Energy Efficient Optimization for Computation Offloading in Fog Computing System

Zheng Chang*, Zhenyu Zhou†, Tapani Ristaniemi*, and Zhisheng Niu‡
*University of Jyvaskyla, Faculty of Information Technology, P. O. Box 35, FI-40014 Jyvaskyla, Finland
†School of Electrical and Electronic Engineering, North China Electric Power University, Beijing, China, 102206
‡Department of Electronic Engineering, Tsinghua University, 100084 Beijing, China
Email: zheng.chang@jyu.fi, zhenyu_zhou@ncepu.edu.cn, tapani.ristaniemi@jyu.fi, niuzhs@tsinghua.edu.cn

***Abstract*—In this paper, we investigate the energy efficient computation offloading scheme in a multi-user fog computing system. We consider the users need to make the decision on whether to offload the tasks to the fog node nearby, based on the energy consumption and delay constraint. In particular, we utilize queuing theory to bring a thorough study on the energy consumption and execution delay of the offloading process. Two queuing models are applied respectively to model the execution processes at the mobile device (MD) and fog node. Based on the theoretical analysis, an energy efficient optimization problem is formulated with the objective to minimize the energy consumption subjects to execution delay constraints. In order to address the formulated problem, an alternating direction method of multipliers (ADMM)-based distributed algorithm is proposed. Extensive simulation studies are conducted to demonstrate the effectiveness of the proposed scheme and the superior performance over the other existed schemes can be observed.**

***Index Terms*—computation offloading; resource allocation; fog computing; energy efficiency**

I. Introduction

With the rapid development of Information and Communication Technology (ICT) industry, mobile devices (MD) have become an indispensable part of our daily life as they can provide convenient communications almost anytime and anywhere. It is expected that billions of MDs will be connected to the Internet in the following 5 years, which boosts the intriguing concept of the Internet-of-Things (IoT) where all smart objects, such as wearable devices, smart vehicles, smart phones, sensors, laptops, cameras and industrial components, are connected via a network or Internet and empowered with distributed or centralized data analytic capability. However, due to the restrictions of the MDs on size, weight, battery life, and heat dissipation, the gap between the capability of limited local computing resources and demand for executing complex applications is gradually increasing [1]. Many computational-intensive and delay-intensive mobile applications have poor performance when they are executed on the MDs, especially for IoT devices which are particularly limited with transmission power, storage, and computing resources.

Recent study shows that the mobile cloud computing (MCC) technology provides a promising opportunity to overcome the limitation of hardware and obtain energy saving for the MDs in the IoT by offloading the computational-intensive tasks to the cloud for execution [2] [3]. By offloading different components of mobile applications to the cloud server, the performance of mobile applications can be greatly improved and the energy consumption of the MDs can be significantly reduced [4]. However, it is worth mentioning that the traditional central cloud is usually remotely located and far away from their users. Thus, for latency-sensitive mobile applications, such as high quality video streaming, mobile gaming and so on, offloading to the distant central cloud may not be a perfect solution. To overcome these disadvantages, fog computing, also known as "cloud at the edge", emerges as an alternative proximity solution to provide pervasive and agile computation services for the MDs at anytime and anywhere, and support future cloud services and applications, especially to the IoT applications with strict requirement of latency and high resilience [5].

The idea of using fog computing brings both computational and radio resource more closer to the MDs, thus improving scalability from both computation and radio aspects [6]. In the MCC, computation offloading has attracted significant attention in recent years. In [7], the authors develop a Markov decision process (MDP)-based optimal offloading algorithm for the MD in an intermittently connected cloudlet/fog system, considering the users' local load and availability of cloudlets. Recently, the fog/edge computing system has also been applied to the development of vehicular network that is considered as a typical application area of IoT . In [8], the authors propose predictive offloading scheme for the vehicular edge computing networks. The authors of [9] investigate the contract theoretic approaches for addressing the offloading problem in the vehicular edge computing system.

It can be noticed that the computational resource in the fog node cannot be treated as sufficiently as the traditional central cloud. Moreover, offloading the requests/tasks to the cloud/fog can save the energy at the MDs, but it unavoidably incurs corresponding delay including waiting time at the fog and the communication time between the fog and MD. Thus, the how to save energy consumption and provide energy efficient offloading algorithm with delay constraint for the MDs needs to be addressed. In this paper, our aim is to investigate such a issue in a fog computing system, and propose optimal offloading and power allocation policies. The main contribution of this paper is summarized as follows:

- First, we present an energy consumption optimization problem with explicit consideration of delay performance,

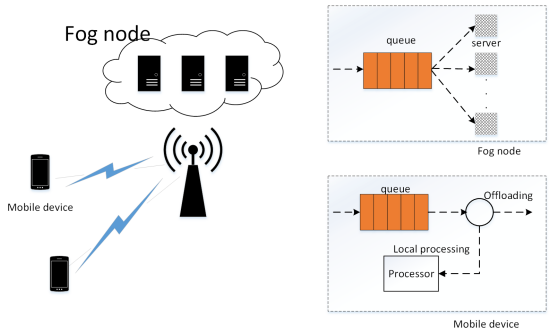

Fig. 1. The model of fog computing system

including the energy consumption and delay in local execution process, computational task transmission process, fog execution and transmission process, and central cloud execution and transmission process, which can thoroughly complement the analysis on the fog computing system.

- In particular, both wireless transmission and computing capabilities are explicitly and jointly considered. We also consider the heterogeneity of the queue at different network elements, e.g., the queue at the MD is considered as a $M/M/1$ queue, the one at the fog is considered as a $M/M/c$ queue which is rarely studied in the previous work about MCC.
- A energy efficient optimization problem is formulated, which involves minimizing the energy consumption by finding the optimal offloading probability and transmit power, subjects to the delay and power constraints. By using nonlinear optimization and ADMM-based method, we are able to transform the optimization problem and address it in a distributed manner.

The reminder of this paper is organized as follows. The system model and assumptions are presented in Section II. In Section III, we present the formulated problem and its transformed form. A distributed algorithm is proposed in Section IV to find the optimal solution. The simulation results are presented to verify the proposed scheme in Section V, and finally we conclude our work in Section VI.

II. System Model and Assumption

A. System Model

The considered fog computing system is presented in Fig. 1. We assume that there are N MDs, and we denote the set of MDs as $\mathcal{N}$. We also consider a fog node, which can connect the MDs via a small cell base station (SBS) in the system. Each MD executes an application, such as mobile gaming, image/video processing etc, and generates a series of homogeneous computation tasks. We use $\varpi_i = \{\theta_i, \zeta_i, \tau_i\}$ to describe each computation task, where θ_i is the data size (e.g., in bits) of each request generating from the MD i to be executed remotely, ζ_i is the needed computational resources and τ_i is the deadline (maximum latency required) of the computation task. For each MD i, the task ϖ_i can be processed locally or offloaded to the fog to execute.

We consider the MD is with single server and the fog node is with c servers, which means that the resources of fog node are limited. For each MD, it can offload a portion or whole of its requests to the fog computing center through the wireless channel, where the transmission suffers from interference generated by other MDs. We assume that the requests generated from MD i, are assumed to follow a Poisson process with an average arrival rate of λ_i. It is also assumed that service times of the MD and fog node have an exponential distribution. Correspondingly, the traffic model at the MD as a $M/M/1$ queue and the one at the fog node as a $M/M/c$ queue. The MD chooses to offload the service request with a probability $p_i^f, 0 \leq p_i^f \leq 1$.

B. Local Computing Model

Accordingly, the service requests which are processed locally also follow a Poisson process with average rate of $(1 - p_i^f)\lambda_i$, and it is considered as local execution rate. We can observe that when the value of p_i^f becomes larger, more requests are delivered to the fog while less requests are processed locally. Let u_i^m denotes the computing capability of MD i, which is comparable to ζ_i. We also assume that l_i^m denotes the normalized workload on the MD i which represents the percentages of CPU that have been occupied. For example, $l_i^m = 0$ indicates that the CPU is totally idle. When considering a $M/M/1$ queue at the MD, the average response time T_i^m for locally processing requests at MD i is expressed as follows:

$$T_i^m\left(p_i^f\right) = \frac{\zeta_i}{u_i^m\left(1 - l_i^m\right) - \left(1 - p_i^f\right)\lambda_i}. \tag{1}$$

Moreover, the energy consumption $E_i^m(p_i^f)$ for locally executing the requests for MD i can be given as follows:

$$E_i^m\left(p_i^f\right) = \kappa_i T_i^m\left(p_i^f\right) = \frac{\kappa_i \zeta_i}{u_i^m\left(1 - l_i^m\right) - \left(1 - p_i^f\right)\lambda_i}, \tag{2}$$

where κ_i is the coefficient denoting the local computation power consumption of MD i, which is related to the intrinsic nature of the CPU and the complexity of these requests. For the sake of simplicity, we assume κ_i is constant during the computation process.

C. Communications Model

When MD i transmits the data/computation task to the fog node, we can obtain the uplink data rate for computation offloading of MD i as follows:

$$R_i = W \log_2\left(1 + \frac{P_i h_i}{\omega_0}\right), \tag{3}$$

where W is the channel bandwidth and P_i is the transmission power of the MD i. h_i is the channel gain between MD i and the SBS. ω_0 denotes the noise power. We consider the

orthogonal multiple access here so there is no interference among the MDs. From (3), we can obtain the transmission time of MD i for offloading the data as follows:

$$T_i^t\left(p_i^f, P_i\right) = \frac{\theta_i}{R_i}. \tag{4}$$

As one can observe, the energy consumption of MD i comprises of two parts: (1) energy consumption of the MD for local service request processing; (2) energy consumption for transmitting data to the SBS. Then, we denote the energy consumption for transmitting the requests from the MD to the SBS is $E_i^t(p_i^f, P_i)$, which can be given as follows:

$$E_i^t\left(p_i^f, P_i\right) = P_i T_i^t\left(p_i^f\right) = \frac{P_i\theta_i}{R_i}. \tag{5}$$

Therefore, the total power consumption can be expressed as

$$E_i\left(p_i^f, P_i\right) = \left(1 - p_i^f\right) E_i^m\left(p_i^f\right) + p_i^f E_i^t\left(p_i^f, P_i\right). \tag{6}$$

The expression of (6) can be found in (7).

D. Fog Computing Model

The service requests which are offloaded to fog node follow a Poisson process with an average rate of $p_i^f\lambda_i$ and it is denoted as the offloading rate. It can be noticed that the computing resource of the fog node may be adequate for running several mobile requests simultaneously, but insufficient for executing too many requests which results in longer delay. Accordingly, we assume that there are c homogeneous servers deployed in the fog node. The service rate for each server is denoted as u^f. The requests from different MDs in the system are pooled together with a total rate λ_p^f. According to the properties of the Poisson process, λ_p^f is given as follows:

$$\lambda_p^f = \sum_{i=1}^{N} \lambda_i p_i^f. \tag{8}$$

Based on the analysis of $M/M/c$ queue at the fog node and Erlang's Formula [10], we define

$$\rho^f = \frac{\lambda_p^f}{cu^f}. \tag{9}$$

Therefore, the average waiting time of each request at the fog node, which contains the queuing time and execution time, is denoted as follows

$$T_{wait}^f\left(p_i^f\right) = \frac{C\left(c, \rho^f\right)}{cu^f - \lambda_p^f} + \frac{1}{u^f}, \tag{10}$$

where [10]

$$C\left(c, \rho^f\right) = \frac{\frac{c\rho^f}{c!(1-\rho^f)}}{\sum_{k=0}^{c-1}\frac{(c\rho^f)^k}{k!} + \frac{c\rho^f}{c!(1-\rho^f)}}. \tag{11}$$

Assuming u_b^f is the transmission/processing rate of the fog node, we can obtain the expected time T_b^f for the execution results waiting in the fog before they are completely delivered out as follows:

$$T_b^f\left(p_i^f\right) = \frac{1}{u_b^f - \lambda_p^f}. \tag{12}$$

The time and energy consumption for the MD to receive the results can be ignored, due to the fact that for many applications (e.g., face recognition), the size of the computation outcome in general is much smaller than the one of input data. From the above analysis, we can obtain the execution time of MD i, which is denoted as follows

$$\begin{aligned} T_i\left(p_i^f, P_i\right) = &\left(1 - p_i^f\right) T_i^m\left(p_i^f\right) + p_i^f T_i^t\left(p_i^f, P_i\right) \\ &+ p_i^f\left(T_{wait}^f(p_i^f) + T_b^f(p_i^f)\right). \end{aligned} \tag{13}$$

III. Problem Formulation and Transformation

A. Problem Formulation

To this end, with above analytic results on the energy consumption and delay, we are able to formulate a energy efficient optimization which involves minimizing energy consumption subjects to the delay constraint. First, we consider a utility function, $U(p_i^f, P_i) = \sum_i^N E_i(p_i^f, P_i)$. As we can see, for executing the computational task, the energy consumption is above zero. Then, the formulated problem can be given as follows:

$$\mathbf{P1}: \quad \min_{\{p_i^f, P_i\}} U(p_i^f, P_i), \tag{14}$$

subject to

$$\begin{aligned} \mathbf{C1}: &\quad \left(1 - p_i^f\right)\lambda_i < u_i^m\left(1 - l_i^m\right), \\ \mathbf{C2}: &\quad \lambda_p^f < cu^f, \\ \mathbf{C3}: &\quad 0 < P_i < P_i^{max}, \\ \mathbf{C4}: &\quad T_i\left(p_i^f, P_i\right) \le \tau_i, \\ \mathbf{C5}: &\quad 0 \le p_i^f \le 1. \end{aligned} \tag{15}$$

In (15), **C1** shows that the request arrival rate of local execution should not exceed the MD's processing rate. **C2** denotes that the actual processing rate at the fog should not exceed the service rate of the fog node. **C3** makes sure that the transmit power is smaller than the maximum transmit power and **C4** puts a constraint on the delay.

B. Problem Transformation

It can be noticed from (7) that the formulated problem is in a fractional form. It can be found that there is no standard approach for solving non-convex optimization problems. In order to make the problem tractable, we transform the objective function and approximate the transformed objective function in order to simplify the problem. First, before transforming the formulated problem from the fractional form to subtractive form, we can arrive the following theorem

Theorem 1. *The objective function (14) is quasi-convex function w.r.t. to the power allocation P_i and offloading variables p_i^f, respectively.*

$$E_i\left(p_i^f, P_i\right) = \underbrace{\frac{\overbrace{\left(1-p_i^f\right)\kappa_i\zeta_i}^{f_1(p_i^f)}}{u_i^m\left(1-l_i^m\right)-\left(1-p_i^f\right)\lambda_i}}_{f_3(p_i^f)} + \underbrace{\frac{\overbrace{P_i p_i^f \theta_i}^{f_2(p_i^f,P_i)}}{W\log_2\left(1+\frac{P_i h_i}{\omega_0}\right)}}_{f_4(P_i)} = \frac{f_1(p_i^f)f_4(P_i)+f_3(p_i^f)f_2(p_i^f,P_i)}{f_3(p_i^f)f_4(P_i)}. \tag{7}$$

The convexity of (14) can be proved using the positive definiteness of the Hessian matrix of the function. Thus, as a result, the unique global optimal solutions for p_i^f and P_i exist and the optimal point can be obtained by using the bisection method with high complexity. We can also apply the nonlinear fractional programming method [11] to solve the formulated problem in the followings.

First, we define the global optimal solution q_i^* for MD i can be expressed as

$$q_i^* = \min_{p_i^f, P_i} \frac{U_{i,1}(p_i^f, P_i)}{U_{i,2}(p_i^f, P_i)}. \tag{16}$$

where $U_{i,2}(p_i^f, P_i) = f_1(p_i^f)f_4(P_i) + f_3(p_i^f)f_4(p_i^f, P_i)$ and $U_{i,2}(p_i^f, P_i) = f_3(p_i^f)f_4(p_i^f, P_i)$. As we can see, finding $\sum q_i^*$ equals to find the optimal solution of **P1**. To this end, we are now ready to introduce the following Theorem.

Theorem 2. *The optimal solution q_i^* can be obtained iff*

$$\min_{p_i^f, P_i} U_{i,1}(p_i^f, P_i) - q^* U_{i,2}(p_i^f, P_i) = 0. \tag{17}$$

Theorem 2 gives a necessary and sufficient condition w.r.t. optimal solutions and its proof is similar to the one in [11]. Particularly, for the considered optimization problem with an objective function in fractional form, there exists an equivalent optimization problem with an objective function in subtractive form, i.e., $U_{i,1}(p_i^f, P_i) - q_i^* U_{i,2}(p_i^f, P_i)$, and both formulations result in the same solutions. To achieve the optimal q_i^*, the iterative algorithm with guaranteed convergence in [11] can be applied and it is given in Alg. 1.

Algorithm 1 Iterative Algorithm for Obtaining q^*

```
Set maximum tolerance δ;
while (!Convergence) do
    Solve the problem (18) for a given q and obtain sub-channel and power allocation {(p_i^f)', P_i'};
    if U_{i,1}((p_i^f)', P_i') − q_i U_{i,2}((p_i^f)', P_i') ≤ δ then
        Convergence = true;
        return {(p_i^f)*, P_i^*} = {(p_i^f)', P_i'} and obtain q_i^* by (16);
    else
        Convergence = false;
        return Obtain q_i = U_{i,2}((p_i^f)', P_i')/U_{i,2}((p_i^f)', P_i');
    end if
end while
```

IV. Distributed Solution via ADMM

As one can see, during the iteration in Alg. 1, in order to achieve q^*, we need to address the following problem with q:

$$\mathbf{P2}: \min_{p_i^f, P_i} \mathcal{V}(p_i^f, P_i) = \sum_{i=1}^{N}\left(U_{i,1}(p_i^f, P_i) - q_i U_{i,2}(p_i^f, P_i)\right), \tag{18}$$

s.t. **C1**-**C5**.

In the following, we aim to address the formulated problem. As each MD needs to make the decision on offloading and power allocation, to make the problem tractable and solvable, we will introduce an ADMM-based solution to address the formulated problem in a distributed manner.

A. ADMM-based solution

As we can see, ADMM is a simple but powerful algorithm that is well suited to address distributed convex optimization problem. Following the approaches in [12], we introduce local copies of the global optimal resource allocation policies. Each local variable can be considered as the preference of each MD about the resource allocation solution. Let us introduce a set of new variables to represent the local variables. Firstly, local variable of P_i is denoted as $\tilde{P}_i$. As p_i^f in $\mathbf{P2}$ is not separable with respect to different MD, to apply ADMM to problem $\mathbf{P2}$, this coupling must be handled appropriately. Therefore, we also define local variable of p_i^f is denoted as $\tilde{p}_i^f$. Correspondingly, there exists an equivalent formulation of problem $\mathbf{P2}$ as follows,

$$\mathbf{P3}: \min_{\tilde{p}_i^f, \tilde{P}_i} \mathcal{V}\left(\tilde{p}_i^f, \tilde{P}_i\right) \tag{19}$$

s.t.

$$\begin{aligned}
\widetilde{\mathbf{C1}}: &\ \left(1-\tilde{p}_i^f\right)\lambda_i < u_i^m\left(1-l_i^m\right), \\
\widetilde{\mathbf{C2}}: &\ \sum_{i=1}^{N}\lambda_i\tilde{p}_i^f < cu^f, \\
\widetilde{\mathbf{C3}}: &\ 0 < \tilde{P}_i < P_i^{max}, \\
\widetilde{\mathbf{C4}}: &\ T_i\left(\tilde{p}_i^f, \tilde{P}_i\right) \le \tau_i, \\
\widetilde{\mathbf{C5}}: &\ 0 \le \tilde{p}_i^f \le 1,
\end{aligned} \tag{20}$$

To this end, the primal non-convex optimization problem P1 is transformed into a suboptimal convex problem P3. By means of the local variables $\tilde{P}_i$ and $\tilde{p}_i^f$, let us define a feasible local variable set for each MD,

$$\mathcal{F}_i = \left\{ \tilde{P}_i, \tilde{p}_i^f \,\middle|\, \widetilde{\mathbf{C1}}, \widetilde{\mathbf{C2}}, \widetilde{\mathbf{C3}}, \widetilde{\mathbf{C4}}, \widetilde{\mathbf{C5}} \right\} \tag{21}$$

where the associated local function g_i defined as follow,

$$g_i(\tilde{P}_i, \tilde{p}_i^f) = \begin{cases} U_{i,1}(\tilde{p}_i^f, \tilde{P}_i) - q_i^* U_{i,2}(\tilde{p}_i^f, \tilde{P}_i), \tilde{P}_i, \tilde{p}_i^f \in \mathcal{F}_i, \\ \infty, \qquad\qquad otherwise. \end{cases} \tag{22}$$

By such, the global consensus problem of $\mathbf{P3}$ can be rewritten as,

$$\begin{aligned} \mathbf{P4}: \quad & \min \quad \sum_{i=1}^{N} g_i\left(\tilde{P}_i, \tilde{p}_i^f\right), \\ & s.t. \quad \tilde{p}_i^f = p_i^f \qquad 1 \le i \le N. \end{aligned} \tag{23}$$

B. Problem solving via ADMM

According to [7], $\mathbf{P4}$ is a global consensus problem. We can then apply ADMM for approaching the problem $\mathbf{P5}$. The initial step is that an augmented Lagrangian with corresponding global consensus constraints should be formed. Let λ_i, $\forall i \in [1, ..., N]$ be the Lagrange multipliers corresponding to consensus constraints in $\mathbf{P4}$. The augmented Lagrangian for $\mathbf{P4}$ is,

$$\begin{aligned} & \mathcal{L}_\rho\left(\left\{\tilde{P}_i, \tilde{p}_i^f\right\}, \{P_i\}, \{\lambda_i\}\right) \\ & = \sum_{i=1}^{N} g_i\left(\tilde{P}_i, \tilde{p}_i^f\right) + \sum_{i=1}^{N} \lambda_i\left(\tilde{p}_i^f, p_i^f\right) \\ & + \frac{\rho}{2} \sum_{i=1}^{N} \left(\tilde{p}_i^f - p_i^f\right)^2, \end{aligned} \tag{24}$$

where λ_i is the vector of the Lagrange multipliers and $\rho \in R_{++}$ is a positive constant parameter for adjusting the convergence speed of the ADMM [7]. Based on the iteration of ADMM with consensus constraints [7], the ADMM method applied to $\mathbf{P4}$ consists of following sequential optimization steps as shown (25)-(27).

C. Distributed Resource Allocation Algorithm

With the above analysis, the proposed distributed resource allocation algorithm can be summarized into following steps.

1) $\{\tilde{P}_i, \tilde{p}_i^f\} - update$: In the first step, power allocation strategies are separable across each MD i. Therefore, $\{\tilde{P}_i, \tilde{p}_i^f\} - update$ can be decomposed into N subproblems, which can be solved locally. Thus, MD i can obtain the optimization solution at iteration t in (25).
2) $\left\{p_i^f\right\}$-update and $\{\lambda_i\}$-update: Compared with the updating of local variables, $\left\{p_i^f\right\}$-update and $\{\lambda_i\}$-update are simple since they are un-constrained quadratic optimization problems. We refer to (26) and (27) to solve the specific update process.
3) Stop criteria and convergence: Based on the discussion in [7], our ADMM-based algorithm iterates satisfy residual convergence, objective convergence and dual variable convergence as $t \to \infty$, because our objective function $\sum_{i=1}^{N} g_i\left(\tilde{P}_i, \tilde{p}_i^f\right)$ is closed, proper and convex and the Lagrangian $\mathcal{L}_\rho$ has saddle point.

As we can see, the iterative distributed optimization process is done via alternating the direction of the variables by sequentially solving several parallel subproblems.

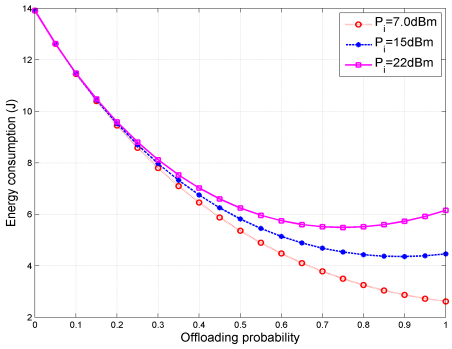

Fig. 2. The impact of offloading probability on energy consumption

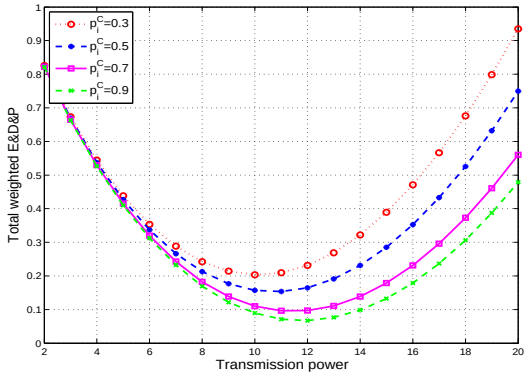

Fig. 3. The impact of transmission power on E&D&P

V. Performance Evaluations

In the simulations, we assume that the maximum transmit power of the MD is 23 dBm. The number of servers in the fog node is $c = 4$. We also consider there are 10 MDs in the system, if not specified.

First, we investigate the impact of offloading probability p_i^f and transmission power P_i on the energy consumption. Some of the simulation parameters are modified from [6]. In Fig. 2, we investigate the impact of offloading probability p_i^f on the energy consumption at different transmit powers. As we can see that at a certain transmit power, the energy consumption decreases with the increased offloading probability at the beginning. This is mainly because that when offloading probability increases, more and more requests are offloaded to the fog node. As the transmit energy consumption is less than the local computing energy consumption, the MD's energy consumption decreases at the beginning. However, when the offloading probability goes higher, the transmit energy consumption increases and dominants the overall energy

$$\left\{\tilde{P}_i, \tilde{p}_i^f\right\}^{[t+1]} := \arg\min \left\{ g_i\left(\tilde{P}_i, \tilde{p}_i^f\right) + \lambda_i\left(\tilde{P}_i - {P_i}^{[t]}\right) + \frac{\rho}{2}\left(\tilde{P}_i - {P_i}^{[t]}\right)^2 \right\}, \tag{25}$$

$$\{p_i^f\}^{[t+1]} := \arg\min \left\{ \sum_{i=1}^{N} \lambda_i^{[t]}\left(\tilde{p}_i^{f[t+1]} - p_i^f\right) + \frac{\rho}{2}\sum_{i=1}^{N}\left(\tilde{p}_i^{f[t+1]} - p_i^f\right)^2 \right\}, \tag{26}$$

$$\{\lambda_i\}^{[t+1]} := \{\lambda_i\}^{[t]} + \rho\left(\tilde{p}_i^{f[t+1]} - p_i^{f[t]}\right). \tag{27}$$

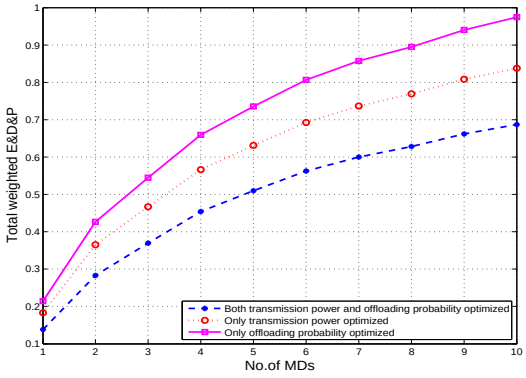

Fig. 4. Comparison among among different schemes

consumption. For example, when $P_i = 22$ dBm and $p_i^f = 0.9$, the energy consumption is higher than the case when $p_i^f = 0.7$. From Fig. 2, the benefits on energy consumption of using fog node can be observed. Meanwhile, the necessity of optimizing the offloading probability and transmit power.

In Fig. 3, we investigate the impact of transmit power on the total energy consumption at different offloading probabilities. At first, with increased transmit power , the energy consumption decreases, which mainly due to the short transmission time. After the energy consumption reaches the minimum at a certain value of transmit power, the energy consumption increases all for cases with different value of p_i^f. Moreover, it can be found that when offloading probability $p_i^f = 0.9$, the energy consumption is higher than the case when $p_i^f = 0.7$, which also evidences the observations in Fig. 2.

We compare our proposed scheme with other schemes presented in [6], [7]. In our scheme, we have optimized both offloading probability and transmit power to minimize the energy consumption while the method in [6] can be viewed as the one only optimizes the offloading probability and the one in [7] only optimizes the transmit power. In Fig. 4, we vary the number of MDs in the system and plot the normalized energy consumption for three schemes. We can see that our method can achieve a better performance in energy consumption by jointly optimizing the offloading probability and transmit power, which demonstrates the comprehensiveness and validity of this study.

VI. CONCLUSION

In this paper, we have investigated the problem of energy efficient optimization for a fog computing system. Specifically, we have derived the analytic results of energy consumption and delay performance with assumption of two different queue models at mobile devices and fog node. By leveraging the obtained results, we have optimized the offloading probability and transmit power for the MDs to minimize the energy consumption with delay constraint. The performance evaluations are presented to illustrate the effectiveness of the proposed scheme and demonstrate the superior performance over the other existed schemes.

REFERENCES

[1] X. Sun and N. Ansari, "EdgeIoT: mobile edge computing for internet of things," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 22-29, Dec. 2016.

[2] Z. Jiang, and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *IEEE Access*, vol. 3, pp. 2306-2316, 2015.

[3] X. Guo, L. Liu, Z. Chang, and T. Ristaniemi, "Data offloading and task allocation for cloudlet-assisted ad hoc mobile clouds," *Wireless Networks*, 2016, DOI :10.1007/s11276-016-1322-z.

[4] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang and W. Wang, "A survey on mobile edge networks: convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757-6779, 2017.

[5] X. Huang, R. Yu, J. Kang, Y. He, and Y. Zhang, "Exploring mobile edge computing for 5G enabled software defined vehicular networks", *IEEE Wireless Communications*, accepted

[6] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89-103, Aug. 2015.

[7] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516-2529, Dec. 2015.

[8] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile edge computing for vehicular networks: A promising network paradigm with predictive off-loading", *IEEE Vehicular Technology Magazine*, vol. 12, no.2, pp.36-44, June 2017.

[9] K. Zhang, Y. Mao, S. Leng, S. Maharjan, A. Vinel, and Y. Zhang, "Contract-theoretic approach for delay constrained offloading in vehicular edge computing networks", *ACM/Springer Mobile Networks and Applications (MONET)*, accepted, 2017.

[10] L. Kleinrock, Queueing systems, volume i: theory, pp. 101?103, 1975

[11] W. Dinkelbach, "On Nonlinear Fractional Programming," *Management Science*, vol. 13, pp. 492-498, Mar. 1967.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, Jan. 2011.