Joint resource allocation and offloading strategies in cloud enabled cellular networks

Mohamed Kamoun Wael Labidi Mireille Sarkiss
mohamed.kamoun@cea.fr wael.labidi@cea.fr mireille.sarkiss@cea.fr
CEA, List Communication System Laboratory 91191 Gif sur yvette Cedex

***Abstract*—The numerous features installed in recent mobile phones opened the door to a wide range of applications involving localization, storage, photo and video taking and communication. A significant number of applications involve user generated content and require intensive processing which limits dramatically the battery lifetime of featured mobile terminals. Mobile cloud computing has been recently proposed as a promising solution allowing the mobile users to run computing-intensive and energy parsimonious applications. This new feature requires new functionalities inside the cellular network architecture and needs appropriate resource allocation strategies which account for computation and communication in the same time. In this paper we present promising options to upgrade 4G architecture to support these new features. We also present two resource allocation strategies accounting for both computation and radio resources. These strategies are devised so that to minimize the energy consumption of the mobile terminals while satisfying predefined delay constraints. We compare online learning based solutions where the network adapts dynamically to the application that is run on mobile terminals, and pre-calculated offline solutions which are employed when a certain level of knowledge about the application and the channel conditions is available at the network side. We show, that even with imperfect knowledge about the application, pre-calculated offline strategies offer better performance in terms of energy consumption of mobile terminals.**

I. Introduction

Recent mobile terminals include multiple features like localization, photo and video taking as well as significant storage capabilities. These features paved the way to multiple types of applications involving these functionalities. In a significant number of cases, these applications require intensive computation which results in a significant degradation of battery lifetime. In the same time cloud computation and storage platforms have emerged recently with pay-as-you-need service model to offer economical access to computation and storage resources. It has been shown that cloud computation services can offer significant energy savings to mobile users [8]. Several frameworks have been developed to enable cloud computing in mobile devices (cf [5], [7] and references therein). Most of these frameworks assumed that the computation resources are located outside the mobile operator network and such option suffers from non negligible delays which limits the range of applications that can leverage remote resources.

Recently it has been proposed to install computation resources in the edge of cellular networks and more precisely in the eNBs or home eNBs in an LTE based network [13]. This option allows the mobile operator to offer an added value service with better latency figures and to tune the cloud architecture to the popular applications that are commonly run by mobile users so that to alleviate their power consumption [8]. Besides, such architecture allows the operator to save bandwidth resources on the backhaul compared to the scenario where the cloud service is located outside its network. These advantages can be leveraged only through an appropriate resource allocation strategy that accounts for both radio and computing resources in the same time. The aim of such strategy is to partition the mobile application code into elementary procedures and to offload certain parts so that to minimize the power consumption of the terminals while guaranteeing proper functioning. The partitioning strategy needs to account for the channel conditions and also for the availability of radio resources.

Code partitioning strategies for mobile applications with power optimization objective have been addressed in several contributions. The power consumption of the cloud infrastructure has been considered in [10] where the authors proposed job allocation strategies so that to satisfy offloading demands coming from mobile users.

For the terminal side, various strategies has been investigated. The authors in [14] considered the optimization of terminal hardware setting (CPU frequency) along with the data rate of the wireless link so that to make best decision between two options: internal processing and offloading. Others optimization frameworks based on code partitioning have been proposed in [6] and [3]. With, these frameworks, the application is considered as a consequent calls to elementary functions. Only a subset of elementary functions can be offloaded. For each offloadable function, a decision is made between internal processing and offloading. The decision is made so that to minimize the total power consumed to run the whole application. The models that are considered in the aforementioned contributions allow for two options only: offloading or internal processing. Also, they only consider instantaneous optimization. In this paper we consider an optimization framework for offloadable applications involving repetitive processing of user generated content. We consider the scenario where the mobile network (more precisely the eNB) is responsible for joint allocation of radio and computing resources. This allocation is performed based on channel quality indicators and queue status reports that are sent by the terminal, and is designed so that to satisfy an average delay constraint. The proposed optimization framework considers between three decisions: internal processing, offloading and idle. A prediction of channel conditions is taken into account so that to make the best decision in terms of average power consumption while respecting a predefined average delay constraint. Two joint radio resource allocation and offloading strategies are investigated. The first approach which is based on online learning, adapts dynamically to the application properties and the channel conditions. The second strategy called pre-calculated offline is based on prior knowledge of the application properties and the statistical behaviour of radio environment. Both strategies are compared and bench-marked with single decision

978-1-4673-6432-4/15/$31.00 ©2015 IEEE

strategies. Section II presents the system model and the required upgrade to the 4G architecture to support the offloading features. Section III presents two radio resource and offloading strategies for mobile applications that process user generated data. These strategies are based on offline and online dynamic programming tools and allows to account for application and the statistical channel properties. Section IV compares the aforementioned strategies and present benchmark results of these strategies with respect to local processing and all offload approaches.

II. System model and architecture design

A. Offloading models for mobile users

We consider a 4G cellular network where the eNodeBs (eNB) have been equipped with computation and storage resources. These resources are employed to run computation intensive applications on behalf of the mobile terminals that are attached to the network. Compared to the architecture where computation resources are available in an independent cloud provider, the adopted option allows better latency performance. To take advantage of the computation resources installed in the eNBs, mobile applications need to include native offloading features. Representative examples of applications that can benefit from offloading are those that process user generated data like photo and video processing, speech recognition and speech synthesis. Typical implementation for these applications employ a remote procedure call (RPC) interface [9] where each procedure makes a decision between running a local implementation of the corresponding task and offloading the same task to a remote computing device. In what follows we will focus on this model.

B. Architectural components and signalling strategies

In order to leverage the computation resources located at the edge of the network (eNBs), appropriate signalling and architectural components have to be implemented in both realms: cloud computation infrastructure and radio access network. We propose to add a new software component called cloud computing management unit (CCMU) in the eNBs [13]. This entity controls the offloading decisions jointly with radio resource allocation. It will answer for offloading requests that are sent by mobile applications. Besides, an application layer channel has to be defined between the applications that are run on mobile terminals and the CCMU. This channel can employ offloading protocols such as LIPA and SIPTO which have been devised to offer a layer 3 link between mobile users and devises located in the local network to which the base stations are attached [12] [1].

Figure 1 draws a protocol stack using the LTE user plane stack which includes additional components and protocols. These main objective of these new components is to allow user applications to communicate with computation resources in the eNB. The establishment of a data plane link between the user and the eNB can be performed during the initial attachment of the user to its cell. Once this link is established, the user equipment can offload tasks to the computing unit within the eNB using

[1]Note that the term offloading for LIPA and SIPTO protocols is different from the offloading meant here (cf [12] for more details)

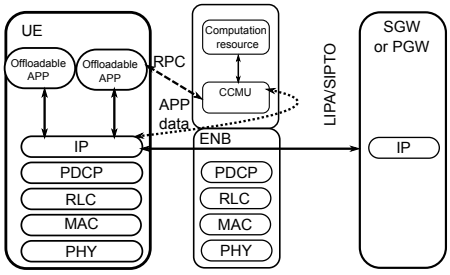

Fig. 1. Integration of cloud computation with LTE

a remote procedure call execution model. With this model, a program is divided into a finite or infinite series of procedure calls. These procedures are run by the mobile application and include offloading requests. Upon positive answer (offloading accepted), they wait until the result of is sent back by the CCMU. Upon negative answer (offloading demand rejected) they run a local implementation that runs on the user equipment (UE) processor. We assume that the mobile application processes data that is generated locally in the UE and that the CCMU has a complete knowledge of radio parameters and cloud computation resource availability. With cloud enabled eNBs, the CCMU performs job scheduling along with radio resource management. In the following we will focus on the design of joint allocation of computation resources and radio resources so that to minimize the energy footprint of the application while respecting a user defined quality of service (QoS) constraint.

III. Dynamic programming for joint allocation of computing and radio resources

A. Main assumptions and problem formulation

We consider a single cell, single user scenario where a UE is attached to a computing enabled eNB. The local data that is generated at the UE can be either processed locally or offloaded to the eNB. We assume that the decision to run local processing or to ask for offloading is made at regularly spaced epochs. A request is sent from the mobile application to the CCMU located in the eNB which will make a decision that is sent back to the terminal.

The epochs can be aligned with the frame duration parameters of the LTE network and with the times where radio scheduling messages are executed. The time slot between two successive actions is denoted T and the user content is generated according to a Poisson law with rate λ. The buffer size and the data that is generated are expressed in the same unit called here packet. At the beginning of each time slot, the UE sends a report on the channel quality that it experiences with respect to the serving eNB and also its buffer size. In the LTE standard, these data are sent through the CQI (channel quality information) and the BSR (buffer size report) messages [1]. We assume that the eNB knows the power consumption model of the UE. This information can be sent during the initial attachment. The eNB is assumed to have sufficient computation resources to offload the entire data that is generated by the user.

The main objective of the CCMU is to minimize the average power consumption of the user equipment while satisfying a quality of service constraint expressed in terms

of average delay experienced by the packets that are generated at the UE side before being processed.

Let $s_k = (b_k, x_k)$ be the last status report sent by the user equipment to the eNB at time k. b_k is the buffer status report and x_k is the channel quality information. It is easy to see that s_k is a discrete time Markov chain. Based on s_k, the eNB will send a scheduling decision a_k which can be one of three types: i)to offload a_k packets, ii) to process internally in the UE a_k packets iii)to stay idle. The decision will be made based on the statistical knowledge of the states that are experienced by the UE as well as the latest report reflecting the current state. Its aim is to minimize the average consumed power while satisfying pre-defined delay constraints. [2].

This scheduling problem corresponds to a constrained Markov decision process problem for which five quantities have to be defined:

- **State space** $\mathcal{S}$: the state space is given by the pairs $s = (b, x)$ (BSR and CQI in LTE). We assume that b is the exact buffer occupation of the UE and that x is obtained by quantizing the normalized fading state using a predefined quantization grid. The normalization is made with respect to the path-loss experienced between the UE and the eNB. The cardinality of $\mathcal{S}$ is given by $(1 + B) \times X$ where B is the maximal buffer size of the UE and X is the number of channel quantization intervals. For $s \in \mathcal{S}$ we denote by $x(s)$ and $b(s)$ the corresponding channel and buffer status respectively.
- **The action space** : the action space $\mathcal{A}$ comprises three main types of actions
  1) To offload a packets to the eNB $a \in [1..O]$. O is the maximum number of packets that can be offloaded within a time duration T.
  2) To process a packets internally $a \in [1...I]$. I is the maximum number of packets that can be processed by the user within a time duration T.
  3) To stay idle (neither local nor remote processing) and wait until the next decision $a = 0$

  The size of $\mathcal{A}$ is given by $O + I + 1$
- **Cost function** f : The cost function is the average power consumed for the processing of each packet. It will be expressed later.
- **Constraint functions** : The constraint functions correspond to the average delay experienced by each packet before being processed and to the overflow probability.
- **The transition probability matrix** given by $\mathcal{P}(s^{'}|s, a)$ which is the probability to go from state s to state $s^{'}$ when action $a \in \mathcal{A}$ is executed. We assume that channel realizations are independent and identically distributed. For simplicity we account only for the distribution of the quantized channel state denoted $\psi(x)$. $\mathcal{P}(s^{'}|s, a)$ is thus given by

$$\mathcal{P}(s^{'}|s, a) = \psi(x(s^{'}))e^{-\lambda}\frac{\lambda^{b(s^{'})-b(s)+a}}{(b(s^{'}) - b(s) + a)!}$$

The objective of the computing enabled eNB is to a find a strategy μ which minimizes the average consumed power while satisfying the average delay constraints. A strategy μ is a mapping between the state space $\mathcal{S}$ and the action space $\mathcal{A}$

The average cost of a strategy μ is the corresponding average cost incurred by the user equipment. It is given by:

$$f(\mu) = \lim_{N\to\infty}\frac{1}{N}\mathbb{E}_{\mu}\left[\sum_{n=1}^{n=N} g(s_n, a_n)\right] \quad (1)$$

$\mathbb{E}_{\mu}[.]$ is the expectation under strategy μ, i.e. under the condition that $\forall n,\ a_n = \mu(s_n)$. $g(s, a)$ is the instantaneous cost incurred by the user when performing action a in state s. This function is calculated as follows:

- Internal processing: the energy consumption of the user is supposed to be a linear function of the number of packets a that are processed internally during the time slot duration T.

  $$g(s, a) = a \times P_0 \times T \quad (2)$$

  were $E_0 = P_0 \times T$ is the energy needed to process one packet. P_0 is the power consumed by the terminal when processing one packet locally.
- Offloading: The energy consumed to process a packets using a remote procedure call to the eNB is the sum of three terms (see equation (3))
  1) Energy needed to send a packets to the eNB
  2) Energy needed to receive the processed packets.
  3) Energy consumed when the UE is waiting for the result of the remote procedure call.

  All packets that are generated by the application are assumed to have the same size L_i. The result of the processing of 1 packet is assumed to have size L_o. For simplicity, in (3) only the radiated power is considered in the power consumption. In the general case, the numerator of the first term in equation (3) can be replaced by any concave function of the radiated power.

$$g(s, a) = \overbrace{\frac{aL_iP_u}{W\log_2\left(1 + \frac{P_u x}{WN_0}\right)}}^{(1)} + \overbrace{\frac{aL_oP_r}{W_{\text{DL}}\log_2(1 + \frac{P_e x}{W_{\text{D}}N_0})}}^{(2)} + \overbrace{aT_wP_w}^{(3)} \quad (3)$$

- Idle $g(s, a = 0) = P_{\text{idle}}T\ \forall s \in \mathcal{S}$;

P_u is the transmit power employed by the user. W is the bandwidth allocated to uplink (UL) transmissions, and W_{DL} is the bandwidth allocated to DL transmissions when the packets are offloaded. x is the channel quality information, N_0 is the Gaussian noise experienced by the user in the DL

[2]the power consumption of the eNB is not considered here

and by the eNB in the UL [3]. P_w is the power consumed when the user is in waiting state. In this state the DL receiver is turned on even if no data is received. T_w is the time spent by the eNB to process one packet. P_e is the transmit power of the eNB when sending the result to the UE. P_r is the power consumed by the UE when receiving DL information. When the user reports a state $s = (b, x)$, the action to offload a packets is admissible only if $a \leq b$ and $\exists P_u \in [\,0 \quad P_{\max}\,]$ such that:

$$\frac{aL_i}{W \log_2\left(1 + \frac{P_u x}{W N_0}\right)} + \frac{aL_o}{W_{\text{DL}} \log_2(1 + \frac{P_e x}{W_{\text{DL}} N_0})} + aT_w \leq T \tag{4}$$

Where $P_{\max}$ is the maximum transmit power of the UE. The decision on a allows to calculate the transmit power at the UE side. In order to minimize the transmit power, P_u is chosen so that to respect (4) with equality.

The average delay experienced by each new generated packet before being processed can be expressed as the average size of the UE buffer (Little law) which is given by:

$$h_d(\mu) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_\mu \left[\sum_{n=1}^{n=N} b(s_n) \right]$$

Since the UE has a finite size buffer, the overflow events have to be considered. The employed strategy will be devised so that to keep the probability of buffer overflow under a predefined threshold δ. This probability is given by:

$$h_o(\mu) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_\mu \left[\sum_{n=1}^{n=N} \mathbf{1}_{(b(s_n)=B)} \right]$$

The objective of the CCMU is to minimize the average energy consumed per slot while guaranteeing that the average buffer size is under a predefined quantity $\overline{Q}$ and that the probability of buffer overflow is lower that a predefined threshold δ. This corresponds to the following optimization problem:

$$\begin{aligned} &\mu^\star = \min f(\mu) \\ &\text{s.t. } h_d(\mu) \leq \overline{Q} \text{ and } h_o(\mu) \leq \delta \\ &a_n = \mu(s_n)\ \forall n \in \mathbb{N} \end{aligned} \tag{5}$$

The problem (5) is an infinite horizon average cost problem [4], [2] for which offline and online solutions have been proposed [2], [4]. In the following we investigate two options to solve (5): online an pre-calculated offline solutions. The online solution relies on a Lagrangian relaxation on top of a learning strategy. It offers the advantage of requiring low level of information on the application properties and the channel statistics. The offline solution relies on a prior knowledge of the channel statistics and the application properties (rate of generated data). It offers the advantage of optimality when these conditions are met. Both solutions rely on a prior knowledge of the power consumption model of the UE.

[3]We assume that UL and DL transmission experience the same channel and the same noise

B. Online solution with post-decision state framework

The problem (5) is a constrained optimization problem on the space of strategies. Using a Lagrangian relaxation and applying a saddle point argument [11], this problem can be written as:

$$(\mu^\star, \lambda_d^\star, \lambda_o^\star) = \underset{\lambda_d \geq 0, \lambda_o \geq 0}{\operatorname{argmax}} \ \underset{\mu}{\operatorname{argmin}}\, \mathcal{L}(\lambda_d, \lambda_o, \mu)$$

where

$$\begin{aligned} &\mathcal{L}(\lambda_d, \lambda_o, \mu) = f(\mu) + \lambda_d(h_d(\mu) - \overline{Q}) + \lambda_o(h_o(\mu) - \delta) \\ &= \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_\mu \left[\sum_{n=1}^{N} r_{\lambda_d, \lambda_o}(s_n, a_n) \right] \\ &\text{where} \\ &r_{\lambda_d, \lambda_o}(s_n, a_n) = g(s_n, a_n) \\ &+ \lambda_d(b(s_n) - \overline{Q}) + \lambda_o(\mathbf{1}_{b(s_n)=B} - \delta) \end{aligned} \tag{6}$$

Where λ_d and λ_o are the Lagrange multipliers corresponding to the delay and overflow constraints respectively. For fixed values of λ_d and λ_o, the minimization in (6) is a standard average cost dynamic programming problem which can be solved using relative value iteration (RVIA) methods [11], [4]. The RVIA algorithm is an iterative approach which allows to calculate the value function $V^\star_{\lambda_d, \lambda_o}(s)$ satisfying the following Bellman equation [4]:

$$V_{\lambda_d,\lambda_o}(s) = \min_{a\in\mathcal{A}} r_{\lambda_d, \lambda_o}(s, a) + \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s, a) V_{\lambda_d,\lambda_o}(s') - \gamma \tag{7}$$

where γ is the optimal cost per stage [4], [11]. For fixed values of λ_d and λ_o, the optimal policy is given by

$$\mu^\star_{\lambda_d,\lambda_o}(s) \in \underset{a\in\mathcal{A}}{\operatorname{argmin}} r_{\lambda_d, \lambda_o}(s, a) + \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s, a) V^\star_{\lambda_d,\lambda_o}(s') \tag{8}$$

Solving (8) requires to know the transition probabilities $\mathcal{P}(s'|s, a)$. These probabilities have the particular form of $\mathcal{P}(s'|s, a) = \mathcal{P}(s'|z(s, a))$ where $z(s, a) = (b(s) - a, x(s))$. $z(s, a)$ is called post-decision state and can be seen as the status of terminal after decision a has been made in the status s, but before the new arrival and the new channel realization. A value function on the space of post-decision states $\widetilde{\mathcal{S}} = \{z(s, a), s \in \mathcal{S}, a \in \mathcal{A}\}$ can be defined as:

$$\widetilde{V}_{\lambda_d,\lambda_o}(\widetilde{s}) = \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|\widetilde{s}) V_{\lambda_d,\lambda_o}(s') \tag{9}$$

$\widetilde{V}_{\lambda_d,\lambda_o}(\widetilde{s})$ satisfy the following Bellman equation:

$$\widetilde{V}_{\lambda_d,\lambda_o}(\widetilde{s}) = \sum_{s'\in\mathcal{S}} \mathcal{P}(s|\widetilde{s}) \min_{a'\in\mathcal{A}} \left(r_{\lambda_d, \lambda_o}(s', a') \widetilde{V}_{\lambda_d,\lambda_o}(z(s', a')) \right) \tag{10}$$

In practice, neither $\mathcal{P}(s'|s, a)$ nor $\mathcal{P}(s|\widetilde{s})$are available at the eNB. However the formulation (10) is more adapted to an online solution since the min operator is located inside the averaging operator [11]. This allows to replace the averaging operation in (10) with a stochastic approximation iterative update. We will employ here the same online algorithm developed in [11] but with two constraints rather than only one. We will use the same algorithm as [11] (equations 22,23,24) except that we update two Lagrange multipliers λ_d and λ_o using two update sequences $e_d(n)$ and $e_o(n)$ satisfying conditions (18) and (21) in [11] rather than only one.

C. Pre-calculated offline strategy

In a significant number of cases, the arrival rate of packets can be known or estimated in advance with a certain accuracy. Besides, in low mobility scenarios the channel variations can be acquired over a sufficiently long period before running the application. When such information is available, pre-calculated offline strategy offers a promising option compared to online approaches. In fact, the CCMU can pre-calculate an application dependent strategy for various values of arrival rates and for common channel distributions. A dedicated signalling can be devised inside the application so that to fetch the strategy from the CCMU before starting to process data. This option allows to minimize the signalling overhead caused by useless offloading requests which receive negative answer. In this way, the offloading decision is always made by the application after an initial approval by the CCMU.

The calculation of an offline randomized strategy relies on weighting each (state action) pair with a given probability called the occupation measure. This measure represents the probability to visit each (state, action) pair according to the devised strategy [2].

Let β be the probability distribution of the initial state. For an infinite length trajectory, the occupation measure $\rho_\mu(s,a)$ of a sate action (s,a) pair when strategy μ is employed corresponds to the fraction of epochs where the state s is visited and the action a is performed under the condition that the initial state of the trajectory is drawn according to the probability distribution β. This quantity is given by:

$$\rho_\mu(s,a) = \lim_{N\to\infty} \frac{1}{N}\mathbb{E}\left[\sum_{n=1}^{n=N} \mathbf{1}_{(s_n=s,\mu(s_n)=a)}\right] \quad (11)$$

Using occupation measures $\rho_\mu(s,a)$, (5) can be written as a linear programming problem given by [2]:

$$\rho^\star = \underset{\rho}{\operatorname{argmin}} \sum_{s\in\mathcal{S},a\in\mathcal{A}} g(s,a)\rho(s,a)$$

$$\text{s.t.} \sum_{s\in\mathcal{S},a\in\mathcal{A}} \rho(s,a)b(s) \le \overline{Q} \text{ and } \sum_{a\in\mathcal{A}s\in S|b(s)=B} \rho(s,a) \le \delta$$

and

$$\sum_{a\in\mathcal{A},s^{'}\in\mathcal{S}} \rho(s^{'},a)(\mathbf{1}_{(s^{'}=s)} - P(s|s^{'},a)) = 0 \ \forall s\in\mathcal{S} \quad (12)$$

where the last line in (12) results from the Markov property of the process $(s_n,a_n)\ n\in\mathbb{N}$

An optimal strategy can be obtained from $\rho^\star(.,.)$ the solution of the problem (12) in the following way. When the user is in state s, choose a random action a according to the following probability:

$$p_s(a) = \frac{\rho^\star(s,a)}{\sum_{a'\in\mathcal{A}}\rho^\star(s,a')} \quad (13)$$

IV. Numerical Experiments

In this section we compare the aforementioned strategies for various configurations. We consider an single LTE cell using a bandwidth $W_{\text{DL}} = 5$MHz. Only one user is considered. We assume that UL transmissions for this user are allocated $W = 500$KHz. The time slot T is taken equal to 2ms. All packets that are processed have identical size $L_i = 500$bits and the result of processing of each packet has size $L_o = 5000$bits. The maximum buffer size is $B = 50$packets and the average queue constraint is $\overline{Q} = 10$packets.

All powers are normalized by the signal to noise ratio experienced by the eNB when unitary power signal is sent at the UE side. This is equivalent to normalizing by WN_0/α where α is the path-loss of the channel between the UE and the eNB without fading. The channel state x corresponds to the fading process. The time spent by the eNB to process one packet is $T_w = 0.1$ms. When more than 1 packets are offloaded to the eNB, they are processed serially as shown in (3). The numerical values of the powers are given by: $P_0 = 0.7, P_e = 20, P_w = 0.5, P_{\text{idle}} = 0.2$. The tolerated overflow probability is $\delta_{=}10^{-5}$. The channel coefficient x is assumed to follow an exponential distribution with average $\frac{1}{2}$. x is quantized with respect of the following intervals expressed in dB $[-13, -8.5, -5.4, -3.3, -1.6, -0.08, 1.57, 3.18]$. The maximum number of packets that can be offloaded during one time slot has been set to $O = 5$ and the maximum number of packets that can be processed locally during one time slot has been set to $I = 1$.

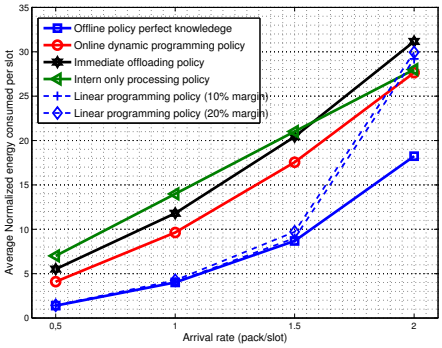

Fig. 2. Average power with online and pre-calculated offline strategies

Figure 2 draws the normalized average energy per time slot for offline (with perfect knowledge of λ and channel distribution) and online strategies. We compare these strategies with static policies which maintain the same average buffer length. With immediate offloading policy, the terminal offloads all packets exceeding the average buffer size constraint $\overline{Q}$. With internal only policy it processes all packets exceeding the average buffer size constraint. The curves named "Linear programming policy (10% margin)" and "Linear programming policy (20% margin)" draw the average energy when the UE employs a pre-calculated (offline) strategy that has been devised with an overestimated arrival rate (10% and 20% of excess). One can see that the pre-calculated offline strategy outperforms the online approach by more than 50% for low and moderate arrival rates and offers similar performance for high arrival rates. The online strategy offers between 20 and 30% of gain compared to static policies. One can also see that the performance of pre-calculated offline strategies with 10% and 20% margins are close to those achieved by offline policy with perfect knowledge for low and moderate arrival rates. With the considered channel distribution the maximal rate of offloading is 2.5packets/slot. When the

arrival rate approaches this value both pre-calculated offline and online strategies have the same performance as the immediate offloading policy. In fact, with the considered channel distribution, the instantaneous energy cost per packet is better for offloading (compared to internal processing) in more than 70% of cases. With high arrival rates the optimal strategy can no longer take benefits from the channel diversity by choosing the idle decision when the channel is in bad states.

Figures 3 and 4 draw the average status occupation for pre-calculated offline, online and immediate offloading strategies for arrival rates of $1\text{packet}/\text{slot}$ and $2\text{packets}/\text{slot}$ respectively. One can see that the pre-calculated offline strategy favours the offloading decision for almost all packets, and alternate this decision with idle state for "bad" channel conditions. This is due to the fact that such strategy knows perfectly the channel distribution and will take better benefits from good channel conditions contrary to the online strategy which relies on incomplete knowledge of the channel distribution. The internal processing option is chosen by the offline strategy only for high arrival rates (2 packets/slot). However, the online strategy makes use of internal processing for more than 10% of cases with $1\text{packet}/\text{slot}$ and for about 20% of cases with $2\text{packets}/\text{slot}$.

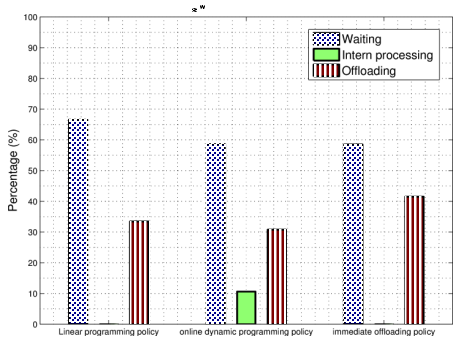

Fig. 3. Average status occupation with $\lambda = 1\text{packet}/\text{slot}$

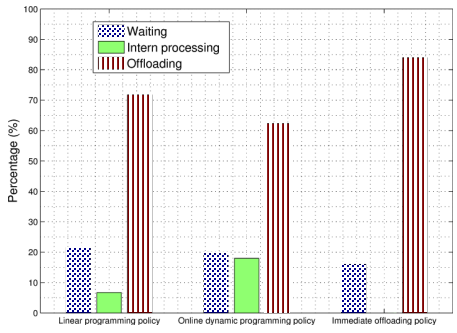

Fig. 4. Average status occupation with $\lambda = 2\text{packets}/\text{slot}$

V. Conclusion

In this paper we investigated an upgrade of 4G architecture where cloud computing resources are placed in the edge of the network (in the eNBs). We presented architectural components that are needed for this upgrade as well as the required signalling that has to be devised to enable such feature. We considered the scenario where the computing resources that are installed in the eNBs are leveraged to offload mobile applications that process user generated data. Joint radio resource allocation and offloading strategies have been investigated to leverage this new feature. These policies have be designed so that to minimize the average power consumed by the user equipment while satisfying predefined delay constraints. Two types of strategies have been studied: online and pre-calculated offline. Both have shown significant gains with respect to static strategies. It has been shown that pre-calculated offline strategies offer significant gains compared to to online policies even if they are designed with a certain margin in order to account for imperfect knowledge of the application properties. Offline strategies offer the advantage to require lower signalling overhead.

Acknowledgment

This work was funded by the European Community 7th Frame- work Program Project ICT-TROPIC, under grant nr. 318784.

References

[1] 3rd generation partnership project; technical specification group radio access network; evolved universal terrestrial radio access (e-utra); medium access control (mac) protocol specification (release 8). Technical report, 3GPP, May 2008.

[2] Eitan Altman. *Constrained Markov Decision Processes*. Chapman and Hall, 1999.

[3] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo. Joint allocation of computation and communication resources in multiuser mobile cloud computing. In *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*, pages 26–30, June 2013.

[4] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 4th edition, 2005.

[5] Yi Ding, Teemu Savolainen, Jouni Korhonen, Sasu Tarkoma, Pan Hui, and Markku Kojo. Nao: A framework to enable efficient mobile offloading. In *Proceedings of the Workshop on Posters and Demos Track*, PDT '11, pages 8:1–8:2, New York, NY, USA, 2011. ACM.

[6] Dong Huang, Ping Wang, and D. Niyato. A dynamic offloading algorithm for mobile computing. *Wireless Communications, IEEE Transactions on*, 11(6):1991–1995, June 2012.

[7] S. Kosta, A. Aucinas, Pan Hui, R. Mortier, and Xinwen Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *INFOCOM, 2012 Proceedings IEEE*, pages 945–953, March 2012.

[8] K. Kumar and Yung-Hsiang Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51–56, April 2010.

[9] Alexander Lenk, Markus Klems, Jens Nimis, Stefan Tai, and Thomas Sandholm. What's inside the cloud? an architectural map of the cloud landscape. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing*, CLOUD '09, pages 23–31, Washington, DC, USA, 2009. IEEE Computer Society.

[10] Shaolei Ren and M. van der Schaar. Efficient resource provisioning and rate selection for stream mining in a community cloud. *Multimedia, IEEE Transactions on*, 15(4):723–734, June 2013.

[11] N. Salodkar, A. Bhorkar, A. Karandikar, and V.S. Borkar. An on-line learning algorithm for energy efficient delay constrained scheduling over a fading channel. *Selected Areas in Communications, IEEE Journal on*, 26(4):732–742, May 2008.

[12] C.B. Sankaran. Data offloading techniques in 3gpp rel-10 networks: A tutorial. *Communications Magazine, IEEE*, 50(6):46–53, June 2012.

[13] Tropic. Tropic: Distributed computing, storage and radio resource allocation over cooperative femtocells.

[14] Yonggang Wen, Weiwen Zhang, and Haiyun Luo. Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones. In *INFOCOM, 2012 Proceedings IEEE*, pages 2716–2720, March 2012.