Interference-Aware Downlink and Uplink Resource Allocation in HetNets with D2D Support

Interference-Aware Downlink and Uplink Resource Allocation in HetNets with D2D Support

Francesco Malandrino, *Member, IEEE,* Zana Limani, Claudio Casetti, *Member, IEEE,* Carla-Fabiana Chiasserini, *Senior Member, IEEE*

Abstract

We address the resource allocation problem in an LTE-based, 2-tier heterogeneous network where in-band D2D communications are supported under network control. The different communication paradigms share the same radio resources, thus they may interfere. We devise a dynamic programming approach to efficiently schedule download and upload traffic, by (i) efficiently matching communicating endpoints and (ii) assigning radio resources in an interference-aware manner while accounting for the characteristics of the content to be delivered. To this end, we develop an accurate model of the system and apply approximate dynamic programming to solve it. Our solution allows us to deal with realistic, large-scale scenarios. In such scenarios, we compare our approach to today's networks where eICIC techniques and proportional fairness scheduling are implemented. Results highlight that our solution increases the system throughput while greatly reducing energy consumption. We also show that D2D mode, established either in the downlink or uplink, can effectively support delivery of highly popular content without significantly harming macrocell or microcell traffic, leading to increased system capacity. Interestingly, we find that D2D mode can also be a low-cost alternative to microcells.

Index Terms

Resource allocation, heterogeneous cellular networks, device-to-device communication.

F. Malandrino, Z. Limani, C. Casetti and C.-F. Chiasserini are with the Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Torino, Italy, email: name.lastname@polito.it.

I. Introduction

The rapid increase in cellular data traffic due to the proliferation of wireless gadgets and evolution of smartphones is posing a serious challenge to today's mobile cellular networks. The key question remains: will next-generation cellular networks be ready to rise to the occasion?

It is clear that there is a limit to what today's networks with the current cellular infrastructure can achieve both in terms of capacity and coverage. Indeed, they are not flexible enough for the ever-changing behaviour of users who now have access to a diverse range of applications on their smartphones and tablets, and who are ultimately more demanding in terms of user experience.

A cost-effective way to overcome these challenges is the deployment of Heterogeneous Networks (HetNets), i.e., adding low-cost base stations across the coverage area of the traditional cellular infrastructure. This results in a multi-tier network with smaller cells such as micro, pico and femtocells overlaying the traditional macrocells. Such a strategy promises to improve both capacity and coverage by closing the gap between the access network and the user.

Another expected feature of future networks will be the support of direct device-to-device (D2D) communication between users that are in proximity of each other [1]–[3]. Such D2D links will most likely be established under operator control, as foreseen by the 3GPP ProSe group working on Release 12 [4].

This feature also opens up a range of opportunities for improving spectrum efficiency and achieving higher data rates, at virtually no cost for the operator. Another very appealing trait of these solutions is that they operate at significantly lower power, making them much more energy-efficient.

A major issue in these types of complex networks is interference management, both between different tiers within the heterogeneous architecture and between infrastructure-to-device and device-to-device communication. To achieve high spectrum utilization and exploit the opportunities offered by D2D and HetNets, these solutions must coexist in harmony between each other and the traditional cellular network, while sharing the same frequency spectrum.

Techniques to mitigate cross-layer interference in HetNets are already available, e.g., ICIC (Inter-Cell Interference Coordination). However, intelligent resource allocation techniques must be developed to fully reap the benefits offered by HetNets and D2D support.

In this paper, we address these challenges and propose an interference-aware resource scheduling algorithm for an LTE-based, two-tier HetNet with D2D support. We consider that D2D will

take place within the LTE bands, in what is often called "in-band underlay" mode [5], where D2D opportunistically accesses the same spectrum resources used by the other nodes in the cellular network. Indeed, as shown in [5], the in-band underlay D2D mode outperforms the overlay mode in terms of achieved throughput. In principle, D2D communications can take place in either the uplink or the downlink resources. Currently, it is widely accepted that uplink resources should be used [6], since, at present, traffic is significantly heavier in downlink than in uplink. However, it is expected that in the future traffic will be much less asymmetric, then the use of downlink resources will represent a viable option. In both scenarios, D2D can cause significant interference to normal infrastructure-to-device communications, either to nearby receiving UEs when implemented in the downlink bands, or to nearby receiving base stations (BSs) when deployed in the uplink bands. Without proper management of this interference, D2D communication may easily end up doing more harm than good.

We therefore address and compare both D2D scenarios, and propose a resource allocation procedure based on approximate dynamic-programming. The procedure itself is adaptable to both downlink and uplink D2D scenarios, it is updated every subframe and is efficient enough to be applied to large-scale scenarios. The performance of our approach is numerically evaluated and compared to standard resource scheduling algorithms adopted in today's cellular networks, employing interference mitigation techniques. Results highlight that the proposed approach is apt at fully exploiting the potential of both the heterogeneity of the network and D2D support, while consuming far less energy. Results further reveal that D2D interactions act inherently as an additional layer in the heterogeneous network, thereby potentially reducing the need for deploying more microcells. Finally, while the uplink and downlink scenarios provide similar performance in current traffic load conditions, the downlink will become preferable as the upload and the download traffic tend to even out.

In summary, our key contributions are as follows:

- We provide a realistic network model for a HetNet with in-band underlay D2D support, which captures the relationship between the different tiers of the HetNet, as well as the relationship between base stations, users and users' traffic requests. A realistic model is used to obtain an accurate representation of the inter-tier and inter-cell interference.
- We formulate the resource allocation problem in such a HetNet using a Dynamic Programming model and solve it using Approximate Dynamic Programming (ADP) techniques. The

algorithm we propose solves this very hard problem efficiently enough so that it can be run in real time. Furthermore, our algorithm provides a solution that takes into account the interference caused between the different tiers and the nodes in the network, as well as the nature and type of the data traffic demanded by the users.

- We evaluate two distinct scenarios depending on whether D2D communications are allowed to use the downlink or uplink resources, and then solve the resource allocation problem for both directions of data traffic, namely upload and download, by jointly scheduling the downlink and uplink resources.
- Finally, using numerical evaluation, we show that our approach significantly outperforms the *de facto* standard approach, the proportional fair scheduling (PF). In particular, our results highlight the gain in performance due to D2D, especially in the presence of viral traffic, namely content items that experience a sudden surge in demand in a short period of time.

The remainder of this paper is organized as follows: after discussing related work in Sec. II, we introduce the system under study and formulate our problem in Sec. III. The network model is presented in Sec. III-B. Sec. IV outlines the dynamic programming approach and our ADP solution. Results derived in a realistic scenario are shown in Sec. V. Finally, we draw our conclusions in Sec. VI.

II. Related work

The deployment of a multi-tier network where cells use the same radio resources is highly beneficial since it allows traffic offloading from macrocells to smaller cells [7]. However, such scenario imposes the adoption of ICIC techniques, for which a good survey can be found in [8]. Additionally, eICIC specifications in 3GPP Rel. 10 [9] foresee the use of the Cell Range Expansion (CRE) in LTE systems. Such a technique involves adding a positive range expansion bias to the pilot downlink signal strength received from microcells so that more users connect to them. Then, in order to mitigate the interference between overlaying macro- and microcells, macrocells may periodically mute their downlink transmissions in certain subframes (called almost blank subframes - ABSs). By using ABSs for edge users, microcells can significantly improve their performance [10]. In our work, we do not focus on eICIC techniques; rather, we take a scenario implementing them as our term of comparison. Unlike the above mentioned works, we assume the presence of an area controller that issues resource allocation and scheduling

instructions to BSs, through high-speed optical fiber connectivity [11], [12]. Also, we assume both I2D and D2D communication paradigms in all cells.

The integration of D2D communication in cellular networks and its applications are investigated in [13]. This work presents a conceptual framework for the formulation of problems such as peer discovery, scheduling, and resource allocation. Another good survey on D2D proximity services as foreseen by 3GPP can be found in [1]. This work provides a comprehensive overview on some of the key functionalities of D2D communication and design challenges concerning the integration of such communication within cellular networks. The authors of [1] focus on two main use cases of D2D, as envisioned by 3GPP, namely for public safety and traffic offloading. In the latter use case, which is the one we also focus on in our work, it is expected that D2D will be established with support from infrastructure nodes, simplifying tasks such as synchronization and discovery, and enabling dynamic resource allocation for D2D links [14].

The problem of resource allocation is also studied in [15]–[17], although only macrocells and D2D mode are considered therein. Additionally, in [15] the D2D pairs wishing to exchange data are given at the outset (i.e., unlike our work, [15] does not address the endpoint association problem). In [16] the authors seek an optimal resource allocation scheme for cellular networks with D2D support, comparing orthogonal and non-orthogonal resource sharing modes. As numerical results show, D2D coupled with non-orthogonal resource sharing mode ensures the highest gains in terms of sum rate and per-cell throughput. Authors of [15], [17] formulate resource allocation as a mixed integer optimization problem, which is notoriously hard to solve, with [15] also presenting a greedy heuristic.

The work in [5] further compounds the problem by investigating the selection of the most suitable communication mode, still in a single-tier scenario with D2D. There, an analytical model is proposed, based on the assumption that the positions of BSs and users can be modeled as a Poisson point process. In [18], we address the resource allocation problem in a network scenario similar to the one presented in this paper, however only downlink resources are considered in the allocation procedure. In contrast, here we jointly address the uplink/downlink resource scheduling and, furthermore, we consider separate scenarios for D2D operating in the downlink and uplink resources. Preliminary results of our work, considering the resource scheduling problem for both upload and download traffic are presented in [19]. Beside the different methodology and scope of the studies above, we stress that our work addresses HetNets including macrocells, microcells

and D2D. While [5] derives an optimal factor of spectrum partition between cellular and D2D communication, we aim at determining the endpoint that should serve each user and an efficient data scheduling on a single radio resource basis.

III. Problem Formulation

Resource scheduling management in LTE networks is an important function that aims at efficiently allocating downlink and uplink channel resources to UEs, in order to meet as many of their expectations while optimizing network performance. According to the LTE specifications [20], the minimum resource scheduling unit is referred to as a resource block (RB). One RB consists of 12 subcarriers (each 15 kHz wide) in the frequency domain and one subframe (1-ms long) in the time domain. Radio resource allocation is updated every subframe.

We consider the resource allocation problem in a two-tier LTE heterogeneous network comprising several microcells nested within the traditional larger macrocells deployed in an urban setting. The macro and microcells are served by BSs, referred to as macroBSs and microBS, respectively. In LTE, a macroBS is often referred to as eNode B. It has been shown that the resource allocation problem in LTE-based networks is NP-hard [15], [17]. The problem is further exacerbated when introducing additional infrastructure layers in the heterogeneous architecture and enabling D2D. In such a complex scenario the resource scheduling strategy needs to address not only the problem of which resources to allocate to which UE, but also which potential endpoint among the various choices (macroBS, microBS or another UE) should serve a UE.

A. Scenario and assumptions

In this subsection we detail the system scenario and any assumptions we make about the D2D-enabled HetNet under study. As was previously mentioned, the network model that we use is based on a two-tier architecture. The network is made up of a set of first tier macroBSs that control the macrocells and a set of microBSs deployed within their coverage area that control the second tier microcells. We define the coverage of a BS (either macroBS or microBS), as the area where the received strength of the BS pilot signal is higher than a given threshold, namely -70 dBm in our case [21]. A UE under the coverage of both a macroBS and a microBS can be served by either of them.

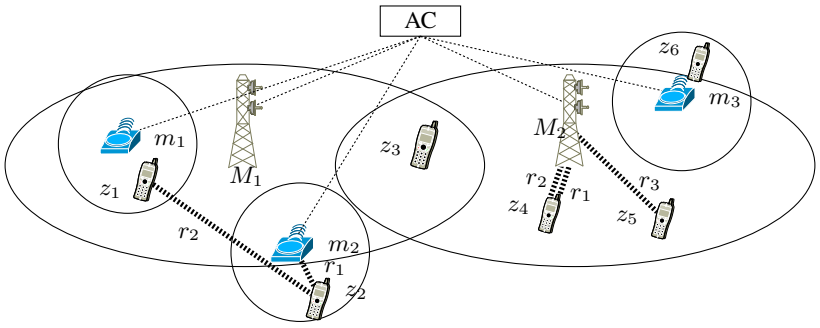


Fig. 1. An example scenario. UEs are denoted by $z_1, \ldots, z_6$, macroBSs by M_1, M_2 and microBSs by m_1, m_2, m_3. Solid lines denote coverage areas. Dotted lines correspond to RBs used by a pair of endpoints.

In general, we assume that our HetNet supports Frequency Division Duplexing (FDD), with uplink and downlink using two different portions of the spectrum, however our formulation can be easily extended to the Time Division Duplexing (TDD) case. Radio resource allocation is performed and updated every subframe by the Area Controller (AC) in the core network, which assists BSs in radio resource allocation and traffic scheduling. The AC collects information on the channel quality from the BSs and on content items that users wish to upload/download. Thus, BSs are only concerned with propagation and spectrum aspects, while they are oblivious to higher-layer demands. From the collected information, the AC determines (i) which endpoint (among the possible ones: macroBS, microBS, or UE) should serve each user, and (ii) which RB(s) to employ for such communication. Decisions taken by the AC are issued to BSs, which forward them to UEs. The fact that the AC performs the resource allocation task in a centralised manner is in contrast with the distributed schemes that are adopted in today's cellular networks, where eNodes B are in charge of resource scheduling. However, given the expected complexity of future cellular networks, it is widely anticipated that a more centralised structure will be adopted [22]. An example scenario of such a network is shown in Fig. 1, while Fig. 2 shows the roles of ACs, BSs, and users.

We assume that BSs have optical fiber connectivity to the core network, as envisioned by operators and network manufacturers [11], [12]. This assumption is reasonable given the new, complex tasks and the ever-increasing amount of traffic that the cellular infrastructure is expected to handle.

As envisioned by recent trends and standardization activities, we consider network-controlled

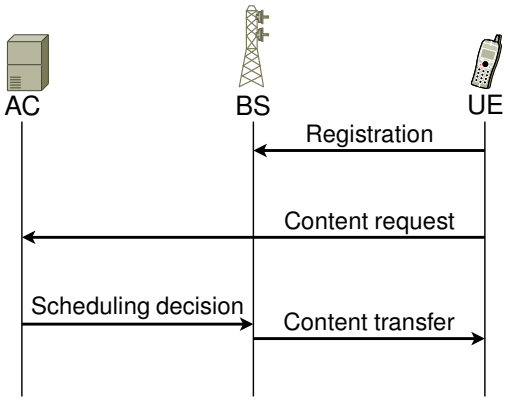


Fig. 2. The role of the area controller in our system model, in the case of content download. Users register with BSs, as in current LTE networks. Content requests, however, are directed to the AC, which makes content-aware scheduling decisions. Such decisions are pushed to BSs, which are then in charge of enacting them.

D2D communication [4], [14], [16]. This implies that, not only can synchronization and security issues be easily solved, but also UE pairs can be efficiently scheduled so as to use cellular resources even at high traffic load.

We assume in-band, underlay, deployment of D2D communication, that is, spectrum resources are shared between devices using the D2D communication paradigm and the cellular infrastructure. However, we consider two possible scenarios: one in which D2D interactions leverage the downlink radio resources (downlink scenario), and the other in which D2D takes place in the uplink spectrum (uplink scenario). To this end, and in order to ensure efficient allocation of all radio resources, in either scenario, we consider both download and upload traffic scheduling.

We mainly focus on unicast data transfers and assume that UEs can be served download traffic by only one endpoint at a time, and similarly they can transmit upload data to only one endpoint at a time. Considering the most popular types of terminals, we also assume UEs to be half-duplex, i.e., they cannot transmit and receive at the same time. This implies that, e.g., in the downlink scenario a UE receiving information from the cellular infrastructure cannot simultaneously serve another UE.

In Sec. V we compare the performance of the proposed system to a distributed scenario reflecting today's networks, where D2D is not supported and UEs are always served according to the proportional fairness algorithm, by the BS whose received signal is the strongest.

B. Network Model

We denote the collective set of BSs by $\mathcal{B}$ and the specific sets of macroBSs and microBSs by $\mathcal{B}_M$ and $\mathcal{B}_m$, respectively. The BSs serve a number of users with mobile equipment that are within the service coverage area. We refer to these users shortly as UEs and denote the total set of UEs by $\mathcal{Z}$.

1) Interactions and requests: Within the network the following interactions are supported: interactions between BSs and UEs and vice-versa, and interactions between two UEs or D2D communications.

BSs and UEs interact with each other using assigned radio resources. RBs allocated to each interaction can be selected from two distinct sets of RBs: the set of downlink radio resources $\mathcal{R}_d$, and the set of uplink resources $\mathcal{R}_u$. The RB allocation is valid for one subframe, therefore the time is divided into a set $\mathcal{K}$ of 1 ms time steps.

The goal of the interactions is to transfer the content requested by the UEs. The content requests may be made in both directions: upload or download. In fact, a UE can make a download request for a specific content item from a finite download content set denoted as $\mathcal{C}_d$. One of the BSs, or, when D2D is enabled, a UE caching the requested content item, can serve the request. Similarly, a UE may wish to upload a specific content item to the Internet, through the cellular infrastructure (i.e., macroBSs or microBSs). We denote by $\mathcal{C}_u$ the set of content items that UEs may wish to upload.

The allocation of RBs depends on the type of the communicating endpoints since the different RB sets may be used only for specific types of interactions. Namely, BS-UE interactions may be allocated RBs only from the $\mathcal{R}_d$ set, while UE-BS interactions may be allocated RBs only from the $\mathcal{R}_u$ set. For D2D communications, the RB set to be used shall depend on whether D2D is established in the uplink or downlink band. Indeed, since we assume that D2D links are used for download traffic offloading, i.e., they may serve to provide UEs with requested content items if available in the vicinity, the choice of the RB set is constrained by the traffic direction.

We define the time step at which a UE becomes interested in downloading/uploading a specific content item as an input parameter called *want-time* and denote it as $w_c(z) \in \mathcal{K}$. Furthermore, each content item $c \in \mathcal{C}_d \cup \mathcal{C}_u$ is characterized by a certain size l_c and a maximum delay D_c with which it should be delivered to its destination. These parameters are known to the scheduler and are used as input parameters to the scheduling algorithm.

During each interaction a certain amount of data is transferred between the two communicating endpoints. We define the variable $\chi_c^k(e_1, e_2)$ to indicate the amount of content c transferred between e_1 and e_2 at time step k. Then, for each content item $c \in \mathcal{C}_d \cup \mathcal{C}_u$ a user is interested in, we define a variable $h_c^k(z)$, which denotes the total amount of content c downloaded/uploaded by user z until time step k. Note that $h_c^k(z)$ is a non-decreasing quantity bounded by the size of the content item, i.e., $0 \leq h_c^k(z) \leq h_c^{k+1}(z) \leq l_c, \forall k \geq 0, \forall c \in \mathcal{C}_d \cup \mathcal{C}_u$.

The relationship between the two quantities is the following:

$$\begin{aligned} h_c^{k+1}(z) &\leftarrow h_c^k(z) + \sum_{e \in \mathcal{B} \cup \mathcal{Z}} \chi_c^k(e, z), \forall c \in \mathcal{C}_d \\ h_c^{k+1}(z) &\leftarrow h_c^k(z) + \sum_{e \in \mathcal{B}} \chi_c^k(z, e), \forall c \in \mathcal{C}_u \,. \end{aligned} \tag{III.1}$$

A list of symbols and definitions used throughout the paper can be found in Table I.

2) Propagation and interference model: For each interaction, we denote the power with which endpoint $e_1 \in \mathcal{B} \cup \mathcal{Z}$ transmits to endpoint e_2 by $P(e_1, e_2)$. For BS-UE transmissions, the value of such parameter is fixed and depends only on whether e_1 is a macroBS or a microBS, i.e., $P(e_1, e_2) = P(e_1)$ [20]. Conversely, we assume that the transmit power of an UE, both when transmitting to a BS or another UE, is subject to a power control scheme, so that its value may depend on such factors as propagation conditions and positions of either endpoints. Specifically, we use the following power control formula, from [23], to calculate the power of a transmitting UE e_1 towards endpoint e_2, which can be another UE or a BS, at each time step k:

$$P(e_1, e_2)|_{dB} = \min\{P_{max}|_{dB}, 10\log_{10}(M) + P_0|_{dB} + \rho \cdot PL(e_1, e_2)|_{dB} + \Delta_{TF} + f(k)\} \,.$$

P_{max} indicates the maximum power at which a UE can transmit, while M indicates the number of RBs allocated to e_1 at time step k. P_0, ρ, Δ_{TF} and $f(k)$ are cell and user specific configuration parameters, indicating respectively the spectral power density required at the receiver, the path-loss compensation factor, a UE-specific parameter depending on the applied Modulation Coding Scheme (MCS) and a higher-layer closed-loop command to increase/decrease power level. $PL(e_1, e_2)$ indicates the path loss experienced between the two endpoints. In general, it depends on the distance between the two endpoints, on the Line of Sight (LoS) conditions between them as well as on the transmitting and receiving antenna heights. It is calculated according to specific propagation models as detailed below.

TABLE I

LIST OF THE MAIN SYMBOLS USED IN THE SYSTEM MODEL

Symbol	Description
$\mathcal{B}$	Set of BSs
$\mathcal{C}_d, \mathcal{C}_u$	Set of download and upload content items
$\mathcal{R}_d, \mathcal{R}_u$	Set of downlink and uplink radio resources (RBs)
$\mathcal{Z}$	Set of users
l_c	Size of content c
D_c	Maximum delivery delay of content c
$\mathcal{K}$	Set of time steps
$w_c(z)$	Time step at which user z becomes interested in downloading or uploading content c
$h_c^k(z)$	Cumulative amount of data of content c that requester z has downloaded or uploaded until the beginning of time step k
$\delta_r^k(e_1, e_2)$	Amount of data that can be sent from e_1 to e_2 on RB r at time step k
$y_{r,c}^k(e_1, e_2)$	Amount of data of content c transferred from e_1 to e_2 at time step k over RB r
$\chi_c^k(e_1, e_2)$	Amount of data of content c transferred from e_1 to e_2 at time step k (over any possible RB)
$P(e_1, e_2)$	Power with which endpoint e_1 transmits to e_2
$A(e_1, e_2)$	Attenuation of the link between e_1 and e_2
$I_r^k(e_2)$	Interference experienced by endpoint e_2 on RB r
$\mathrm{SINR_r^k(e_1, e_2)}$	Signal to interference and noise ratio experienced by e_2 when receiving from e_1
$PL(e_1, e_2)$	Path loss between e_1 and e_2
N	Noise power (W)

In order to model as accurately as possible the power and interference in our scenario, we denote the total attenuation experienced by the signal between two endpoints as $A(e_1, e_2)$. To precalculate these values, we use the urban propagation models specified in [21] and [24]. Specifically, for macroBS-UE links, the attenuation is given by the following expression:

$$A(e_1, e_2) = \frac{PL(e_1, e_2)}{G_T AP(\theta_{e_1,e_2})} \qquad e_1 \in \mathcal{B}_M, e_2 \in \mathcal{Z}$$

where G_T is the antenna gain of macroBS e_1, and $AP(\theta_{e_1,e_2})$ is the antenna pattern factor, which depends on the angle $\theta(e_1, e_2)$ between the maximum radiation direction of e_1 antenna and the direction between the antenna and UE e_2. $PL(e_1, e_2)$ is the path loss as defined above. To precalculate the path loss for these types of links, the Urban Macro (UMa) propagation model in [21] is used. For microBS-UE links, assuming omnidirectional antennas with unitary gain, the attenuation between two endpoints depends solely on the path loss value, i.e., $A(e_1, e_2) =$

$PL(e_1, e_2)$, which is precomputed using the Urban Micro (UMi) propagation model in [21]. For UE-BS, the path loss is calculated using the respective reversed propagation model used in the downlink. For D2D links, path loss is modeled using the UMi propagation model specified in [24] and a correction factor that compensates for the low antenna height of the transmitter.

In general, we see that the attenuation of the signal depends on the type and position of communicating antennas, as well as their propagation conditions (LoS). However, since both power and attenuation values are simply input values to our problem formulation, the propagation and power control models can be easily extended to accommodate additional assumptions, while obtaining a remarkable level of realism.

Given the transmit power and the attenuation factor, the useful power received at e_2 from source e_1 is $P(e_1, e_2)/A(e_1, e_2)$. Similarly, every other node pair (e, z) communicating on the same RB where e_2 is receiving, causes a certain amount of interference to e_2 given by $P(e, z)/A(e, e_2)$. Thus, the total amount of interference experienced by e_2 on RB r is:

$$I_r^k(e_2) = \sum_{\substack{(e,z) \text{ use } r \text{ at } k \wedge \\ e:\, A(e,e_2)>0}} P(e, z)/A(e, e_2)\,,$$

while the signal to noise plus interference ratio (SINR) is yielded by

$$\mathrm{SINR}_r^k(e_1, e_2) = \frac{P(e_1, e_2)/A(e_1, e_2)}{N + I_r^k(e_2)}. \tag{III.2}$$

where N denotes the noise power. We can finally map the SINR onto the amount of data that can be transferred from e_1 to e_2 using RB r during step k. We indicate this amount by $\delta_r^k(e_1, e_2)$, and we determine its value based on experimental measurements [25].

Clearly, the value $\delta_r^k(e_1, e_2)$ places a strict limitation on the amount of data that can be transferred between the two endpoints, which we defined earlier as $\chi_c^k(e_1, e_2)$. The relationship between the two can be described by the following inequality:

$$\sum_{c \in \mathcal{C}_d \cup \mathcal{C}_u} \chi_c^k(e_1, e_2) \leq \sum_{r \in \mathcal{R}_d \cup \mathcal{R}_u} \delta_r^k(e_1, e_2). \tag{III.3}$$

In (III.3), strict inequality holds when e_1 is a serving UE and the total amount of data it is caching for e_2 is smaller than what could be transferred over the link between the two nodes.

Additionally, we define the intermediary variable $y_{r,c}^k(e_1, e_2)$ to indicate the amount of content c transferred between e_1 and e_2 over RB r. For $y_{r,c}^k(e_1, e_2)$, a similar inequality holds:

$$y_{r,c}^k(e_1, e_2) \leq \delta_r^k(e_1, e_2) \qquad \forall c \in \mathcal{C}_d \cup \mathcal{C}_u\,. \tag{III.4}$$

IV. A Dynamic Programming Approach to Resource Allocation

We now present the model we develop using standard dynamic programming methodology, in order to tackle the resource allocation problem formulated in the previous section. As shown by previous work [15], [17], the problem of radio resource allocation in LTE-based systems is NP-hard, even when less complex scenarios than ours are considered. Thus, we resort to approximate dynamic programming in order to solve the model in realistic, large-scale scenarios.

A. The dynamic programming model

Dynamic programming is an optimization technique based on breaking a complex problem into simpler, typically time-related, subproblems. Since scheduling in LTE systems occurs every subframe, we solve the resource allocation problem every time step k. A dynamic programming model consists of the following elements (denoted by bold-face Latin letters) [26]:

- the *state variable*, $\mathbf{s}^k$, which describes the state of the system at time k;
- the *action* set, $\mathbf{A}^k = \{\mathbf{a}^k\}$, i.e., all possible decisions that can be taken at time k;
- an exogenous (and potentially stochastic) *information process*, accounting for information on the system becoming available at time k;
- the *cost* of an action, $\mathbf{C}(\mathbf{s}^k, \mathbf{a}^k)$, i.e., the immediate cost due to the selected action, given the current state;
- the *value*, $\mathbf{V}(\mathbf{s}^k, \mathbf{a}^k)$, of ending up at a new state $\mathbf{s}^{k+1}$, determined by the current state and action; such value is given by the cost associated with the optimal system evolution from $\mathbf{s}^{k+1}$.

Table II summarizes these quantities, their meaning in our system and the symbols we use for them. Fig. 3 shows how each of them is used in the model.

In particular, in our case the system state at generic time k is given by the set of duplets: $\mathbf{s}^k = \{h_c^k(z), w_c(z)\}_{u,c}$. Each duplet refers to a different UE-content pair, z and c, and includes (i) the amount $h_c^k(z)$ of the content transferred to or by the UE and (ii) the want-time $w_c(z)$. Clearly, at time k we only know those want-times $w_c(z) \leq k$.

An action is a set of triplets, each defining which endpoint e_1 should transmit to which other receiving endpoint e_2 and using which RB r, i.e., $\mathbf{a}^k = \{(e_1, e_2, r)\}$. In simpler terms, an action is a realization of resource allocation. Note that, since we are considering both download and upload traffic, $e_1, e_2 \in \mathcal{B} \cup \mathcal{Z}$.

TABLE II

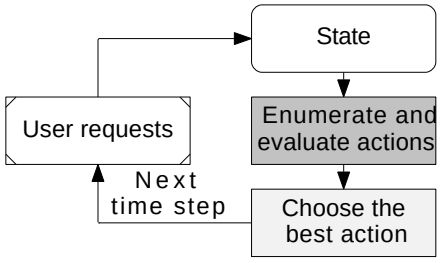
LIST OF SYMBOLS USED IN THE DYNAMIC PROGRAMMING MODEL

Quantity and symbol	Description
Current state $\mathbf{s}^k$	Set of duplets, each referring to a different UE-content pair. A duplet includes the amount of content c already transferred to/by z, $h_c^k(z)$, and the want-time $w_c(z)$ if no greater than k
Action to take $\mathbf{a}^k$	Set of triplets indicating which pairs of endpoints (e_1, e_2) should communicate on which RB, i.e., (e_1, e_2, r)
Exogenous information	Want-times $w_c(z)$
Cost $\mathbf{C}(\mathbf{s}^k, \mathbf{a}^k)$	Ratio of the amount of content still to be retrieved to the remaining time before the deadline for content delivery expires
Value $\mathbf{V}(\mathbf{s}^k, \mathbf{a}^k)$	Total (expected) costs due to the system future evolution
$\tilde{\mathbf{A}}^k$	Auxiliary action space, i.e., set of values expressing the level of preference associated to each type of endpoint

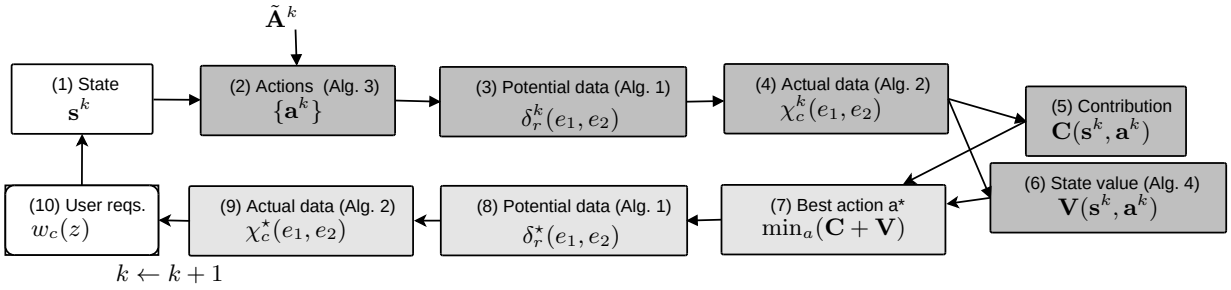
The dynamic programming model works as shown in Fig. 3 (a): for each time step we enumerate and evaluate the possible actions, select (and enact) the best one, and move to the next time step. At this point, we become aware of which content items have been recently requested, hence we can determine the next system state.

Fig. 3 (b) offers a more detailed view. The starting point is given by the current state $\mathbf{s}^k$ and the set of actions describing the possible resource allocations (steps 1 and 2 in the figure). The latter step is further elaborated in the next section. For each action, we compute the potential (δ) and, then, the actual (χ) amount of data that can be transferred between every pair of endpoints (steps 3–4), using the algorithms we provide below. Given the variables χ, we can update the total amount of data that each requesting UE z can download/upload by the beginning of the next time step using (III.1).

For each action $\mathbf{a}^k$, we can then evaluate the cost $\mathbf{C}(\mathbf{s}^k, \mathbf{a}^k)$ the system incurs if $\mathbf{a}^k$ is selected (step 5 in Fig. 3 (b)). We define such cost as the sum over all requesters and content of the ratio of the amount of data still to be transferred to the time before the content delivery deadline



(a)



(b)

Fig. 3. Dynamic programming: (a) Main steps involved; (b) Detailed view. Given the current state (1), the set of possible actions can be determined (2). For each action, the potential (3) and actual (4) amount of content transferred between the pairs of endpoints can be computed. These values are further used to compute the cost (5) of an action, and to estimate the value of the state it leads to (6). The latter two figures are used (7) to select the best action. The resulting data transfers (8-9), along with the users that just became interested in a content, define the next state. The description of the notations appearing in the flow diagram can be found in Tables I and II.

expires, i.e.,

$$\mathbf{C}(\mathbf{s}^k,\mathbf{a}^k) = \sum_{c\in\mathcal{C}_d}\sum_{\substack{z\in\mathcal{Z}:\\ w_c(z)\le k}} \frac{l_c - \left(h_c^k(z) + \sum_{e\in\mathcal{B}\cup\mathcal{Z}}\chi_c^k(e,z)\right)}{w_c(z)+D_c-k} + \sum_{c\in\mathcal{C}_u}\sum_{\substack{z\in\mathcal{Z}:\\ w_c(z)\le k}} \frac{l_c - \left(h_c^k(z) + \sum_{e\in\mathcal{B}}\chi_c^k(z,e)\right)}{w_c(z)+D_c-k}\,. \tag{IV.1}$$

By the above definition, a lower cost is therefore obtained for those allocation strategies, $\mathbf{a}^k$, assigning more resources to transfers that are closer to their completion deadline.

The value $\mathbf{V}(\mathbf{s}^k,\mathbf{a}^k)$ (step 6 in Fig. 3 (b)) is yielded by the sum of the costs $\mathbf{C}(\mathbf{s}^{k+1},\mathbf{a}^{k+1}) + \mathbf{C}(\mathbf{s}^{k+2},\mathbf{a}^{k+2}) + \ldots$. In other words, it is the cost that will be paid in the future, after the system has reached state $\mathbf{s}^{k+1}$. State values do not normally admit a closed-form expression. In standard

dynamic programming [26, Ch. 3], they are computed by accounting for all possible states and actions, typically leading to an exceedingly high complexity in non-toy scenarios. We address such an issue by applying approximate dynamic programming as described in the following section.

Once $\mathbf{C}(\mathbf{s}^k, \mathbf{a}^k)$ and $\mathbf{V}(\mathbf{s}^k, \mathbf{a}^k)$ have been computed for all actions, the action $\mathbf{a}^*$ minimizing the cost $\mathbf{C}(\mathbf{s}^k, \mathbf{a}^k)+\mathbf{V}(\mathbf{s}^k, \mathbf{a}^k)$ is selected (step 7 in Fig. 3 (b)). Given $\mathbf{a}^*$, the corresponding actual amount of transferred data can be calculated (steps 8-9). This, along with fresh information on user requests (step 10), leads to the next state $\mathbf{s}^{k+1}$.

Next, we detail how to compute the potential $\delta_r^k(e_1, e_2)$ (Algorithm 1) and actual $\chi_c^k(e_1, e_2)$ (Algorithm 2) amount of data, while taking into account the interference due to the spatial reuse of radio resources. It is worth stressing that, the processes we describe below have a very low computational complexity, namely $O(|\mathcal{Z}|)$, while maintaining a high level of realism.

Algorithm 1 Computing the amount δ of data that can be potentially transferred

Require: $\mathbf{a}^k$

1: $I_r^k(z) \leftarrow 0, \forall u \in \mathcal{Z}$, $\forall r \in \mathcal{R}_d \cup \mathcal{R}_u$
2: **for all** $(e_1, e_2, r) \in \mathbf{a}^k$ **do**
3: **for all** $z \in \mathcal{Z} \setminus \{e_1, e_2\}$ **do**
4: $I_r^k(z) \leftarrow I_r^k(z) + \mathbb{1}_{A(e_1,z)>0} P(e_1, e_2)/A(e_1, z)$
5: **for all** $(e_1, e_2, r) \in \mathbf{a}^k$ **do**
6: $\mathrm{SINR}_r^k(e_1, e_2) \leftarrow \frac{P(e_1,e_2)}{A(e_1,e_2)(N+I_r^k(e_2))}$
7: $\delta_r^k(e_1, e_2) \leftarrow$ **sinr_to_delta**$(\mathrm{SINR}_r^k(e_1, e_2))$
8: **return** $\delta_r^k(e_1, e_2)$

Algorithm 1 is used in steps 3 and 8 in Fig. 3 (b). In line 4, we account for the fact that every active endpoint pair may create interference at other users on the particular used RB. All interference values are computed within the first loop. The second loop computes the SINR (line 6) and maps it onto the amount of data that can be transferred on RB r during time step k (line 7). We perform such mapping by using the experimental values in [25].

Algorithm 2 instead refers to steps 4 and 9 in Fig. 3 (b). The algorithm takes as input the action $\mathbf{a}^k$ and the amount of data $\delta_r^k(e_1, e_2)$ that can be potentially transferred as a consequence of this action (computed in the previous step through Algorithm 1). Then, for each triplet in $\mathbf{a}^k$, it defines the set of potentially transferable content items $\mathcal{T}$ in lines 3–6. This is done firstly

Algorithm 2 Computing the amount χ of data being actually transferred

Require: $\mathbf{a}^k$, $\delta_r^k(e_1, e_2)$

1: $\chi_c^k(e_1, e_2) \leftarrow 0$, $y_{r,c}^k(e_1, e_2) \leftarrow 0, \forall c, e_1, e_2, r$
2: **for all** $(e_1, e_2, r) \in \mathbf{a}^k$: $\delta_r^k(e_1, e_2) > 0$ **do**
3: **if** $r \in \mathcal{R}_u \wedge e_2 \in \mathcal{B}$ **then**
4: $\mathcal{T} \leftarrow \{c \in \mathcal{C}_u : w_c(e_1) < k \wedge h_c^k(e_1) < l_c\}$
5: **else**
6: $\mathcal{T} \leftarrow \{c \in \mathcal{C}_d : w_c(e_2) < k \wedge h_c^k(e_2) < l_c\}$
7: $c^\star \leftarrow \arg\min_{c \in \mathcal{T}:} w_c$
8: **if** $c^\star \in \mathcal{C}_d$ **then**
9: $y_{r,c^\star}^k(e_1, e_2) \leftarrow \min \{h_{c^\star}^k(e_1) - h_{c^\star}^k(e_2), \quad \delta_r^k(e_1, e_2)\}$
10: **else if** $c^\star \in \mathcal{C}_u$ **then**
11: $y_{r,c^\star}^k(e_1, e_2) \leftarrow \min \{l_{c^\star} - h_{c^\star}^k(e_1), \quad \delta_r^k(e_1, e_2)\}$
12: $\chi_{c^\star}^k(e_1, e_2) \leftarrow \chi_{c^\star}^k(e_1, e_2) + y_{r,c^\star}^k(e_1, e_2)$
13: **return** $\chi_c^k(e_1, e_2), y_{r,c}^k(e_1, e_2)$

to meet the scenario-imposed constraint that UE-BS communication may use only uplink RBs. Secondly, it satisfies the constraint that a UE can only play one role at a given time step k, either as an uploader, downloader or a server. As a consequence, download and upload content cannot be mixed. The content to be transferred is selected in line 7, giving priority to incompletely transferred content items that were requested first. In particular, in lines 9 and 11, for each item the amount transferred on RB r, $y_{r,c^\star}^k(e_1, e_2)$, is determined. This amount is given by the minimum between the amount of data that source e_1 still has for destination e_2 and the amount of data that can be accommodated in the RB[1]. Finally, the χ-value is obtained by summing the y values over all RBs (line 12).

Notwithstanding the low complexity implied by the computation of the δ and χ quantities, standard dynamic programming itself is affected by the well-known "curse of dimensionality" [26], which makes it impractical for all but very small scenarios. In our scenario, this problem is caused mainly by the exceedingly large set of possible actions and the aforementioned complexity in the evaluation of the future cost $\mathbf{V}$. As an example, consider the set $\mathbf{A}^k$ of possible actions that can be taken at time step k, which includes all possible sets of (e_1, e_2, r) triplets.

[1]The computation of this amount assumes that content is downloaded in order, i.e., from the first to the last byte. It does not hold for p2p applications, however file transfers and multimedia streaming do behave this way.

There are $|\mathcal{B} \cup \mathcal{Z}||\mathcal{B} \cup \mathcal{Z}||\mathcal{R}_d \cup \mathcal{R}_u|$ such tuples and, thus, a total of $2^{|\mathcal{B}\cup\mathcal{Z}||\mathcal{B}\cup\mathcal{Z}||\mathcal{R}_d\cup\mathcal{R}_u|}$ possible actions $\mathbf{a}^k \in \mathbf{A}^k$. Some of these actions can be discarded as meaningless, e.g., allocating RBs to a UE that has already completed its transfer. Others, e.g., having a UE receive from more than one endpoint in the same time step, or receiving a content while transmitting, are ruled out by technology constraints [20]. Furthermore in III-B, we laid down scenario-imposed rules regarding which links can use which sets of RBs, which also eliminate a great deal of triplet combinations. However, the very fact that the size of $\mathbf{A}^k$ grows exponentially with the number of UEs, BSs and RBs makes a standard dynamic programming model not scalable. For a similar reason, the evaluation of $\mathbf{V}$ stemming from $\mathbf{A}^k$ is exceedingly cumbersome. Indeed, one should consider all possible system evolutions starting from the current state, by selecting at each future time step the optimal action. Thus, we resort to ADP and propose the algorithms below so as to efficiently generate and rank actions, hence find a solution with low computational complexity.

B. The ADP solution

Recall that the immediate cost $\mathbf{C}$ of each action can be evaluated with very low complexity, thanks to Algorithms 1 and 2. Thus, in order to ensure scalability, it is sufficient to act along two directions: (i) making the number of actions to be evaluated at each time step smaller and independent of the number of UEs and BSs, and (ii) reducing the complexity of evaluating the future cost $\mathbf{V}$ of an action. Of course, it is not possible to achieve such a result while keeping the optimality guarantee. However, such an approach has been shown to be very effective [26, Ch. 1], as also confirmed by our performance evaluation in Sec. V. Below, we describe how we tackle the two issues.

1) Reducing the action space: The procedure described here is performed in the second step of the dynamic programming model in Fig. 3 (b), where we define the set of all possible actions $\mathbf{A}^k$. Considering that the subsequent steps in the model have to be performed for every action $\mathbf{a}^k$ in the set, reducing this set implies reduction in the complexity of the procedure as a whole.

To do so, we define an auxiliary action space $\tilde{\mathbf{A}}^k$, whose size is much smaller than the original action space $\mathbf{A}^k$ and, more importantly, does not grow with the number of UEs or BSs. Then, we show a deterministic (and computationally efficient) way to map an action $\tilde{\mathbf{a}}^k \in \tilde{\mathbf{A}}^k$ of the auxiliary action space into an action $\mathbf{a}^k \in \mathbf{A}^k$. It follows that the actions we evaluate (steps 5–7 in Fig. 3 (b)) are only those $\mathbf{a}^k \in \mathbf{A}^k$ that have a correspondence in $\tilde{\mathbf{A}}^k$.

Algorithm 3 Mapping α-triplets into actions

Require: $\tilde{\mathbf{a}}^k = (\alpha_M, \alpha_m, \alpha_D)$

$\mathcal{W} \leftarrow \{z \in \mathcal{Z} \text{ s.t. } \exists c \in \mathcal{C}_d \cup \mathcal{C}_u \colon w_c(z) < k \wedge h_c^k(z) < l_c\}$
sort $\mathcal{W}$ **by** $w_c(z)$
for all $z \in \mathcal{W}$ **do**
  for all e, r **do**
    compute $y_{r,c}^k(e, z), \forall c \in \mathcal{C}_d$
    compute $y_{r,c}^k(z, e), \forall c \in \mathcal{C}_u$ (Algorithm 2)
    if $e \in \mathcal{B}$ **then**
      if $r \in \mathcal{R}_d$ **then**
        $e_1 \leftarrow e$, $e_2 \leftarrow z$
        $\sigma(e_1, e_2, r) \leftarrow \sum_{c \in \mathcal{C}_d} y_{r,c}^k(e_1, e_2)$
      if $r \in \mathcal{R}_u$ **then**
        $e_1 \leftarrow z$, $e_2 \leftarrow e$
        $\sigma(e_1, e_2, r) \leftarrow \sum_{c \in \mathcal{C}_u} y_{r,c}^k(e_1, e_2)$
      if $e \in \mathcal{B}_M$ **then**
        $\sigma(e_1, e_2, r) \leftarrow \sigma(e_1, e_2, r) \cdot \alpha_M$
      if $e \in \mathcal{B}_m$ **then**
        $\sigma(e_1, e_2, r) \leftarrow \sigma(e_1, e_2, r) \cdot \alpha_m$
    if $e \in \mathcal{Z}$ **then**
      $e_1 \leftarrow e$, $e_2 \leftarrow z$
      if $r \in \mathcal{R}_u$ **then**
        $\sigma(e_1, e_2, r) \leftarrow \sum_{c \in \mathcal{C}_d} y_{r,c}^k(e_1, e_2)$
      else
        $\sigma(e_1, e_2, r) \leftarrow 0$
      $\sigma(e_1, e_2, r) \leftarrow \sigma(e_1, e_2, r) \cdot \alpha_D$
  $e_1^\star, e_2^\star \leftarrow \arg\max_{e_1, e_2} \sum_r \sigma(e1, e2, r)$
  $r^\star \leftarrow \arg\max_r \sigma(e_1^\star, e_2^\star, r)$
  $t_{curr} \leftarrow 0$, $t_{new} \leftarrow 0$
  for all $(e_t, e_r, \rho) \in \mathbf{a}^k$ and $c \in \mathcal{C}_d \cup \mathcal{C}_u$ **do**
    compute $\delta_\rho^k(e_t, e_r)$ **and** $y_{\rho,c}^k(e_t, e_r)$ (Algorithms 1-2)
    $t_{curr} \leftarrow t_{curr} + y_{\rho,c}^k(e_t, e_r)$
  for all $(e_t, e_r, \rho) \in \mathbf{a}^k \cup (e_1^\star, e_2^\star, r^\star)$ and $c \in \mathcal{C}_d \cup \mathcal{C}_u$ **do**
    compute $\delta_\rho^k(e_1, e_2)$ **and** $y_{\rho,c}^k(e_t, e_r)$ (Algorithms 1-2)
    $t_{new} \leftarrow t_{new} + y_{\rho,c}^k(e_t, e_r)$
  if $t_{new} > t_{curr}$ **then**
    $\mathbf{a}^k \leftarrow \mathbf{a} \cup (e_1^\star, e_2^\star, r^\star)$
return $\mathbf{a}^k$

To determine the auxiliary action space, we proceed as follows: we ask ourselves what kind of choice has the highest relevance in a system such as ours. The most significant one is to rank transfer paradigms, i.e., using macroBSs, microBSs or D2D – and test which combination of them yields the highest throughput and carries the least interference. We thus rank the "importance" of each paradigm by a triplet of real values $\alpha_M, \alpha_m, \alpha_D \in [0, 1]$. These values indicate which endpoints should be preferably used, as shown in Algorithm 3, and each triplet represents an auxiliary action $\tilde{\mathbf{a}}^k$. For the set of auxiliary actions to be manageable, we need to discretize each value in the α triplet. The set $\tilde{\mathbf{A}}^k$ is thus finite and we can control its size by choosing the granularity of each α. This is our tuning knob for scalability purposes.

Algorithm 3 takes as input an action $\tilde{\mathbf{a}}^k$ and maps it onto an action $\mathbf{a}^k$ (line 36). Its logic is straightforward: we serve requesting users, starting from the neediest ones, selecting the most effective endpoint.

More specifically, in line 1, we identify the set $\mathcal{W} \subseteq \mathcal{Z}$ of active requesting users, i.e., users with an incomplete transfer. This set is sorted (line 2) by the want-time $w_c(z)$, so the earliest requests are given higher priority. Then, for each user $z \in \mathcal{W}$, we loop over the potential endpoints e and RBs r that e and z may use to transfer content (line 4), and asses all potential combinations of z, e and r that make sense and are allowed in the scenario. Note that we have explicitly reflected all the scenario-imposed constraints in lines 7–20. Most importantly in line 20, we set the constraint on which set of RBs may be used for D2D communications. Namely, since we consider two different scenarios, this can be either the uplink ($\mathcal{R}_u$) or the downlink ($\mathcal{R}_d$) RB set. We can switch between the two scenarios by simply changing this line in the algorithm.

Next, for each combination we define the transmitting endpoint e_1 and the receiving endpoint e_2, which depend on the traffic direction. For each triplet (e_1, e_2, r), we compute a score σ, which is initialized to the amount of data (computed by Algorithm 2) that e_1 may transfer to e_2. Each score σ is then weighted by the α-coefficient corresponding to the type of endpoint e, setting priorities over the different possible data transfer paradigms. As an example, the α-coefficients give us leverage to encourage D2D transfers by setting a high value for α_D, or to limit the usage of macroBSs to users that have no other means to be served by setting α_M to a low value. In line 25, we select the pair (e_1, e_2) corresponding to the highest sum of scores over all possibles RBs. Notice that by selecting only one pair in line 25, we honor the technology constraint by

which each user can either download, upload or serve data from at most one source and to at most one destination in a given time step. In the following line, we assign to the endpoint pair $(e_1^\star, e_2^\star)$ the RB that maximizes their σ score. However, before conclusively including the selected triplet $(e_1^\star, e_2^\star, r^\star)$ in the allocation strategy yielded by $\mathbf{a}^k$, we check whether it increases the total amount of data transferred in the network or not (lines 28–34). While verifying that, we resort again to the interference-aware Algorithms 1 and 2 to compute the δ and y values. If the amount of data grows, the triplet is added to action $\mathbf{a}^k$ (line 35). In conclusion, we stress that the size of the auxiliary action space $\tilde{\mathbf{A}}$ is small and it is independent of the number of UEs and BSs. Thus, we have achieved our scalability goal.

Algorithm 4 Estimating the value of a state

Require: $\mathbf{s}^k$, $\mathbf{a}^k$, $\{\mathbf{a}^{k+1}, \dots, \mathbf{a}^K\}$

1: $v \leftarrow 0$
2: **for** $q = k+1 \rightarrow K$ **do**
3: **for all** $(e_1, e_2, r) \in \mathbf{a}^q$ **do**
4: **compute** $\delta_r^q(e_1, e_2)$ **using** Algorithm 1
5: **for all** (e_1, e_2): $\exists\, \delta_r^q(e_1, e_2) > 0$ **do**
6: **for all** $c \in \mathcal{C}_d \cup \mathcal{C}_u$: $w_c(z) \leq k \wedge h_c^q(z) < l_c$ **do**
7: **compute** $\chi_c^q(e_1, e_2)$ **using** Algorithm 2
8: $\hat{h}_c^{q+1}(e_2) \leftarrow \hat{h}_c^q(e_2) + \chi_c^q(e_1, e_2)$, $\forall c \in \mathcal{C}_d$
9: $\hat{h}_c^{q+1}(e_1) \leftarrow \hat{h}_c^q(e_1) + \chi_c^q(e_1, e_2)$, $\forall c \in \mathcal{C}_u$
10: **compute** $\mathbf{C}(\mathbf{s}^q, \mathbf{a}^q)$
11: $v \leftarrow v + \mathbf{C}(\mathbf{s}^q, \mathbf{a}^q)$
12: **return** $\mathbf{V}(\mathbf{s}^k, \mathbf{a}^k) = v$

2) Evaluating the state values: To evaluate an action, it is important to compute the value of the state $\mathbf{s}^{k+1}$ the action leads to. As already stated, the value of a state corresponds to the sum of the costs we will pay due to future actions, if these are chosen optimally. Clearly, if we set $\mathbf{V}(\mathbf{s}^k, \mathbf{a}^k) = 0$ for all actions, i.e., we select the action that seems more profitable at the current step, we end up adopting a greedy strategy. However, in network scenarios where D2D is allowed, accounting for future actions may be of particular relevance: e.g., transmitting to some users at a faster pace, so that they can act as serving UEs later, may be beneficial to the whole network.

It follows that we need to compute the value function $\mathbf{V}$ accurately enough, while keeping the

complexity low. To do so, we resort to the methodology typically used in ADP. Such methodology [26, Ch. 9] implies that, at each step k, we fix the sequence of future actions, starting from state $\mathbf{s}^{k+1}$. We apply this procedure to our problem as described in Algorithm 4.

The algorithm takes as input: (i) the current state $\mathbf{s}^k$ and the current action to be evaluated $\mathbf{a}^k$ (i.e., the two elements determining next step $\mathbf{s}^{k+1}$), and (ii) the future actions that we expect will be taken. In order to compute the latter, we start by assuming that the conditions experienced by a user do not change during the transfer time. This is a fair assumption since, as shown by Figs. 4(b),(c) and 5(b),(c) in Sec. V, users complete their transfer in hundreds of ms, hence the movement of pedestrian users during content transfer is negligible. Also, note that the procedure for computing the value function $\mathbf{V}$ is repeated at every time step k. We feed such information to a machine learning model, so as to compute future actions $\{\mathbf{a}^{k+1}, \ldots, \mathbf{a}^K\}$ [26, Ch. 9].

Next, we exploit the estimated information on the system to compute, at each future time step $q > k$, the δ and χ values for each communication foreseen by action $\mathbf{a}^q$ (lines 4 and 7) using the algorithms presented in Sec. IV-A.

In lines 8 and 9, for each step $q > k$, given the previous state and the χ values, we apply (III.1) and update the amount of data of content c, h_c^q, that each downloader/uploader can retrieve/transfer until step q. Then, we use the quantities χ and h to evaluate the cost of action $\mathbf{a}^q$. Note that we cannot predict future user requests, however, due to the short time span before a transfer completion, their number is limited. Additionally, their deadline will be further away in time,[2] hence their impact is minimal (see (IV.1)). At last, $\mathbf{V}(\mathbf{s}^k, \mathbf{a}^k)$ is calculated by summing all future cost contributions (line 12).

C. Solution complexity

As mentioned before, to meet the scalability requirements, the algorithms must be of sufficient low-complexity. Applying ADP, this requirement is indeed met. With reference to Fig. 3 (b), and assuming that the dominant factor is the number of users, the complexity is as follows: step (2), $O(2^{|\mathcal{Z}|})$ with plain dynamic programming, which reduces to $O(|\mathcal{Z}|)$ using Algorithm 3. Steps (3) and (4), $O(|\mathcal{Z}|)$. Step (5), $O(1)$. Step (6), $O(|\mathbf{A}|^k)$ with plain dynamic programming, which reduces to $O(|\mathcal{Z}|)$ with Algorithm 4.

[2]Recall that Algorithm 4 is repeated at every time step k.

V. RESULTS

We evaluate the performance of our approach in the two-tier scenario that is typically used within 3GPP for LTE network evaluation [27]. The scenario comprises a service network area of 12.34 km^2, covered by 57 macrocells and, unless otherwise specified, 228 microcells. Macrocells are controlled by 19 three-sector BSs; the macroBSs inter-site distance is set to 500 m. MicroBSs are deployed over the network area, so that there are 4 non-overlapping microcells per macrocell. A total of 3420 users are present in the area. In particular, in order to have a higher user density where microcells are deployed, 10 users are uniformly distributed within 50 m from each microBS. The rest of the users are uniformly distributed over the remaining network area. Users move according to the cave-man model [28], with average speed of 1 m/s.

In line with [21], [27], [29] we assume a transmitting power of 43 dBm for macroBSs, and 30 dBm for microBSs, and antenna height values of 25 m and 10 m, respectively. For the macroBS antenna we further assume the antenna gain to be 14 dBi and the maximum attenuation 20 dB, while the microBS antennas are omnidirectional with 0 dBi gain. UE transmitting power is controlled, using the following values for the cell and user configuration parameters: $P_{max}|_{dB} = 23$ dBm, $P_o|_{dB} = -70$ dBm, $\rho = 0.7$ and $\Delta_{TF} = 0$. Closed loop control is disabled. We assume the antenna height of the UE to be 1.5 m.

All network nodes operate over a 10 MHz band at 2.6 GHz for downlink and at 2.5 GHz for uplink, hence we have $|\mathcal{R}_d| = |\mathcal{R}_u| = 50$ RBs.

As already mentioned, the signal propagation for BS-UE links is modelled according to ITU specifications for urban environment [21] and for D2D according to the specifications in [24], while the SINR is mapped onto per-RB throughput values using the experimental measurements in [25]. The noise power level is set at -174 dBm/Hz, according to [21]. The energy consumption of the network nodes is instead computed according to [29].

Users may require content for download or upload from a set of 21 different items, belonging to three categories: e-books (10 items), videos (10 items), or viral content (1 item). Their characteristics and intervals between user requests are summarized in Table III. We highlight that video and viral items have stricter constraints on delivery time. Content items from the e-book and video category may be requested either for download or upload. The viral content item, on the other hand, is modeled as being in high demand during a narrow time interval to

TABLE III

CONTENT TYPES

Feature	eBook	Video	Viral
Dowload request rate [items/s]	$1e-3$	$1e-3$	$5e-2$
Upload request rate [items/s]	$0.5e-3$	$0.5e-3$	–
Size [Mbit]	12	3	3
Deadline [steps]	4000	1000	1000
Request interval [steps]	1–1000	1–1000	41–60

mimic content becoming suddenly popular, hence users may request it only for download. We consider the traffic load to be asymmetric with upload traffic being half the download one.

The scheduling decisions issued by the AC are valid for one subframe, therefore the resource allocation algorithm is performed every 1 ms. While applying our ADP approach, we consider that the values of the $\alpha_M, \alpha_m, \alpha_D$ parameters, are discretized as $\{0.1, 0.2, \ldots, 1\}$. Additional experiments with values exhibiting finer granularity have shown negligible improvement.

We will additionally consider two different D2D deployment scenarios, depending on which of the two RB sets (downlink or uplink) D2D communications are allowed to share with the cellular infrastructure. We shall refer to them as the DL scenario, when D2D operates in the downlink portion of the spectrum, and the UL scenario when it operates in the uplink portion.

We compare our approach against a system implementing the 3GPP eICIC with a microcell bias of 15 dB and the ABS model where macroBSs are silent in 1 out of every 2 subframes [30]. In the latter, D2D mode is not supported and UEs connect to the BS from which they receive the strongest pilot signal. At the BSs, traffic is scheduled according to the proportional fairness (PF) algorithm, which is standard in today's LTE networks [20]. In the following, we will refer to this benchmark scenario as PF.

The comparison between ADP and PF for both scenarios is shown in Figs. 4–6. In particular, Fig. 4(a) and Fig. 5(a) show that ADP allows the transfer of more data than the state-of-the-art (around 7%), while using over 40% less energy. Such a gain can be attributed to the lower usage of macrocells (characterized by very high transmit power), in favour of microcells and D2D. In the plot, the possible endpoints are differentiated by using different colors: black for macroBSs, gray for microBSs and red for UEs. Note that the energy consumption due to D2D mode is

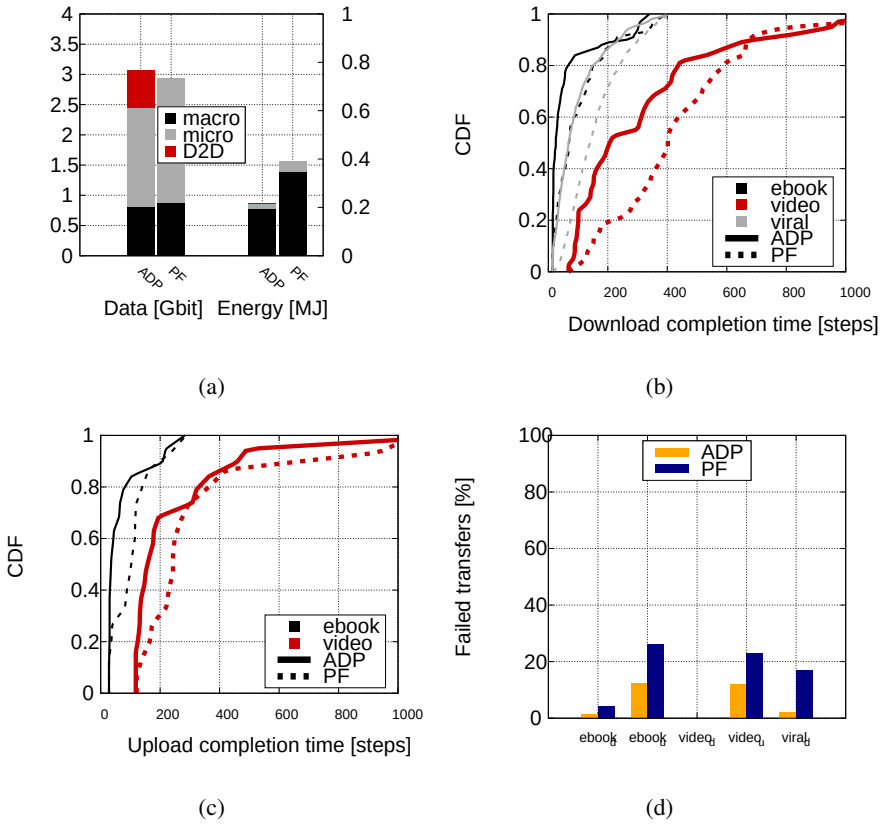


Fig. 4. DL scenario. ADP vs. PF: (a) total amount of transferred data and consumed energy, (b) CDF of the download completion time, (c) CDF of the upload completion time, (d) failed transfers.

negligible and can be barely seen in the plot. Also, under both ADP and PF, transmissions from microBSs are more efficient than those from macroBSs, as the former carry a higher amount of data at a much lower energy cost.

Figs. 4(b)–(c) and Figs. 5(b)–(c) depict the CDF of the completion time of successful downloads (b) and uploads (c), for the different content categories (differentiated by the different colors). A download/upload is successful if it can be completed by the corresponding deadline. Comparing ADP (solid lines) to PF (dotted lines), we notice that in general ADP outperforms PF in terms of ensuring faster content delivery both for uploads and downloads, regardless of the D2D scenario. This is especially true for viral and video content, which have stricter deadlines. Indeed, in ADP, the cost $\mathbf{C}$ in (IV.1) accounts for content deadlines, giving higher priority to those content transfers that are closer to their completion deadline. In particular, for video content, ADP is able to provide a far lower completion time for at least 90% of the successful

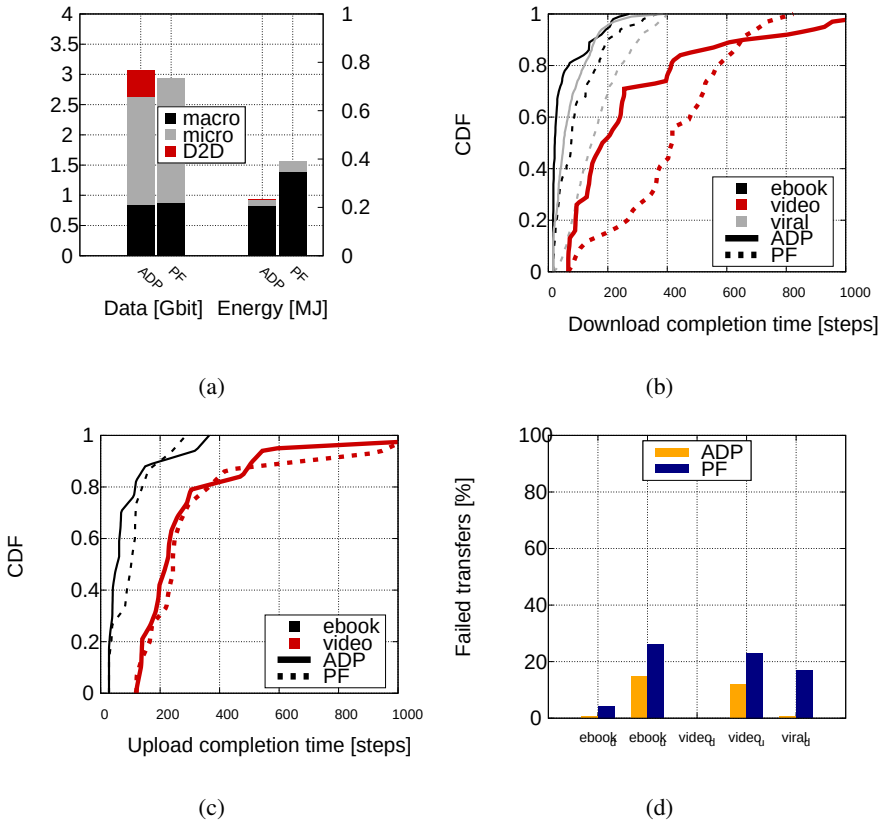
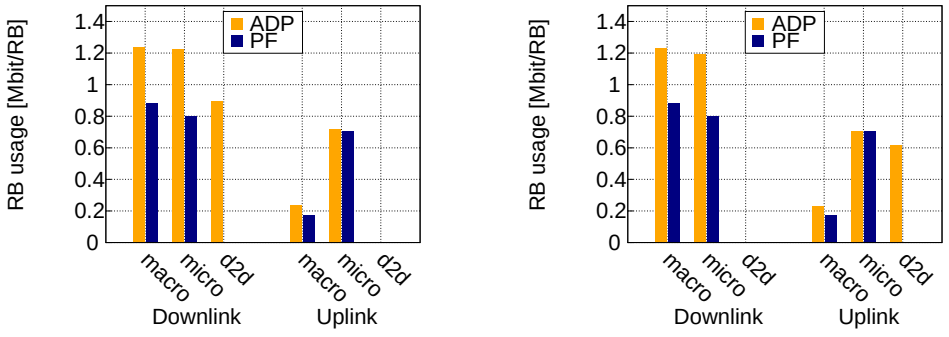


Fig. 5. UL scenario. ADP vs. PF: (a) total amount of transferred data and consumed energy, (b) CDF of the download completion time, (c) CDF of the upload completion time, (d) failed transfers.

downloads, than PF. These results are closely related to the percentage of failed transfers shown in Fig. 4(d) and Fig. 5(d), where the performance of ADP and PF are differentiated by using the orange and blue color, respectively. Clearly, ADP guarantees higher success rates than PF for all content categories and in both traffic directions, but the contrast is most dramatic for viral content. This can also be attributed to the fact that D2D is heavily used by ADP to deliver this type of content.

Fig. 6 highlights the improvement that ADP offers in terms of usage of radio resources, both downlink and uplink, compared to PF. Observe that, on average, in downlink ADP can transmit much more data per RB than PF. We observe a gain of around 35-40% in RB usage efficiency for macrocells and around 50% gain for microcells. The reason for such behavior is that our interference-aware approach is far more efficient in matching potential endpoints than the PF based system. In other words, ADP scheduling yields higher values of SINR at the receiving



(a) (b)

Fig. 6. RB usage: (a) DL scenario, (b) UL scenario.

endpoints, hence higher data rates per RB. In the DL scenario, the amount of data per RB is especially high for D2D links, which is remarkable considering that UEs transmit at significantly lower power than microBSs or macroBSs.

By looking at Figs. 4–6, we also notice the differences in performance between the DL and UL scenario. In terms of RB usage, the values of data transferred per RB are in general higher in the DL scenario than in the UL scenario. This is mainly due to the fact that the overall achievable data rates for a certain value of SINR are higher in the downlink than the uplink, according to the experimental measurements used in our evaluation. Nonetheless, D2D in the UL scenario is significantly more efficient in using RBs compared to UE-macroBS links, and comparable to UE-microBS links. This impacts also the overall amount of data that ADP is able to transfer through D2D in the UL scenario compared to the DL scenario, as can be noticed by comparing Fig. 4(a) and Fig. 5(a). While the overall amount of transferred data is similar in both scenarios, the amount transferred by D2D is slightly higher in the DL. In the UL scenario, the slack is picked up by microcells, which causes a slight increase in energy consumption. We therefore conclude that the UL and DL scenarios provide similar performance in current traffic load conditions, however the DL scenario will become preferable as the upload and the download traffic tend to even.

In the same scenarios as above, we now halve the number of microcells from 228 to 114, i.e., 2 microcells per macrocell. The most noticeable effect is that, with ADP, D2D communication steps up to compensate for the missing microBSs, as shown in Figs. 7(a) and (b). Instead, PF falls

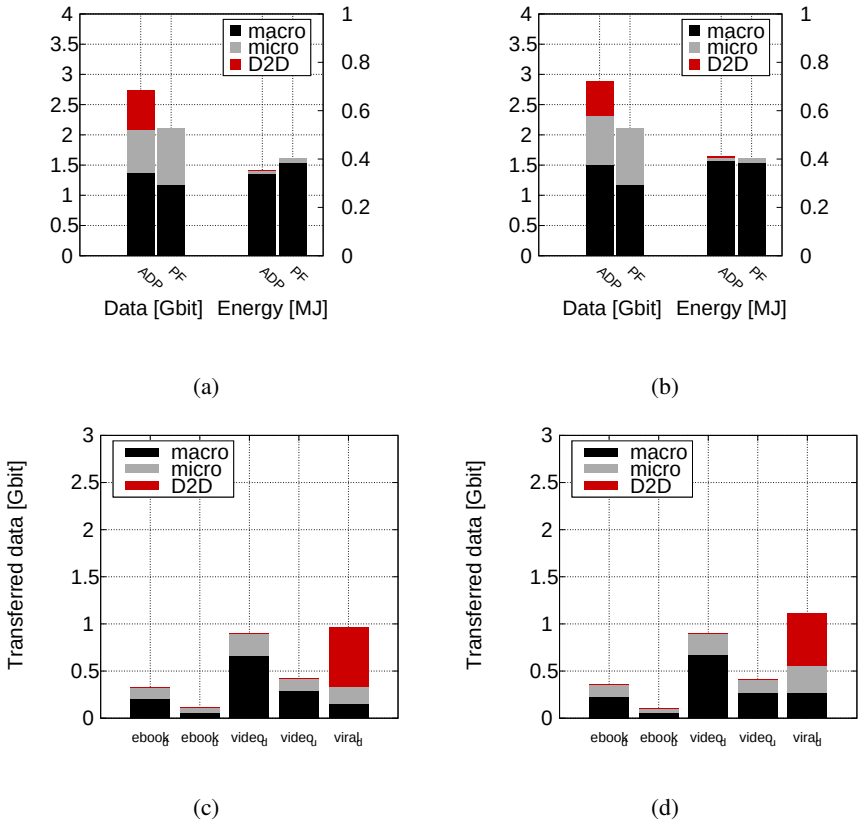


Fig. 7. Halving the number of microcells: amount of transferred data and consumed energy, in the DL scenario (a), and in the UL scenario (b); amount of transferred data by ADP, in the DL scenario (c), and in the UL scenario (d).

considerably short of providing the same throughput as before. Indeed, comparing these plots to Figs. 4(a) and 5(a), ADP exhibits a mere 10% drop in transferred data in the DL scenario, and around 6% drop in the UL scenario, with respect to 28% for PF. Energy consumption increases for both approaches, achieving similar levels for both. As expected, ADP tends to favour content with stricter time constraints (viral and video), at the expense of e-books. For sake of brevity, we omit plots comparing other metrics, which however confirm the above observations.

Summary. Thanks to a lesser usage of macrocells, our proposed scheme enables the transfer of 7% more data than PF, at an energy cost that is reduced by over 40%. ADP also provides a completion time that is significantly lower than PF, for most of the data transfers. Particularly striking is the success rate of viral content delivery: thanks to D2D communications, ADP exhibits 0-2% failures versus 18% of PF. As for the efficiency in RB usage, the interference-aware scheduling performed by ADP leads to an improvement of 35-40% over PF for macrocells and

of around 50% for microcells. Another interesting finding is the advantage of accommodating D2D communications in the uplink or in the downlink bandwidth. Under current traffic load conditions, the two options are equally effective. However, as upload and download traffic tend to even out, using the downlink bandwidth will be preferable. Finally, D2D is found to be an effective, low-energy and low-cost replacement for microcell deployment.

VI. Conclusions

We considered a 2-tier, LTE-based network, supporting D2D communication. We devised an interference-aware solution to the problem of uplink and downlink radio resource allocation, to efficiently accommodate the download and upload traffic in such a complex network. For each traffic request, our algorithm selects which endpoint should serve a user, and allocates the radio resources for such communication, in an energy-aware and spectrum-efficient manner. In particular, we presented approximate dynamic programming algorithms to generate and rank possible resource allocation decisions. This way, we obtained a low-complexity solution that can deal with realistic, large-scale scenarios. In addition, we evaluate two possible approaches to in-band, network-controlled D2D implementation, and assess the performance of our solution for both cases. Results show that our solution combined with D2D outperforms the state-of-the-art used in today's networks both in terms of overall throughput and user experience. Furthermore, we highlight that D2D mode can be a valid, low-cost alternative to microcells in supporting traffic with little energy consumption.

References

[1] L. Xingqin, J. Andrews, A. Ghosh, R. Ratasuk, "An overview of 3GPP device-to-device proximity services," *IEEE Comm. Mag.*, vol. 52, no. 4, 2014.

[2] L. Lei, X. Shen, M. Dohler, C. Lin, Z. Zhong, "Queuing models with applications to mode selection in device-to-device communications underlaying cellular networks," *IEEE Trans. on Wireless Communications,* vol. 12, no. 12, 2013.

[3] L. Song, D. Niyato, Z. Han, E. Hossain, "Game-theoretic resource allocation methods for device-to-device (D2D) communication," *IEEE Wireless Comm. Mag.*, in press, http://arxiv.org/abs/1403.5723.

[4] 3GPP RP-122009, "Study on LTE device to device proximity services," in 3GPP TSG RAN Meeting #58, Dec. 2012.

[5] X. Lin, J. G. Andrews, A. Ghosh, "A comprehensive framework for device-to-device communications in cellular networks," http://arxiv.org/abs/1305.4219.

[6] Qualcomm, "Creating a digital sixth sense with LTE Direct", July 2014.

[7] H. Elsawy, E. Hossain, D. I. Kim, "HetNets with cognitive small cells: user offloading and distributed channel access techniques," *IEEE Comm. Mag.*, vol. 51, no. 6, 2013.

[8] G. Boudreau, J. Panicker, N. Guo, R. Chang, N. Wang, S. Vrzic, "Interference coordination and cancellation for 4G networks," *IEEE Commun. Mag.*, vol. 47, no. 4, 2009.

[9] 3GPP Std. Rel. 10, "Enhanced Inter-Cell Interference Control (ICIC) for non-Carrier Aggregation (CA) based deployments of heterogeneous networks for LTE," RP-100383, June 2013.

[10] S. Deb, P. Monogioudis, J. Miernik, J.P. Seymour, "Algorithms for enhanced inter-cell interference coordination (eICIC) in LTE HetNets," *IEEE/ACM Trans. on Networking,* vol. 22, no. 1, 2014.

[11] Freescale white paper, "Next-generation wireless network bandwidth and capacity enabled by heterogeneous and distributed networks," July 2013.

[12] Ericsson white paper, "It all comes back to backhaul," 2012.

[13] G. Fodor, E. Dahlman, G. Mildh, "Design aspects of network assisted device-to-device communications," *IEEE Comm. Mag.*, vol. 50, no. 3, 2012.

[14] L. Lei, Z. Zhong, C. Lin, X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Comm.*, vol. 19, no. 3, 2012.

[15] M. Zulhasnine, C. Huang, A. Srinivasan, "Efficient resource allocation for device-to-device communication underlaying LTE network," *IEEE WiMob*, 2010.

[16] C.-H. Yu, K. Doppler, C. B. Ribeiro, O. Tirkkonen, "Resource sharing optimization for device-to-device communication underlaying cellular networks," *IEEE Trans. on Wireless Comm.*, vol. 10, no. 8, 2011.

[17] F. Malandrino, C. Casetti, C.-F. Chiasserini, "A fix-and-relax model for heterogeneous LTE-based networks," *IEEE Mascots*, 2013.

[18] F. Malandrino, C. Casetti, C.-F. Chiasserini, Z. Limani "Fast Resource Scheduling in HetNets with D2D Support," *IEEE INFOCOM*, 2014.

[19] F. Malandrino, C. Casetti, C.-F. Chiasserini, Z. Limani "Uplink and Downlink Resource Allocation in D2D-Enabled Heterogeneous Networks," *IEEE WCNC Hetnets*, 2014.

[20] S. Sesia, I. Toufik, M. Baker (Eds.), *LTE – The UMTS long term evolution: From theory to practice*, Wiley, 2009.

[21] ITU-R, "Guidelines for evaluation of radio interface technologies for IMT-Advanced", *Report ITU-R M.2135-1,* Dec. 2009.

[22] "C-RAN - The road towards green RAN", China Mobile Research Institute White Paper, 2011.

[23] 3GPP Technical Specification 36.213, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," Release 11, 2013.

[24] J. Meinilä, P. Kyösti, L. Hentilä, T. Jämsä, E. Suikkanen, E. Kunnari, M. Narandžić, "D5.3: WINNER+ final channel models," *Wireless World Initiative New Radio WINNER+*, 2010.

[25] D. Martín-Sacristán, J. F. Monserrat, J. Cabrejas-Peñuelas, D. Calabuig, S. Garrigas, N. Cardona, "3GPP long term evolution: Paving the way towards next 4G," *Waves*, 2009.

[26] W. B. Powell, *Approximate dynamic programming*, Wiley, 2011.

[27] 3GPP Technical Report 36.814, "Further advancements for E-UTRA physical layer aspects," 2010.

[28] D. J. Watts, *Small worlds: The dynamics of networks between order and randomness,* Princeton University Press, 1999.

[29] FP7 IP EARTH project, "Deliverable D2.3: Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," https://www.ict-earth.eu/.

[30] A. Ghosh, N. Mangalvedhe, R. Ratasuk, B. Mondal, M. Cudak, E. Visotsky, T. A. Thomas, "Heterogeneous cellular networks: From theory to practice," *IEEE Comm. Mag.*, vol. 50, no. 6, 2012.