therefore, provided that two UABs are alternated to bring services to the users, the second UAB could start flying as soon as the first one reaches the HB to recharge.

An example of the scenario where only UUs are shown is reported in Figure 5.1. In the example, TBSs are placed on a square lattice and the location of HB is coincident with the central TBS. Note, that UUs happen to be distributed along the cell edges, this fact makes sense since users in this positions are the most susceptible to bad channel quality and interference generated by nearby TBSs. In fact, the proposed model can be applied to every kind of distribution of UUs, caused by a certain distribution of TBSs. For instance, a more realistic scenario w.r.t. the one shown in Figure 5.1 is a scenario where TBSs are not regularly distributed, implying a clusterized distribution of UUs.

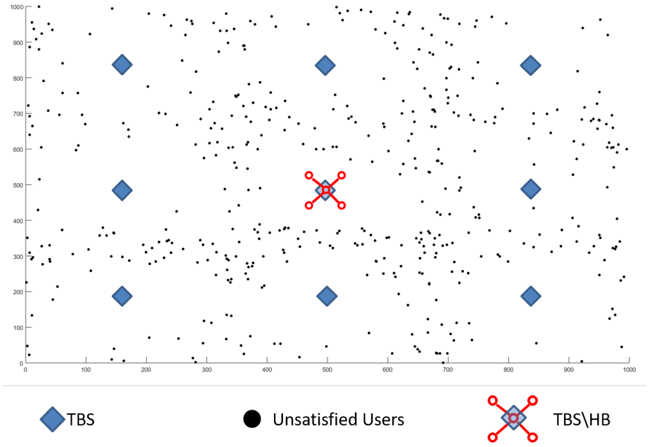

Figure 5.1: Scenario Example

We stress that the communications at ground level between TBSs and users are not considered. What matters in this work is the distribution of UUs that is taken as input as well as users' activation/expiration sequence. In fact, these input can be considered as generated by scheduling and transmissions dynamics happening at ground level. For instance, the UUs distribution shown in the example reported in Figure 5.1, was kindly provided by the authors of [31] where the ground level is accounted for via simulations of radio channel and scheduling.

Traffic Model

As stated above, initially, the UAB is located in its HB, meanwhile, at ground level, users are generating traffic for the download of a video content. The TBSs in the area schedule users according to some algorithms in order to satisfy requests of service. We assume that, in order to schedule users, the TBSs acquire information on user positions and on user expiration time, which is a requirement on the maximum time a user is willing to wait to get the required service. Once the TBSs have done their best to satisfy users, the UAB comes into the game trying to serve those users remained unsatisfied. Indeed, all the information (position and expiration time) about UUs are wirelessly communicated to the UAB. With this information, the UAB can run an algorithm to find a trajectory such that:

- as many UUs as possible are served within their expiration time,

- the UAB is able to come back to the HB before battery expiration.

An important and realistic fact that we consider is that while the UAB is flying the TBSs continue to serve users and to communicate to the flying UAB the information needed about new UUs. This assumption in fact allows to consider a dynamic scenario where a user tries to access a TBS and in case it cannot be satisfied, it becomes an UU for UAB, that will try to serve it within the expiration time. After the end of the expiration time a user will become expired. In practice, to generate the traffic, UUs are distributed in an area and each of them is assigned to an activation time, an expiration time and a demand in bits. A UU can be served by the UAB when it is an Active Unsatisfied User (AU), i.e. when it is activated and not yet expired. An AU becomes satisfied if all its demand can be served by the UAB.

Channel Model

The links to consider in our scenario are of three types: UAB-user, UAB-TBS, TBS-user. As already mentioned, what happens on the ground is out of the scope of this work, consequently the channel TBS-user is not considered.
For what concerns the link UAB-TBS, we assume it to have infinite capacity. Obviously, this assumption is unrealistic, however it is motivated by the fact that LoS condition is often achieved in the UAB-TBS link. Moreover, a different assumption adds further complexity to the framework we built, in any case it is currently being faced (together with other improvements) for an extension of this work.
On the other hand, whatever channel models could be used for the UAB-user channel. In particular, for sake of simplicity, we use the following channel model:

$$P_{rx}^{dBm} = \begin{cases} P_{tx}^{dBm} - 10\beta \log_{10}(d) - k & \text{if } d < r \\ 0 & \text{Otherwise} \end{cases} \tag{5.1}$$

Where:

- r is the transmission range in meters of the UAB, intended as the maximum distance for which the received power at the user is higher than a threshold P_{th}, i.e. $P_{rx} > P_{th}$,
- β is the propagation exponent,
- $k = 10\beta \log_{10}(\frac{4\pi}{\lambda})$ is a constant, where λ is the wavelength in meters.

Then, the capacity (C) is evaluated as the Shannon capacity:

$$C = B_c \log_2(SNR + 1) \tag{5.2}$$

where:

$$SNR = \frac{P_{rx}}{NR_b} \tag{5.3}$$

In equations 5.2 and 5.3:

- B_c is the signal bandwidth in $[Hz]$,
- N is the monolateral noise spectral density in $[W/Hz]$,

- R_b is the bit rate in $[bits/sec]$.

One further assumption is that for any AU, the UAB needs to know the channel in the link UAB-user. Luckily, as already commented in Section 5.2, we could use an offline flight of the UAB during off-peak hours to scan the considered area in order to reconstruct a radio-map as proposed in [42]. For the sake of simplicity we also assume that the UAB disposes of an infinite number of Radio Resources that are orthogonal to the radio resources of the TBSs. This fact means that all the users covered by the UAB can be served without suffering of interference caused by TBSs.

5.4 Framework and Problem Modeling

To model the problem, we define a space, time and speed discretization. The plane (at a given hight) where the UAB can move is discretized in a squared grid with a certain space-step to which the UAB movement is constrained. The intersection points of the grid define the Turning Points (TPs) for the UAB. Let C be the set of TPs in the considered area (TPs corresponding to HB are duplicated, for convenience, as the UAB departs from HB and returns at HB). Notice, that the UAB will move from one TP to an adjacent TP, i.e., it will move in one of the four directions North, South, East or West (movements along the diagonals are not allowed). Time is also discretized: in particular, we choose the time-step as the ratio between the space-step and ξ_{max}, the maximum allowed speed for the UAB. The time horizon is defined according to the UAB battery capacity: let T be the set of time instants in the considered time horizon. We consider discrete speeds for the UAB in the range $[\xi_{min}, \xi_{max}]$, where ξ_{min} represents the minimum allowed speeds for the UAB. In particular, to avoid working with time-steps of different duration, we select as allowed discrete speeds ξ_{max} and a given number of speeds in the range $[\xi_{min}, \xi_{max}]$ such that the speed is obtained as ratio of the space-step over a multiple of the time-step. In other words, the speeds are selected so that the time needed for the UAB to move from a TP to an adjacent one is a multiple of the time-step.

Given the described space, time and speed discretization, we model the considered problem on a directed time-space graph $G = (V, A)$. The set of nodes V contains an artificial source σ and an artificial sink τ (introduced for notation convenience), as well as a node v_{ct} for each TP $c \in C$ and time instant $t \in T$ that the UAB can choose. The set of arcs A contains: (i) artificial starting arcs from σ to the nodes $v_{HB,t}$, representing the departure time instants of the UAB from HB, (ii) artificial ending arcs from the nodes $v_{HB,t}$ to τ, representing the arrival time instants of the UAB at HB, (iii) and travel arcs $(v_{ct}, v'_{c't'})$ representing the movement of the UAB from TP c at time instant t towards TP c' at time instant t' by using one of the selected speeds.

Notice that unsatisfied users $u \in UU$ are not necessarily placed in TPs. Therefore, we need to take into account which unsatisfied users are served by the UAB when it moves between two TPs along an arc $a \in A$. In particular, we assume that the UAB, while moving along an arc $a = (v_{ct}, v'_{c't'})$ serves the unsatisfied users it covers when it is in c. Therefore, through the radio map and by knowing the position of all the unsatisfied users, the UAB is able to estimate the capacity it can provide to the users when it is in a certain TP. Since each arc $a = (v_{ct}, v'_{c't'})$ has a time duration $l_a = t' - t$, by knowing the capacity in bits per time-unit, it is possible to derive how many bits are received by each covered users in the period l_a. We let $r_{uc(a)}$ represent the capacity (in bits per time unit, see equation (5.2)) that the UAB can provide to unsatisfied user $u \in UU$ while moving from TP $c(a)$ (starting node of arc a) along arc $a \in A$. Notice that, depending on the user demand d_u ($u \in UU$), the UAB can need to traverse several arcs providing capacity to the user before completely serving it. Only if the demand is completely satisfied, the user is counted as served.

Finally, we need to take into account the users activation and expiration. Let a_u be the activation time and e_u the expiration time of user $u \in UU$. Since the UAB cannot provide capacity to users that are expired or not active, we set to 0 the $r_{uc(a)}$ of all arcs $a = (v_{ct}, v'_{c't'}) \in A$ such that $t < a_u$ (i.e., the user u is still not active) and of all arcs $a = (v_{ct}, v'_{c't'}) \in A$ such that $t > e_u$ (i.e., the user u is already expired).

The considered problem corresponds to the well-known NP-hard Orienteering Problem (OP) [43] in the following particular case: all the unsatisfied users have demand $d_u = 1$, $u \in UU$ (i.e. a user can be served by the UAB in a single travel), each unsatisfied user is placed in a TP and, for each $u \in UU$, $r_{uc(a)} \neq 0$ for only one arc $a \in A$ (i.e. each unsatisfied user can be served only by traversing a specific arc), the expiration time of all users coincides with the battery capacity ($e_u = B$, $u \in UU$), all users are active at the beginning of the time horizon ($a_u = 0$, $u \in UU$), and the UAB has only one allowed speed. Indeed, OP calls for determining a Hamiltonian cycle whose total edge weight does not exceed a given threshold, while visiting a subset of nodes with maximum total profit. In the studied problem, the profit is obtained by the served users: in particular, we associate a priority p_u to all unsatisfied users $u \in UU$, and the goal is to maximize the weighted sum of the served users. The threshold on the total edge weight corresponds to the battery capacity. Differently from OP, in the considered problem, there is the possibility of serving a user more than once, the users can be served by traversing different arcs and it is not necessary to visit the user in order to serve it, the UAB can travel at different speeds, and users become active or expired at different time instants. Since the studied problem generalizes OP, it is also NP-hard.

We propose an Integer Linear Programming (ILP) model to solve the problem, based on the defined time-space graph $G = (V, A)$. Let $s(a)$ be time instant of the starting node and $e(a)$ the time instant of the ending node of arc $a \in A$. Let B be the UAB battery capacity. Finally, let $\delta^+_{(v)}$ represent the set of outgoing arcs and $\delta^-_{(v)}$ the set of ingoing arcs from/to node $v \in V$.

We introduce two types of binary variables:

$$y_u = \begin{cases} 1 & \text{if user } u \in UU \text{ is served} \\ 0 & \text{otherwise} \end{cases}$$

$$x_a = \begin{cases} 1 & \text{if arc } a \in A \text{ is chosen in the UAB trajectory} \\ 0 & \text{otherwise} \end{cases}$$

Integer Linear Problem Formulation:

$$\max \sum_{u \in UU} p_u y_u \tag{5.4}$$

s.t.:

$$\sum_{a \in A: s(a) \geq a_u e(a) \leq e_u} r_{uc(a)} l_a x_a \geq d_u y_u \quad \forall\, u \in UU \tag{5.5}$$

$$\sum_{a \in \delta^+_{(\sigma)}} x_a = 1 \tag{5.6}$$

$$\sum_{a \in \delta^-_{(v_{ct})}} x_a = \sum_{a \in \delta^+_{(v_{ct})}} x_a \quad \forall\, v_{ct} \in V \setminus \{\sigma, \tau\} \tag{5.7}$$

$$x_a \leq \sum_{a' \in \delta^-_{(\tau)} : e(a') - s(a) \leq B} x_{a'} \quad \forall\, a \in \delta^+_{(\sigma)} \tag{5.8}$$

$$y_u \in \{0, 1\} \quad \forall\, u \in UU \tag{5.9}$$

$$x_a \in \{0, 1\} \quad \forall\, a \in A \tag{5.10}$$

The objective function (5.4) maximizes the served users by weighting them according to the priority. In practice, the priority can be used if we want to give more importance to some users than others. For instance in a "freemium" business model where users paying premium services have to be served by the UAB with higher priority, or in a case where nodes can be either people or machines and people should be served with higher priority.

Constraints (5.5) impose to consider a user as served only if its demand is completely satisfied within the time window of the user (i.e. after it is active and before it is expired).

Constraints (5.6) and (5.7) impose to select a feasible path for the UAB from the source to the sink of G, i.e., the UAB starts from and goes back to HB.

Constraints (5.8) guarantee that the path ends at τ before expiration of the UAB battery, i.e. the UAB is able to return to HB. In particular, we must select an arc from σ such that the time difference between the arrival time at τ and the departure time from σ is smaller or equal to the battery capacity.

Finally the last constraints (5.9) and (5.10) assure that the variables are binary.

The main issues related to the solution of this model are the following. The first one is the size of the time-space graph both in terms of memory needed to store it and for the complexity it implies, since the number of variables and constraints depend on the size of G. In particular, the number of different speeds, the size of space- and time-steps, the number of users and the battery capacity deeply affect the computation time needed to solve the model. Similarly, the size of the matrix containing elements $r_{uc(a)}$ also depends on the number of arcs and on the number of users. We wish to mention that we made the assumption that the UAB, before starting its flight, knows the activation and expiration times of users for the entire duration of its battery. However, in practice, users become active depending on which users the TBSs is not able to satisfy. Therefore, a realistic algorithm should learn on the go the activation and expiration of the users without the possibility of taking the optimal decision before starting the flight. However, regardless of these issues, the optimal solution of the ILP model can used as an upper bound in the design of realistic algorithms.

5.5 Heuristic Algorithm: W-Rolling Horizon (W-RH)

The W-Rolling Horizon algorithm consists in setting a time window (W) and repeatedly solve a slightly modified version of the ILP proposed in Section 5.4. In particular, the UAB starts from a starting-TP and solves the ILP by setting the battery endurance at W stopping in a destination-TP [2] that allows the UAB to come back to the HB in a time $B - W$. Then, a new iteration starts considering the previous destination-TP as the new starting-TP and again W as the new UAB battery then the modified model is solved again. This procedure goes on until $B - W = W$, at this point, the ILP is solved for the last time ensuring the UAB comes back to the HB.
The advantage of this approach is that at every iteration it is possible to dynamically rebuild the space-time graph that is going to have reduced dimension and complexity, leading to a much quicker resolution of the problem. On the other hand, the solution will be sub-optimal.
In the results section we will show the effect of different values of W.

5.6 Results

In the simulation performed, the radio parameters are set as shown in Table 5.1. Table 5.2 shows the parameters setting for what concern the framework. Notice that a realistic value for the UAB battery is around 30 minutes; unfortunately we are going to use much lower values because of the impact of that value on the time needed to find the solution and on the memory required, in particular for the optimal ILP (which is in fact, NP-hard).

P_{tx}	9 dBm
r	300 m
β	3
λ	12 cm
B_c	180 KHz
N	10^{-20} W/Hz
R_b	30 Kbit/sec

Table 5.1: Channel Parameters

To study the performance of the proposed UAB-aided network, we analyzed three metrics, namely, the number of served users, the sum throughput that is $\sum_{u \in U} y_u d_u / B$ and the time needed to find a solution. In particular we are going to show results by varying B, the user expiration time (E) and the window W for the W-RH algorithm.
Notice that the solution obtained through the optimal model it is not necessarily the optimal one since, being the problem NP-hard, the time required to reach the optimal solution increases quickly by increasing the dimension of the input instance (i.e. the dimension of the space-time graph and the number of users). In fact, we stopped the solver as soon as the relative gap between the incumbent

[2] The "come-back-Home" constraints are removed

Area Size	1 Km^2
Space Step	100 m
Time Step	5 s
Drone Speeds	{20, 10} m/s
#UUs	500
Expiration Time (E)	60-100 s
Battery (B)	120-240 s
Activation Time	Uniform[0;B] s
Demand	Uniform[2;200] Mbit

Table 5.2: Framework Parameters

solution and the incumbent estimated upper bound was 30%. Note that, the found solution could even be an optimal solution even if the gap is not 0. This is caused by the fact that while the solver cannot find a better solution, it tries to decrease the upper bound to match the value of the incumbent solution, but we stop it before it matches since this process is too slow. The solution obtained will be referred to as "the x% gap solution
In Table 5.3 the optimal model is considered. The first column reports the Time To Solve (T2S), that is the time needed to the solver to find a 30% gap solution, actually proving that the gap is 30% by lowering the estimated upper bound. The second column shows the Time To Best (T2B), which is the time needed to the solver to find a solution that will be proven to be a 30% gap solution after some processing (taking a time equal to T2S - T2B). The last two columns report the value of the incumbent solution found when the solver is stopped after 2 minutes versus the case 30% gap solution is found.

B, E	T2S [sec]	T2B [sec]	Incumbent - 2 min.	Incumbent - Gap 30%
120, 60	5.85×10^4	8.60×10^3	244	257
120, 120	335	325	235	282
240, 60	8.36×10^4	6.38×10^4	211	304
240, 120	1067	10^3	176	355

Table 5.3: ILP related metrics

Notice that, when the battery is low, 2 minutes are enough to get a number of satisfied users that is very close to the one provided by the 30% gap solution in a time that is considerably higher than 2 minutes. However, the same consideration does not hold when B increases, where 2 minutes are not enough to get close to the 30% gap solution.
Looking at Table 5.3, one may wonder why a lower E implies an higher T2S. Table 5.4 contains the answer to this question. The last column of Table 5.4 represents the number of useful arcs, intended as those arcs that, if crossed by the UAB, allow the UAB to provide a non 0 capacity to the covered AUs. First, notice that fixed B, the space-time graph is the same as denoted by the number of arcs and vertices. Nonetheless, the number of useful arcs is lower for lower E. Consider that the UAB have to choose carefully where to go, indeed, the wrong choice may preclude the UAB to traverse other useful arcs that are too far. Thus, making the wrong decision when E is lower will exclude some

useful arcs from the “few” available, while an higher E (and the consequent higher number of useful arcs) gives more chances to the UAB to compensate for a wrong decision. To sum up, a lower number of useful arcs calls for a more careful choice implying an higher T2S.

B, E	$\lvert A\rvert$	$\lvert V\rvert$	#Constraints	$\#r_{uc(a)} \neq 0$
120, 60	16080	2365	2864	716820
120, 120	16080	2365	2864	919737
240, 60	37200	5269	5768	916492
240, 120	37200	5269	5768	1503399

Table 5.4: My caption

In Figure 5.2, the comparison between the 30% gap solution and one that results from the W-RH approach in terms of satisfied users, is presented. In particular, four cases are shown for different settings of B and E. For a given setting of B and E, the W-RH solution provided in Figure 5.2 is the best out of all the solutions obtained by changing W. Notice also that the result provided in Figure 5.2 is relative to a specific scenario and it is not averaged (again, because of the time required to solve the optimal model), differently from all the other results reported in this section.
From Figure 5.2 it can be seen that an increase of either B or E allows the drone to serve more users. Moreover it can be appreciated that the solution obtained through the optimal model always provide better solutions.

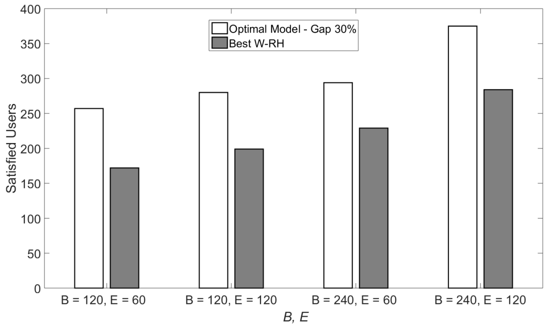

Figure 5.2: Comparison between optimal model and W-RH

All the next figures present results averaged over 100 scenarios where the placement of users is fixed but the activation/expiration of users and the demand of users is randomized.

Figures 5.3, 5.4 and 5.5 show results for the W-RH when B is set to 120 seconds and the E is set to 60 and to 120 seconds in terms of, respectively, satisfied users, sum throughput and time to solve (T2S). In particular, from Figure 5.3, it can be seen that it exist an optimal setting for W, which is $W = 25$ sec. This behavior is due to the fact that the UAB does its best in the following window but it does not know what happens next. As consequence, if it finds a region where a lot of AU are present, it will move there regardless of

other users that are going to become active in the next window. In turn, once the UAB is in a certain TP, it might not be able to move in other good regions because of the "go-back-Home" constraints. Moreover, from the same figure it is evident that higher E allows the UAB to serve more users.
Consistently, in Figure 5.4 it is shown how the sum throughput is in line with the number of satisfied users.
Figure 5.5 underline the NP-hard nature of the problem, showing that a bigger input instance caused by an higher W implies an inherently increasing T2S. Nonetheless, the T2S setting E to 60 or 120 seconds, is about the same.

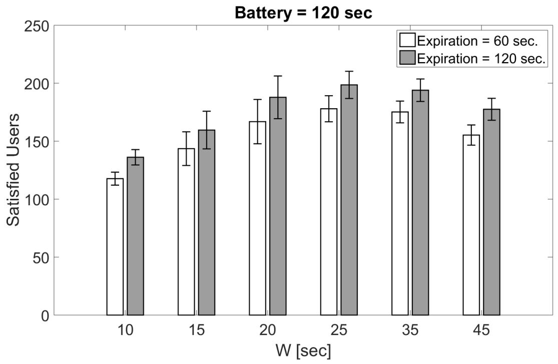

Figure 5.3: Satisfied Users for $B = 120$, $E = 60, 120$ and varying W.

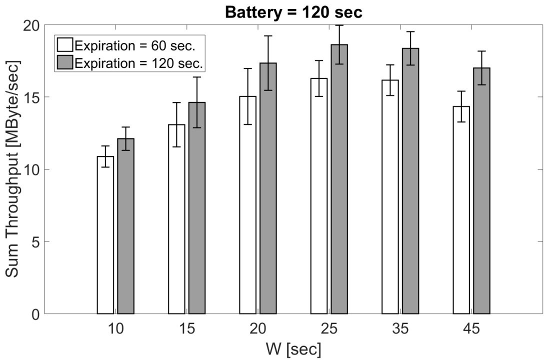

Figure 5.4: Sum Throughput for $B = 120$, $E = 60, 120$ and varying W.

Figures 5.6, 5.7 and 5.8 show results for the W-RH when B is set to 240 seconds and the E is set to 60 and to 120 seconds in terms of, respectively, satisfied users, sum throughput and T2S.
A comparison between Figure 5.3 and Figure 5.6, highlights that, fixed E, an increase of B imply an higher number of satisfied users as expected. However, the best setting of W is different and depends on the value of B. In particular when $B = 240$ sec. the best value for W is 35 sec. Furthermore, it can be noted a sudden drop in terms of satisfied users when $W > 45$ for the setting $B = 240$ sec. and $E = 60$ sec. This effect caused by the fact that (i) the activation time is uniformly randomly generated over B and (ii) by the small E that cause AUs to become expired quickly. In fact, it is clear that in the case $B = 240$, $E = 60$

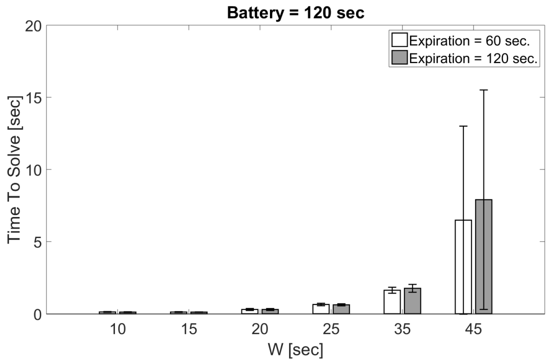

Figure 5.5: Time To Solve for $B = 120$, $E = 60, 120$ and varying W.

(i) there will be less AUs per time step and (ii) a lot of users will expire without the possibility for the UAB to reach them in time.
As before, the sum throughput shown in Figure 5.7 is in line with the number of satisfied users.

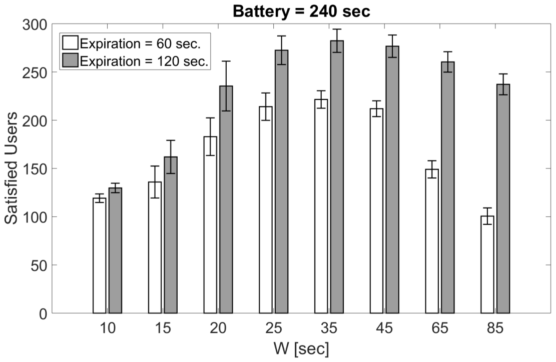

Figure 5.6: Satisfied Users for $B = 240$, $E = 60, 120$ and varying W.

For what concerns T2S when $B = 240$, in Figure 5.8 we neglect to show in the plot the results for $W = 65$ and $W = 85$. This values are reported in table 5.5. Notice that for the case $E = 60$, the drop in satisfied users when $W > 45$, make it redundant to show the neglected cases. On the other hand when $E = 120$, there are two reasons why we do not report the values for $W = 65$ and $W = 85$ in the bar chart. The former is that the T2S in these cases is too high w.r.t. settings where W is smaller, thus a bar chart with al the values would be unreadable. The latter is that since the optimal value of W is 35, it does not make sense to use higher W leading to worse results while requiring significantly more T2S.

Indeed for $E = 60$, as shown in Figure 5.6, the performance are far from the best one when $W = 65$ or 85. Indeed, as aforementioned, the On the other hand, when $E = 120$ The former is that the T2S in these cases is too high, thus unreadable in the same bar chart. The latter is that, since the optimal value of W is 35, it does not make sense to use higher W leading to worse results while

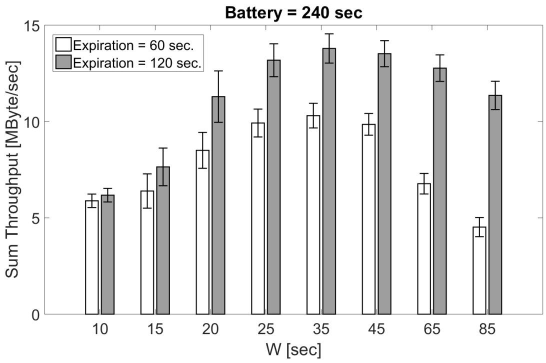

Figure 5.7: Sum Throughput for $B = 240$, $E = 60, 120$ and varying W.

B, E, W	T2S [sec]
240, 60, 65	11.39
240, 60, 85	17.5
240, 120, 65	136.4
240, 120, 85	1671

Table 5.5: Time To Solve

requiring significantly more time.

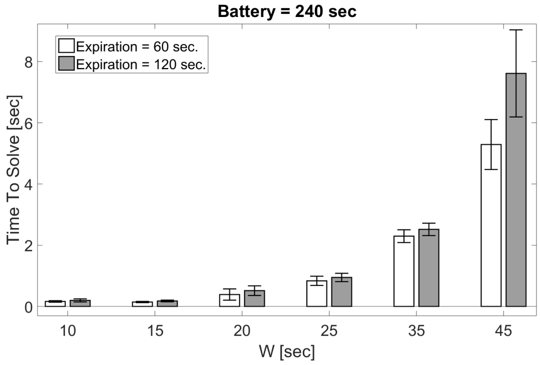

Figure 5.8: Time To Solve for $B = 240$, $E = 60, 120$ and varying W.

Part III

Device to Device Communication

Chapter 6

MILP Based Radio Resource Assignment for Device to Device Communications

This part of the dissertation regards paper [9]

6.1 Introduction

Capacity has always been a problem for cellular network operators especially after integrating data services into their networks. One emerging solution to this problem is the use of Device-to-Device (D2D) communications, that is enabling direct communication between devices bypassing the base station (BS), hence exploiting one of the devices involved in the D2D link as a relay between the BS and the other device.

Recent studies have revealed that a lot of traffic in cellular networks is due to duplicated downloads of the same content; for instance, the top 10% of videos on YouTube, accounts for 80% of all views [44].

According to the concept of homophily, in fact, people being in the same community, hence eventually in the proximity of each other, tend to have the same interests, thus sharing and downloading the same contents. Since the D2D paradigm empowers the traditional networks by exploiting users proximity, it seems a natural solution to the described problem.

The positive effects of D2D proximity are manyfold. Since mobile users are expected to be within the vicinity of each other, they use lower levels of transmit power, thus prolonging the battery life of the mobile phones, achieving better throughput and possibly higher spectral efficiency due to local reuse of eNB resources [45]. On the other hand, the activation of D2D links may cause interference to cellular users served by the base station and viceversa, thus there is the need of interference management algorithms to prevent degradation of the QoS expected by users.

Motivated by the interest shown by organizations like 3GPP in the topic [46], this paper studies via simulations a network inter-operated by cellular and D2D users. The focus is mostly on: 1) proposing an algorithm formulated as a *Mixed Integer Linear Program* for the mode selection, which is the selection of the D2D link to activate; 2) studying the interference introduced by the D2D links, both in the case of using uplink or downlink resources, together with the related impact on the network performance.

6.2 Literature Overview

In [47] the authors propose to reuse cellular uplink resources for D2D communication and they develop a power control mechanism according to which D2D UEs use the signal power received in the downlink frame to determine their pathloss to the base station in order to scale their transmit power such that they can communicate with each other directly during the uplink frame while causing only minimal interference to the base station.
In [48] it is studied the uplink interference between D2D and cellular UEs. In the proposed mechanism to mitigate interference, D2D UEs read the resource block allocation information from the control channel and transmit using a resource block assigned to a cellular UE not in proximity. Moreover, information about the expected interference generated by D2D UEs on cellualar on a specific resource block is broadcasted to all D2D UEs. Thus D2D UEs are able to adjust their transmit power accordingly.
An interference mitigation scheme is proposed in [49]. In the paper, the authors focus on mitigating the interference between D2D communication and cellular networks for downlink resource sharing. In particular, it is considered a resource allocation scheme by setting interference limited area for both the D2D transmitter and the D2D receiver, aiming to control the mutual interference between D2D and cellular UEs. Moreover, to guarantee quality of service to cellular UEs the eNB is in charge to the resource assignment and it can force D2D transmitters to reduce their transmission power.
A different approach to interference is proposed in [50], [51] and [52] where dedicated resources are allocated for D2D, obviously in this way it is possible to get rid of interference but less radio resources are available for the communication. In contrast with these works, we consider both uplink and downlink in order to compare performance and select the best option in terms of network throughput. Moreover, in our work the selection of the nodes to be coupled in a D2D link is made by solving a Mixed Integer Linear Program (MILP).

6.3 System Model

6.3.1 Reference Scenario

We consider a circular macro cell with radius R, and an eNB placed in the center (see Fig. 6.1). Inside the cell N users (UE) are randomly and uniformly distributed. Our interest is on downlink transmissions, from eNB to UEs, through a direct link (no D2D case), or through a two-hop communication (D2D case). In both cases (D2D used and not) radio resources are assigned by the eNB (see Section 6.4).

When D2D links are activated the N UEs are divided in three categories: relays, targets and not D2D users. Relays and targets are D2D-enabled nodes, that is each relay transmits to exactly one target, performing a D2D link. Consequently, we have N_{D2D} nodes split in $N_{\text{R}} = \frac{1}{2}N_{\text{D2D}}$ relays and $N_{\text{T}} = \frac{1}{2}N_{\text{D2D}}$ targets. In addition, we will denote as UE_{I} the $N_{\text{I}} = N - N_{\text{D2D}}$ users not coupled in any D2D link.

In order to account for interference that may be caused by D2D links over UEs, these UE_{I}, referred as interference users, are cellular users that communicate with the eNB reusing the same radio resources assigned to relays for the D2D communication. Obviously, if UE_{I} need downlink services and the D2D links are scheduled with uplink resources, there is no interference to account for; the same can be said if UE_{I} need uplink services while D2D links are scheduled with downlink radio resources. Conversely, interference plays a role when UE_{I} reuse the resources already in use by D2D links. It will be shown in numerical results that the reuse of uplink or downlink resources has different impact on the performance.

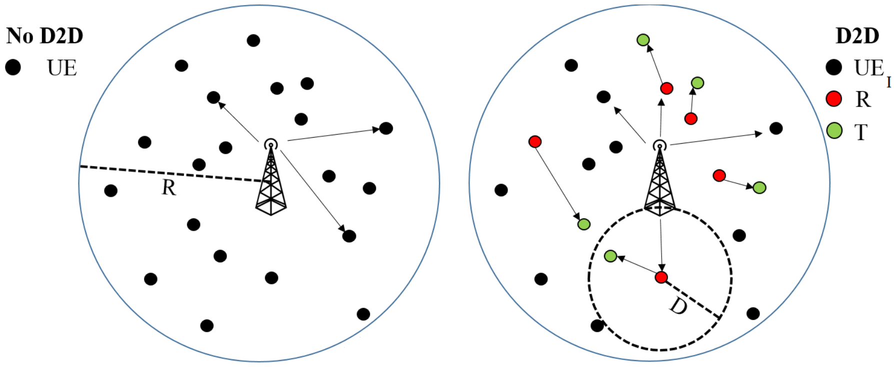

Figure 6.1: Reference scenario.

6.3.2 Channel Model

Two propagation models were used in the simulations for the communication between UEs and for the communication with the eNB. The propagation models used are (i) the Urban Macro propagation model (UMa) [53] for eNB-UE and (ii) the ITU-R P1411-6 proposed in 3GPP meetings [54] for UE-UE.

In all cases the channel is modeled also considering Line-Of-Sight (LOS) or Non Line-Of-Sight (NLOS) condition, according to the equations reported in Table A.2.11.2-8 in [53].

6.3.3 LTE and Power Control

In LTE, the available bandwidth is split in sub-bands of 15 kHz centered at the respective sub-carriers used for OFDMA (Orthogonal Frequency-Division Multiple Access) in downlink and for SC-FDMA (Single Carrier Frequency-Division Multiple Access) in uplink. The number of resource blocks (RB) available in the system depend on the number of subcarriers: 12 subcarriers correspond to a resource block. We consider a bandwidth of 20 MHz corresponding to 100

RBs. In LTE this number of RBs is available at each subframe of 1 ms.
Once the RBs are assigned to users, according to the Channel Quality Indicator (CQI) and to the Signal-to-Noise-Ratio (SNR), which are periodically reported to the eNB by all the connected users, the eNB can select a suitable Modulation and Coding Scheme (MCS) for the transmission [55]. In particular, to each range of SNR a MCS is associated.

In our Power Control (PC) scheme we check which is the best MCS that can be exploited transmitting the maximum possible power, then we reduce it to the minimum level needed to use the same MCS. Consequently, even though the transmission power is reduced, the SNR remains in the range associated to the best MCS achievable.
Finally according to 3GPP tables [56], from the knowledge of the MCS used and of the number of RBs assigned to the considered user in the current subframe, it is possible to extract the Transport Block Size (TBS). This is an indication of throughput, since it provides the amount of bit that can be carried in a transport block in a subframe of 1 ms.

6.4 MILP for D2D Mode Selection

When a user needs to get some proximity services, it has first to discover the available surrounding UEs that provide such services and this is called *discovery phase*. After discovery, the UE will connect to one of the discovered proximate UEs and the *communication phase* starts. Currently, D2D is still being standardized by 3GPP, as proximity-based services (ProSe) where devices detect their proximity and subsequently trigger different services. There are two types of ProSe discovery [57]. In this work we consider the *Network-Assisted ProSe discovery* where the network fully controls UE operations and the eNB is in charge of selecting the couple of nodes to link in D2D communication. Moreover, since detection reliability is extremely important in D2D, we decided to assign dedicated radio resources to the discovery phase, while the communication phase shares resources with the cellular network.
For sake of simplicity, we assume the only D2D links that could be created are the so-called disjoint D2D direct links, where each node is part of just one D2D link and it can be only a target or only a relay.
We assume the eNB knows the SNR of each couple of UEs in the cell and their positions. The decision whether to activate and assign resources to a D2D link is referred to as mode selection and it is taken by solving a Mixed Integer Linear Program (MILP), optimizing the following objective function:

$$\min \quad \sum_{i=1}^{N} \sum_{j=1}^{N} -\left(\alpha_1 \gamma_{ij} + \alpha_2 \gamma_i\right) x_{ij} \tag{6.1}$$

Subject to

$$
\begin{aligned}
(\gamma_{ij} - \gamma_j)x_{ij} &\leq 0 & i = 1,..,N; j = 1,..,N \qquad (6.2)\\
x_{ij} &\leq \Delta_{ij} & i = 1,..,N; j = 1,..,N \qquad (6.3)\\
\sum_{j=1}^{N} x_{ij} &\leq 1 & i = 1,..,N \qquad (6.4)\\
\sum_{i=1}^{N} x_{ij} &\leq 1 & j = 1,..,N \qquad (6.5)\\
\sum_{i=1}^{N} x_{ij} + \sum_{j=1}^{N} x_{ji} &\leq 1 & i = 1,..,N \qquad (6.6)\\
x_{ij} &= 0 & i = j \qquad (6.7)\\
x_{ij} &\in [0; 1] & i = 1,..,N; j = 1,..,N \qquad (6.8)
\end{aligned}
$$

where:

- i and j are the indices of relays and targets respectively.
- $x_{ij} = 1$ if the D2D i-j link is created, 0 otherwise.
- γ_{ij}: SNR perceived by the target on the link relay-target.
- γ_i: SNR perceived by the relay on the link eNB-relay.
- Δ_{ij} is a matrix such that $\Delta_{ij} = 1 \Leftrightarrow \mathit{distance}(i, j) < D$: a D2D link can be created if and only if relay and target are at most at distance D (assured by constraint (6.3)).
- α_1 and α_2 are weighting parameters.

In the proposed MILP, constraint (6.2) allows the creation of a D2D link if and only if γ_{ij} is larger than the SNR of the direct link target-eNB. In this way we try to avoid the creation of a D2D link that would be worse than the respective cellular one. Constraints from (6.4) to (6.6) assure that the D2D link created are disjoint D2D. Notice also that setting a discovery range is also a simple way to account for sociality as described in Section 6.1, indeed devices (i.e., people) communicate only if they are physically close to each other, thus eventually being also socially close. Finally, in the MILP interference is not considered because this would imply an high complexity in the scheduling that may be excessive considering that in LTE scheduling is performed every ms.

6.5 Communication and Resources Assignment

In the case of no D2D, the eNB assigns to the N UEs the downlink radio resources using a simple round robin scheduler. Then, the MCS is selected and power control is applied.

In the case of D2D we distinguish two phases in the communication. During the first phase all the UE$_\mathrm{I}$ and the relays are assigned downlink RBs in order to perform content download. After this first phase, the MILP performs mode selection and the relays switch from reception to transmission toward the target (we recall that we assume half-duplex). To analyse the role of interference in the two cases when the relay-target communication uses uplink or downlink RBs, the second phase is different depending on the considered case.

- *D2D commmunication over downlink RBs*: downlink RBs assigned to relays in the first phase are used during the second phase for the commmunication relay-target. Nonetheless, the same RBs are considered available from the eNB that reassign them to $\mathrm{UE_I}$. In particular the RBs assigned to a certain relay during the first phase, are assigned to the furthest $\mathrm{UE_I}$ during the second phase.
- *D2D commmunication over uplink RBs*: after the first phase, both relays and $\mathrm{UE_I}$ release their downlink RBs and they are assigned uplink RBs. In particular, the uplink RBs assigned to a relay are assigned also to the furthest $\mathrm{UE_I}$.

Notice that also in the D2D transmission PC is used; this could be very effective, since the D2D link can be very short.
Notice that the policy described for the RBs assignment, in principle should foster the performance in case of use of downlink radio resources because of the interfering links involved.
To shed some light on the implications of using uplink or downlink RBs consider Figure 6.2. When downlink RBs are assigned to relays for the D2D transmission the useful links are relay-target and eNB-$\mathrm{UE_I}$, but the same transmissions generate interference towards $\mathrm{UE_I}$ and target, respectively. On the other hand, when uplink RBs are assigned to D2D communication the useful links are relay-target and $\mathrm{UE_I}$-eNB, in this case, interference is generated towards eNB (by relay) and target (by $\mathrm{UE_I}$).

6.6 Performance Metrics

We evaluate performance in terms of network throughput ratio, denoted as η, which is the ratio between the network throughput achieved when D2D is used and when it is not. In particular, the network throughput is defined as the sum of the number of bits transmitted from the eNB to the users in the cell in a reference time interval, T. The latter is obtained by scaling the TBS (see section 6.3.3) to T.

By denoting as $\Sigma^{(\mathrm{net})}_{\mathrm{NoD2D}}$ the network throughput in the case of no D2D used, we have (see Figure 6.3(a)):

$$\Sigma^{(\mathrm{net})}_{\mathrm{NoD2D}} = \sum_{i=1}^{N} B_i/T \tag{6.9}$$

where B_i are the bits received in a time T by the i-th UE.

We now evaluate the network throughput when D2D is used and when interference is not present. For $\mathrm{UE_I}$ the throughput is evaluated as before. On the other hand, for targets and relays we can not use the same approach, also because half-duplex is considered. Considering a single D2D link, the relay receives a certain amount of bits B_R from the eNB then it switches to transmission

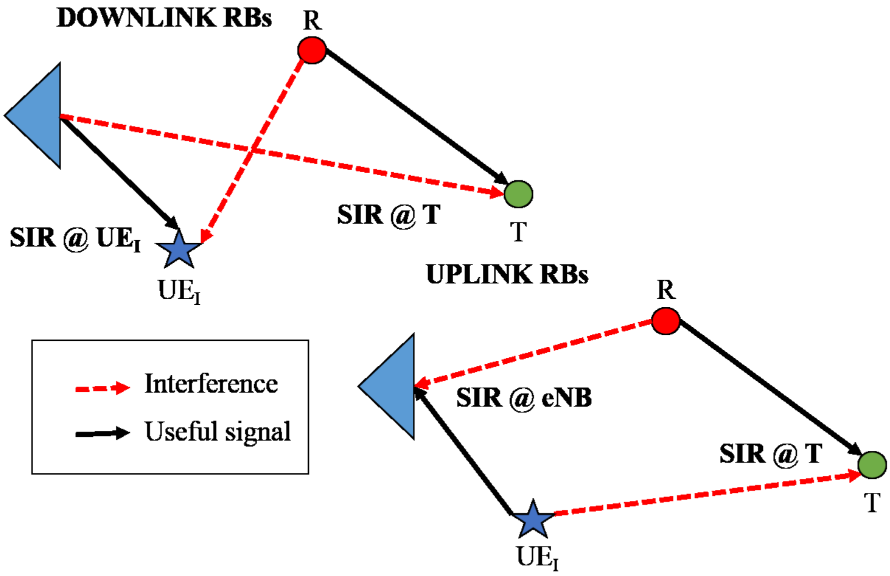

Figure 6.2: Interference using uplink or downlink resources.

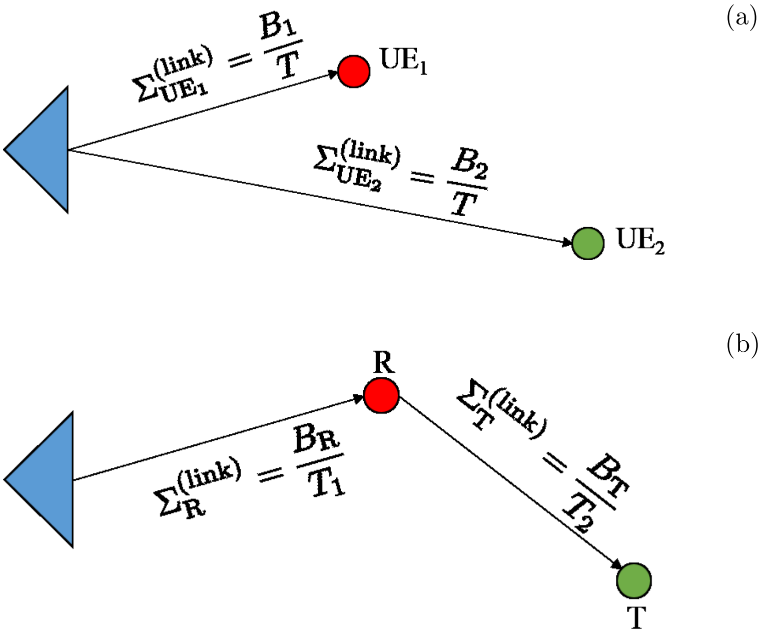
(a)
$\Sigma^{(\mathrm{link})}_{\mathrm{UE}_1} = \frac{B_1}{T}$
$\mathrm{UE_1}$
$\Sigma^{(\mathrm{link})}_{\mathrm{UE}_2} = \frac{B_2}{T}$
$\mathrm{UE_2}$
(b)
R
$\Sigma^{(\mathrm{link})}_{\mathrm{R}} = \frac{B_{\mathrm{R}}}{T_1}$
$\Sigma^{(\mathrm{link})}_{\mathrm{T}} = \frac{B_{\mathrm{T}}}{T_2}$
T

Figure 6.3: Throughput per UE (a) and per D2D link (b).

mode (it stops receiving) and it starts transmitting the same B_{R} bits toward the target. To have a fair comparison with the case when D2D is not used, we find the value of B_{R} such that the entire transmission process eNB-relay-target lasts for a time equal to T. B_{R} can be found solving the linear system:

$$\begin{cases} B_{\mathrm{R}} = \Sigma_{\mathrm{R}}^{(\mathrm{link})} T_1 \\ B_{\mathrm{T}} = \Sigma_{\mathrm{T}}^{(\mathrm{link})} T_2 \quad \Rightarrow B = T \dfrac{\Sigma_{\mathrm{R}}^{(\mathrm{link})} \Sigma_{\mathrm{T}}^{(\mathrm{link})}}{\Sigma_{\mathrm{R}}^{(\mathrm{link})} + \Sigma_{\mathrm{T}}^{(\mathrm{link})}} \\ B_{\mathrm{R}} = B_{\mathrm{T}} = B \\ T = T_1 + T_2 \end{cases} \tag{6.10}$$

where (see Figure 6.3(b)) $\Sigma_{\mathrm{R}}^{(\mathrm{link})}$ and $\Sigma_{\mathrm{T}}^{(\mathrm{link})}$ are the throughput in the links eNB-relay and relay-target, respectively. B_{R} is the number of bits transmitted from eNB to relay in time T_1 and B_{T} is the number of bits transmitted from relay to target in time T_2. As a consequence, the throughput of a single D2D link is given by $\Sigma_{\mathrm{D2D}} = \frac{2B}{T}$. Finally, the overall network throughput, having a number N_{I} of UE$_{\mathrm{I}}$ and a number $\frac{1}{2}N_{\mathrm{D2D}}$ of D2D links, can be evaluated as:

$$\Sigma_{\mathrm{D2D}}^{(\mathrm{net})} = \frac{\sum_{i=1}^{N_{\mathrm{I}}} B_i}{T} + \sum_{j=1}^{\frac{1}{2}N_{\mathrm{D2D}}} \Sigma_{\mathrm{D2D}_j} \tag{6.11}$$

By dividing the network throughout of eq. (9) and that of eq. (11) we get the network throughput ratio, η, in the absence of interference.
Now, in order to compare the performance when uplink or downlink resources are used, and to check the impact of interference on the D2D performance, we derive the network throughput ratio in the presence of interference. To distinguish from the case of no interference, we now denote the network throughput ratio as η_{DWN} for the case of use of the downlink resources, and as η_{UP} for the case where uplink resources are used. The latter are given by:

$$\eta_{\mathrm{DWN}} = \frac{1}{\Sigma_{NoD2D}^{(net)}} \left(\sum_{i}^{N_{\mathrm{R}}} \Sigma_{\mathrm{R}_i}^{(\mathrm{link})} + \sum_{i}^{N_{\mathrm{T}}} \Sigma_{\mathrm{T}_i}^{(\mathrm{link})} (1 - \mathcal{F}_{\mathrm{DWN}}^{\mathrm{T}}) + \sum_{i}^{N_{\mathrm{I}}} \widetilde{\Sigma}_{\mathrm{I}_i}^{(\mathrm{link})} (1 - \mathcal{F}_{\mathrm{DWN}}^{\mathrm{I}}) \right) \tag{6.12}$$

$$\eta_{\mathrm{UP}} = \frac{1}{\Sigma_{NoD2D}^{(net)}} \left(\sum_{i}^{N_{\mathrm{R}}} \Sigma_{\mathrm{R}_i}^{(\mathrm{link})} + \sum_{i}^{N_{\mathrm{T}}} \Sigma_{\mathrm{T}_i}^{(\mathrm{link})} (1 - \mathcal{F}_{\mathrm{UP}}^{\mathrm{T}}) + \sum_{i}^{N_{\mathrm{I}}} \widetilde{\Sigma}_{\mathrm{I}_i}^{(\mathrm{link})} (1 - \mathcal{F}_{\mathrm{UP}}^{\mathrm{eNB}}) \right) \tag{6.13}$$

where $\mathcal{F}$ is the failure rate and it is the the probability that the signal-to-interference ratio (SIR) is lower than a threshold, SIR$_{\mathrm{th}}$. This failure rate is different if we use downlink of uplink resources. In particular, referring to Fig. 6.2, in the case of reuse of downlink resources the interfered links are eNB-target and relay-UE$_{\mathrm{I}}$, thus we define $\mathcal{F}_{\mathrm{DWN}}^{\mathrm{T}}$ and $\mathcal{F}_{\mathrm{DWN}}^{\mathrm{I}}$. Reusing uplink resources, the interfered links are UE$_{\mathrm{I}}$-target and relay-eNB, thus we define $\mathcal{F}_{\mathrm{UP}}^{\mathrm{T}}$ and $\mathcal{F}_{\mathrm{UP}}^{\mathrm{eNB}}$. Finally, $\widetilde{\Sigma}_{\mathrm{I}}$ indicates the throughput of UE$_{\mathrm{I}}$ after the D2D communication started. We recall that UE$_{\mathrm{I}}$ before D2D transmission and relay are not interfered, hence for these type of nodes we consider the failure rate to be zero.

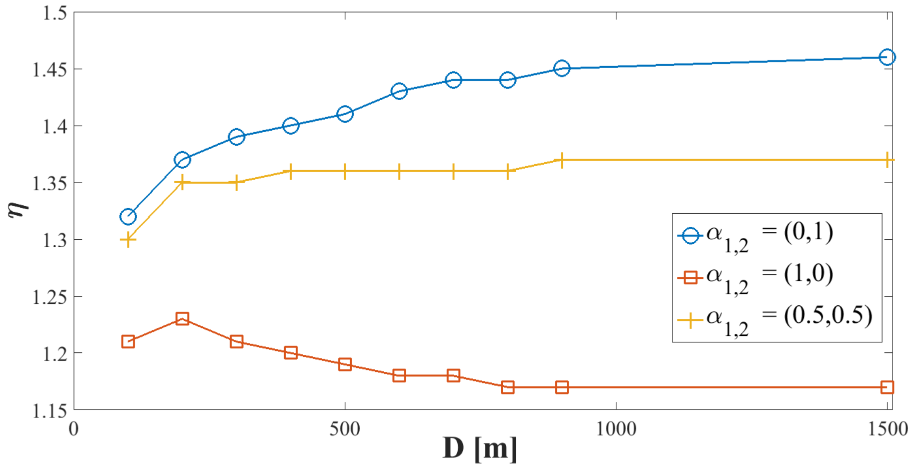

Figure 6.4: η vs D for $R = 750$ m and 100% D2D capable devices.

6.7 Results

In our simulations we set $N = 100$ and $T = 1$ sec. and we consider the scenario parameters: $R = 500, 750$ m and D from 100 m up to $2R$ m, moreover we show results for different percentages of D2D capable devices out of the 100 users. For each configuration of R and D, 100 scenarios were generated and simulated in order to get averages. In each scenario different values of α_1 and α_2 of the MILP were evaluated; for the sake of conciseness only the relevant cases are reported.

In Figure 6.4 we show η as a function of D, when $R = 750$ m and for three settings of $\alpha_{1,2}$. Passing from $D = 100$ m to $D = 200$ m η always increases because there are few cases in which a relay finds eligible targets. Increasing D above 200 m, this effect becomes negligible, thus it is possible to appreciate the effect of the MILP with different settings of $\alpha_{1,2}$. The case $\alpha_{1,2} = (0, 1)$ is the best one, suggesting that in a half-duplex D2D transmission, the link limiting the throughput is the link eNB-relay. Indeed, the opposite case, $\alpha_{1,2} = (1, 0)$, is always the worst since $\alpha_2 = 0$ and we do not control the SNR of the link eNB-relay. In this latter case, an increase of D causes a drop of η. Indeed, even with no control on the link eNB-relay, D acts as an upper bound for the distance relay-target helping in keeping an high SNR.
To evaluate the impact of R in Table 6.1 we report the case $R = 500$ m. As can be seen, D2D is still beneficial, even though η is decreased w.r.t. to the case of $R = 750$ m. This happens because with a smaller cell radius the radio links between eNB and UEs tend to be shorter, hence they tend to be characterized by a better channel. In such a situation there is small space for the improvements eventually introduced by D2D. The same effect is underlined by the fact that the impact of $\alpha_{1,2}$ on the performance is less notable.

All the result presented in the following were obtained by setting $R = 750$ m, $D = 500$ m and SIR$_{th} = 2$ dB.

In Figure 6.5 it is shown the variation of η by changing the percentage of D2D capable devices. In all the cases the best performance are reached for $\alpha_{1,2} = (0, 1)$. As expected the more device are D2D capable the better

R [m]	D [m]	$\alpha_{1,2} = (0, 1)$	$\alpha_{1,2} = (0.5, 0.5)$	$\alpha_{1,2} = (1, 0)$
500	100	1.26	1.25	1.18
	300	1.24	1.19	1.10
	1000	1.26	1.20	1.09

Table 6.1: η for 100% D2D capable devices.

performance we get in terms of η. However we recall that η does not account for interference, thus Figure 6.6 and 6.7 are presented in the following to show the effect of interference.

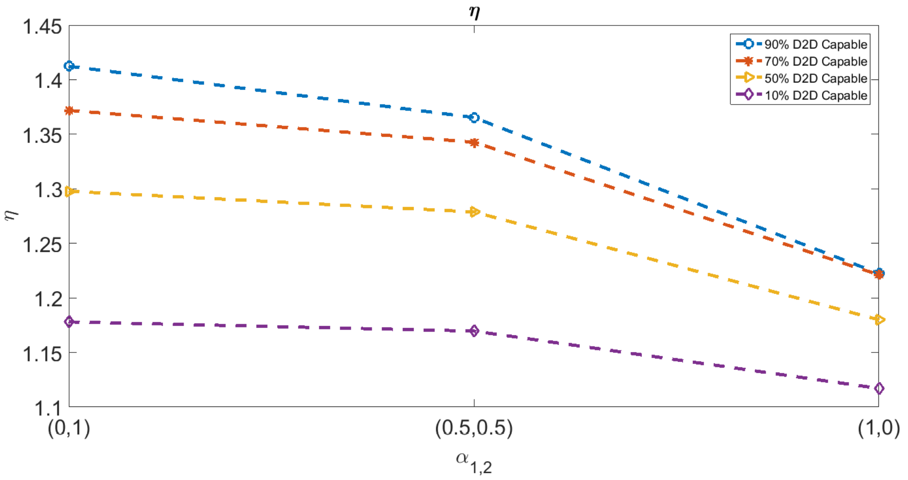

Figure 6.5: η for different $\alpha_{1,2}$ settings and for different percentages of D2D capable devices.

In Figure 6.6 it is shown that the assignment of downlink RBs is, in general, beneficial even accounting for interference. Indeed, η_{DWN} is always greater than 1 and it improves when we consider a higher percentage of D2D capable devices. Nonetheless, obviously, the values of η_{DWN} are worse than the values of η of about 0.1 because of interference. Moreover, notice that when only 10% out of the 100 devices are D2D capable, the higher value of η_{DWN} is obtained for the setting $\alpha_{1,2} = (1, 0)$ that is usually the worst case, as previously explained. This fact happens because having only 10% D2D capable devices, few D2D link are created, and what happen for $\alpha_{1,2} = (1, 0)$ is that relay and target tend to have a good channel, while there is no control on the link eNB-relay which is the real bottle neck. In this situation, a small improvement in throughput is obtained through D2D, but on the other hand D2D links create very little interference toward cellular users.

To study what happens in case of uplink RBs assignment for D2D communication, consider Figure 6.7. The first conclusion that we can draw by looking at the figure is that in general it is better to assign downlink RBs for D2D communication since the improvement w.r.t the non D2D case are higher. Not only, in Figure 6.7 there are values of η_{UP} lower than 1. Indeed, for the cases where the D2D capable devices are 50% or less, the performance degradation introduced

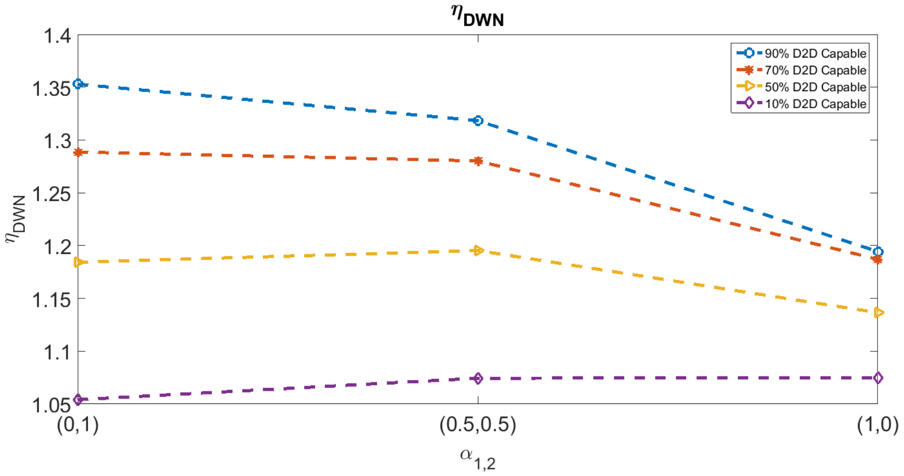

Figure 6.6: η_{DWN} for different $\alpha_{1,2}$ settings and for different percentages of D2D capable devices.

by the presence of interference is stronger than the positive effect introduced by the use of D2D communication and in such cases is better not to use D2D at all. Finally notice that using uplink RBs, the best result is achieved when 70% devices are D2D capable and not when they are 90% as before; again in this case the effect of interference is not negligible, even if the network throughput remains better than the case without D2D.

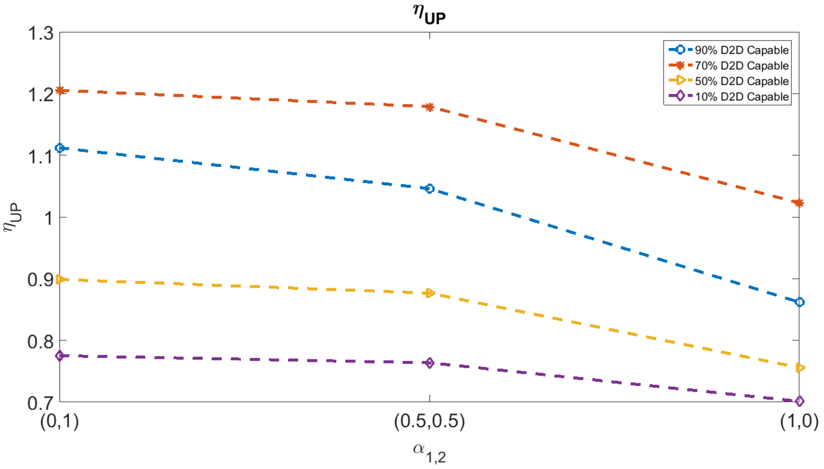

Figure 6.7: η_{UP} for different $\alpha_{1,2}$ settings and for different percentages of D2D capable devices.

6.8 Conclusions

In this section we modeled a cellular scenario evaluating the advantages of D2D activation on top of the cellular network. We stress that the scenario is tuned to the ones used for D2D within 3GPP RAN WG1. In particular, we propose a method for smart radio resource assignments through the selection of the right set of D2D links to activate. Interference is analyzed in case of radio resource reuse by D2D communication. The results clearly show the advantage of using D2D communications in terms of network throughput. It is also shown how the choice of a well designed scheduling strategy has a strong impact on the interference created in the network by the activation of D2D links.

Part IV

Femto-caching

Chapter 7

Femto-caching

7.1 Introduction

Current cellular networks struggle to keep up with the trends of data traffic growth. As reported in [1], by 2020, 30.6 exabytes per month will be generated and about 75% will be caused by video related traffic. Hence, it is widely agreed that overlapping layers of small cells (e.g. pico, femto) will be mandatory to improve the data rate experienced by the user from radio transmission. Undoubtedly, this densification imposes new challanges in the design of the backhaul that will receive significantly more traffic per m^2. Moreover, this traffic coming from the edge is usually received through links that are often wireless or in any case with limited capacity. The threat is for the backhaul to become a bottleneck, since it cannot provide enough capacity.
The idea that came out from researchers to cope with this issue is to bring the content closer to the end user, avoiding an access to the backhaul when not needed. Thinking about video related content, this task could be accomplished by equipping the femto cells (that are very close to the users) with a (femto-) cache devoted to store popular contents and eventually to serve directly a user requiring a content already stored in the cache, without the need of accessing the backhaul. From another point of view we could say that the link capacity is actually replaced with storage capacity at the edge of the network. Indeed, this paradigm would make possible to avoid an access to the backaul, provided that the required content is already stored at small cell level. One can argue that in any case it is necessary to download the content to store in the femto-cache, thus accessing the backhaul for this purpose. Even if this is true, since the time scale of the variation of a video content popularity is in the order of the hours if not of the days, it would make sense to download the contents during off-peak hours to reduce traffic during the peak hours.
If this was not enough to convince the reader about this approach, check the following general example. Consider a popular video on YouTube, and assume a very trivial policy that fills a femto-cache with the most popular contents. Since, as already said, the popularity of a content varies in hours or days, imagine how many times the same popular content is requested to a femto-cell. That said, if there is no femto-cache, each request corresponds to a backhaul access, if the femto-cache is present instead, all the requests can be simply served by the

femto-cell itself with the only burden of one single download from the backhaul. As presented the problem can look pretty simple. Indeed as far as we consider a single femto-cell non overlapping with other femto-cells, the problem is as simple as it seems. Unfortunately, we cannot say the same when there are areas with partially overlapping coverage of different femto-cells. One of the first works on this field was [58] which coined the term "Femto-Caching". The complexity stands in the fact that a user covered by multiple femto-cells has no " diversity-gain" if all the covering femto-cells store the same most popular contents. For instance (figure 7.1) a user covered by two femto-cell (storing M contents each) would maximize his gain if the stored contents are the $2M$ most popular contents. Indeed in this case the user will see an "effective memory" of $2M$ that is double the size of a case where each femto-cell store the most popular contents. The problem of femto-Caching calls for finding the best allocation of contents in the femto-caches, given a scenario where it is known the position of cells and users and the popularity of all the contents present in a given catalogue.

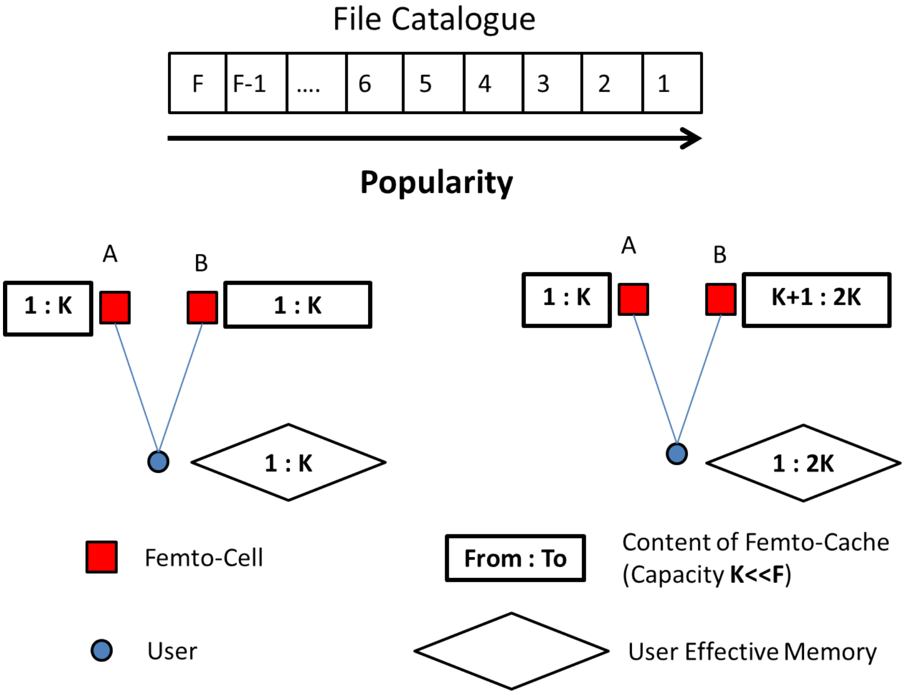

Figure 7.1: Femto-Caching Problem: example describing *"Effective Memory"*

Before entering in further details, it is important to stress which are the metrics of merit when talking about caching (see [59]). In general there exists two lines of works:

- **Femto-Caching**: aims to reduce backhaul traffic. The main metrics are (Cache-) Hits and (Cache-) Misses, that are the number of times a user DO/DOES NOT find the requested content already in the cache of a covering femto-cell. This line of work calls for finding the possibly optimal placement of contents in the caches.

- **Cache-Aided Communication**: aims to improve the transmission rate on the radio access channel neglecting the impact of Cache Misses. In this line of work several techniques, referred to as Coordinated Multi-Point (CoMP) transmission, are considered. Examples of CoMP transmission are MU-MIMO or Joint Transmission techniques.

In the following, only the Femto-Caching approach is considered.
This activity has been performed during a period abroad in France at the research center EURECOM.

7.2 Survey and Research Gap

In this section, I will describe the most important caching algorithm and policies already present in the literature. Then, I will underline what are the gaps not filled yet by researchers. Finally, I will outline the early work done on the topic complemented with preliminary results.

Belady's MIN Algorithm for Single Cache

This algorithm was proposed in 1966 in [60] by Belady, obviously for a different application, briefly described hereafter. Consider a computer program that has to execute instructions and that can execute them at no cost only if they are present in a cache (wich has finite size). Whenever a new instruction needs to be executed, it can be executed at no cost if it is already in the cache or it can be added to the cache, by paying a cost, provided that there is still space or that an instruction already present in the cache is removed and replaced by the new one. Hence, once the cache is full the problem is to understand *i*) if it worth it to store the new incoming instruction by removing another one already in the cache, and if yes, *ii*) which instruction have to be substituted with the new incoming one. The objective is to minimize the overall cost.
Now, if we consider a single femto-cache or several non overlapping femto-caches (that in fact is the same), the problem is the same where the instructions are the contents, the computer is the femto-cell trying to serve the user requests (alias an instruction that wants to be executed) and the cost is the cost of retrieving and storing the new requested content.
The Belady's MIN algorithm provides the optimal solution for the single cache case. Unfortunately, for the femto-Cache scenario the algorithm is unpractical since it requires the knowledge of the incoming request sequence from t_0 to infinity.
The idea of this algorithm is very simple: postpone as much as possible the next replacement in the cache. A pseudo code of the algorithm is given in 1. Unfortunately, is not easy at all to adapt this algorithm to a femto-caching scenario with possible multi-coverage of users.

7.2.1 Least Frequently Used (LFU)

Differently from the MIN algorithm, the LFU policy is a trivial policy that is applicable in practice. The idea is simply to keep track of the number of requests received per each content (cached or not), keeping in cache the most frequent contents. In particular, the cache is continuously updated as follows. Imagining

Algorithm 1 Belady's MIN Algorithm: Pseudo Code

```
loop {Process next request}
  $r \leftarrow$ content requested by incoming request
  $t \leftarrow$ time of the next request of $r$
  if Cache is not full then
    add $r$ to Cache
    go to loop
  end if
  $\epsilon = t, \quad s = \emptyset$
  for all $c \in$ Cache do
    $\tau \leftarrow$ time of the next request of $c$
    if $\tau > \epsilon$ then
      $\epsilon = \tau, \quad s = c$
    end if
  end for
  if $s = \emptyset$ then
    go to loop
  else
    Replace $s$ with $r$
  end if
end loop
```

the cache as a queue, the content in first position is the Most Frequently Used and all the following contents are sorted by decreasing popularity (number of request received for a content) till the last position that is occupied by the Least Frequently Used content. The name LFU, comes from the fact that the least frequently used content is the one to be removed from the cache upon arrival of a request for a more popular content (i.e. a content that received more request in the past). Obviously, after a learning phase, this policy simply stores in each cache the most popular contents. This is the reason why LFU is often used as a benchmark.
The two main problem of LFU are that:

- the concept of effective memory is not exploited,
- if the popularity of the contents in the catalogue varies over time, depending on the speed of this variation, the learning phase have to be repeated periodically. Solutions to this second problem were investigated by using techniques like sliding windows and others well known methods.

Notice that when the transient caused by the training phase typical of all the dynamic policies or algorithms is not the focus of the study, the LFU algorithm is often replaced by the *Most Popular* policy, that, exploiting the a priori knowledge of the contents popularity,simply stores in all the caches the most popular files. This is done due to the easy implementation of the *Most Popular* policy.

7.2.2 Single-Least Recently Used (S-LRU)

As for the LFU policy the S-LRU is a simple policy applicable in practice. In S-LRU, each content cached is labeled with the time-stamp of the last request

for that content, then upon arrival of a request for a not-cached content, this new content is surely inserted in the cache by removing the least recent content present in the cache. Similarly to LFU, the chache is continuously update by sorting from the most recent used content to the least recent one. The first position of the cache is called Most-Recent-Used position while the last is called Least-Recent-Used position.
The main advantages with respect to LRU are:

- the information to keep track of are only relative to contents present in the cache. There is no need to know anything about not-cached contents.
- S-LRU is less sensible to popularity variations, it will adapt seamlessly, without the need for windowing strategies.

Nonetheless, the concept of effective memory is still not exploited, each femto-cell works without any kind of collaboration with the others.

7.2.3 Spatial Multi-Least Frequently Used (M-LRU)

The main contribution of the M-LRU policy, proposed in [61] by A. Giovanidis and A. Avranas, is to modify the LRU policy to profit from the multi-coverage of users.
The main idea is that a user can check all the caches of covering BSs for the requested content, and download it from any one that has it in its cache. Hence, cache updates and content insertions can be done in a more efficient way than just applying single-LRU independently to all caches. In particular two versions of M-LRU are proposed (quoting [61]):

- **M-LRU-One**: Action is taken only in one cache out of the covering m. (a. Update) If the content is found in a non-empty subset of the m caches, only one cache from the subset is used for download and, for this, the content is moved to the Most-Recently-Used position. (b. Insertion) If the object is not found in any cache, it is inserted only in one while its Least-Recently-Rsed object is evicted. This one cache can be chosen as the closest to the user, a random one, or from some other criterion.
- **M-LRU-All**: Insertion action is taken in all m caches. (a. Update) If the content is found in a non-empty subset of the m caches, all caches from this subset are updated. (b. Insertion) If the object is not found in any cache it is inserted in all m. A variation based on q-LRU can be proposed, where the object is inserted in each cache with probability $q > 0$.

An example of M-LRU is presented in figure 7.2.

Notice that in [61] it is neglected the impact of the communication needed for this policy to work. In fact, some local communication between femto-caches covering the same user and between those femto-caches and the user itself is needed. Indeed, it is necessary to decide who is going to act as a server. Moreover, notice that the eviction/insertion actions are not controlled since the user position implies a certain action to be taken from the femto-caches. This means, that what happens inside the network is not controlled from network operators. Finally, the concept of effective memory is exploited only implicitly, indeed, the

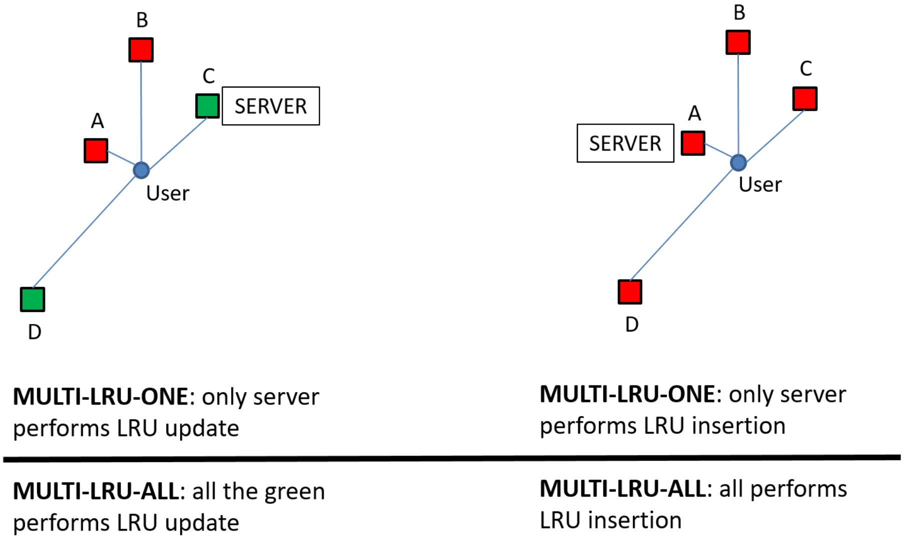
B
C
SERVER
A
User
D
B
C
A
SERVER
User
D
MULTI-LRU-ONE: only server performs LRU update
MULTI-LRU-ONE: only server performs LRU insertion
MULTI-LRU-ALL: all the green performs LRU update
MULTI-LRU-ALL: all performs LRU insertion

Figure 7.2: Example of M-LRU. A user covered by the four femto-cells A,B,C and D, issues a request for a content. On the left hand side it is represented a situation where only C and D have the required content. On the right hand side, no cell is caching the required content. In this example, the selected server is the closest.

effect of this policy is in the end to increase content diversity in a set of femto-cells covering a user. However, there is no communication between femto-cells to agree on which cache has to store a certain content, in order to maximize the Hit probability.

The results provided in [61], shows that for static popularity, M-LRU-ONE is always performing better than M-LRU-ALL. Moreover, when the average number of covering femto-cell per user is lower than a scenario dependent threshold, LFU has better performance; on the opposite, by increasing the density of femto-cells in order to have on average an higher number of covering femto-cells per user, M-LRU-ONE gets better than LFU.

7.2.4 Greedy Algorithm

The Greedy algorithm was proposed in 2012 in [62]. The power of this Algorithm is that it guarantees an allocation of files in femto-caches that provide an hit probability greater or equal to the half of the optimal one. However,the algorithm is unpractical because it has to be run by an oracle entity with full knowledge of: (i) relative position femto-cells - users (better described in the following), (ii) file catalogue, (iii)popularity of all the files and (iv) content of each femto-cache at each iteration of the algorithm. Moreover, this algorithm is not dynamic, in the sense that given a femto-cell distribution and a user distribution, one allocation of files in caches is found, but as soon as the user distribution, or the catalogue, or the file popularity change the provided allocation does not give guarantees anymore.

To formalize the problem and to give a description of the algorithm I will introduce some notation.

- $h = 1...H$ are the femto-cells.
- $u = 1...U$ are the users.
- $f = 1...F$ are the files in the catalogue of size F.
- $\mathcal{G}$ is the bipartite interaction graph where the nodes in the left side of the graph are femto-cells and the nodes in the right side of the graph are users. An edge e_{hu} exists between a femto-cell h and a user u if u is covered by h.
- $\mathcal{N}_u$ represents the femto-cells covering users u.
- C_h is the set of files cached in femto-cell h.
- $\mathcal{E}_u = \bigcup_{h \in \mathcal{N}_u} C_h$ is the effective memory seen by user u
- M is the femto-cell capacity in terms of number of files a cache can store.
- P_f popularity of file f.

  Given this definitions, the problem to solve to get the optimal solution for the femto-caching problem is the one described in equations (7.1)-(7.3).

$$\max \quad \Psi = \sum_{u=1}^{U} \sum_{f \in \mathcal{E}_u} P_f \tag{7.1}$$

Subject to

$$|C_h| \leq M, \qquad \forall h \tag{7.2}$$

$$\mathcal{A}_u = \bigcup_{h \in \mathcal{N}_u} C_h, \qquad \forall u \tag{7.3}$$

Unfortunately, as proven in [58], the problem is NP-Complete. However, it is possible to formulate the very same problem as the maximization of a submodular monotone function subject to matroid constraints [1]. The power of the new formulation is that classical results on approximations of submodular functions [63] established that the greedy algorithm provides solutions that achieve at least $\frac{1}{2}$ of the optimal value. Having said that, let me define Q as an $F \times H$ matrix where the element $Q_f^h = 1$ if file f is cached in femto-cache h and 0 otherwise. Indeed, the matrix Q is a representation of the state of all the femto-caches in the system. Finally, notice that the objective function $\Psi(\cdot)$(equation (7.1)) is actually function of Q and $\mathcal{G}$, that is $\Psi(Q, \mathcal{G})$. Since $\mathcal{G}$ is have to be invariant until the algorithm provides a solution, in the following the dependence on it will be omitted.
Algorithm 2, describe the Greedy algorithm for femto-caching.

[1] A detailed description can be found in [58]

Algorithm 2 Greedy Algorithm

```
**Initialize:**
$Q_f^h = 0 \quad \forall f, \forall h$
$i = 0$ {*comment:* iterations. An element of $Q$ is set per iteration.}
$\Psi(Q(i)) = 0\{Q(i)$ is $Q$ at the i-th iteration$\}$
$\Delta_{\text{best}} = 0, \quad f_{\text{best}} = \emptyset, \quad h_{\text{best}} = \emptyset$
**for** $(i = 1;\ i < F \times H;\ i++)$ **do** {*comment:* $F \times H$ iterations to fill $Q$.}
  $Q(i) = Q(i-1)$
  **for** $(h = 1;\ h < H;\ h++)$ **do**
    **for** $(f = 1;\ f < F;\ f++)$ **do**
      $Q_f^h(i) = 1$
      $\Delta_i = \Psi(Q(i-1)) - \Psi(Q(i)) \quad \{$*comment:*$\Delta_i \geq 0\}$
      **if** $(\Delta_i > \Delta_{\text{best}})$ **then**
        $\Delta_{\text{best}} = \Delta_i, \quad f_{\text{best}} = f, \quad h_{\text{best}} = h$
      **end if**
      $Q(i) = Q(i-1)$
    **end for**
  **end for**
  **if** $(f_{\text{best}} \neq \emptyset$ **and** $h_{\text{best}} \neq \emptyset)$ **then**
    $Q_f^h(i) = 1$
  **end if**
**end for**
```

In algorithm 2 you can see that at the beginning Q is empty, then at each iteration one file f is cached in a cache h ($Q_f^h(i)$ set to 1). The pair (f, h) to choose is the one increase the most Δ, which is the marginal value of the objective function. To conclude notice, that a part of the complexity of the algorithm is hidden behind Ψ. In fact at each iteration the calculation of Ψ is required, and it has to be done according to equation (7.1), which in turn requires an iteration over the U user and a further nested iteration over the effective memory of the user considered.

7.2.5 Exact Potential Game (EPG) Approach

The EPG approach was proposed in [64]. The main idea behind this approach is to (i) formulate the femto-caching problem as a game, (ii) show that the game is actually an Exact Potential Game, (iii) design an algorithm able to find Nash equilibria for the EPG. Having to deal with an EPG allows to exploit a couple of known results: firstly, the EPG has at least one Nash equilibrium and each of them is a feasible solution for the original problem, secondly, the best Nash equilibrium is an optimum for the original femto-caching problem.

The main problem of this approach stands in the fact that the EPG has **at least one** pure Nash equilibrium. Indeed, it is not possible to know a priori how many Nash equilibria the problem has. In fact, what is needed is the best Nash equilibrium, but what can be found running the proposed algorithm is the best *among the found solutions.* So, in principle if the algorithm runs a "sufficient" number of iterations finding "enough"[2] or all the solutions, it would

[2] The number of solution is "enough" if among them there is the optimal one

be possible to find the optimal solution, however in practice, it is impossible to know if “enough” or all the solutions were found. On the other hand the main value of this scheme is that, differently from the Greedy algorithm, it allows to find a solution in a completely distributed way, where each femto-cell, as a player of the game, takes decision on what to cache by itself, based on the local knowledge of the overlapping femto-cells caches. However, as the Greedy algorithm, the EPG algorithm should be re-ran every time the distribution of the user or the content popularity changes.
First of all, it is interesting to see how the femto-caching problem formulation is written in order to show that it can be considered as an EPG. Consider Figure 7.3 where the interaction graph is represented. In the figure E_1 represent the set of edges between users and femto-cells, while E_2 is the set of edges connecting a file f with femto-caches h if f is stored in h.

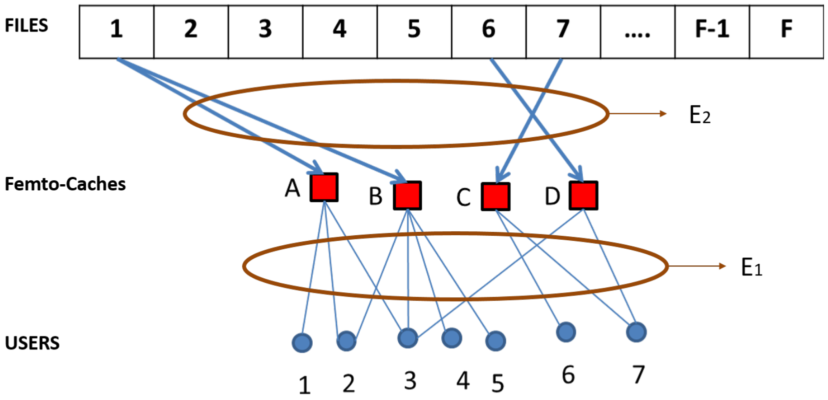

Figure 7.3: An example of interaction graph. To be precise, $\mathcal{G}$ as defined in section 7.2.4 and as it will be used in the following is the bipartite graph having femto-caches and users as vertices and E_1 as edges

Notice that E_1 is an input while E_2 is the solution to the problem. Defining

- $\mathcal{H}$ as the set of all the femto-caches,
- $x_{fh} = 1$ if f is stored in h and $x_{fh} = 0$ otherwise,
- $x_{hu} = 1$ if user u is covered by h and $x_{hu} = 0$ otherwise,

he problem can be written as:

$$\max_{x_{fh}} \quad \Psi = \sum_{u=1}^{U} \sum_{f}^{F} P_f \cdot \max_{h \in \mathcal{H}} \{x_{fh} x_{hu}\} \tag{7.4}$$

Subject to

$$\sum_{f}^{F} x_{fh} \leq M, \qquad \forall h \in \mathcal{H} \tag{7.5}$$

Notice that constraint (7.3) (expressing the concept of effective memory) in the original femto-caching problem formulation (described in section 7.2.4), is here embedded in the objective function (7.4), as $\max_{h \in \mathcal{H}}\{x_{fh}x_{hu}\}$. This smart formulation allows to prove that (7.4) can be considered as the potential function of an EPG. Giving the formal definition of EPG and of potential function would require the definition of concepts from game theory which is outside the scope of this dissertation, nonetheless hereafter are given enough information to understand what is an EPG and why/how it can by exploited to design the algorithm described at the end of this section. Recalling the definitions of matrix Q and of $\mathcal{G}$ from section 7.2.4, it holds $\Psi = \Psi(Q, \mathcal{G})$, where Ψ is the global objective function defined in (7.4). It is now defined also a local objective function:

$$\mathcal{L}_h = \sum_{u \in \mathcal{N}_h} \sum_{f}^{F} P_f \cdot \max_{h \in \mathcal{H}}\{x_{fh}x_{hu}\} \tag{7.6}$$

where $\mathcal{N}_h = \{u : (h, u) \in E_1\}$, is the set of users covered by femto-cell h. Notice that it holds $\mathcal{L}_h = \mathcal{L}_h(Q^h, Q^{O(h)}, \mathcal{G})$, where Q^h is the h-th column of matrix Q which describes the content of femto-cache h and $O(h) = \{h' : \exists u \in \mathcal{N}_h \cap \mathcal{N}_{h'}\}$ is the set of femto-cells overlapping with h (i.e. the set of femto-cells covering at least one of the users covered by h)[3].

In [64] it is proven that for all the possible allocation of files in femto-cache h it holds:

$$\mathcal{L}_h(\mathbf{s}) - \mathcal{L}_h(\widehat{\mathbf{s}}) = \Psi(Q) - \Psi(Q(h \rightarrow \widehat{\mathbf{s}})) \tag{7.7}$$

Where

- $\mathbf{s}$ is a binary vector of size F, representing a realization of Q^h.
- $\mathcal{L}_h(\mathbf{s})$ is the local utility of h when h stores the files marked by a 1 in $\mathbf{s}$.
- $Q(h \rightarrow \widehat{\mathbf{s}})$ indicates the matrix obtained form Q by replacing column h with a different realization $\widehat{\mathbf{s}}$ of the same column (simply means to change the files stored in femto-cahce h).

Having said that, equation (7.7) states that if an allocation of file in femto-cache h improves the local utility of femto-cache by a certain amount, also the value of the global objective function (7.4) increase of the very same amount. This condition is enough to state that the femto-caching problem is an EPG with potential function (7.4).

The very last definition before describing the EPG algorithm is the definition of *Best Response*. The Best Response of femto-cache h is the allocation of files in femto-cache h that maximizes the utility $\mathcal{L}_h$ of femto-cell h, given $Q^{O(h)}$. Since there could be more Best Responses for a given h and $Q^{O(h)}$, a Best Response set is referred to as $BR_h = \{\mathbf{s}_h^{br1}, \mathbf{s}_h^{br2}, ..., \mathbf{s}_h^{brn}\}$.

The EPG algorithm (shown in Algorithm 3), exploits the EPG properties by trying to maximize the local utility of a femto-cache. However, to avoid to get stuck in local minima, there is a probability following a Boltzmann distribution that a femto-cache changes its allocation from a good allocation or even from an optimal one to a worse one. In particular according to the Boltzmann

[3] Notice also that the dependence of $\mathcal{L}$ from $\mathcal{G}$ is more stringent than necessary. Indeed, just the knowledge of the subgraph of $\mathcal{G}$ induced by $\{h \cup \mathcal{N}_h \cup O(h)\}$ is needed.

Algorithm 3 EPG Algorithm: Pseudo Code

```
Initialize:
$i = 0$ {$i$ is the iteration: $Q(i)$ represents $Q$ at the $i$-th iteration}
$Q(0)$ is known i.e. $\mathbf{s}_h(0) \quad \forall h \in \mathcal{H}$ is known
Iteration:
for each $i = 1, 2, 3, ..., \text{MAXLOOP}$ do
    1. Selcet Players: randomly select an Indipendent Set $\mathcal{K}(i)$ of non overlapping femto-cells
    2. Explore New Strategy:
    for all $h \in \mathcal{K}(i)$ do
        - calculate $\mathcal{L}_h(\mathbf{s}_h(i))$ by necessary communications with overlapping femto-cells
        - find $BR_h$
        - pick a new candidate file allocation $\widehat{\mathbf{s}}_h(i)$ as follows:
        a) if $\mathbf{s}_h(i) \notin BR_h \;\Rightarrow\; \widehat{\mathbf{s}}_h(i)$ is randomly selected in $BR_h$
        b) if $\mathbf{s}_h(i) \in BR_h \;\Rightarrow\; \widehat{\mathbf{s}}_h(i)$ is randomly selected among whatever possible allocation.
        - calculate $\mathcal{L}_h(\widehat{\mathbf{s}}_h(i))$ using the local knowledge of $Q(0)$ and of $\mathcal{G}$
        3. Update Strategy as follows:
        $\begin{cases} \text{Prob}\{\mathbf{s}_h(i+1) = \widehat{\mathbf{s}}_h(i)\} = \frac{\exp\{\beta\mathcal{L}_h(\widehat{\mathbf{s}}_h(i))\}}{\Phi} \\ \text{Prob}\{\mathbf{s}_h(i+1) = \mathbf{s}_h(i)\} = \frac{\exp\{\beta\mathcal{L}_h(\mathbf{s}_h(i))\}}{\Phi} \end{cases}$
        Where $\Phi = \exp\{\beta\mathcal{L}_h(\widehat{\mathbf{s}}_h(i))\} + \exp\{\beta\mathcal{L}_h(\mathbf{s}_h(i))\}$, and $\beta$ is a learning parameter.
    end for
end for
```

distribution, the higher the value of the local utility provided by the current allocation, the lower is the probability to change it. The parameter β, balances the trade-off between extensive search space and convergence. An higher β will boost the algorithm toward an optimum, but it will make it harder to escape from that optimum and obviously this is a positive fact only if the optimum found is not a local one.

7.3 Research Gap

The idea of using caches directly placed at the edge is nowadays studied under multiple point of views and for different applications as described. However, usually the study of a policy or of an algorithm for content placement considers only cache hit as metric to measure performance or to perform comparisons. Actually, it is important to notice that every time a cache is updated, an access to the backhaul needs to be performed in order to download the required content. Thus, since one of the objectives of femto-caching is to soften the load on the backhaul, it is important to consider the traffic generated on the backhaul by the caching policy. To give a simple example, LRU policy needs to update the cache, by downloading a content from the backhaul, every time there is a cache miss. On the other hand, the LFU policy access the backhaul only when there is a cache miss causing, in the required file popularity, a change which is big enough to become more popular than a file already cached. In fact, after the learning phase, LFU will access rarely the backhaul (unless there is a change in the overall file popularity distribution).

A second point to stress is that it is not possible yet to know how close a policy is to the optimal. Obviously, finding an algorithm to get the optimal solution would be a disruptive result. Actually, even if it was not possible to find the optimal solution, it would be a great result to find a way to get the value of the optimal solution in order to have an upper bound for comparison even if only in terms of hit probability.

Finally as underlined from table 7.1, all the dynamic policies are actually very simple and they do not exploit the concept of effective memory, as well as some sort of local coordination. As a consequence, it seems fairly reasonable to try to fill this gap with a dynamic policy smarter than those presented in the previous sections.

Policy	**Dynamic**	**Distributed**	**Optimality**
MIN	**Y**	**Y**	**Optimal (Single Cache)**
LFU	**Y**	**Y**	/
S-LRU	**Y**	**Y**	/
M-LRU	**Y**	**Y**	/
Greedy	**N**	**N**	$\geq 1/2$ **Optimal**
EPG	**N**	**Y**	**Optimality not granted**

Table 7.1: Caching Algorithms Comparison

7.3.1 Proposed Policy and Early Results

In this section, I will present some preliminary results I have obtained by working on the topic of femto-caching. In particular, the first result is of theoretical nature, while the second is about the design of a dynamic version of the EPG policy.

Theorem on Popular File Caching

As it should be clear by now, one of the main issues when dealing with the femto-caching problem is the complexity due to the nature of the problem itself and to the big numbers involved. As example, the file catalogue can be the catalogue of all the videos on YouTube rather than the contents present on a video streaming platform like Netflix. Hence, we are talking about thousands or millions of contents. The consequence is highlighted by the Greedy and the EPG Algorithms, where an iteration on all the files present in the catalogue is needed in order to check the improvement brought by adding or not a certain file. Intuitively, one could guess that it is useless to check whether to cache or not a file with low popularity, in order to reduce the problem complexity. The next theorem embraces this intuition and formalizes it.

Theorem 1 (Theorem on Popular File Caching)
Given an instance $\widetilde{\mathbf{Q}}$ of $\mathbf{Q}$, if $\widetilde{\mathbf{Q}}$ is an optimal allocation of files in caches for a given interaction graph $\mathcal{G}$, then any femto-cache h stores files taken from the set of the $M(1 + |O(h)|)$ most popular files, where M is the cache size of each

femto-cache and $|O(h)|$ *represents the number of femto-caches overlapping with* h*.*

In other terms, defining $A = \{f : 1 \leq f \leq M(1 + |O(h)|\}$ $, itholds : \widetilde{\mathbf{Q}} : \Psi(\widetilde{\mathbf{Q}}) = \max_{\mathbf{Q}} \Psi(\mathbf{Q}) \Rightarrow \widetilde{Q}_h \underset{M}{\subseteq} A \quad \forall h = 1, ..., H(7.8)$

The proof of this theorem is given in Appendix A.

Dynamic EPG Policy

In this section I will present a dynamic version of the EPG policy (DynEPG), presenting some preliminary results obtained till now.
Consider the typical scenario for the femto-cell problem:

- Users are distributed in according to a Poisson Point Process with density λ_u.
- Femto-cells are distributed in according to a Poisson Point Process with density λ_f.
- Femto-cells are equipped with caches of memory M and their covers an circular area with radius R referred to as transmission range.
- A file catalogue composed of F files is generated. Each file has a popularity assigned according to a Zipf Distribution.
- Time is discretized in time units. At each time unit N uniformly randomly chosen users generates a request.
- The content requested by users are generated according to file popularity i.e. a more popular file has higher probability to be requested.

In particular I define as $Zone(Z)$ a partition of the considered area such that any user inside that partition is covered by the same set of femto-caches. An example is given in Figure 7.4.

Notice that we can now talk about Effective Memory associated to a Zone. Indeed, any user placed in the considered Zone can access the same Effective Memory, intended as the set of all the contents cached by the femto-cells associated to the considered Zone.

Communication Protocol For the DynEPG an important part is the communication protocol that will allow femto-cells to dynamically acquire local information useful to take caching decisions. Figure 7.5 shows the communication protocol.

As you can see from Figure 7.5, once a user broadcasts a content request, all the covering femto-cells answer to the user with a packet called Request To Send (RTS) packet. The RTS packet contains the ID of the transmitting femto-cell, plus an acknowledgement (ACK) or a negative acknowledgement (NACK) if the considered femto-cell have already the requested file or not, respectively. Then, the user, upon reception of all the RTS packets, will create a Clear To Send (CTS) packet. The CTS packet is composed of all the RTS packets received, plus an additional field containing an indication on which femto-cell is designated as the server (depending on whatever metric: distance, channel quality,...). Once

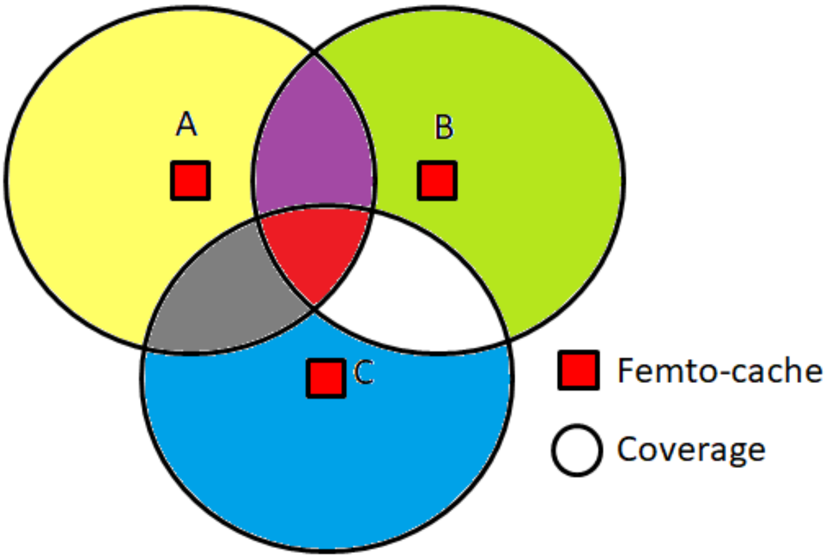

Figure 7.4: Example of Zones generated by three overlapping femto-cells. Different colors indicate different zones.

the CTS is ready, it is transmitted in broadcast by the user. Firstly, all the femto-cells receiving the CTS will know who has to serve the user. Secondly, they will learn some useful information (specified in the following) about the Zone from which the CTS was received. Indeed, notice that by definition, all the femto-cells receiving the user request and the CTS are involved in the same Zone. Notice that for the Multi-LRU policy to work, it is required the same communication protocol with the only exception that the CTS need to contain only the server field.
Thanks to this communication protocol, after a training phase, a femto-cell learns:

- All the Zones where it is involved, by checking for each user request the relative CTS and in particular the set of femto-cells present in the CTS.
- The density of user per each Zone where it is involved, by counting the number of incoming request from each Zone.
- The file catalogue and the files popularity (exactly as it happens in LFU, i.e. by observing and counting the incoming requests).
- A guess on the Effective Memory related to each Zone where it is involved (described in the following).

Guessing the Zone Effective Memory

The core idea of this policy is to try to exploit some local knowledge to infer the Zone effective memory for each Zone. To do that each femto-cell takes advantage of the communication protocol described. To understand the mechanism consider one femto-cell, which keeps track of the effective memory of each zone

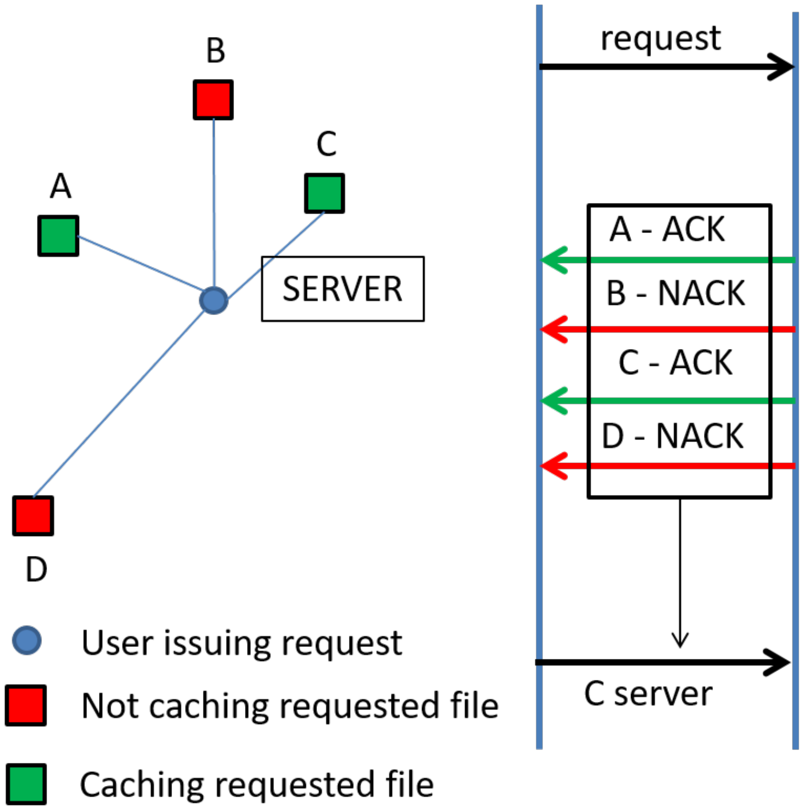

Figure 7.5: Communication Protocol for DynEPG.

it is involved in. Once an incoming user request triggers the communication previously described, the considered femto-cell waits for the CTS. Then, by analyzing the CTS, it detects the Zone from which the request is coming and it updates the Effective memory for that Zone as follows:

The policy will be described in detail in the following, however, let me anticipate that a femto-cell decides on what files to cache or evict depending on the Effective Memory of the Zones where it is involved. In fact, assume that a femto-cell C stores a guess about a certain Zone Z, it can happen that, by the time a new request comes from Z, some of the femto-cell involved in Z have changed their cache allocation because of what happened in other Zone where they are involved. As consequence, the guess of C on Z can be wrong and for sure obsolete. This motivate the statement in line 6 of the previous algorithm and it is the reason why we talk about guess rather than knowledge of the Effective Memory associated to a Zone.

DynEPG Policy and Results

The DynEPG policy starts from the fact that, now that we have defined the concept of Zone, we can rewrite the local Utility function 7.6 in the form:

Algorithm 4 DynEPG Zone Effective Memory update

```
Incoming request for file $f$ from Zone $Z$
if (CTS contains at least one ACK) then
    add $f$ to $Z$ Effective Memory
else
    if ($f$ already present in $Z$ Effective Memory) then
        remove $f$ from $Z$ Effective Memory
    end if
else
    do nothing
end if
```

$$\mathcal{L}_h = \sum_{u \in \mathcal{Z}_h} \sum_{f}^{F} P_f \cdot x_{fz}^{h} \tag{7.9}$$

Where $\mathcal{Z}_h$ indicates the set of all the Zones where the h-th femto-cell is involved and x_{fz} is a binary value equal to 1 if file f is present in the Effective Memory associated to Zone z as guessed by femto-cell h.
With this new definition of the local utility, every time a new request comes it triggers all the receiving femto-cells that will start the communication protocols together with the updates of the guess on the Zone Effective Memory. Then, once the server is selected by the CTS packet, the server frmto-cell will run the EPG algorithm described in Algorithm 3 using as local utility function the one described in equation 7.9. Notice, that there is non need of direct communication between femto-cells to discover the memory allocation of neighbouring femto-cells, as instead it is required for plain EPG. Indeed, all the information needed are the one obtained by overhearing CTS packets.

Preliminary Results Table 7.2 shows the parameters of the scenario considered to analyze performances of DynEPG policy. In particular, two relevant scenarios are considered by setting the mean number of femto-cells covering a user (Mean Femto-cell To User Ratio - $MFUR$) to 3.23 for the first scenario and to 7 for the second one.

$MFUR$	$\{3.23, 7\}$
F/M	100
Zipf Exponent	0.8
Requests/TimeUnit	20

Table 7.2: Scenario Settings

Figures 7.6 and 7.7 present a comparison between some of the caching policies described and the DynEPG in terms of Hit Probability and they are related respectively to $MFUR = 3.23$ and $MFUR = 7$. As you can see from Figure 7.6, for a low $MFUR$ the Pop algorithm outperforms M-LRU-ONE by about 10%. In the same case, also DynEPG works better than M-LRU-ONE but the improvement with respect to Pop is far from being relevant. On the other hand, the results achieved by the benchmarks Greedy and EPG (very close to each other) are still much better than all the dynamic policies shown.

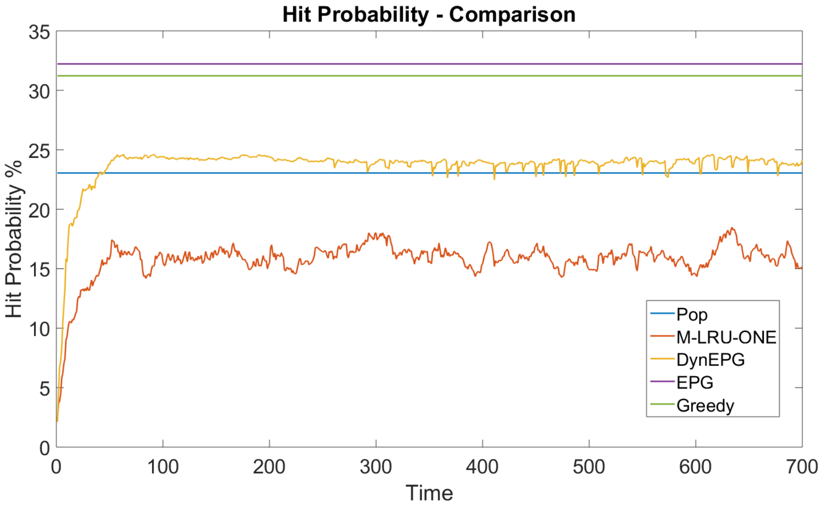

Figure 7.6: $MFUR = 3.23$ - Comparison between caching polices in terms of Hit Probability.

Figure 7.7 considers a case where the $MFUR$ is higher, and as expected the Pop policy is worse that the M-LRU-ONE. Moreover, while the DynEPG policy can still match the performance of M-LRU-ONE, the performance of the Greedy and of EPG policies (also in this case very close to each other) are much better.

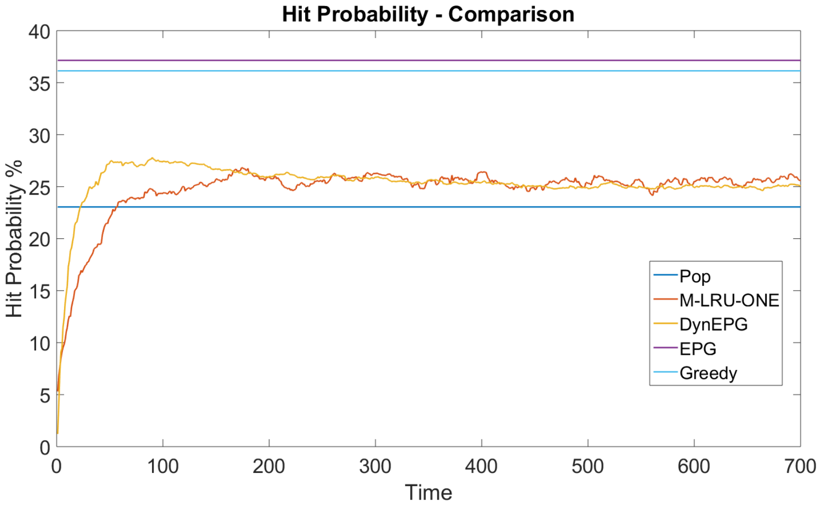

Figure 7.7: $MFUR = 7$ - Comparison between caching polices in terms of Hit Probability.

7.3.2 Conclusions

Unfortunately, because of the approaching of the end of my Ph.D., I could not investigate more on this topic. At the best of my knowledge, there is no dynamic policy able to outperform Pop and M-LRU-ONE by a sensible margin in regime of high $MFUR$ and low $MFUR$ respectively. However, the gap between the analyzed dynamic policies and the EPG or Greedy algorithms is such to let researchers believe that there should be some dynamic policy exploiting some local information and collaboration leading to better results.

Part V

Conclusions

Chapter 8

Conclusions

Current Radio Networks were designed and improved during the years to provide the services and the performance required by the users and the markets. In fact, regardless the progressively better technologies, the aim of the design was mainly one: provide more coverage and more capacity getting closer and closer to the Shannon limit. On one hand, this approach translates every request of service in request of data with no attention to the kind of data required. On the other end, the deployment of the network in terms of Base Stations and related hardware has always been static resulting in the impossibility for the networks to be ubiquitous by serving users wherever and whenever seamlessly.

Nowadays more than ever, some killer applications related to video streaming (e.g. Youtube, Netflix, and all the social networks in general), accounts for the majority of traffic generated. At the same time, other relevant type of application are related to Machine to Machine and to IoT or to Industry 4.0 in general. This kind of data traffic, is extremely different from video related traffic, thus, it should be treated differently to exploit its characteristics by designing cellular networks accordingly.

Connectivity and internet access is given as granted. It is common to get upset when we are not able to use some applications on our smart phones because of lack of connectivity. The fact is that connectivity is not seen anymore as something new that is desirable to have, but it is seen as a must, consequently the exception nowadays is not to have it. The good news is that this means that engineers have done a good job making radio connectivity pervasive and indispensable. The other side of the coin is that, as of today, capacity is expected even in places and situation where it is hard to bring connectivity, e.g. rural areas, disaster situation, war zones.

In this dissertation and during my Ph.D., I proposed ideas and solutions to face these problems through approaches that are parallel to the traditional ones.

In particular, I studied Delay Tolerant Networks through experimental approach and I designed routing protocol evaluating performance through simulations. I have also studied the use of Unmanned Aerial Vehicles in radio networks, first in DTNs and then by a theoretical approach in order to optimize the UAV trajectory to serve as many users as possible in a cellular network. DTNs, and the use of UAVs in radio networks, meet the need for more flexible and ubiquitous networks.

The other line of my research had the aim of improving network capacity with a particular care on the characteristics of the data traffic to be satisfied. In particular the work on femto-caching exploits as the characteristic for video contents the inherent popularity. Also Device to Device communication has been looked at by considering relay and target as two users requesting the same content (so again popularity) from the Base Station and cooperating to improve capacity.

In all the works done, it was possible to increase network performance showing that next generation networks should be more than just a capacity increase.

Finally as last recap, my work started from an experimental characterization of sociality for DTN, moving to the routing protocol design for the same kind of network in a realistic environment (namely the city center of Bologna). After, it was also analyzed the impact of using a UAV in a similar DTN scenario. In order for a DTN to work, relaying of data has to be performed between devices, this is the logic that made me curious about D2D communication, even if, in fact, my work about D2D is far from the DTN world.

Motivated by the work of UAV in DTN scenario, I have studied through operational research tools, a mathematical framework to optimize UAV route.

Aside from all these topic, I have also started a work about femto-caching for which I have shown only preliminary results due to the approaching end of my Ph.D.

Chapter 9

My Two Cents

To conclude this dissertation and this three years of Ph.D., I would like to give my view about my expectations on the future of radio networks and telecommunications more in general. Participating to several conferences in these years made clear to me that it is ongoing a shift from telecommunications to other adjacent areas. The last conferences I have participated to, in principle about telecommunications, was actually talking about artificial intelligence rather than big data or data mining or in other cases, about optimization of existing telecommunication concepts through tools outside the telecommunication world. This fact made me think that telecommunications as we know it is going toward an end for a simple reason: the network simply works. In the next 2 generations of cellular networks (15 to 20 years) the network will be completely “fluid”, in the sense that all the infrastructures will consist only of peaces of general purpose metal programmed with really smart self-adapting software. In such a way, the network itself will be able to readapt to the traffic demand condition when there is the need to do it; what we will need to do is just maintenance and few things more. Even if I depicted a scenario that is not very comforting for us as telecommunication engineers, this sounds to me as what was called the “telecommunication engineers dream” during the first hour of my bachelor course.

Nonetheless, there are topics that I think were not explored enough. As an example, one is based (again) on a work of Shannon. One year after his seminal paper about communication theory, Shannon wrote, in 1949, another paper regarding communication under adversarial impact [65]. The communication theory community has always cared about physical constraints dictated by nature (e.g. noise or unintended interference etc.). However, it has paid much less attention to security and optimization of communication systems against impact of adversaries. I strongly believe that in the years to come the main problems in communication will be related to security. To give a self-explanatory example think about cryptocurrencies like bitcoin. Notice that in this field it will not be possible to reach a saturation point for research for a very simple reason: the adversary is doing his own research to counteract our countermeasures.

Moreover, there are fields where telecommunication expertise could be of great use, in particular all that disciplines where it is possible to identify a transmitter, a receiver, and to statistically characterize the channel. An example of that is the work proposed in [66], were the authors were able to send information by modulating the power electronic components, such that a distributed power system (as e.g. microgrid) can operate and exchange messages without relying on external communication systems. In other words, communication without using dedicated electronic circuits. Other emerging examples of this concepts are related to the recently established areas of molecular communications and nano-networks.

Appendix A

Appendix

A.0.1 Theorem on Caching

Definitions:

- $h = 1, ..., H$ denotes femto-caches.
- $u = 1, ..., U$ denotes users.
- $f = 1, ..., F$ denotes files.
- P_f is the popularity of file f.
- $\mathcal{F} = [1, 2, ..., F]$ is a vector of size F describing the file catalogue. Note that the vector is sorted in decreasing order of popularity of the files, that is: $if\ P_{f'} > P_{f''} \Rightarrow f' < f''$.
- M denotes the capacity of each femto-cache, namely, the number of files it can cache.
- $\mathbf{Q}$ is a vector of size H whose elements are sets Q_h of size M. The h-th element of $\mathbf{Q}$ is Q_h and it represents the content of the h-th femto-cache.
- The operator $\underset{\mathrm{M}}{\subseteq}$ denotes "subset of size M".
- The operator $O(h)$ denotes the set of femto-caches that cover at least one of the users covered by cache h.
- $\mathcal{G}$ is the interaction graph as defined in chapter IV.
- $\Psi(\mathbf{Q}, \mathcal{G}) = \sum_{u=1}^{U} \sum_{f=1}^{F} P_f x_{fh}$, where x_{fh} is a binary variable such that $x_{fh} = 1$ if file f is cached in femto-cache h and $x_{fh} = 0$ otherwise. In the following, since $\mathcal{G}$ is given and invariant, the dependence on it will be omitted.

Theorem 2 (Theorem on Popular File Caching)
Given an instance $\widetilde{\mathbf{Q}}$ of $\mathbf{Q}$, if that is an optimal allocation of files in caches for a given interaction graph $\mathcal{G}$, then any femto-cache h stores files taken from the set of the $M(1 + |O(h)|$ most popular files.

In other terms, defining $A = \{f : 1 \leq f \leq M(1 + |O(h)|)\}$*, it holds :*

$$\widetilde{\mathbf{Q}} : \Psi(\widetilde{\mathbf{Q}}) = \max_{\mathbf{Q}} \Psi(\mathbf{Q}) \Rightarrow \widetilde{Q}_h \underset{M}{\subseteq} A \quad \forall h = 1, ..., H \tag{A.1}$$

Proof. Assume that $\widetilde{\mathbf{Q}}$ is a vector representing an optimal allocation, i.d.
$\widetilde{\mathbf{Q}} : \Psi(\widetilde{\mathbf{Q}}) = \max_{\mathbf{Q}} \Psi(\mathbf{Q})$.
It is defined $\widehat{\mathbf{Q}}$ as: $\widehat{\mathbf{Q}} = \widetilde{\mathbf{Q}}(\widetilde{Q}_h \rightarrow \widehat{Q}_h)$ where the notation $\widetilde{\mathbf{Q}}(\widetilde{Q}_h \rightarrow \widehat{Q}_h)$ denotes the modified vector $\widetilde{\mathbf{Q}}$ where the element $\widetilde{Q}_h$ is replaced by the set $\widehat{Q}_h$.
We can write:

$$\left.\begin{aligned} \Psi(\widetilde{\mathbf{Q}}(\widetilde{Q}_h \rightarrow \emptyset)) + \Delta\widetilde{Q}_h = \Psi(\widetilde{\mathbf{Q}}) \\ \Psi(\widehat{\mathbf{Q}}(\widehat{Q}_h \rightarrow \emptyset)) + \Delta\widehat{Q}_h = \Psi(\widehat{\mathbf{Q}}) \end{aligned}\right\} \quad \Rightarrow \quad \boxed{\Delta\widetilde{Q}_h \geq \Delta\widehat{Q}_h} \tag{A.2}$$

Note that in equation A.2 holds since $\Psi(\widetilde{\mathbf{Q}}) \geq \Psi(\widehat{\mathbf{Q}})$ by construction. Moreover, in equation A.2 equality holds if:

- $\widetilde{Q}_h = \widehat{Q}_h$, but this is impossible by construction;
- $\Psi(\widetilde{\mathbf{Q}}) = \Psi(\widetilde{\mathbf{Q}}(\widetilde{Q}_h \rightarrow \emptyset))$, that is if $\widetilde{Q}_h$ does not bring any improvement, however, in that case $\Psi(\widehat{\mathbf{Q}}) = \Psi(\widetilde{\mathbf{Q}}) = \Psi(\widetilde{\mathbf{Q}}(\widetilde{Q}_h \rightarrow Q_h))$ for any Q_h. Hence, it is always possible to choose $\widetilde{Q}_h \underset{\mathrm{M}}{\subseteq} A$ satisfying the theorem.

Defining

- $\mathcal{N}_h$ as an operator denoting the set of users covered by h
- $\mathcal{N}_u$ as an operator denoting the set of femto-caches covering user u.
- $\omega_{u,h} = \{\bigcup_{h^* \in \{\mathcal{N}_u \setminus h\}} \widetilde{Q}_{h^*}\}$ as the effective memory of user u as if it was not covered by femto-cache h. In other words, given a femto-cache h covering a user u, $\omega_{u,h}$ is the set of files cached by all the femto-caches covering u but h.

The values of $\Delta\widetilde{Q}_h$ and $\Delta\widehat{Q}_h$ can be written in the form:

$$\begin{aligned} \Delta\widetilde{Q}_h &= \sum_{u \in \mathcal{N}_h} \sum_{f \in \{\widetilde{Q}_h \setminus \widetilde{\omega}_{u,h}\}} P_f \\ \Delta\widehat{Q}_h &= \sum_{u \in \mathcal{N}_h} \sum_{f \in \{\widehat{Q}_h \setminus \widehat{\omega}_{u,h}\}} P_f \end{aligned}$$

Notice that we do not multiply by x_{fh} because the nested sum is iterated over a set of files that by definition are cached in h, hence $x_{fh} = 1$ for all the files in the set.
Now, I am going to construct a $\widetilde{\mathbf{Q}}$ contradicting the thesis of the theorem (expression A.1), showing that in this case an absurd is reached by violation of equation A.2.

BY CONTRADICTION:

$$\widetilde{Q}_h = \{\widetilde{Q}'_h \bigcup \widetilde{Q}''_h\} \quad \text{where} \quad \begin{cases} \widetilde{Q}'_h \underset{\epsilon}{\subset} A \\ \widetilde{Q}''_h \underset{\mathrm{M}-\epsilon}{\subset} \{\mathcal{F}\} \setminus A \end{cases} \tag{A.3}$$

In equation A.3 the parameter ϵ is any number smaller that M-1, namely $\epsilon \in \{0, ..., \mathrm{M}\}$.
Now, given this generic construction for $\widetilde{Q}_h$, I am going to prove that it is always possible to find an alternative allocation of files in helper h, namely $\widehat{Q}_h$ such that equation A.2 is contradicted.

$$\widehat{Q}_h = \{\widehat{Q}'_h \bigcup \widehat{Q}''_h\} \quad \text{where} \quad \begin{cases} \widehat{Q}'_h = \widetilde{Q}'_h \\ \widehat{Q}''_h \underset{\mathrm{M}-\epsilon}{\subset} A \setminus A^*_h \end{cases} \tag{A.4}$$

Note that $A^*_h = A \setminus \left\{ \widehat{Q}'_h \bigcup_{h^* \in O(h)} Q_{h^*} \right\}$.

To construct $\widehat{Q}''_h$, the condition $|A^*_h| \geq \mathrm{M} - \epsilon$ have to be satisfied, and in fact in the worst case it holds:

$$|A^*_h| = |A| - (\epsilon + \mathrm{M}|O(h)|) = \mathrm{M} + \mathrm{M}|O(h)| - \epsilon - \mathrm{M}|O(h)| = \mathrm{M} - \epsilon$$

It is now possible to write:

$$\Delta\widetilde{Q}_h = \sum_{u \in \mathcal{N}_h} \left(\sum_{f \in \{\widetilde{Q}'_h \setminus \omega_{u,h}\}} P_f + \sum_{f \in \{\widetilde{Q}''_h \setminus \omega_{u,h}\}} P_f \right) \tag{A.5}$$

$$\Delta\widehat{Q}_h = \sum_{u \in \mathcal{N}_h} \left(\sum_{f \in \{\widehat{Q}'_h \setminus \omega_{u,h}\}} P_f + \sum_{f \in \{\widehat{Q}''_h \setminus \omega_{u,h}\}} P_f \right) \tag{A.6}$$

Since $\widetilde{Q}'_h = \widehat{Q}'_h$:

$$\Delta\widetilde{Q}_h - \Delta\widehat{Q}_h = \sum_{u \in \mathcal{N}_h} \left(\sum_{f \in \{\widetilde{Q}''_h \setminus \omega_{u,h}\}} P_f - \sum_{f \in \{\widehat{Q}''_h \setminus \omega_{u,h}\}} P_f \right)$$

By definition of A, it holds $P_f \geq P_{f'}$, $\forall f \in A$, $\forall f' \in \{\mathcal{F}\} \setminus A$. In conclusion, by construction of $\widehat{Q}''_h$ and $\widetilde{Q}''_h$, since $|\widehat{Q}''_h| = |\widetilde{Q}''_h|$:

$$\sum_{f \in \{\widetilde{Q}''_h \setminus \omega_{u,h}\}} P_f \leq \sum_{f \in \{\widehat{Q}''_h \setminus \omega_{u,h}\}} P_f \quad \Rightarrow \quad \boxed{\Delta\widetilde{Q}_h \leq \Delta\widehat{Q}_h} \tag{A.7}$$

Equation A.7 contradicts equation A.2, except for the case of equality which happens only if $\epsilon = \mathrm{M} \Rightarrow \widetilde{Q}_h = \widehat{Q}_h$.

□

Bibliography

[1] CISCO. C.V. Forecast. *"Cisco visual networking index: Global mobile data traffic forecast update 2015-2020"*, 2016. February.

[2] V. Chandrasekhar, J. G. Andrews, and A. Gatherer. Femtocell networks: a survey. *IEEE Communications Magazine*, 46(9):59–67, September 2008.

[3] Qualcomm. [online]. `http://www.qualcomm.com/media/documents/wireless-networks-1000x-more-spectrum-especially-small-cells`.

[4] Qualcomm. [online]. `http://www.qualcomm.com/media/documents/wireless-networks-1000x-more-small-cells`.

[5] Laura Galluccio, Beatriz Lorenzo, Savo Glisic, Chiara Buratti, Colian Giannini, and Roberto Verdone. Epidemic information dissemination in opportunistic scenarios: a realistic model. In *2015 European Conference on Networks and Communications: Track 7: Special Sessions (EuCNC 2015 - Track 7- Special Sessions)*, pages 636–640, Paris, France, June 2015.

[6] L. Galluccio, C. Giannini, B. Lorenzo, S. Glisic, C. Buratti, and R. Verdone. Epidemic information dissemination in opportunistic scenarios: a realistic model obtained from experimental traces. In *2015 International Symposium on Wireless Communication Systems (ISWCS)*, pages 381–385, Aug 2015.

[7] C. Giannini, A. A. Shaaban, C. Buratti, and R. Verdone. Delay tolerant networking for smart city through drones. In *2016 International Symposium on Wireless Communication Systems (ISWCS)*, pages 603–607, Sept 2016.

[8] C. Giannini, P. Calegari, C. Buratti, and R. Verdone. Delay tolerant network for smart city: Exploiting bus mobility. In *2016 AEIT International Annual Conference (AEIT)*, pages 1–6, Oct 2016.

[9] C. Giannini, C. Buratti, and R. Verdone. Milp-based radio resource assignment for device to device communications. In *2017 29th International Teletraffic Congress (ITC 29)*, volume 2, pages 1–6, Sept 2017.

[10] C. Giannini, C. Buratti, , V. Cacchiani, and R. Verdone. Maximizing The Number of Served Ground Users From Unmanned Aerial Base Stations. IEEE *Transaction on Vehicular Technology, March 2018 [submitted]*

[11] D. Karamshuk, C. Boldrini, M. Conti, and A. Passarella. Human mobility models for opportunistic networks. *IEEE Communications Magazine*, 49(12):157–165, December 2011.

[12] Xiaolan Zhang, Giovanni Neglia, Jim Kurose, and Don Towsley. Performance modeling of epidemic routing. In *Proceedings of the 5th International IFIP-TC6 Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, NETWORKING'06, pages 827–839, Berlin, Heidelberg, 2006. Springer-Verlag.

[13] B. Lorenzo, S. Glisic, L. Galluccio, and Y. Fang. Adaptive infection recovery schemes for multicast delay tolerant networks. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 4420–4426, Dec 2013.

[14] T. Karagiannis, J. Y. Le Boudec, and M. Vojnovic. Power law and exponential decay of intercontact times between mobile devices. *IEEE Transactions on Mobile Computing*, 9(10):1377–1390, Oct 2010.

[15] Mirco Musolesi and Cecilia Mascolo. Designing mobility models based on social network theory. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(3):59–70, July 2007.

[16] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Transactions on Communications*, 36(7):834–844, Jul 1988.

[17] Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. *IEEE/ACM Transactions on Networking*, 14(1):27–40, Feb 2006.

[18] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, pages 145–158, New York, NY, USA, 2004. ACM.

[19] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, 2000.

[20] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, WDTN '05, pages 252–259, New York, NY, USA, 2005. ACM.

[21] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, July 2003.

[22] J. Burgess, Brian Gallagher, D. Jensen, and B.N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11, April 2006.

[23] A.G. Voyiatzis, J. Gialelis, and D. Karadimas. Dynamic cargo routing on-the-go: The case of urban solid waste collection. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2014 IEEE 10th International Conference on*, pages 58–65, Oct 2014.

[24] T. Zimmermann, H. Wirtz, O. Punal, and K. Wehrle. Analyzing metropolitan-area networking within public transportation systems for smart city applications. In *New Technologies, Mobility and Security (NTMS), 2014 6th International Conference on*, pages 1–5, March 2014.

[25] Michael Doering, Tobias Pögel, and Lars Wolf. Dtn routing in urban public transport systems. In *Proceedings of the 5th ACM Workshop on Challenged Networks*, CHANTS '10, pages 55–62, New York, NY, USA, 2010. ACM.

[26] Shusen Yang, U. Adeel, and J.A. McCann. Selfish mules: Social profit maximization in sparse sensornets using rationally-selfish human relays. *Selected Areas in Communications, IEEE Journal on*, 31(6):1124–1134, June 2013.

[27] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[28] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.

[29] Frans Ekman, Ari Keränen, Jouni Karvo, and Jörg Ott. Working day movement model. In *MobilityModels '08: Proceeding of the 1st ACM SIGMOBILE workshop on Mobility models*, pages 33–40, New York, NY, USA, 2008. ACM.

[30] F. Mohammed, A. Idries, N. Mohamed, J. Al-Jaroodi, and I. Jawhar. Uavs for smart cities: Opportunities and challenges. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 267–273, May 2014.

[31] S.Mignardi and R. Verdone. On the performance improvement of a cellular network supported by an unmanned aerial base station. In *2017 ITC29: Soft5 Workshop, Genova, Italy*, Sept 2017.

[32] Qixing Feng, J. McGeehan, E. K. Tameh, and A. R. Nix. Path loss models for air-to-ground radio channels in urban environments. In *2006 IEEE 63rd Vehicular Technology Conference*, volume 6, pages 2901–2905, May 2006.

[33] A. Al-Hourani, S. Kandeepan, and S. Lardner. Optimal lap altitude for maximum coverage. *IEEE Wireless Communications Letters*, 3(6):569–572, Dec 2014.

[34] A. Merwaday and I. Guvenc. Uav assisted heterogeneous networks for public safety communications. In *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 329–334, March 2015.

[35] E. Yanmaz. Connectivity versus area coverage in unmanned aerial vehicle networks. In *2012 IEEE International Conference on Communications (ICC)*, pages 719–723, June 2012.

[36] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, 20(8):1647–1650, Aug 2016.

[37] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah. Drone small cells in the clouds: Design, deployment and performance analysis. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2015.

[38] S. Koulali, E. Sabir, T. Taleb, and M. Azizi. A green strategic activity scheduling for uav networks: A sub-modular game perspective. *IEEE Communications Magazine*, 54(5):58–64, May 2016.

[39] P. Ladosz, H. Oh, and W. H. Chen. Optimal positioning of communication relay unmanned aerial vehicles in urban environments. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1140–1147, June 2016.

[40] V. Sharma, M. Bennis, and R. Kumar. Uav-assisted heterogeneous networks for capacity enhancement. *IEEE Communications Letters*, 20(6):1207–1210, June 2016.

[41] C. Giannini, A. A. Shaaban, C. Buratti, and R. Verdone. Delay tolerant networking for smart city through drones. In *2016 International Symposium on Wireless Communication Systems (ISWCS)*, pages 603–607, Sept 2016.

[42] J. Chen, U. Yatnalli, and D. Gesbert. Learning radio maps for uav-aided wireless networks: A segmented regression approach. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[43] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

[44] Meeyoung Cha, Haewoon Kwak, Pablo Rodriguez, Yong-Yeol Ahn, and Sue Moon. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Transactions on Networking (TON)*, 17(5):1357–1370, 2009.

[45] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl. Device-to-device communication as an underlay to LTE-advanced networks. *IEEE Communications Magazine*, 47(12):42–49, December 2009.

[46] Feasibility study for proximity services (prose)(release 12). In *3GPP TR 22.803 v.12.2.0, Release 12*, 2012.

[47] B. Kaufman and B. Aazhang. Cellular networks with an overlaid device to device network. In *Proc. Systems and Computers 2008 42nd Asilomar Conf. Signals*, pages 1537–1541, October 2008.

[48] T. Peng, Q. Lu, H. Wang, S. Xu, and W. Wang. Interference avoidance mechanisms in the hybrid cellular and device-to-device systems. In *Proc. Indoor and Mobile Radio Communications 2009 IEEE 20th Int. Symp. Personal*, pages 617–621, September 2009.

[49] X. Chen, L. Chen, M. Zeng, X. Zhang, and D. Yang. Downlink resource allocation for device-to-device communication underlaying cellular networks. In *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pages 232–237, Sept 2012.

[50] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi. Design aspects of network assisted device-to-device communications. *IEEE Communications Magazine*, 50(3):170–177, March 2012.

[51] J. C. F. Li, M. Lei, and F. Gao. Device-to-device (d2d) communication in mu-MIMO cellular networks. In *Proc. IEEE Global Communications Conf. (GLOBECOM)*, pages 3583–3587, December 2012.

[52] B. Zhou, H. Hu, S. Q. Huang, and H. H. Chen. Intracluster device-to-device relay algorithm with optimal resource utilization. *IEEE Transactions on Vehicular Technology*, 62(5):2315–2326, June 2013.

[53] Evolved Universal Terrestrial Radio Access. Further advancements for e-utra physical layer aspects," 3gpp. *Technical Specification Group, RAN TS36*, 814, 2010.

[54] Recommendation ITU-R P ITU-R. 1411-5: Propagation data and prediction methods for the planning of short-range outdoor radio communication systems and radio local area networks in the frequency range 300mhz to 100 ghz. *ITU-R Rec*, 2009.

[55] Giuseppe Piro, Nicola Baldo, and Marco Miozzo. An lte module for the ns-3 network simulator. In *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, pages 415–422. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[56] 3GPP. Physical layer procedures. In *3GPP Tech. Spec. TR 36.213, Release 12*, 2015.

[57] Alcatel-Lucent and Alcatel-Lucent Shangai Bell. Device discovery for d2d proximity services - r1 - 130954. In *3GPP TSG RAN WG1 Meeting #72bis*, 2013.

[58] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire. Femtocaching: Wireless video content delivery through distributed caching helpers. In *2012 Proceedings IEEE INFOCOM*, pages 1107–1115, March 2012.

[59] A. Tuholukova, G. Neglia, and T. Spyropoulos. Optimal cache allocation for femto helpers with joint transmission capabilities. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–7, May 2017.

[60] L. A. Belady. A study of replacement algorithms for a virtual-storage computer. *IBM Systems Journal*, 5(2):78–101, 1966.

[61] Anastasios Giovanidis and Apostolos Avranas. Spatial multi-lru caching for wireless networks with coverage overlaps. *SIGMETRICS Perform. Eval. Rev.*, 44(1):403–405, June 2016.

[62] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire. Femtocaching: Wireless video content delivery through distributed caching helpers. In *2012 Proceedings IEEE INFOCOM*, pages 1107–1115, March 2012.

[63] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, Dec 1978.

[64] Y. Tan, Y. Yuan, T. Yang, Y. Xu, and B. Hu. Femtocaching in wireless video networks: Distributed framework based on exact potential game. In *2016 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1–6, July 2016.

[65] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, Oct 1949.

[66] M. Angjelichinoski, Č. Stefanović, P. Popovski, H. Liu, P. C. Loh, and F. Blaabjerg. Multiuser communication through power talk in dc microgrids. *IEEE Journal on Selected Areas in Communications*, 34(7):2006–2021, July 2016.

List of Publications

1. Laura Galluccio, Beatriz Lorenzo, Savo Glisic, Chiara Buratti, Colian Giannini, and Roberto Verdone. *Epidemic information dissemination in opportunistic scenarios: a realistic model*. In 2015 European Conference on Networks and Communications: Track 7: Special Sessions (EuCNC 2015 - Track 7- Special Sessions), pages 636-640, Paris, France, June 2015.

2. L. Galluccio, C. Giannini, B. Lorenzo, S. Glisic, C. Buratti, and R. Verdone. *Epidemic information dissemination in opportunistic scenarios: a realistic model obtained from experimental traces*. In 2015 International Symposium on Wireless Communication Systems (ISWCS), pages 381-385, Aug 2015.

3. C. Giannini, P. Calegari, C. Buratti, and R. Verdone. *Delay tolerant network for smart city: Exploiting bus mobility*. In 2016 AEIT International Annual Conference (AEIT), pages 1-6, Oct 2016.

4. C. Giannini, C. Buratti, and R. Verdone. *Milp-based radio resource assignment for device to device communications*. In 2017 29th International Teletrffic Congress (ITC 29), volume 2, pages 1-6, Sept 2017.

5. C. Giannini, A. A. Shaaban, C. Buratti, and R. Verdone. *Delay tolerant networking for smart city through drones*. In 2016 International Symposium on Wireless Communication Systems (ISWCS), pages 603-607, Sept 2016.

6. C. Giannini, C. Buratti,V. Cacchiani, and R. Verdone. *Maximizing The Number of Served Ground Users From Unmanned Aerial Base Stations*. IEEE Transactions on Vehicular Tecnology, submitted March 2018.