2017 IEEE 56th Annual Conference on Decision and Control (CDC)
December 12-15, 2017, Melbourne, Australia

QoS aware Dynamic Pricing and Scheduling in Wireless Cloud Computing

Zhifei Wang, Jibing Wu, Yahui Wu*, Su Deng, Hongbin Huang
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, 410073 China
* Corresponding author: wuyahui@nudt.edu.cn

***Abstract*—This paper considers the Quality-of-Service (QoS) aware pricing and scheduling problem in wireless cloud computing, where the service provider provides a set of services to users through wireless communication. In this process, the provider announces a price for each service according to the system state and queue length. By collecting the service demand from users and observing the system state, the provider allocates some resources to serve the demand. Due to the dynamic of both the demand and system state, it is hard for the provider to make price and schedule the resources optimally. This paper first formulates the problem as a mathematical model. Then, it proposes the QoS aware dynamic pricing and scheduling algorithm (QDPSA). The QDPSA algorithm only depends on the current system state and queue length and can maximize the average profit of the operator. Simulations show that we can get better tradeoff between the profit and queue length through setting a control parameter. In addition, our results also demonstrate that the service with higher value of QoS coefficient can achieve shorter queue length, i.e. shorter response time.**

I. INTRODUCTION

Mobile devices have become an essential part of human life in recent years [1]. Correspondingly, the mobile applications and services increase rapidly [2]. However, it is very challenging to deliver highly sophisticated applications on mobile devices due to the limited resources, such as battery, processing power and network bandwidth [3]. To improve the performance of mobile applications, wireless cloud computing is designed to overcome some of these limitations by offloading the mobile applications to the remote cloud or service provider [4]. In this paradigm, users pay for executing applications and the service provider allocates some resources to serve these demands, thereby yielding some cost. The mission of the service provider is to design a pricing and scheduling algorithm to maximize its profit.

In this paper, we consider the Quality-of-Service(QoS) aware dynamic pricing and scheduling algorithm to maximize the profit in wireless cloud computing system, as shown in Fig.1. The system can provide users a variety of applications, such as scientific computing, visual search, and batch image processing. In this paper, we focus on the delay insensitive services, which account for 70% of all workloads of the system [5]. Suppose that there are K types of delay insensitive services with different QoS requirements available to the users, and QoS refers to average queueing length. At each time, the service provider observes the service queue and the system state and chooses a set of prices and announces them to all users. The users react to the current price with certain demand Next, the service provider allocates server resources to serve the demands, resulting in K service queues. The above process yields some cost. We are trying to explore the pricing and scheduling algorithm so as to maximize time average profit of the service provider while ensuring queue stability and QoS requirement.

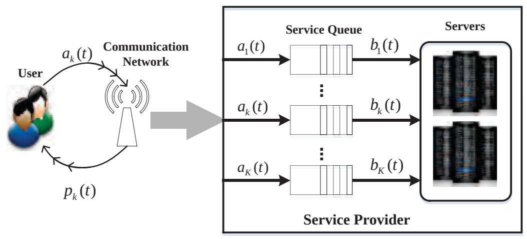

Fig.1. The pricing and scheduling system of wireless cloud computing

Pricing and scheduling problem has been widely studied in the area of computer network [6][7], smart grid [8][9], WiFi Markets [10], and cloud computing[11][12][13], etc. For example, in [6], a new price-based two-tier iterative resource allocation algorithm in wireless ad hoc networks is proposed. The algorithm converges to a global network optimum. In [13], the authors address the issue of how to intelligently manage the resources in a shared cloud database system and present a cost aware resource management system. Some work tried to tackle the dynamic pricing and scheduling problem in wireless environment. By taking both pricing and scheduling into consideration, Hande et al [14] generalize the well-known utility maximization based rate allocation model to incorporate pricing of content providers. Huang and Neely [15] develop an online algorithm that jointly solves the dynamic pricing and scheduling problem for an Access Point (AP) in a wireless network, which can achieve any average profit that is arbitrarily close to the optimum. Ren et al [16] focus on scheduling and pricing in wireless cloud computing and develop a provably-efficient Dynamic Scheduling and Pricing (Dyn-SP) algorithm. Dyn-SP produces a close-to-optimal average profit while bounding the service queue length. To our best knowledge all above works fail to consider different QoS, requirements of different services for the dynamic pricing and scheduling problem in wireless cloud computing. Some works [17] focus on the QoS aware scheduling problem in cloud environment. However, they do not consider the pricing problem and dynamic environment simultaneously. We are trying to develop a QoS aware dynamic pricing and scheduling algorithm to achieve optimal profit. The main contributions of this paper can be summarized as follows:

- This paper proposes a discrete time model to formulate the dynamic pricing and scheduling problem in wireless cloud computing and this model is very general.
- This paper proposes the QDPSA algorithm for solving the above problem. This algorithm can be implemented with unknown system statistics a priori and achieve optimal profit. More importantly, the QDPSA algorithm can satisfy different QoS requirements. These conclusions are validated by our simulations

The rest of the paper is organized as follows. Section II presents the system model formulation. The QDPSA algorithm is developed in Section III. Performance of the QDPSA algorithm is evaluated with simulations in Section IV. Section V concludes this paper.

II. System Model

We consider a discrete-time model in which the operator of the service provider announces the service price and schedules resources to serve the demand at each slot. Key notations used in this paper are listed in table I. We omit the description of "at slot *t*" for brevity.

TABLE I. List of Notatioins

Notations	***Description***
$a_k(t)$	Demand of type-*k* service
$b_k(t)$	Service rate of type-*k* service
$q_k(t)$	Backlog of type-*k* service
$p_k(t)$	Price announced by operator for type-*k* service
$\boldsymbol{c}(t)$	Service cost vector
$Q_k(t)$	QoS aware backlog of type-*k* service
$u(t)$	Profit of the system
$S(t)$	System state

A. Demand Modelling

Let $p_k(t)$ denotes the price for type-*k* service advertised to the mobile users. Assume that $p_k(t)$ belongs to a compact *P* set for all *t* and is bounded by $[0, p_{max}]$. The demand $a_{i,k}(t)$ for type-*k* batch service of user *i* depends on the current price $p_k(t)$ and it can be modelled as a function of $p_k(t)$,

$$a_{i,k}(t) = \hat{a}_{i,k}(p_k(t)) \tag{1}$$

The service provider gathers the demand of each service from the network. Intuitively, the demand of service incurs in the service provider depends on the user demand and the communication network state,

$$a_k(t) = \hat{a}_k(a_{1,k}(t), ..., a_{i,k}(t), ..., a_{I,k}(t), S_N(t)) \tag{2}$$

Where *I* is the number of users, and $S_N(t)$ stands for the network state, the network state is time-varying for various reasons such as the users' distances to the respective base station. And we assume that the network state process is a finite state ergodic Markov chain with state space S_N. Combing Eq.1 and Eq.2, and without causing ambiguity, we can obtain,

$$a_k(t) = \hat{a}_k(p_k(t), S_N(t)) \tag{3}$$

Moreover, we assume that there exists a maximum value $a_{\max}$ such that $a_k(t) \le a_{\max}$ for all *t* and all *k* and $\hat{a}_k(p_{\max}, S_N(t)) = 0$. Given the current $p_k(t)$ and $S_N(t)$, $a_k(t)$ is independent of history information.

B. Service Rate Formulation

Let $b_k(t)$ be the service rate, the amount of batch that can be served, at slot *t* and $S_D(t)$ be the service provider state S_D. Suppose that $S_D(t)$ is a finite state ergodic Markov chain with state space S_D. The system state, which is denoted by *S*, consists of the network state and the service provider state, that is, $S(t)=(S_N(t), S_D(t))$. In this case, we can obtain $S=S_N \times S_D$, which is unknown a priori.

The operator of the service provider should allocate some resource to each demand, which yields certain cost. We assume that the cost of serving type-*k* service on slot *t* is $c_k(t)$ and there exists some finite value c_{max}, which satisfies $0 \le c_k(t) \le c_{max}$ for all *k* and all *t*. Further, we denote vector $\boldsymbol{c}(t)$ as $(c_1(t), c_2(t), .., c_K(t))$, and assume that the cost vector belongs to some set of cost *C*, that is, $\boldsymbol{c}(t) \in C$. The service rate is determined by the cost vector and service provider state,

$$b_k(t) = \hat{b}_k(\boldsymbol{c}(t), S_D(t)) \tag{4}$$

The service provider state refers to many aspects of the service provider, such as the resources that can allocated and the electricity price, and so on. We assume that the service rate $b_k(t)$ can be continuously split and is bounded by a maximum value b_{max}.

Since $S(t)=(S_N(t), S_D(t))$, we further simplify the Eq.3 and Eq.4, yielding the following two equations.

$$a_k(t) = \hat{a}_k(p(t), S(t)) \tag{5}$$

$$b_k(t) = \hat{b}_k(\boldsymbol{c}(t), S(t)) \tag{6}$$

C. Queue Dynamic and Stability

Denote $q_k(t)$ as the queue backlog of type-*k* service at service provider at slot *t*. For type-*k* service, future queue dynamic $q_k(t)$ is driven by the user demand $a_k(t)$ and service rate $b_k(t)$ according to the following dynamic equation,

$$q_k(t+1) = \max[q_k(t) - b_k(t), 0] + a_k(t) \tag{7}$$

Where $a_k(t)$ and $b_k(t)$ are defined by Eq.5 and Eq.6. Let $\boldsymbol{q}(t)$ be the queue backlogs vector $(q_1(t), q_2(t), ..., q_K(t))$.

We introduce the concept of queue stability similar to [18], a discrete time process $\{q_k(t)\}$ is mean rate stable if,

$$\lim_{t\to\infty}\frac{\mathbb{E}\{q_k(t)\}}{t}=0 \tag{8}$$

D. *Formulation of the Problem*

At slot *t*, the total profit of the service provider is,

$$u(t)=\sum_{k=1}^{K}p_k(t)a_k(t)-\sum_{k=1}^{K}c_k(t) \tag{9}$$

Where the first term and second term of right-hand-side represent the income and the total cost of the service provider at slot *t*, respectively. In fact, $u(t)$ can be expressed as $u(t)=\hat{u}(p(t),\boldsymbol{c}(t),S(t))$. Define $\overline{u}(t)$ as the expectation of $u(t)$ over the first *t* slots under a particular algorithm,

$$\overline{u}(t)=\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{u(\tau)\} \tag{10}$$

Where the expectation is over the randomness of the system state and the random policies. Similarly, we can define the expectations $\overline{a}_k(t)$, $\overline{b}_k(t)$. The problem can be formulated as,

$$\text{Minimize:}\quad \limsup_{t\to\infty}\ -\overline{u}(t) \tag{11}$$

$$\text{Subject to:}\quad 1)\ \lim_{t\to\infty}\frac{\mathbb{E}\{q_k(t)\}}{t}=0\ \ \forall k \tag{12}$$

$$2)\ p_k(t)\in P \tag{13}$$

$$3)\ \boldsymbol{c}(t)\in C \tag{14}$$

The objective is to maximize the average total profit by setting the decision variables $p_k(t)$ and $\boldsymbol{c}(t)$ in each slot.

E. *Slackness Condition*

Let $-u^{opt}$ be the infimum value of Eq.11 over all feasible pricing and scheduling policies. Our work is based on the following slackness condition,

Slackness Condition: If the problem Eq.11-Eq.14 is feasible, there exists a positive value $\varepsilon>0$ and another value $\phi(\varepsilon)$ that is related to ε and an *S*-only algorithm $p^*(t)$ and $\boldsymbol{c}^*(t)$ that satisfies,

$$\mathbb{E}\{-\hat{u}(p_k^*(t),\boldsymbol{c}^*(t),S(t))\}=\phi(\varepsilon) \tag{15}$$

$$\mathbb{E}\{\hat{a}_k(p_k^*(t),S(t))\}\le\mathbb{E}\{\hat{b}_k(\boldsymbol{c}^*(t),S(t))\}-\varepsilon\ \ \forall k \tag{16}$$

Where the *S*-only algorithm is the algorithm that makes stationary and randomized pricing and scheduling decisions every slot *t* based only on the observed $S(t)$. The slackness condition Eq.16 is naturally satisfied by our formulation, as the service provider can always set $p(t)=p_{\max}$ such that $a_k(t)=0$. Thus, the Eq.15 can be achieved by some algorithm.

III. QoS AWARE DYNAMIC PRICING AND SCHEDULING ALGORITHM

In the present algorithms, the real queue backlog often plays the role of decision variable. However, these algorithms consider all service queues to be equally important and fail to consider different QoS requirements. To overcome the shortcoming, we introduce the concept of QoS aware queue and further propose the QDPSA algorithm.

A. *QoS Aware Queue Dynamics*

For type-*k* service, we define the QoS aware queue backlog $Q_k(t)$ as a linear function of the real queue length $q_k(t)$. This linear function satisfies that if $q_k(t)$ equals zero, the corresponding $Q_k(t)$ also equals zero. Therefore, we can obtain,

$$Q_k(t)=\mu_k q_k(t) \tag{17}$$

Where we call μ_k as QoS aware coefficient. The QoS aware coefficient is a positive constant, such that if $q_k^m(t)<q_k^n(t)$, then we have $Q_k^m(t)<Q_k^n(t)$. In fact, the QoS aware queue can be any function of the real function, not just linear function, which is left for future work.

Let $A_k(t)$ be $\mu_k a_k(t)$ and $B_k(t)$ be $\mu_k b_k(t)$, which are called as the QoS aware demand and service rate, respectively. Obviously, they can be formulated as $A_k(t)=\hat{A}_k(p(t),S(t))$ and $B_k(t)=\hat{B}_k(p(t),S(t))$. Combing Eq.7 and Eq.17 and plugging $A_k(t)$ and $B_k(t)$ yield the QoS aware queue dynamics,

$$\begin{aligned}Q_k(t+1)&=\mu_k q_k(t+1)\\&=\mu_k(\max[q_k(t)-b_k(t),0]\ +a_k(t))\\&=\max[Q_k(t)-B_k(t),0]\ +A_k(t)\end{aligned} \tag{18}$$

In this case, the total profit can be expressed as,

$$u(t)=\sum_{k=1}^{K}p_k(t)a_k(t)-\sum_{k=1}^{K}c_k(t)=\sum_{k=1}^{K}p_k(t)A_k(t)/\mu_k-\sum_{k=1}^{K}c_k(t) \tag{19}$$

In the rest part of this paper, we will show that different μ_k can indeed achieve different QoS requirements.

B. *QoS Aware Dynamic Pricing and Scheduling Algorithm*

Instead of using the real queue, we introduce the QoS aware queue to simulate the dynamic pricing and scheduling algorithm. The resulting algorithm consists of two parts.

QoS aware dynamic pricing algorithm: At the beginning of every slot *t*, the service provider observes the network state and all $Q_k(t)$ and chooses the price $p_k(t)$ to minimize the following expression,

$$\sum_{k=1}^{K}\hat{A}_k(p_k(t),S(t))(Q_k(t)-\frac{Vp_k(t)}{\mu_k}) \quad (20)$$

QoS aware dynamic scheduling algorithm: At the beginning of every slot *t*, the service provider observes the service provider state and all $Q_k(t)$ and chooses the cost vector $\boldsymbol{c}(t)$ to minimize the following expression,

$$\sum_{k=1}^{K}[Vc_k(t)-Q_k(t)\hat{B}_k(\boldsymbol{c}(t),S(t))] \quad (21)$$

The parameter $V > 0$ is a control variable which can be tuned to different values to trade the queue backlog for the service provider's long-term profit.

C. Performance Analysis

We will show how the QDPSA algorithm stabilizes the queues and obtains the optimal profit. We first extend the slackness condition in the context of QoS aware queue.

Extensive Form of Slackness Condition. If the problem Eq.11-Eq.14 is feasible, there exists a positive value $\varepsilon^Q > 0$ and another value $\phi^Q(\varepsilon^Q)$ that is related to ε^Q and an *S*-only algorithm $p^*(t)$ and $\boldsymbol{c}^*(t)$ that satisfies,

$$\mathbb{E}\{-\hat{u}(p^*(t),\boldsymbol{c}^*(t),S(t))\}=\phi^Q(\varepsilon^Q) \quad (22)$$

$$\mathbb{E}\{\hat{A}_k(p^*(t),S(t))\}\le\mathbb{E}\{\hat{B}_k(c^*(t),S(t))\}-\varepsilon^Q \quad \forall k \quad (23)$$

We derive Eq.22 and Eq.23 from the original slackness condition. Let μ_{min} be the minimum value of all μ_k, and let ε^Q be $\mu_{\min}\varepsilon$. Plugging $\varepsilon=\varepsilon^Q/\mu_{\min}$ into Eq.15 yields,

$$\mathbb{E}\{-\hat{u}(p^*(t),\boldsymbol{c}^*(t),S(t))\}=\phi(\varepsilon)=\phi(\varepsilon^Q/\mu_{\min})=\phi^Q(\varepsilon^Q)$$

Multiplying μ_k in both sides of Eq.16 and plugging $\varepsilon^Q=\mu_{\min}\varepsilon$ into the resulting inequality yields,

$$\begin{aligned}
&\mu_k\mathbb{E}\{\hat{a}_k(p_k^*(t),S(t))\}\le\mu_k\mathbb{E}\{\hat{b}_k(\boldsymbol{c}^*(t),S(t))\}-\mu_k\varepsilon\\
&\Rightarrow\mathbb{E}\{\hat{A}_k(p_k^*(t),S(t))\}\le\mathbb{E}\{\hat{B}_k(\boldsymbol{c}^*(t),S(t))\}-\mu_k\varepsilon\\
&\qquad\qquad\le\mathbb{E}\{\hat{B}_k(\boldsymbol{c}^*(t),S(t))\}-\mu_{\min}\varepsilon\\
&\Rightarrow\mathbb{E}\{\hat{A}_k(p_k^*(t),S(t))\}\le\mathbb{E}\{\hat{B}_k(\boldsymbol{c}^*(t),S(t))\}-\varepsilon^Q
\end{aligned}$$

The following theorem shows that the QDPSA algorithm can stabilize all queues and achieve $O(1/V)$-optimal average profit, while the queue length is bounded by $O(V)$.

Theorem 1. If the problem Eq.11-Eq.14 is feasible and the slackness condition is satisfied, then the QDPSA algorithm with any $V > 0$ stabilizes the system, with a resulting average backlog bound given by,

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}\{q_k(t)\}\le\frac{B^Q+V[\phi^Q(\varepsilon^Q)+u^{opt}]}{\varepsilon^Q\mu_{\min}} \quad (24)$$

Where $\phi^Q(\varepsilon^Q)$ is defined in Eq.22 and B^Q is defined as,

$$B^Q\triangleq\frac{a_{\max}^2+b_{\max}^2}{2}\sum_{k=1}^{K}\mu_k^2 \quad (25)$$

Furthermore, the average profit bound is given by,

$$\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}\{u(t)\}\ge u^{opt}-\frac{B^Q}{V} \quad (26)$$

Proof. Define the Lyapunov function as the half of sum of the squares of the individual queue backlogs,

$$L(\boldsymbol{Q}(t))\triangleq\frac{1}{2}\sum_{k=1}^{K}Q_k(t)^2 \quad (27)$$

Define the one-step conditional Lyapunov drift $\Delta(L(\boldsymbol{q}(t)))$ as follows,

$$\Delta(L(\boldsymbol{Q}(t)))\triangleq\mathbb{E}\{L(\boldsymbol{Q}(t+1))-L(\boldsymbol{Q}(t))\,|\,\boldsymbol{Q}(t)\} \quad (28)$$

Furthermore, define the drift-plus-penalty expression as $\Delta(L(\boldsymbol{Q}(t)))+V\mathbb{E}\{-u(t)\,|\,\boldsymbol{Q}(t)\}$, and we can obtain the following lemma.

Lemma 1. Suppose that *S*(*t*) is i.i.d. over slots. Under any pricing and scheduling algorithm, for all *t*, all possible values of $\boldsymbol{Q}(t)$, and all parameters $V\ge0$, the upper bound of the drift-plus-penalty expression satisfies,

$$\begin{aligned}
&\Delta(L(\boldsymbol{Q}(t)))+V\mathbb{E}\{-u(t)\,|\,\boldsymbol{Q}(t)\}\\
&\le B^Q+V\mathbb{E}\{-u(t)\,|\,\boldsymbol{Q}(t)\}+\sum_{k=1}^{K}Q_k(t)\mathbb{E}\{A_k(t)-B_k(t)\,|\,\boldsymbol{Q}(t)\}
\end{aligned} \quad (29)$$

Proof. Squaring the queue dynamic equation Eq.18 and using the facts that $\max[Q_k(t)-B_k(t),0]^2\le[Q_k(t)-B_k(t)]^2$ and $\max[Q_k(t)-B_k(t),0]\le Q_k(t)$ yields,

$$\begin{aligned}
&Q_k(t+1)^2=(\max[Q_k(t)-B_k(t),0]\;+A_k(t))^2\\
&\qquad\le[Q_k(t)-B_k(t)]^2+A_k(t)^2+2Q_k(t)A_k(t)\\
&\Rightarrow\frac{Q_k(t+1)^2-Q_k(t)^2}{2}\le\frac{A_k(t)^2+B_k(t)^2}{2}+Q_k(t)(A_k(t)-B_k(t))
\end{aligned} \quad (30)$$

Taking conditional expectations of Eq.30 and summing over $k\in\{1,...,K\}$, yields a bound on $\Delta(L(\boldsymbol{Q}(t)))$. Adding $V\mathbb{E}\{-u(t)\,|\,\boldsymbol{Q}(t)\}$ to both sides proves the result.

Every slot *t*, the QDPSA minimizes the right-hand-side of drift-plus-penalty expression (30) over all algorithms, so we have,

$$\begin{aligned}
&\Delta(L(\boldsymbol{Q}(t)))+V\mathbb{E}\{-u(t)\,|\,\boldsymbol{Q}(t)\}\\
&\le B^Q+V\mathbb{E}\{-u^*(t)\,|\,\boldsymbol{Q}(t)\}+\sum_{k=1}^{K}Q_k(t)\mathbb{E}\{A_k^*(t)-B_k^*(t)\,|\,\boldsymbol{Q}(t)\}
\end{aligned} \quad (31)$$

Where $-u^*(t)$, $A_k^*(t)$, $B_k^*(t)$ are the resulting values under any alternative algorithm (*p**(*t*), ***c****(*t*)). Suppose that the slackness condition is satisfied. Consider the *S*-only algorithm (*p**(*t*), ***c****(*t*)) that yields Eq.22 and Eq.23. Because *S*(*t*) is i.i.d. and the algorithm is only based on the current system state *S*(*t*), the values of $u^*(t)$, $A_k^*(t)$, $B_k^*(t)$ are independent of current queue backlogs ***Q***(*t*), that is,

$$\mathbb{E}\{-u^*(t)\,|\,\boldsymbol{Q}(t)\} = \mathbb{E}\{-u^*(t)\} \tag{32}$$

$$\mathbb{E}\{A_k^*(t) - B_k^*(t)\,|\,\boldsymbol{Q}(t)\} = \mathbb{E}\{A_k^*(t) - B_k^*(t)\} \tag{33}$$

Plugging Eq.22-Eq.23 and Eq.32-Eq.33 into (29) yields,

$$\Delta(L(\boldsymbol{Q}(t))) + V\mathbb{E}\{-u(t)\,|\,\boldsymbol{Q}(t)\} \le B^Q + V\phi^Q(\varepsilon^Q) - \varepsilon^Q \sum_{k=1}^{K} Q_k(t) \tag{34}$$

Taking an expectation of Eq.34 with respect to the distribution of ***Q***(*t*) yields,

$$\begin{aligned} &\mathbb{E}\{L(\boldsymbol{Q}(t+1)) - L(\boldsymbol{Q}(t))\} + V\mathbb{E}\{-u(t)\} \\ &\quad \le B^Q + V\phi^Q(\varepsilon^Q) - \varepsilon^Q \sum_{k=1}^{K} \mathbb{E}\{Q_k(t)\} \end{aligned} \tag{35}$$

Summing over $\tau \in \{0,\ldots,t\text{-}1\}$ and dividing by *t* yields,

$$\begin{aligned} &\frac{\mathbb{E}\{L(\boldsymbol{Q}(t)) - L(\boldsymbol{Q}(0))\}}{t} + \frac{V}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{-u(\tau)\} \\ &\quad \le B^Q + V\phi^Q(\varepsilon^Q) - \frac{\varepsilon^Q}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}\{Q_k(\tau)\} \end{aligned} \tag{36}$$

Rearranging the items, dividing by ε^Q , we can obtain,

$$\begin{aligned} &\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}\{Q_k(\tau)\} \\ &\le \frac{B + V\phi^Q(\varepsilon^Q)}{\varepsilon^Q} + \frac{V}{\varepsilon^Q t}\sum_{\tau=0}^{t-1}\mathbb{E}\{-u(\tau)\} + \frac{\mathbb{E}\{L(\boldsymbol{Q}(0))\}}{\varepsilon^Q t} \end{aligned} \tag{37}$$

Taking limits as $t \to \infty$, we can obtain,

$$\begin{aligned} &\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}\{Q_k(t)\} \le \frac{B^Q + V[\phi^Q(\varepsilon^Q) + u^{opt}]}{\varepsilon^Q} \\ &\Rightarrow \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}\{q_k(t)\} = \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\frac{\mathbb{E}\{Q_k(t)\}}{\mu_k} \\ &\qquad \le \frac{1}{\mu_{\min}}\limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{k=1}^{K}\mathbb{E}\{Q_k(t)\} \\ &\qquad \le \frac{B^Q + V[\phi^Q(\varepsilon^Q) + u^{opt}]}{\varepsilon^Q \mu_{\min}} \end{aligned} \tag{38}$$

According to Theorem 4.5 of [18], there exists an *S*-only pricing $p_k^*(t)$ and scheduling ***c****(*t*) ∈ *C* algorithm that satisfies,

$$\mathbb{E}\{-\hat{u}(p^*(t), \boldsymbol{c}^*(t), S(t))\} \le -u^{opt} + \delta^Q \tag{39}$$

$$\mathbb{E}\{\hat{a}_k(p^*(t), S(t))\} \le \mathbb{E}\{\hat{b}_k(\boldsymbol{c}^*(t), S(t))\} \ \forall k \tag{40}$$

Now we fix $\delta^Q > 0$, and consider the *S*-only algorithm that yields Eq.39-Eq.40. Plugging Eq.32-Eq.33 and Eq.39-Eq.40 into the drift bound Eq.29 and taking $\delta^Q \to 0$ yields,

$$\Delta(L(\boldsymbol{Q}(t))) + V\mathbb{E}\{-u(t)\,|\,\boldsymbol{Q}(t)\} \le B^Q - Vu^{opt} \tag{41}$$

According to the above equation and Theorem 4.2 in [18], we can conclude that all queues are mean rate stable. Furthermore, similar to the proof of queue bound, we can get profit bound, as described in Eq.26.

IV. Simulation

We call the BDPSA as the basic dynamic pricing and scheduling algorithm, in which all services are considered to be equally important. In this section, we perform simulations for BDPSA and QDPSA algorithms and compare the corresponding results of both algorithms. Set $\mu_1 = 1.5$ and $\mu_2 = 1$. Further, set $p_k(t) = \sqrt{\mu_k}\,p(t)$. In our algorithms, the demand function and service rate function can be any form. All the following parameters are set same as those in [15]. Assume that the network state and service provider state are selected in {*Good*, *Bad*} and appear with equal probability. For simplicity, we only consider two types of service and the arrival rates of both services are the same under certain network state. Further, the demand function is given by,

$$\hat{a}_k(p_k(t), (Bad, \cdot)) = \begin{cases} 4 & 0 \le p(\mathrm{t}) \le 1 \\ -6p(t) + 10 & 1 \le p(\mathrm{t}) \le \dfrac{3}{2} \\ -\dfrac{2}{17}p(t) + \dfrac{20}{17} & \dfrac{3}{2} \le p(\mathrm{t}) \le 10 \end{cases} \tag{42}$$

$$\hat{a}_k(p_k(t), (Good, \cdot)) = \begin{cases} 10 - p(\mathrm{t}) & 0 \le p(t) \le 2 \\ -6p(t) + 20 & 2 < p(t) \le 3 \\ -\dfrac{1}{7}p(t) + \dfrac{17}{7} & 3 < p(t) \le 10 \end{cases} \tag{43}$$

The service rate is given by:

$$\hat{b}_k(\boldsymbol{c}(t), (\cdot, Bad)) = \log(1 + c_k(t)) \tag{44}$$

$$\hat{b}_k(\boldsymbol{c}(t), (\cdot, Good)) = \log(1 + 2c_k(t)) \tag{45}$$

The cost $c_1(t)$ and $c_2(t)$ satisfy:

$$c_1(t) + hc_2(t) = c \tag{46}$$

Where $h = 1.01,\ c = 10$.

.We apply above settings into BDPSA and QDPSA algorithms and simulate for 100000 slots, thereby yielding the results. We plot the average backlog v.s. *V* under BDPSA and QDPSA, as shown in Fig.2 and Fig.3.

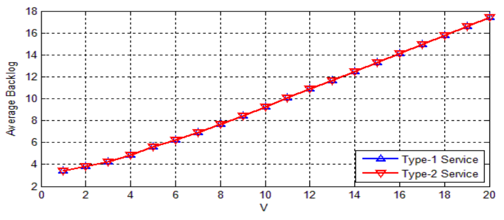

Fig.2. Average backlog of both services under BDPSA v.s. *V*

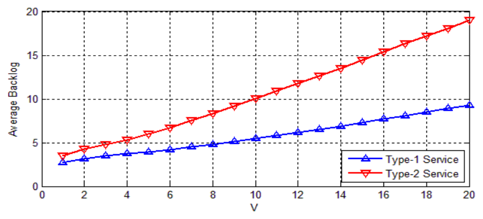

Fig.3. Average backlog of both services under QDPSA v.s. *V*

It can be concluded that the QDPSA algorithm with higher QoS coefficient can eventually reduce the average queue backlog. This is because that in the QDPSA algorithm, all the QoS queues are considered to be equally important, thereby yielding similar QoS queue bakclog and the service with higher QoS coefficient results in lower real queue backlog. Moreover, we find that backlogs of both algorithms increase along with *V*, which is consistent with Eq.24.

Furthermore, we announce higher price for the service with higher coefficient, thereby yielding higher profit, as shown in Fig.4. As *V* grows larger, the profit for both algorithms converges on a constant and the QDPSA can achieve a [O(1/*V*), O(*V*)] profit-backlog tradeoff, which is consistent with Eq.24 and Eq.26.

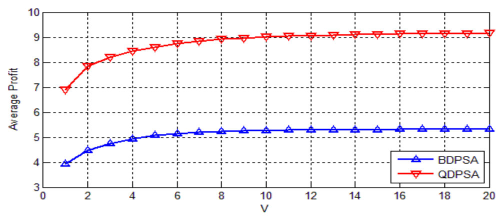

Fig.4. Average profit of BDPSA and QDPSA v.s. V

V. Conclusion

In this paper, we consider the different QoS requirements for different services in wireless cloud computing. By introducing the concept of QoS aware queue, we propose the QDPSA algorithm. QDPSA can be implemented based on the currently available information and achieve the optimal profit. More importantly, the QDPSA algorithm can achieve different QoS requirements.

References

[1] X. Liu, C. W. Yuan, Y. Li, Z. Yang, and B. Cao, "A Lightweight Algorithm for Collaborative Task Execution in Mobile Cloud Computing," Wireless Personal Communications, vol. 86, pp. 579-599, 2016.

[2] Y. Kim, J. Kwak, and S. Chong, "Dual-side dynamic controls for cost minimization in mobile cloud computing systems," in Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on, 2015, pp. 443-450.

[3] S. Yang, D. Kwon, H. Yi, Y. Cho, Y. Kwon, and Y. Paek, "Techniques to Minimize State Transfer Costs for Dynamic Execution Offloading in Mobile Cloud Computing," IEEE Transactions on Mobile Computing, vol. 13, pp. 2648-2660, 2014.

[4] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, et al., "Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications," Wireless Communications, vol. 20, pp. 14-22, 2013.

[5] BMC Workload Automation: Helping Cloud Computing take Flight [Online]. Available: http://documents.bmc.com/products/documents /62/56/286256/286256.pdf.

[6] Y. Xue, B. Li, and K. Nahrstedt, "Optimal resource allocation in wireless ad hoc networks: A price-based approach," IEEE Transactions on Mobile Computing, vol. 5, pp. 347-364, 2006.

[7] B. Wang, Z. Han, and K. R. Liu, "Distributed relay selection and power control for multiuser cooperative communication networks using stackelberg game," IEEE Transactions on Mobile Computing, vol. 8, pp. 975-990, 2009.

[8] R. Tan, V. Badrinath Krishna, D. K. Yau, and Z. Kalbarczyk, "Impact of integrity attacks on real-time pricing in smart grids," in Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security, 2013, pp. 439-450.

[9] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: pricing methods and optimization algorithms," IEEE Communications Surveys & Tutorials, vol. 17, pp. 152-178, 2015.

[10] L. Duan, J. Huang, and B. Shou, "Pricing for local and global Wi-Fi markets," IEEE Transactions on Mobile Computing, vol. 14, pp. 1056-1070, 2015.

[11] P. Xiong, Z. Wang, S. Malkowski, et al. "Economical and Robust Provisioning of N-Tier Cloud Workloads: A Multi-level Control Approach," in International Conference on Distributed Computing Systems, IEEE, 2011, pp.571-580.

[12] P. Xiong, Y. Chi, S. Zhu, et al. "ActiveSLA:a profit-oriented admission control framework for database-as-a-service providers" in ACM Symposium on Cloud Computing, 2011, pp.1-14.

[13] P. Xiong, Y. Chi , S. Zhu, et al. "Intelligent management of virtualized resources for database systems in cloud environment" in IEEE International Conference on Data Engineering, 2011, pp.87-98.

[14] P. Hande, M. Chiang, R. Calderbank, and S. Rangan, "Network pricing and rate allocation with content provider participation," in INFOCOM 2009, IEEE, 2009, pp. 990-998.

[15] L. Huang and M. J. Neely, "The optimality of two prices: Maximizing revenue in a stochastic communication system," IEEE/ACM Transactions on Networking (TON), vol. 18, pp. 406-419, 2010.

[16] S. Ren and M. van der Schaar, "Dynamic scheduling and pricing in wireless cloud computing," IEEE Transactions on Mobile Computing, vol. 13, pp. 2283-2292, 2014.

[17] S. Singh, I. Chana. "QoS-Aware Autonomic Resource Management in Cloud Computing: A Systematic Review" Acm Computing Surveys, vol. 48, pp.1-46.

[18] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," Synthesis Lectures on Communication Networks, vol. 3, pp. 1-211, 2010.