Analysis of the Energy-Performance Tradeoff for Delayed Mobile Offloading

Huaming Wu
Institute for Computer Science
Freie Universität Berlin, Germany
huaming.wu@fu-berlin.de

Katinka Wolter
Institute for Computer Science
Freie Universität Berlin, Germany
katinka.wolter@fu-berlin.de

ABSTRACT

Mobile cloud offloading that migrates heavy computation from mobile devices to powerful cloud servers through communication networks can alleviate the hardware limitations of mobile devices for higher performance and energy saving. Different applications usually give different relative importance to the factors of response time and energy consumption. In this paper, we investigate two types of delayed offloading policies, the partial model where jobs can leave from the slow phase of the offloading process and then executed locally on the mobile device, and the full offloading model, where jobs can abandon the WiFi Queue and then offloaded via the Cellular Queue. In both models we minimise the Energy-Response time Weighted Product (ERWP) metric. We find that jobs abandon the queue very often especially when the availability ratio (AR) of the WiFi network is relatively small. We can optimally choose the reneging deadline to achieve different energy-performance tradeoff by optimizing the ERWP metric. The amount of delay a job can tolerate closely depends on the application type and the potential energy saving for the mobile device. In general one can say that for delay-sensitive applications, the partial offloading model is preferred when having a suitable reneging rate, while for delay-tolerant applications, the full offloading model shows very good results and outperforms the other offloading models when setting the deadline a large value.

CCS Concepts

•**Networks** → **Cloud computing; Mobile networks;** •**Mathematics of computing** → *Mathematical analysis;*

Keywords

Energy-performance tradeoff; queueing model; offloading policies; heterogeneous networks; mobile cloud computing

1. INTRODUCTION

Besides light-weight Internet applications, there is still an increasing demand from mobile users for computation-heavy and energy-hungry applications that are being deployed to mobile devices. Running complex applications on such devices is however challenging due to the strict constraints on their resources, e.g., the limited computational capacity, battery lifetime and network connectivity. Mobile cloud computing aims at combining the strength of cloud computing and the convenience of mobile terminals. Offloading computation-intensive tasks from mobile devices to a capable cloud server via wireless networks is an effective way to alleviate a tussle between resource-constrained mobile devices and resource-hungry mobile applications, and thus boosts the device's performance.

Potential benefits obtained from offloading include response time shortening and energy saving. However, different applications usually have relative energy and performance importance. For delay-tolerant applications (e.g., iCloud, Dropbox, RSS feeds and participatory sensing), response time is less critical and optimising energy usage is more relevant. Some information is not time-critical and its submission to the server may be delayed until the device enters an energy-efficient network [1]. For delay-sensitive applications (e.g., speed chess game, face recognition, video conferencing and vehicular communications), fast response time is of primary concern while energy consumption is less important. The offloading scheme in which cloud services are available with short network latencies (e.g., WiFi networks) can serve in a better way by providing high responsiveness. Thus, there exists a fundamental tradeoff between mean energy consumption and mean response time in expecting applications [2]. Since performance can be defined as the inverse of the mean response time [3], the energy-performance tradeoff has been studied in [4, 5] by deciding whether or not and by means of which communication interface to offload a whole application. Instead, an application can consist of several components (or jobs), and offloading decisions should be made for each. Seamless offloading operation by switching between several transmission technologies has been proposed in [6]. In addition, they examined the tradeoff between energy consumption for WiFi search and transmission rate when the WiFi network was intermittently available. Energy-efficient delayed network selection has been suggested in [2] to optimise the tradeoff between energy usage and delay in data transmission by intentionally deferring data transmission until the device meets an energy-efficient network. Researchers have further suggested the use of "de-

VALUETOOLS 2015, December 14-16, Berlin, Germany

DOI 10.4108/eai.14-12-2015.2262654

layed offloading": if no WiFi connection is available, (some) traffic can be delayed up to a given deadline, or until WiFi becomes available [7].

Mobile devices usually have multiple radio interfaces for data transfer, such as 3G/4G and WiFi with different availabilities, delays and energy costs. Thus, there are several ways to offload tasks to the cloud, e.g., via a costly cellular connection or via intermittently available WiFi [8, 9]. By delaying offloading until WiFi becomes available, there are opportunities to reduce the transmission time while in the meantime bringing extra waiting time. The reduced transmission time is directly translated into battery power saving for the mobile device [7]. However, the delayed offloading is still a matter of debate, since it is not know what extent users would be willing to delay a transmission [10]. In this paper, we try to give an overall guidance of how to balance the time and energy saving for different types of scenarios like delay-tolerant and delay-sensitive applications.We develop a theoretical framework to capture the energy-performance tradeoff by using queueing models with impatient jobs and service interruptions. The models can be used to predict the average performance and energy consumption of mobile offloading under a given network environment deployment condition. The main contributions are as follows:

- Proposing two types of queueing models for delayed mobile cloud offloading systems: the partial offloading model and the full offloading model. A non-delayed offloading model [10] is also introduced here as a comparison.

- Developing an analytical framework for analyzing queueing models with reneging and service interruptions. We obtain closed-form formulas for key performance metrics in the delayed offloading system such as Energy-Response time Weighted Product (ERWP), which combines the advantages of other previously studied metrics.

- Trying to answer the following questions: (i) Given a deadline, what are the expected response time and expected energy consumption as a function of network parameters and the job arrival rate? (ii) How should the deadlines be optimally chosen in order to achieve different energy-delay tradeoffs for specific applications? (iii) Among different offloading models, how to choose the optimal one that achieves the most performance gains based on the ERWP metric.

The remainder of this paper is as follows. In Section 2, we introduce the delayed offloading system and the queueing model as well as the considered metric. In Section 3, we analyse the partial offloading model based on the ERWP metric. The full offloading model is proposed and analysed in Section 4. Section 5 evaluates metrics and models using numerical examples. The paper is concluded in Section 6.

2. SYSTEM OVERVIEW

In delayed offloading, each data transfer is associated with a deadline, and the data transfer is resumed whenever getting in the coverage of WiFi until the transfer is completed [7]. If the transfer does not finish within its deadline, the task will either be executed locally or the cellular networks will finally complete the transfer.

We consider a queueing system for the delayed offloading. The mobile device, the cloud and the wireless networks are represented as queueing nodes to capture the resource contention and delay on the system. The mobile device executes an application with offloadable jobs that can be processed either locally on the processor of the mobile device, or remotely in a cloud infrastructure through offloading. The mobile device, the cellular and WiFi connections are modelled as $M/M/1$-FCFS queues, and the remote cloud is modeled as an $M/M/\infty$ queue, i.e., as a delay center. We denote $1/\mu_m$ and $1/\mu_r$ the expected execution time of jobs on the mobile device and the cloud, respectively. The expected rates to transfer data to the cloud over the cellular network and WiFi are μ_c and μ_w, respectively. The total cost, in terms of energy or response time for processing all the offloadable jobs, is composed of the remote cost (sending some jobs to the cloud and waiting for the cloud to complete them), and the local cost (processing the remaining jobs locally on the mobile device). Our objective is to minimise the mean energy consumption and the mean response time.

The delayed offloading systems involve queueing with reneging and service interruptions. In queueing, reneging means that a job will leave the queue and join another queue after the deadline expires. Service interruption literally means unwilling discontinuity of service in the queue, and this models connection and disconnection periods of a mobile device to WiFi networks in the system [11].

2.1 The WiFi Model

To facilitate the analysis of the mobile offloading systems, we assume that a cellular network is available to mobile users all the time while the availability of a WiFi network depends on the location. Mobile users move in and out of a WiFi coverage area. We model this time variation of the WiFi connection state by the ON-OFF alternating renewal process $(T_{\text{ON}}^{(i)}, T_{\text{OFF}}^{(i)})$, $i \geq 1$, as shown in Fig. 1. The ON periods represent the presence of the WiFi connectivity, while the OFF periods denote the interruption of the WiFi connectivity. During the latter periods data is either not transmitted (the interface is idle) or it is transmitted only through the cellular network. The duration of each ON period $T_{\text{ON}}^{(i)}$, is assumed to be an exponentially distributed random variable and independent of the duration of other ON or OFF periods [10]. Further, the WiFi availability ratio (AR) can be defined as $AR = \frac{\mathbb{E}[T_{\text{ON}}]}{\mathbb{E}[T_{\text{ON}}]+\mathbb{E}[T_{\text{OFF}}]}$.

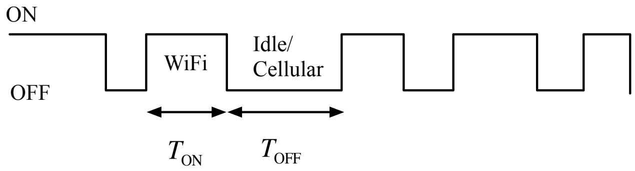


Figure 1: The WiFi network availability model [12]

2.2 Delayed Offloading Models

Accordingly, we build two types of delayed offloading models based on the WiFi network availability model as follows:

- **Partial Offloading Model**: we employ a single queue with two phases (the fast phase with WiFi network and the slow phase with cellular network) to offload jobs to the cloud server. When there is a WiFi connection available, all the offloadable jobs are sent over the WiFi network; otherwise, they are sent over the cellular interface as the cellular network is always available. We set a reneging deadline in the cellular network, if the deadline expires before the job switched over to some WiFi AP, then it is executed locally on the mobile device rather than remotely on the cloud [7]. By doing this, we have partial jobs offloaded to the cloud and the remaining ones processed locally.

- **Full Offloading Model**: when there is a WiFi connection available, all the offloadable jobs are sent over the WiFi network; otherwise, they can be delayed up to a given deadline, or until WiFi becomes available [12]. If the deadline expires before the job can be transmitted over some WiFi AP, then it is offloaded through the cellular network. In this way, we have all the offloadable jobs offloaded to the cloud via the cellular or WiFi network.

2.3 The ERWP Metric

The general cost metric includes energy consumption related costs in addition to the usual performance metrics such as the response time [13]. The response time is the time between the arrival of a job until it completes service and departs. The energy consumption is the energy spent on the mobile device in that period.

We use queueing theory to model the offloading systems according to a new metric named Energy-Response time Weighted Product (ERWP), which is defined as:

$$ERWP = \mathbb{E}[\mathcal{E}]^{\omega} \cdot \mathbb{E}[T]^{1-\omega}, \tag{1}$$

where $\mathbb{E}[T]$ and $\mathbb{E}[\mathcal{E}]$ are the mean response time and mean energy consumption, respectively. ω (ranging between 0 and 1) is a weighting parameter that represents the relative significance of energy consumption and response time for the mobile device. Large ω favors energy consumption while small ω favors response time. Specifically, to focus on performance, ω should be less than 0.5; to focus on power consumption, ω should be greater than 0.5. In some special cases performance can be traded for power consumption and vice versa, therefore we can use the ω parameter to express such special cases preferences for different applications.

We obtain tight optimality results by deriving explicit expressions in mobile cloud offloading systems to capture energy-performance tradeoffs.

3. PARTIAL OFFLOADING MODEL

Figure 2 depicts a delayed offloading model based on the WiFi network availability model [1]. We consider an $M/M/1$ modulated queue in a two-phase (fast and slow) Markovian random environment, with impatient jobs. The jobs are offloaded either via a cellular connection or a WiFi network to the cloud. The single-server queuing system that oscillates between two feasible phases is denoted by f_{ON} and f_{OFF}. The persistence of the system at any phase is governed by a random mechanism: if the system functions at phase f_{ON} it tends 'to jump' to the other phase with Poisson intensity ξ and if the system functions at phase f_{OFF} it tends 'to jump' to the other phase with Poisson intensity η [14].

We assume that offloading jobs arrive at the system according to a Poisson process with rate λ, and the modulating process $f \in \{f_{\mathrm{ON}}, f_{\mathrm{OFF}}\}$ determines the service rates:

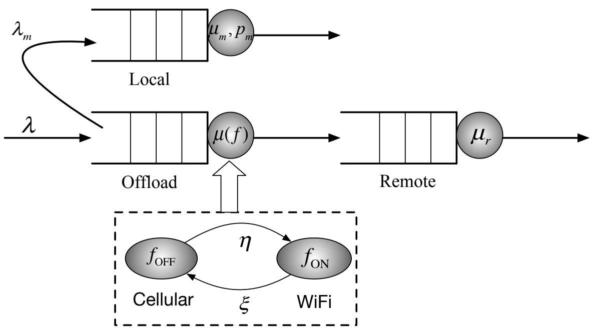


Figure 2: Partial offloading model with cellular and WiFi networks

$$\mu(f) = \begin{cases} \mu_c, & \text{if } f = f_{\mathrm{OFF}} \\ \mu_w, & \text{if } f = f_{\mathrm{ON}} \end{cases}. \tag{2}$$

The average job size is $\mathbb{E}[X]$, the transmission speed of the fast phase (WiFi network) is s_w with service rate $\mu_w = s_w/\mathbb{E}[X]$, and its operating power is p_w when serving jobs and zero whenever idle. Similarly, the corresponding speed for the slow phase (cellular network) is s_c with service rate $\mu_c = s_c/\mathbb{E}[X]$ ($\mu_c \leq \mu_w$), and its operating power is p_c.

When in the slow phase, jobs become impatient. A reneging deadline T_d, is associated with each job in this phase. That is, each job, upon arrival, activates an individual 'impatience timer', exponentially distributed with an reneging rate R. If the system does not change its environment from the slow phase to the fast phase before the deadline expires, the job will be removed from the *Offload Queue* and is assumed to be executed locally on the mobile device rather than offloaded to the cloud [15]. Thus, Fig. 2 demonstrates that the delayed offloading model consists of an *Offload Queue* (with two alternating phases of cellular and WiFi), a *Local Queue* denoting the local processing on the mobile device and a *Remote Queue* representing the remote processing on the cloud server.

The *Offload Queue* alternates its service by means of mutual resets according to the availability of WiFi, which is governed by an interrupted Poisson Process (IPP) with exponentially distributed ON-OFF periods. We model the intermittent availability of WiFi hotspots as a FCFS queue with occasional server break-down [8], either in ON-state where the WiFi network is processing the existing jobs, or in the OFF-state during which the job is serving by the cellular network (the cellular connectivity is assumed to be always available). However, when the job stays in cellular network for too long time, it abandons the *Offload Queue* and is then processed locally on the mobile device. We assume that the sojourn time in a hotspot and the time to move from one hotspot to another are exponentially distributed with parameters ξ (failure rate), and η (recovery rate), respectively. If the job in the *Offload Queue* is completely transmitted before the assigned deadline has expired, we say that the job is successfully offloaded. If offloading fails, the job leaves the *Offload Queue* and join the *Local Queue* on the mobile device for immediate local processing. We call such an event a reneging event [11].

Since there is no waiting time before entering service, the $M/M/\infty$ queue of the Cloud is occasionally referred to as a delay (sometimes pure delay) station, the probability distribution of the delay being that of the service time.

3.1 Queueing Analysis

We use queueing analysis to derive formulas for the average number of jobs for an $M/M/1$ queue operating in a 2-phase network environment. Given the previously stated assumptions, the partial offloading model can be modeled with a 2D Markov chain, as shown in Fig. 3.

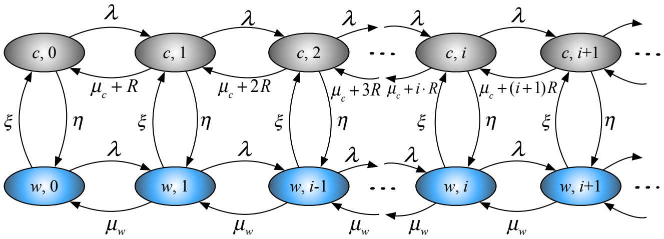


Figure 3: The 2D Markov chain for the partial offloading model with cellular and WiFi networks

The states with cellular network are denoted with $\{c, i\}$, and the states with WiFi connectivity are denoted with $\{w, i\}$. i corresponds to the number of jobs in the system (queuing and in service). During the WiFi phase, the system empties at rate μ_w and during the cellular phase, the system empties at rate $\mu_c + i \cdot R$ since any of the i queued jobs can abandon the *Offload Queue* [12]. Writing the balance equations for the cellular and WiFi phases gives:

$$\begin{aligned}
(\lambda + \eta)\pi_{c,0} &= (\mu_c + R)\pi_{c,1} + \xi\pi_{w,0} && \text{(3a)}\\
(\lambda + \eta + \mu_c + iR)\pi_{c,i} &= \lambda\pi_{c,i-1} + \big(\mu_c + (i+1)R\big)\pi_{c,i+1} \\
&\quad + \xi\pi_{w,i} && \text{(3b)}\\
(\lambda + \xi)\pi_{w,0} &= \mu_w\pi_{w,1} + \eta\pi_{c,0} && \text{(3c)}\\
(\lambda + \xi + \mu_w)\pi_{w,i} &= \lambda\pi_{w,i-1} + \mu_w\pi_{w,i+1} + \eta\pi_{c,i} && \text{(3d)}
\end{aligned}$$

The steady-state probability of finding the offloading system in some region with WiFi unavailability (with only cellular access) is $\pi_c = \frac{\mathbb{E}[T_{\text{OFF}}]}{\mathbb{E}[T_{\text{ON}}]+\mathbb{E}[T_{\text{OFF}}]} = \frac{\xi}{\eta+\xi}$. Similarly, the steady-state probability for the periods with WiFi availability is $\pi_w = \frac{\mathbb{E}[T_{\text{ON}}]}{\mathbb{E}[T_{\text{ON}}]+\mathbb{E}[T_{\text{OFF}}]} = \frac{\eta}{\eta+\xi}$, which equals to the availability ratio AR. The probability generating functions for both cellular and WiFi states are defined as:

$$G_c(z) = \sum_{i=0}^{\infty} \pi_{c,i} z^i \text{ and } G_w(z) = \sum_{i=0}^{\infty} \pi_{w,i} z^i, \quad |z| \leq 1. \quad (4)$$

After some calculation and algebraic manipulations, we obtain:

$$G_w(z)\beta(z) = \eta z G_c(z) - \mu_w(1-z)\pi_{w,0},$$

where $\beta(z) = (\lambda z - \mu_w)(1-z) + \xi z$. The roots z_1, z_2 of the quadratic polynomial $\beta(z) = -\lambda(z - z_1)(z - z_2)$ are $z_{1,2} = \frac{\lambda+\mu_w+\xi\mp\sqrt{(\lambda+\mu_w+\xi)^2-4\lambda\mu_w}}{2\lambda}$ [15].

3.1.1 General Case

Assume the reneging rate $R \neq 0$, we have the partial offloading model as depicted in Fig. 2. According to [15], we obtain:

$$\pi_{c,0} = \frac{RS\xi\kappa_2(1)}{\mu_c(\xi+\eta)(SV-TU)}, \quad (5)$$

$$\pi_{w,0} = -\frac{RT\kappa_2(1)}{\mu_w(\xi+\eta)(SV-TU)}, \quad (6)$$

where we define $S = \int_0^{z_1} \frac{\kappa_1(x)}{\beta(x)}dx$, $T = \int_0^{z_1} \frac{\kappa_1(x)}{x}dx$, $U = \int_{z_1}^{1} \frac{\kappa_2(x)}{\beta(x)}dx$ and $V = \int_{z_1}^{1} \frac{\kappa_2(x)}{x}dx$. Accordingly, $\kappa_1(z)$ and $\kappa_2(z)$ are represented as follows:

$$\kappa_1(z) = e^{-\frac{\lambda z}{R}} z^{\frac{\mu_c}{R}} (z_1 - z)^{\frac{\eta z_1(z_2-1)}{R(z_2-z_1)}} (z_2 - z)^{-\frac{\eta z_2(z_1-1)}{R(z_2-z_1)}}, z \leq z_1,$$

$$\kappa_2(z) = e^{-\frac{\lambda z}{R}} z^{\frac{\mu_c}{R}} (z - z_1)^{\frac{\eta z_1(z_2-1)}{R(z_2-z_1)}} (z_2 - z)^{-\frac{\eta z_2(z_1-1)}{R(z_2-z_1)}}, z \geq z_1.$$

By the definitions of $\kappa_1(z)$, $\kappa_2(z)$ and $\beta(z)$, it follows that $T, U, V > 0$ and $S < 0$. Therefore, $\pi_{c,0}$ and $\pi_{w,0}$ are positive. One can show formally that the system is ergodic. Intuitively, we indicate that the system is always stable since, with any set of parameters $\lambda \geq 0$, $\mu_c \geq 0$, $\mu_w > 0$, $\xi > 0$, $\eta > 0$ and $R > 0$, the abandonment process, whose overall rate increases with the number of jobs, prevents explosion [15]. Alternatively, the system is stable if and only if $\pi_{c,0}$ and $\pi_{w,0}$ are positive, which always holds for the above set of parameters.

Let μ be defined as: $\mu = \pi_c \cdot \mu_c + \pi_w \cdot \mu_w$. According to [15], we obtain:

$$\begin{aligned}
\mathbb{E}[N_c] &= \frac{\lambda - \mu + \mu_c\pi_{c,0} + \mu_w\pi_{w,0}}{R}, && (7)\\
\mathbb{E}[N_w] &= \frac{\eta(\lambda-\mu) + R(\lambda-\mu_w)\pi_w}{\xi R} + \\
&\quad \frac{\eta\mu_c\pi_{c,0} + \mu_w(\eta+R)\pi_{w,0}}{\xi R}. && (8)
\end{aligned}$$

From Fig. 3, the expected number of jobs served per unit of time in the slow and fast phases are $\mu_c(\pi_c - \pi_{c,0})$ and $\mu_w(\pi_w - \pi_{w,0})$, respectively [16]. Therefore, the rate of abandonment due to impatience in the slow phase, λ_{aband}, is given by:

$$\begin{aligned}
\lambda_{\text{aband}} &= \lambda - \mu_c(\pi_c - \pi_{c,0}) - \mu_w(\pi_w - \pi_{w,0})\\
&= \lambda - \mu + \mu_c\pi_{c,0} + \mu_w\pi_{w,0}\\
&= R \cdot \mathbb{E}[N_c], && (9)
\end{aligned}$$

where the abandonment rate is proportional to the reneging rate and the mean number of jobs in the cellular phase.

The rate λ_m that jobs are executed locally on the mobile device must be equal to the abandonment rate, i.e., $\lambda_m = \lambda_{\text{aband}}$. The probability that an arbitrary job arriving to the *Offload Queue* will leave and join the *Local Queue*, i.e., it will be executed locally and will never be offloaded again, is defined as:

$$\Pr\{\text{abandon}\} = \frac{\lambda_{\text{aband}}}{\lambda} = \frac{\lambda - \mu + \mu_c\pi_{c,0} + \mu_w\pi_{w,0}}{\lambda}, \quad (10)$$

where Pr denotes the probability operation.

3.1.2 Extreme Case

Assume the reneging rate $R \to 0$, the partial offloading model as shown in Fig. 2 reduces to a non-delayed offloading model (or on-the-spot offloading [7]), which is depicted in Fig. 4. Since the reneging rate is zero, there will be no *Local Queue* in this model. We use this model as a reference case for comparison purpose with the delayed offloading models.

After solving the balanced equations when setting $R = 0$, we have [17]:

$$g(z)G_c(z) = \pi_{w,0}\xi\mu_w z + \pi_{c,0}\mu_c\big[\xi z + \lambda z(1-z) - \mu_w(1-z)\big],$$

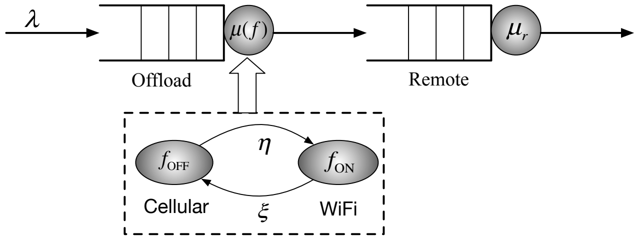


Figure 4: Non-delayed offloading model with cellular and WiFi networks

where $g(z) = \lambda^2 z^3 - \lambda(\eta+\xi+\lambda+\mu_c+\mu_w)z^2 + (\eta\mu_w+\xi\mu_c+\mu_c\mu_w+\lambda(\mu_c+\mu_w))z - \mu_c\mu_w$, and it is proven that $g(z)$ has only one root z_0 in the interval $(0,1)$.

After some algebraic manipulations, we obtain:

$$\pi_{c,0} = \frac{\xi(\mu-\lambda)z_0}{\mu_c(1-z_0)(\mu_w-\lambda z_0)}, \tag{11}$$

$$\pi_{w,0} = \frac{\eta(\mu-\lambda)z_0}{\mu_w(1-z_0)(\mu_c-\lambda z_0)}. \tag{12}$$

Once the values of $\pi_{c,0}$ and $\pi_{w,0}$ have been established, according to Eq. (4), the probability generating functions can be calculated as:

$$G_c(z) = \frac{\xi(\mu-\lambda)z + \pi_{c,0}\mu_c(1-z)(\lambda z-\mu_w)}{g(z)}, \tag{13}$$

$$G_w(z) = \frac{\eta(\mu-\lambda)z + \pi_{w,0}\mu_w(1-z)(\lambda z-\mu_w)}{g(z)}. \tag{14}$$

By using $\mathbb{E}[N_c] = \sum_{i=0}^{\infty} i\pi_{c,i} = dG_c(z)/dz|_{z=1}$ and $\mathbb{E}[N_w] = \sum_{i=0}^{\infty} i\pi_{w,i} = dG_w(z)/dz|_{z=1}$, we get the average number of jobs in the system [17]:

$$\begin{aligned}\mathbb{E}[N] &= \mathbb{E}[N_c] + \mathbb{E}[N_w] \\ &= \frac{\lambda}{\mu-\lambda} + \frac{\mu_c(\mu_w-\lambda)\pi_{c,0} + \mu_w(\mu_c-\lambda)\pi_{w,0}}{(\xi+\eta)(\mu-\lambda)} - \\ &\qquad \frac{(\mu_c-\lambda)(\mu_w-\lambda)}{(\xi+\eta)(\mu-\lambda)}. \end{aligned} \tag{15}$$

3.2 Metric-Based Analysis

The total cost for offloading a job is composed of the cost for sending the job to the cloud and idly waiting for the cloud to complete the job. By Little's Law, $\mathbb{E}[N] = \lambda\mathbb{E}[T]$, the mean response time can be calculated as:

$$\begin{aligned}\mathbb{E}[T] &= \mathbb{E}\big[\mathbb{E}[T_i]\big] = \sum_{i\in\{c,w,m,r\}} \frac{\lambda_i}{\lambda}\mathbb{E}[T_i] \\ &= \frac{1}{\lambda}\sum_{i\in\{c,w,m,r\}} \mathbb{E}[N_i], \end{aligned} \tag{16}$$

where $i \in \{c,w,m,r\}$ represents the cellular phase, the WiFi phase, the mobile device and the remote cloud, respectively. $\mathbb{E}[N_c]$ and $\mathbb{E}[N_w]$ are the average number of jobs in the cellular network and WiFi network as obtained in Eqs. (7) and (8), respectively.

For the local processing, since the arrival rate to the *Local Queue* equals to the abandonment rate of the *Offload Queue*, we have $\lambda_m = R\cdot\mathbb{E}[N_c]$. The work load, i.e., the fraction of time when the server is busy, is denoted as: $\rho_m = \lambda_m/\mu_m$. For an ordinary $M/M/1$-FCFS queue, the average number of jobs on the mobile device is $\mathbb{E}[N_m] = \rho_m/(1-\rho_m)$.

Since there is no waiting time before entering into remote service in the cloud, for an $M/M/\infty$ queue, the average number of jobs in the *Remote Queue* is $\mathbb{E}[N_r] = \lambda_r/\mu_r$, where $\lambda_r = \lambda - \lambda_m$ is the arrival rate to the *Remote Queue*.

A key assumption in our work is that each service operates at a constant power p_i, $\big(i \in \{c,w,m\}\big)$ whenever it is busy, i.e., the mobile device consumes energy only when there are jobs in the system. Since $\mathbb{E}[P] = \lambda\mathbb{E}[\mathcal{E}]$ is the mean power consumption, we can calculate the mean energy consumption for the partial offloading model as:

$$\mathbb{E}[\mathcal{E}] = \mathbb{E}\big[\mathbb{E}[\mathcal{E}_i]\big] = \sum_{i\in\{c,w,m\}} \frac{\lambda_i}{\lambda}\mathbb{E}[\mathcal{E}_i] = \frac{1}{\lambda}\sum_{i\in\{c,w,m\}} \mathbb{E}[P_i]. \tag{17}$$

Since some jobs are remotely executed on the cloud server rather than on the mobile device, we do not need to include such energy consumption. For $i \in \{c,w,m\}$, the corresponding average power consumption can be calculated as:

$$\mathbb{E}[P_i] = p_i \cdot \Pr\{N_i > 0\} = p_i \cdot \rho_i. \tag{18}$$

Since the utilization of the queue is the probability that the server is busy, we have $\Pr\{N_i > 0\} = \rho_i$ [18], i.e., the energy cost is only incurred during the fraction of the time the server is busy.

The energy consumed due to local execution depends on the processing speed of the mobile device. Since the service on mobile device is always available, we have:

$$\mathbb{E}[P_m] = p_m \cdot \Pr\{N_m > 0\} = p_m \cdot \rho_m. \tag{19}$$

The mean energy consumed due to offloading via cellular or WiFi network depends on the transmission power and speed. We have:

$$\mathbb{E}[P_c] = p_c \cdot \Pr\{N_c > 0\} = p_c \cdot \rho_c, \tag{20}$$

$$\mathbb{E}[P_w] = p_w \cdot \Pr\{N_w > 0\} = p_w \cdot \rho_w, \tag{21}$$

where ρ_c and ρ_w are the utilizations of the cellular and WiFi networks, which are equal to the probability that the corresponding network is busy. According to Fig. 3, they can be separately calculated as: $\rho_c = \pi_c - \pi_{c,0}$ and $\rho_w = \pi_w - \pi_{w,0}$.

Further, by substituting Eqs. (16) and (17) into Eq. (1), we can formulate the explicit expressions and the optimization problem of the ERWP metric for the offloading assignment as:

$$R^* = \arg\min_{R} \mathit{ERWP}, \tag{22}$$

we seek the reneging rate R^* such that *ERWP* is minimised.

4. FULL OFFLOADING MODEL

Figure 5 depicts another delayed offloading model based on the WiFi network availability model. All jobs arriving to the system are by default sent to the WiFi interface for offloading. When a job is offloaded to the cloud via a WiFi network, there is queueing due to the transmission speed of the WiFi link. We model the intermittent availability of hotspots as a FCFS queue with occasional server breakdown. The server availability is governed by an IPP with exponentially distributed ON-OFF periods. Specifically, the server is either in ON-state processing the existing jobs, or in OFF-state during which no job receives service. We assume the jobs will abandon the queue during periods without WiFi connectivity.

We assign a reneging deadline for each job (drawn from an exponential distribution). Jobs are serviced in the FCFS order depending on their remaining deadlines (either while

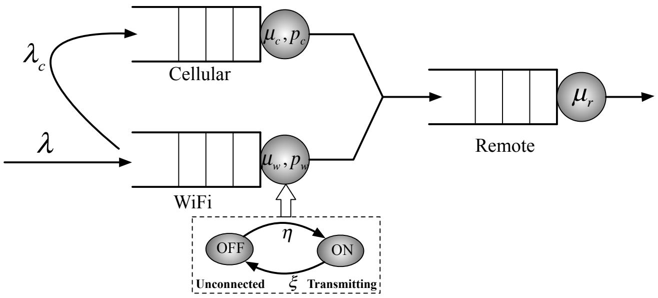


Figure 5: Full offloading model with cellular and WiFi networks

queued or while at the head of the queue, but waiting for WiFi). A job can be served only via WiFi before its deadline. As the queueing system is continuous, it handles transmission at the bit level so that assigning a deadline to a job is equivalent to assigning the same deadline to each bit of the job [7]. When in the OFF-state, jobs become impatient. That is, each job, upon arrival, activates an individual timer, exponentially distributed with an reneging rate R. If the network does not change its environment from the OFF-state to the ON-state before the deadline expires, the job abandons the *WiFi Queue*, and instead, to be offloaded via a cellular network [12]. If the job in the *WiFi Queue* is completely transmitted through WiFi networks before the assigned deadline has expired, we say that the job is successfully offloaded. If offloading fails, the job leaves the *WiFi Queue* and join the *Cellular Queue* in the mobile device for immediate transmission through cellular networks. We call such an event a reneging event.

When the job is offloaded to the cloud via a cellular network, there is queueing due to the transmission speed of the cellular link. Costs arise in terms of transmission delays (queueing and actual transmission time) and transmission energy consumption. Service is always available since the cellular connection is always on. Similarly, the *Remote Queue* is a pure delay station at which jobs spend an exponentially distributed amount of time with mean equal to $1/\mu_r$ time units.

4.1 Queueing Analysis

The *WiFi Queue* refers to offloading jobs from the mobile device to the cloud via a WLAN network, which is modeled as an $M/M/1$-FCFS queue with intermittently available service. When a server recovers, it continues to serve the job whose service has been interrupted, i.e., the work already completed is not lost (cf. data transfers with resume) [8].

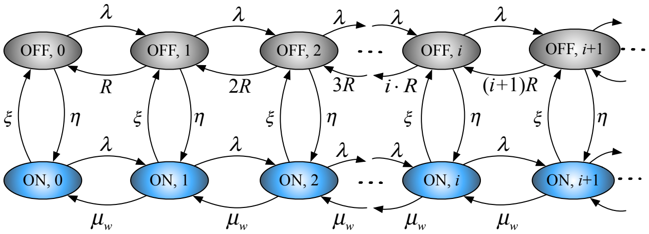


Figure 6: The 2D Markov chain for the WiFi queue

We assume that the service fails from time to time and resumes its operation after a random time. The Markov chain for the *WiFi Queue* is depicted in Fig. 6, which is equivalent to assuming that $\mu_c = 0$, $\pi_{\mathrm{ON}} = \pi_w$ and $\pi_{\mathrm{OFF}} = \pi_c$ in Fig. 3. The states with WiFi connectivity are denoted with $\{\mathrm{ON}, i\}$, and the states with WiFi unconnectivity are denoted with $\{\mathrm{OFF}, i\}$. i corresponds to the number of jobs in the system (queuing and in service). During the ON-state, the system empties at rate μ_w and during the OFF-state, the system empties at rate $i \cdot R$ since any of the i queued jobs can abandon the *WiFi Queue* [12]. Writing the balance equations for this chain gives:

$$(\lambda + \eta)\pi_{\mathrm{OFF},0} = \xi\pi_{\mathrm{ON},0} + R\pi_{\mathrm{OFF},1} \tag{23a}$$

$$\begin{aligned}(\lambda + \eta + iR)\pi_{\mathrm{OFF},i} = \lambda\pi_{\mathrm{OFF},i-1} + (i+1)R\pi_{\mathrm{OFF},i+1} \\ +\xi\pi_{\mathrm{ON},i}\end{aligned} \tag{23b}$$

$$(\lambda + \xi)\pi_{\mathrm{ON},0} = \eta\pi_{\mathrm{OFF},0} + \mu_w\pi_{\mathrm{ON},1} \tag{23c}$$

$$(\lambda + \xi + \mu_w)\pi_{\mathrm{ON},i} = \lambda\pi_{\mathrm{ON},i-1} + \mu_w\pi_{\mathrm{ON},i+1} + \eta\pi_{\mathrm{OFF},i} \tag{23d}$$

After substituting $\mu_c = 0$ into $\kappa_1(z)$ and $\kappa_2(z)$, yields:

$$\kappa_1(z) = e^{-\frac{\lambda z}{R}}(z_1 - z)^{\frac{\eta}{R}\frac{z_1(z_2-1)}{z_2-z_1}}(z_2 - z)^{-\frac{\eta}{R}\frac{z_2(z_1-1)}{z_2-z_1}}, z \le z_1,$$

$$\kappa_2(z) = e^{-\frac{\lambda z}{R}}(z - z_1)^{\frac{\eta}{R}\frac{z_1(z_2-1)}{z_2-z_1}}(z_2 - z)^{-\frac{\eta}{R}\frac{z_2(z_1-1)}{z_2-z_1}}, z \ge z_1.$$

According to [12], we obtain:

$$\pi_{\mathrm{OFF},0} = -\frac{S\xi\kappa_2(1)}{(\xi+\eta)U\kappa_1(0)}, \tag{24}$$

$$\pi_{\mathrm{ON},0} = \frac{R\kappa_2(1)}{\mu_w(\xi+\eta)U}. \tag{25}$$

We further have $\mu = \pi_c \cdot \mu_c + \pi_w \cdot \mu_w = \pi_{\mathrm{ON}}\mu_w$. After substituting the above values in Eqs. (7) and (8), we derive the mean number of jobs in *WiFi Queue* as:

$$\begin{aligned}\mathbb{E}[N_{\mathrm{OFF}}] &= \frac{\lambda - \mu_w(\pi_{\mathrm{ON}} - \pi_{\mathrm{ON},0})}{R}, \\ \mathbb{E}[N_{\mathrm{ON}}] &= \frac{\eta\lambda - \mu_w(\eta + R)(\pi_{\mathrm{ON}} - \pi_{\mathrm{ON},0}) + \lambda R\pi_{\mathrm{ON}}}{\xi R}.\end{aligned}$$

Therefore, the average number of jobs in the *WiFi Queue* can be calculated as:

$$\mathbb{E}[N_w] = \mathbb{E}[N_{\mathrm{OFF}}] + \mathbb{E}[N_{\mathrm{ON}}]. \tag{26}$$

As shown in Fig. 6, the expected number of jobs served per unit of time in the *WiFi Queue* is $\mu_w(\pi_{\mathrm{ON}} - \pi_{\mathrm{ON},0})$. Therefore, the rate of abandonment due to impatience in the OFF periods, λ_{aband}, is given by:

$$\lambda_{\mathrm{aband}} = \lambda - \mu_w(\pi_{\mathrm{ON}} - \pi_{\mathrm{ON},0}) = R \cdot \mathbb{E}[N_{\mathrm{OFF}}]. \tag{27}$$

where the abandonment rate is proportional to the reneging rate and the mean number of jobs in the OFF-state.

The rate of jobs sent back to the cellular network λ_c must be equal to the abandonment rate, i.e., $\lambda_c = \lambda_{\mathrm{aband}}$. The probability that an arbitrary job arriving to the *WiFi Queue* will abandon, i.e., it will be offloaded over a *Cellular Queue*, is defined as:

$$\Pr\{\mathrm{renege}\} = \frac{\lambda_{\mathrm{aband}}}{\lambda} = \frac{\lambda - \mu_w(\pi_{\mathrm{ON}} - \pi_{\mathrm{ON},0})}{\lambda}. \tag{28}$$

4.2 Metric-Based Analysis

By Little's Law, $\mathbb{E}[N] = \lambda\mathbb{E}[T]$, the mean response time can be calculated as:

$$\begin{aligned}\mathbb{E}[T] &= \mathbb{E}\left[\mathbb{E}[T_i]\right] = \sum_{i\in\{c,w,r\}} \frac{\lambda_i}{\lambda}\mathbb{E}[T_i] \\ &= \frac{1}{\lambda}\sum_{i\in\{c,w,r\}} \mathbb{E}[N_i],\end{aligned} \tag{29}$$

where $\mathbb{E}[N_w]$ is the average number of jobs in the *WiFi Queue* as obtained in Eq. (26).

The *Celluar Queue* refers to offloading jobs from the mobile device to the cloud via a cellular network, which is modeled as an $M/M/1$-FCFS queue. Since the arrival rate to the *Celluar Queue* equals to the abandonment rate of the *WiFi Queue*, i.e., $\lambda_c = R \cdot \mathbb{E}[N_{\mathrm{OFF}}]$. The average number of jobs in this queue is given by:

$$\mathbb{E}[N_c] = \frac{\rho_c}{1-\rho_c}, \tag{30}$$

where $\rho_c = \lambda_c/\mu_c$ is the probability that the *Cellular Queue* is busy.

Since all the jobs are offloaded to the remote server in the cloud, for an $M/M/\infty$ queue, the average number of jobs in the cloud server can be calculated as: $\mathbb{E}[N_r] = \lambda/\mu_r$.

The mean energy consumption can be calculated as:

$$\begin{aligned}\mathbb{E}[\mathcal{E}] &= \mathbb{E}[\mathbb{E}[\mathcal{E}_i|i]] = \sum_{i\in\{w,c\}} \frac{1}{\lambda}\mathbb{E}[P_i] \\ &= \frac{1}{\lambda}\sum_{i\in\{w,c\}} p_i \cdot \Pr\{N_i > 0\} = \frac{1}{\lambda}\sum_{i\in\{w,c\}} p_i \cdot \rho_i, \end{aligned} \tag{31}$$

where ρ_w is the fraction of time that WiFi is available to process jobs, and it can be calculated as: $\rho_w = \pi_{\mathrm{ON}} - \pi_{\mathrm{ON},0}$, as the recovery rate $\eta \to \infty$, the availability of WiFi $\pi_{\mathrm{ON}} = AR = \frac{\eta}{\xi+\eta}$ tends to be 1.

Further, by substituting Eqs. (29) and (31) into Eq. (1), we can formulate the optimization of the ERWP metric for the offloading assignment as:

$$R^* = \arg\min_R ERWP, \tag{32}$$

we also seek to find the reneging rate R^* such that *ERWP* is minimised.

5. PERFORMANCE EVALUATION

We consider here a simple scenario where the transmission rate of the cellular network is smaller than that of WiFi, i.e., $s_c < s_w$ and the power consumption when transmitting jobs via the cellular link is larger than the WiFi link, i.e., $p_c > p_w$. Using measurements from real traces collected by [7], the average data rates of the cellular and WiFi networks are set as $s_c = 200$ Kbps and $s_w = 2$ Mbps, respectively. The average duration of WiFi availability period is 52 min ($\xi = 1/52$ min^{-1}), while the average duration with only cellular network coverage is 25.4 min ($\eta = 1/25.4$ min^{-1}). The availability ratio is thus 67%. The mean job size is assumed to be 10 MB. According to the power models developed by [19], we set the power coefficients $p_c = 2.5$ W, $p_w = 0.7$ W and $p_m = 2$ W, respectively. Besides, suppose that the total job arrival rate is $\lambda = 0.5$ packet/min, the mobile service rate $\mu_m = 0.2$ and the cloud service rate $\mu_r = 1$.

An availability ratio of 11% has been reported in [20]. In Fig. 7 as the availability ratio (AR) of the WiFi network increases, the percentage of jobs abandon the *Offload Queue* (for the partial offloading model, refer to Fig. 7(a)) or the *WiFi Queue* (for the full offloading model, refer to Fig. 7(b)) declines rapidly. However, the full offloading model has much higher reneging probability than the partial one under the same deadline T_d. That's because the partial offloading model can use the cellular network to transmit data, and thus the number of jobs waiting in the *Offload Queue* is reduced. On the other hand, as the reneging deadline increases from 60 min to 120 min, jobs have more chance to be offloaded via the WiFi network, and therefore the reneging probability decreases at the lower level of arrival rates. However, at high arrival rates, the reneging probability stays the same under different deadline.

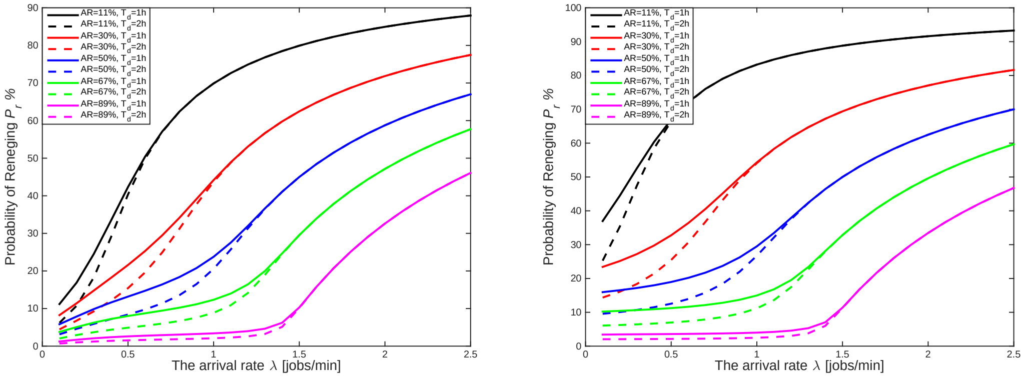
The partial offloading model in Fig. 8(a) has the lowest average response time, since it takes full use of the slow phase of the cellular network during the WiFi is in the unavailable period. For the lower deadlines ($T_d < 40$ min), the mean response time decreases as the deadline arises, since jobs with higher deadlines has more chance to transmit with the fast WiFi network, leading to smaller response time. However, the mean response time increases for higher deadlines, since jobs with lower deadlines leave the queue earlier, leading to smaller queueing delays. From Fig. 8(b), when the reneging deadline is small, the non-delayed offloading model achieves the lowest mean energy consumption among the three models, but as the deadline increases, the full offloading model is much more preferred. This is due to the fact that the WiFi network is much more fast and energy-efficient than the cellular network. The reduced serving time can cause less energy consumption on the mobile device.

We fix the reneging deadline as 120 min. In Fig. 9(a), the mean response time arises with the increase of job arrival rate λ due to the queueing effects. The partial offloading model performs much better than the other two models since it fully uses the unavailable periods of WiFi by offloading jobs with a cellular network, which in turn brings huge energy consumption as shown in Fig. 9(b). The full offloading model is much more energy-efficient than the non-dealyed offloading model at low λ, while at high λ, the non-delayed offloading model saves much more energy. This can be drawn from Fig. 7(b) that as λ increases, more jobs are abandoned from the *WiFi Queue* and are then offloaded via costly cellular network, which result in more energy consumption.

We use the ERWP metric to compare different offloading models. It can be observed from Fig. 10(a) that when ω is small, the partial offloading model can achieve the smallest ERWP value by optimally choosing the reneging rate R, which indicates that when considering response time more important (for delay-sensitive applications), it is better to use the partial offloading model. Otherwise, when considering energy consumption more important than response time (for delay-tolerance applications), the full offloading model is much more preferred, which translates the reduced transmission time from the fast WiFi network into battery power saving for the mobile device. As shown in Fig. 10(b), when the weighting parameter ω is small, as the arrival rate of the offloadable jobs λ increases, all the three offloading models perform worse. However, the non-delayed offloading model is more sensitive to the arrival job rates. The partial offloading model can always achieve the smallest ERWP value, which that when considering response time more important, it is better to use the partial offloading model. Otherwise, when considering energy consumption more important than response time, the full offloading model is much more preferred at lower λ. While at higher rate, the non-delayed offloading model is preferred.

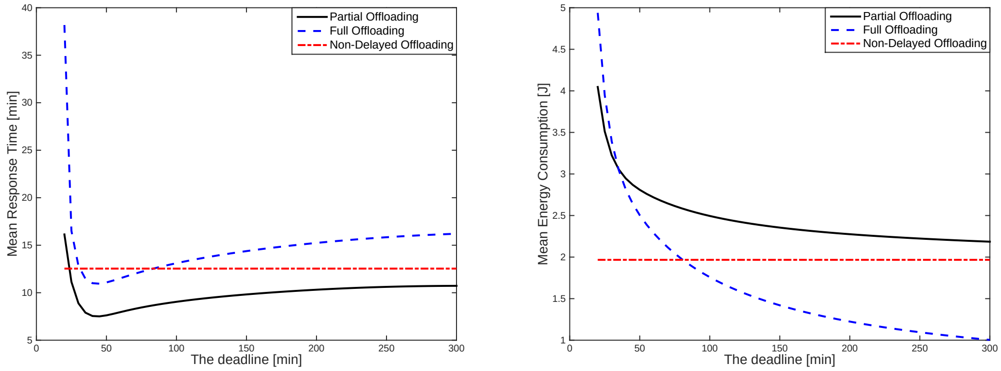
6. CONCLUSIONS

In this paper, we have developed queueing analytic models



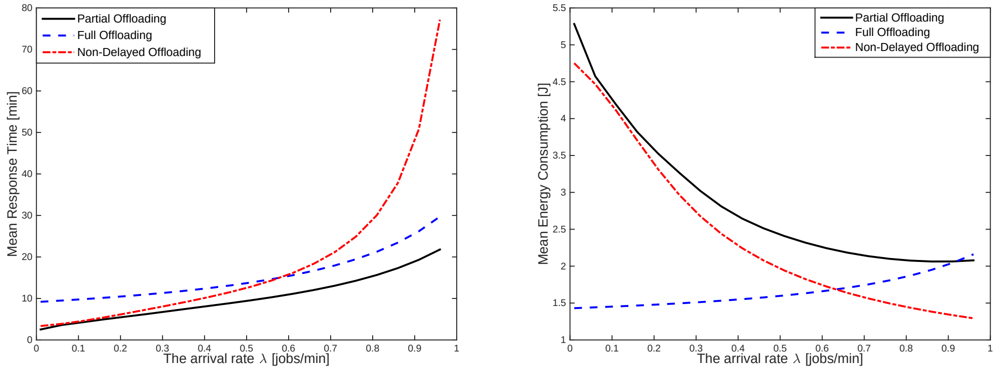
(a) Partial offloading model (b) Full offloading model

Figure 7: The reneging probabilities for the delayed offloading models



(a) Mean Response Time (b) Mean Energy Consumption

Figure 8: Comparison of the offloading models under different deadlines



(a) Mean Response Time (b) Mean Energy Consumption

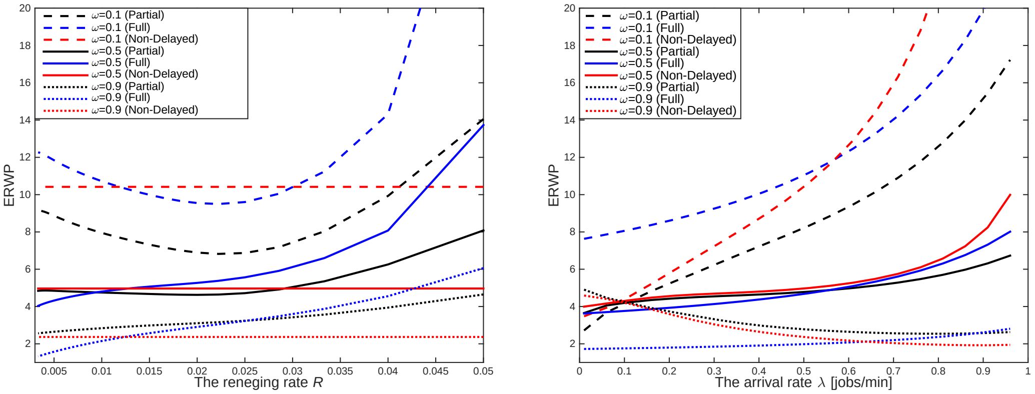
Figure 9: Comparison of the offloading models under different arrival rates

for delayed mobile cloud offloading to leverage the complementary strength of WiFi and cellular networks by choosing heterogeneous wireless interfaces for offloading. We have carried out optimality analysis of the energy-performance tradeoff for mobile cloud offloading systems based on the ERWP metric, which captures both energy and performance metrics and also intermittently available access links.

When the availability ratio (AR) of the WiFi network is relatively small, the percentage of jobs abandon the queue is also very high. We can optimally choose the reneging deadline to achieve different energy-performance tradeoff by optimizing the ERWP metric. We find that for delay-sensitive applications, the partial offloading model is preferred when setting a middle deadline, while for delay-tolerant applications, the full model shows very good results and outperforms the other offloading models when setting the deadline a large value. In general one can say that the partial offloading policy is faster, while the full policy uses less energy.

7. REFERENCES

[1] H. Wu, Y. Sun, and K. Wolter, “Analysis of the energy-response time tradeoff for delayed mobile cloud offloading,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 43, pp. 33–35, Sept. 2015.



(a) Reneging rate

(b) Arrival rate

Figure 10: Comparison of ERWP for the offloading models under different reneging rates and arrival rates

[2] M.-R. Ra, J. Paek, A. B. Sharma, R. Govindan, M. H. Krieger, and M. J. Neely, “Energy-delay tradeoffs in smartphone applications,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 255–270, ACM, 2010.

[3] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, “Optimality analysis of energy-performance trade-off for server farm management,” *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010.

[4] H. Wu, Q. Wang, and K. Wolter, “Tradeoff between performance improvement and energy saving in mobile cloud offloading systems,” in *Communications Workshops (ICC), 2013 IEEE International Conference on*, pp. 728–732, IEEE, 2013.

[5] H. Wu and K. Wolter, “Dynamic transmission scheduling and link selection in mobile cloud computing,” in *Analytical and Stochastic Modeling Techniques and Applications*, pp. 61–79, Springer, 2014.

[6] A. Rahmati and L. Zhong, “Context-for-wireless: context-sensitive energy-efficient wireless data transfer,” in *Proceedings of the 5th international conference on Mobile systems, applications and services*, pp. 165–178, ACM, 2007.

[7] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, “Mobile data offloading: How much can wifi deliver?,” *Networking, IEEE/ACM Transactions on*, vol. 21, no. 2, pp. 536–550, 2013.

[8] E. Hyytiä, T. Spyropoulos, and J. Ott, “Offload (only) the right jobs: Robust offloading using the markov decision processes,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a*, pp. 1–9, IEEE, 2015.

[9] H. Wu, W. Knottenbelt, and K. Wolter, “Analysis of the energy-response time tradeoff for mobile cloud offloading using combined metrics,” in *Teletraffic Congress (ITC 27), 2015 27th International*, pp. 134–142, IEEE, 2015.

[10] F. Mehmeti and T. Spyropoulos, “Performance analysis of “on-the-spot” mobile data offloading,” in *Global Communications Conference (GLOBECOM), 2013 IEEE*, pp. 1577–1583, IEEE, 2013.

[11] Y. Kim, K. Lee, and N. B. Shroff, “An analytical framework to characterize the efficiency and delay in a mobile data offloading system,” in *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing*, pp. 267–276, ACM, 2014.

[12] F. Mehmeti and T. Spyropoulos, “Is it worth to be patient? analysis and optimization of delayed mobile data offloading,” in *INFOCOM, 2014 Proceedings IEEE*, pp. 2364–2372, IEEE, 2014.

[13] H. Wu and K. Wolter, “Tradeoff analysis for mobile cloud offloading based on an additive energy-performance metric,” in *Performance Evaluation Methodologies and Tools (VALUETOOLS), 2014 8th International Conference on*, pp. 90–97, ICST, 2014.

[14] S. Balsamo, G.-L. dei Rossi, and A. Marin, “Queueing networks and conditional product-forms,” in *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*, pp. 204–213, ICST, 2013.

[15] N. Perel and U. Yechiali, “Queues with slow servers and impatient customers,” *European Journal of Operational Research*, vol. 201, no. 1, pp. 247–258, 2010.

[16] U. Yechiali, “Queues with system disasters and impatient customers when system is down,” *Queueing Systems*, vol. 56, no. 3-4, pp. 195–202, 2007.

[17] U. Yechiali and P. Naor, “Queuing problems with heterogeneous arrivals and service,” *Operations Research*, vol. 19, no. 3, pp. 722–734, 1971.

[18] A. Wierman, L. L. Andrew, and A. Tang, “Power-aware speed scaling in processor sharing systems,” in *INFOCOM 2009, IEEE*, pp. 2007–2015, IEEE, 2009.

[19] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, “Energy consumption in mobile phones: a measurement study and implications for network applications,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pp. 280–293, ACM, 2009.

[20] A. Balasubramanian, R. Mahajan, and A. Venkataramani, “Augmenting mobile 3G using WiFi,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 209–222, ACM, 2010.