A Wait-and-See Two-Threshold Optimal Sleeping Policy for a Single Server With Bursty Traffic

Bingjie Leng, *Student Member, IEEE,* Xueying Guo, *Student Member, IEEE,* Xi Zheng, *Student Member, IEEE,* Bhaskar Krishnamachari, *Member, IEEE,* and Zhisheng Niu, *Fellow, IEEE*

***Abstract*—Making idle servers sleep is considered to be a key approach to reduce energy consumption of various information and communication systems. Optimal sleeping policies for a single server have been derived only for non-bursty traffic in prior work. In this paper, for the first time, we study the optimal sleeping operation for a single server with bursty traffic to answer the question of whether server sleeping can bring extra benefit with bursty traffic or not. Key factors including switchover energy consumption as well as delay performance are considered. We formulate the problem as a POMDP (partially observable Markov decision process), and show that it can be solved by observing Time Elapsed Since the Last Arrival (TESLA). The optimal sleeping policy is shown to be a two-threshold policy with a wait-and-see feature, i.e., the server would wait a period of time to see if there are any extra arrivals before switching modes. Simulation results show that with the optimal sleeping mechanism, traffic burstiness can enhance system performance on energy cost and delay penalty.**

***Index Terms*—Bursty traffic, POMDP, sleeping mechanism, two-threshold policy**

I. Introduction

Nowadays, one of the main challenges in communication networks is to reduce energy consumption. In 5G networks, it is claimed that energy consumption and cost per bit have to be reduced by at least 100 times [2], which reveals the great importance of "greening" the networks.

In cellular networks, about 60-80% of the total energy is consumed by base stations (BSs) [3]. Moreover, even with little traffic load, a BS would still consume more than 90% of the peak energy [4]. As a result, BS sleeping mechanisms have been studied widely in recent years and are considered as one of the most efficient methods to enhance energy efficiency. But they may bring extra delay for the system. Sleeping designs and the trade-off between energy and delay are studied in previous work [5]–[7], [16], in which the traffic is modeled as a random memoryless Poisson process. In practice, however, traffic arrivals have a bursty nature, especially for data and video traffic providing multimedia services [8]. Bursty traffic could bring in congestion to the networks and worsen system performance. However, combining sleeping mechanisms and bursty traffic, there might be more chances for a base station to sleep. Therefore, in this work, we study the optimal sleeping policy designed for bursty traffic model and analyze the influence of traffic burstiness to system performance.

In HetNet (heterogeneous networks) and HCN (hyper-cellular networks) [9], switching off small BSs in HetNet or traffic BSs in HCN would not cause a coverage problem, which makes it rational to apply sleeping schemes in these scenarios. As a start, cells with no traffic coverage overlap are considered, in which case, the sleeping operations of different cells are independently considered. Thus, in this work, a single base station is considered. Notice that this work could also be applied to servers in a data center, as these two scenarios share the same demands in energy efficiency and the same nature in traffic burstiness [19]. Hence, in the rest of the paper, a whole BS is considered as a single server.

In [16], the authors discuss multiple hysteresis sleeping mechanisms and their delay performance under Poisson traffic arrivals. In [19], the authors consider a single server and find the optimal sleeping policy with Poisson arrivals by formulating the problem into a continuous-time Markov decision process (MDP), where the system state only changes when there is an arrival or a departure. The optimal policy is proven to be a two-threshold policy, where the queue length in the server is compared to an *active* threshold and a *sleep* threshold, to decide whether the action is to be active, to sleep or to stay at the current operation mode. In this situation, if there are no arrivals or departures for a long time, which would happen with a higher probability under bursty traffic, keeping the server *active* would potentially consume more power. If the server could make decisions in between the queue length changes, we might gain higher energy efficiency with more complex bursty traffic model. In this work, to get the optimal sleeping policy, models based on MDP are considered.

One of the typical models to analyze traffic burstiness is the Interrupted Poisson Process (IPP) [10]–[12]. In [10], the authors studied the sleeping performance of a single server under bursty traffic with IPP model by assuming an N-based policy, where the server goes to sleep when the buffer is empty and wakes up when there are N jobs aggregated in the buffer. They find the optimal transmit power and the waking-up threshold N to minimize the energy consumption, with a certain constraint of delay. However, that work does not consider whether the N-based policy is the optimal policy under bursty traffic. But in this work, the optimal policy is

This work is an extended version of the paper presented at the IEEE Global Communications Conference (IEEE GLOBECOM), Dec. 2016 [1].

Bingjie Leng, Xi Zheng and Zhisheng Niu are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. E- mail: {lengbj14, zhengx14}@mails.tsinghua.edu.cn; niuzhs@tsinghua.edu.cn.

Xueying Guo is currently with the Department of Computer Science, University of California, Davis, CA 95616-8562, USA. E-mail: xyguo@ucdavis.edu.

Bhaskar Krishnamachari is with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, USA. E-mail: bkrishna@usc.edu.

Manuscript received April 11, 2017; revised August 4, 2017.

given and analyzed. Moreover, we assume that the arrival phases are not observable in our system, the probability of which has to be updated periodically with POMDP. As a result, we consider a discrete-time Interrupted Bernoulli Process (IBP), which can reflect the burstiness of traffic. Our result shows that for the optimal policy under bursty arrivals, the *sleep* decision should be made not only based on the queue length, but also based on the burstiness of traffic arrivals.

A POMDP is a generalization of an MDP where we cannot observe the MDP-determined system state directly. Instead, a probability distribution over the state space is recorded and updated. POMDP has been used to analyze the communication systems widely in [13]–[15]. In [14], the authors use POMDP to analyze opportunistic spectrum access. Decisions of channel selection are discussed in [13], [15]. To our best knowledge, our work provides the first optimal solution for the IBP arrivals with POMDP formulation.

In our previous work [1], we only consider a single server with symmetrical switching energy cost, and compare the optimal policy of POMDP formulation with the optimal policy of memoryless traffic given in prior work [19]. However, the two-threshold based structure and the insights are not given and discussed. The optimal policy has not been compared with the optimal policy if the arrival state is observable, which is actually the lower bound of system cost. Furthermore, the trends of thresholds changing with system parameters and delay performance are not analyzed. In this paper, we make the following contributions:

- We formulate this decision problem into a POMDP where asymmetrical switching energy cost, operation energy and delay penalty are considered. Numerical results for the optimal sleeping policy are given.
- The optimal sleeping policy is proven to be a hysteretic policy and a conjecture is given stating that the optimal policy is a two-threshold policy, where the thresholds not only depend on the queue length, but also the time elapsed since the last arrival (TESLA). By numerical results, the policy is found as a wait-and-see policy. The server would wait for a period of time to see if there are any extra arrivals before switching modes.
- The trends how the thresholds change with system parameters are analyzed. It is found that a larger switching cost leads to higher *active* thresholds, which also indicates higher hysteresis. Increasing the traffic burstiness or decreasing delay tolerance causes the server to wake up earlier. When the switching cost is sufficiently large, once the server start to serve tasks, it would never choose to sleep.
- We find the system with no bursty information gives the upper bound of system cost to our POMDP formulation, while the system with sufficient bursty information (the observable arrival phases) provides the lower bound.
- The total cost, energy consumption per bit and delay performance are analyzed by simulation. The system cost decreases with traffic burstiness, which indicates applying sleeping mechanisms to a server with bursty traffic can enhance performance. Even if server is allowed to sleep only when the queue is empty, which yields the minimum delay, more energy can be saved in POMDP formulation compared to the optimal policy given in [19].

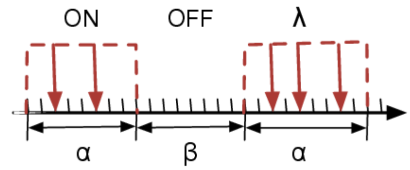


Fig. 1. An example of IBP arrivals with parameters (λ, α, β).

The rest of this paper is organized as follows. The system model and problem formulation are introduced in Section II. The state simplification and optimality equation are given in Section III. We analyze the structure of the optimal policy in Section IV. In Section V, numerical results of the optimal policies and simulations of system performance are analyzed. Conclusions and future directions are presented in Section VI.

II. System Model and Problem Formulation

A. System Model

We consider a single server queue with a buffer size of B. The system is modeled as a discrete-time system where time is divided into time slots and there is at most one arrival or departure at one time slot. The length of each time slot is Δ seconds.

There are two operation modes for the server, *active* mode and *sleep* mode. The server processes the jobs only when it is *active*, with a steady data rate of x bits/s; otherwise, the server is in *sleep* mode and no job could be processed. The server consumes a constant E_{active} energy each time slot when it is active. No energy would be consumed when the server is in the *sleep* mode. Furthermore, asymmetrical switching cost is assumed, i.e., switching the server from *active* to *sleep* mode, or from *sleep* to *active* mode consumes $E_{\text{sw}}^{\text{a}\to\text{s}}$ or $E_{\text{sw}}^{\text{s}\to\text{a}}$ energy, respectively. For example, in [16], the authors claim that $E_{\text{sw}}^{\text{a}\to\text{s}} = 0$ and $E_{\text{sw}}^{\text{s}\to\text{a}} > 0$.

Assume that the job arrivals follow an IBP with parameters (λ, α, β), which includes ON phase and OFF phase. As shown in Fig.1, the time spent in the ON and OFF phases are both geometrically distributed, with expected durations α^{-1} and β^{-1} time slots, respectively. That is, given that the process is in the ON phase (or the OFF phase) at time slot n, it would change to the OFF phase (or the ON phase) with probability α (or β), or would remain in the same phase at the next time slot $n+1$ with probability $(1-\alpha)$ (or $1-\beta$).

In the ON phase, the time interval between two job arrivals during this period also follows a geometric distribution with success probability λ at each time slot, which is also known as the arrival rate. That is, the mean interarrival time in the ON phase is λ^{-1} time slots. There is no job arrival in the OFF phase. Therefore, with varying λ, α and β, traffic with different bursty levels is able to be generated. Based on [17], the average interarrival time, $(\lambda')^{-1}$, is given by

$$(\lambda')^{-1} = \frac{1}{\lambda}\frac{\alpha+\beta}{\beta}. \tag{1}$$

The squared coefficient of variation (SCV) of the interarrival time, C^2, is given by

$$C^2 = 1 + \lambda[\frac{\alpha(2-\alpha-\beta)}{(\alpha+\beta)^2} - 1]. \quad (2)$$

The larger C^2 is, the higher bursty level the arrival process has.

We also assume that the size of each job is exponentially distributed with average l bits. Since the discrete-time system is considered, the service time for each job will be rounded up into a geometrically distributed random variable with mean service time μ^{-1} time slots, where $\mu = 1 - \exp(-\frac{\Delta x}{l})$. Assume that $\mu > \lambda$ so that the traffic load at the busy phase would not exceed 1. Note that with arbitrarily small period of one time slot, the IBP can approximate arbitrarily well the continuous time interrupted Poisson process (IPP), which is one of the most typical models to reflect traffic burstiness.

B. POMDP Formulation

If the state of the arrival process (ON or OFF) can be observed directly, the optimal operation decision problem for the server can be formulated into an MDP, by adding the phase of the arrival process S into the system of MDP problem with memoryless arrivals studied in [19]. Note that $S \in \{\text{ON}, \text{OFF}\}$. However, in general, the actual phase of the arrival IBP process cannot be observed directly. As a result, we formulate this problem as a discrete-time partially observable Markov decision process.

At each time slot, the server takes a certain action based on the current state and the system transits to another state with known transition probabilities. After that, the system would receive an observation based on the new state and the action, which helps us determine the underlying state and make the decision for the next time slot. Finally, the action and the former state incur a cost for the system. Our objective is to decide which action to choose at each time slot to minimize the total expected discounted costs over time. The POMDP formulation of this optimal operation problem is defined as follows.

The actual system state is (S, R, Q, W), where the arrival process is in phase S, the queue length in the server is $Q \in \{0, 1, 2, ..., B\}$ and the operation state of the server is $W \in \{\textit{active}, \textit{sleep}\}$ in the current time slot. R is a Boolean variable that $R = 1$ indicates there is a job arrival in the passing time slot. Note that departures are assumed to happen earlier than arrivals at each time slot. Thus, $R = 1$ indicates there has to be at least one job in the buffer, that is, $Q > 0$. We use S_n, R_n, Q_n and W_n to denote the state at time slot n.

At each time slot, the server takes an action $a \in \Phi$, where the action space $\Phi = \{\textit{active}, \textit{sleep}\}$. After transiting to a new state, an observation (R, Q, W) is taken by the server. All the state elements except the IBP phase S can be observed directly.

The cost function is determined by the current state and the action. We consider the energy consumption including both the switching energy and the operation energy when the server is active. The delay of the jobs are counted in the cost function as a penalty proportional to the sojourn time of each job. By changing the weight between energy consumption and the delay penalty, we are able to model systems with different emphasis on delay tolerance and energy saving.

The solution to this POMDP is a policy that indicates how to choose an optimal action at each state, or correspondingly, at each time slot.

C. Belief-MDP Formulation

In the POMDP formulation, the server would keep a belief of the probability distribution over the state space at the current time slot, upon the action taken and the observation (R, Q, W). It is not hard to find that the only uncertainty in the system state is the IBP phase S. Therefore, we define p to indicate the probability of being in ON phase knowing all the past observations and actions, which is known as the *belief*, and p_n denotes the belief at time slot n. Because the state is Markovian and the ON phase probability can be extracted only from R, but not Q and W, the belief p_n can be updated only based on the previous belief p_{n-1} and R_n. That is,

$$p_n = \Pr\{S_n = \text{ON} | R_n, p_{n-1}\}. \quad (3)$$

To solve this POMDP, we formulate this problem into a belief-MDP where each belief is a system state. As a result, the state of this belief-MDP is $i = (p, R, Q, W)$. If $R_n = 1$, the arrival process has to be in the ON phase, therefore, $p_n = 1$; otherwise, if $R_n = 0$, p_n can be calculated conditionally based on p_{n-1}, α and β. Note that i_n denotes the system state at time slot n.

The action space is still Φ. The action taken at time slot n decides the server's operation mode at time slot $n + 1$, which is expressed by:

$$W_{n+1} = A_\pi(i_n), \quad (4)$$

where π is a certain policy that determines the rules for the server to take an action. $A_\pi(i_n)$ denotes the action taken under state i_n based on policy π.

We define $C(i, A_\pi(i))$ as the cost function received at state i under policy π, which is a weighted sum of energy consumption and delay, given by:

$$C(i, A_\pi(i)) = C_{\text{sw}}(W, A_\pi(i)) + \omega Q + A_\pi(i) E_{\text{active}}, \quad (5)$$

where $C_{\text{sw}}(m, n), m, n \in \Phi$ is the function that calculates the energy consumption caused by the operation mode switching. The expression is given as:

$$C_{\text{sw}}(m, n) = \begin{cases} E_{\text{sw}}^{\text{a}\to\text{s}} & m = \textit{active}, n = \textit{sleep}, \\ E_{\text{sw}}^{\text{s}\to\text{a}} & m = \textit{sleep}, n = \textit{active}. \end{cases} \quad (6)$$

ω in Eq.(5) is the weight of to the delay penalty caused by congestion, which is proportional to the queue length Q, and E_{active} is the constant energy consumption per time slot of an active server.

The objective of choosing the server operation policy is to minimize the expected discounted sum of the costs over time, and the value function is shown below.

$$V(i_0) = \min_\pi \text{E}[\sum_{n=0}^{\infty} \gamma^n C(i_n, A_\pi(i_n))], \quad (7)$$

where γ is the discount factor that indicates how important the future costs are to the value function.

III. Approach to the Optimal Policy

In this section, we first simplify the (p, R, Q, W) state by replacing the (p, R) tuple with TESLA τ. The transition probabilities and the optimality equation are given next.

A. State Simplification

From the definition of p_n in Eq.(3), it is found that if $R_n = 1$, then $p_n = 1$ and if $R_n = 0$, p_n can be calculated by iterations, which is given as follows:

$$\begin{aligned} p_n &= \Pr\{S_n = \mathrm{ON} | R_n = 0, p_{n-1}\} \\ &= \frac{\Pr\{S_n = \mathrm{ON}, R_n = 0 | p_{n-1}\}}{\Pr\{R_n = 0 | p_{n-1}\}} \\ &= \frac{(1-\lambda)\Pr\{S_n = \mathrm{ON} | p_{n-1}\}}{1 - \lambda \Pr\{S_n = \mathrm{ON} | p_{n-1}\}} \\ &= \frac{(1-\lambda)[(1-\alpha-\beta)p_{n-1} + \beta]}{1 - \lambda[(1-\alpha-\beta)p_{n-1} + \beta]}. \end{aligned} \quad (8)$$

Note that $\Pr\{S_n = \mathrm{ON} | p_{n-1}\} = p_{n-1}\Pr\{S_n = \mathrm{ON} | S_{n-1} = \mathrm{ON}\} + (1-p_{n-1})\Pr\{S_n = \mathrm{ON} | S_{n-1} = \mathrm{OFF}\} = (1-\alpha-\beta)p_{n-1} + \beta$. Define a sequence $\{m_k\}$ with $m_0 = 1$, and the iteration between m_k and m_{k-1} is the same as Eq.(8). It is obvious that all the possible values of p_n are included in $\{m_k\}$. Moreover, we find the subscript of the sequence k corresponds to the case TESLA $\tau = k$, which means there have been k time slots with no arrivals. It indicates TESLA covers exactly the same information of the tuple (p, R). Therefore, we are able to simplify the $i = (p = m_k, R, Q, W)$ system state into $i = (\tau = k, Q, W)$ state. It is also shown that the state space of our belief-MDP is countable, as the values of τ are non-negative integers. In the rest of the paper, we use (τ, Q, W) formulation instead.

B. Transition Probability

One of the most important part in the transition probability calculation is to calculate the probability of being in the ON arrival phase at time slot $n+1$, given τ_n. The calculation is given below.

$$\begin{aligned} q_k &\triangleq \Pr\{S_{n+1} = \mathrm{ON} | \tau_n = k\} \\ &= \Pr\{S_{n+1} = \mathrm{ON} | S_n = \mathrm{ON}, \tau_n = k\}\Pr\{S_n = \mathrm{ON} | \tau_n = k\} \\ &+ \Pr\{S_{n+1} = \mathrm{ON} | S_n = \mathrm{OFF}, \tau_n = k\}\Pr\{S_n = \mathrm{OFF} | \tau_n = k\} \\ &= (1-\alpha)m_k + \beta(1-m_k). \end{aligned} \quad (9)$$

Since the system state updates at each time slot, the queue length in the buffer might remain the same in our formulation. The transition probability at state $i_n = (\tau = k, Q, W)$, are discussed in three cases based on the value of Q and given as follows.

(1) For $Q = 0$,

$$\Pr\{i_{n+1} = (0, 1, A_\pi(k, 0, W)) | i_n = (k, 0, W)\} = q_k\lambda,$$
$$\Pr\{i_{n+1} = (k+1, 0, A_\pi(k, 0, W)) | i_n = (k, 0, W)\} = 1 - q_k\lambda.$$

Note that departures take place earlier than arrivals at each time slot. Therefore, when $Q = 0$, τ cannot be 0.

(2) For $Q = B$,

$$\Pr\{i_{n+1} = (0, B, active) | i_n = (k, B, W)\} = q_k\lambda,$$
$$\Pr\{i_{n+1} = (k+1, B, active) | i_n = (k, B, W)\} = (1 - q_k\lambda)(1-\mu),$$
$$\Pr\{i_{n+1} = (k+1, B-1, active) | i_n = (k, B, W)\} = (1 - q_k\lambda)\mu.$$

We assume that if there are B jobs in the buffer at the current time slot, the server will always take the *active* action to process the jobs.

(3) For $0 < Q < B$,

$$\Pr\{i_{n+1} = (0, Q+1, sleep) | i_n = (k, Q, W)\} = q_k\lambda, \quad (10)$$
$$\Pr\{i_{n+1} = (k+1, Q, sleep) | i_n = (k, Q, W)\} = 1 - q_k\lambda, \quad (11)$$
$$\Pr\{i_{n+1} = (0, Q, active) | i_n = (k, Q, W)\} = q_k\lambda\mu, \quad (12)$$
$$\Pr\{i_{n+1} = (0, Q+1, active) | i_n = (k, Q, W)\} = q_k\lambda(1-\mu), \quad (13)$$
$$\Pr\{i_{n+1} = (k+1, Q, active) | i_n = (k, Q, W)\} = (1 - q_k\lambda)(1-\mu), \quad (14)$$
$$\Pr\{i_{n+1} = (k+1, Q-1, active) | i_n = (k, Q, W)\} = (1 - q_k\lambda)\mu. \quad (15)$$

C. Optimality Equation

In the afore-mentioned formulation, we are facing a belief-MDP problem with a finite action space and an infinite but countable state space, because the value of t can be arbitrarily large. Based on [20, pp. 236], it is proven that the *optimality equation* [20, pp. 146] still holds and there exists a stationary optimal policy. The optimality equation is given by:

$$V(i) = \min_{a \in \Phi}\{C(i, a) + \gamma \sum_j P^a_{i \to j} V(j)\}, \quad (16)$$

where a is the action taken by the server and $P^a_{i \to j}$ is the transition probability from state i to state j. The two terms on the right hand side of Eq.(16) denotes the immediate cost at the current time slot and the expected future cost, respectively. As a result, the optimal operation policy π^* is

$$A_{\pi^*}(i) = \arg\min_{a \in \Phi}\{C(i, a) + \gamma \sum_j P^a_{i \to j} V(j)\}, \forall i. \quad (17)$$

IV. Properties of Optimal Policy

In this section, it is shown that the optimal policy under IBP traffic model has similar properties with the optimal policy under memoryless traffic model, which is proven to be a two-threshold policy in [19]. Firstly, we prove that the optimal policy has the *hysteretic* property in Section IV-A. Furthermore, we give a conjucture on the optimality of two-threshold based policy, by bringing in the optimality analysis and proofs for the MDP problem with known arrival state in Section IV-B.

A. Hysteretic Property

Due to the switching cost between operation modes, the server should not change modes too frequently. This leads to *hysteretic* property, which is defined as follows.

***Definition** 1:* A policy π is hysteretic if for some $b \in \Phi$, $A_\pi(\tau, Q, b) = a$ implies $A_\pi(\tau, Q, a) = a$, $a \in \Phi$.

Even though there is switching cost, with certain τ and Q, if the server chooses to switch to the better mode, then the server would stay in this "better" mode if it is already in this status. Theorem 1 and the corresponding proof are given accordingly.

***Theorem** 1:* The optimal policy defined by Eq.(17) is hysteretic.

Proof: Eq.(17) can be rewritten as

$$A_{\pi^*}(i) = \arg\min_{a\in\Phi}\{\sigma(W, a) + \eta(\tau, Q, a)\}, \tag{18}$$

where

$$\sigma(W, a) \triangleq C_{\text{sw}}(W, a), \tag{19}$$

$$\eta(t, Q, a) \triangleq \omega Q + aE_{\text{active}} + \gamma \sum_j P^a_{i\to j} V(j). \tag{20}$$

It is found that function σ has the following properties:

$$\sigma(c, d) \le \sigma(c, e) + s(e, d), \quad \forall c, d, e \in \Phi, \tag{21}$$

$$\sigma(c, c) = 0, \quad \forall c \in \Phi. \tag{22}$$

According to Lemma 1 in [18], the optimal policy is hysteretic. ■

B. Two-threshold Based Policy

In the following analysis, to dig deeper into the structure of the optimal policy, we start with investigating the optimal policy structure of an MDP with system state defined as $\{(S_n, Q_n, W_n), n \ge 0\}$, where the arrival phase is observable, instead of the belief-MDP $\{(\tau_n, Q_n, W_n), n \ge 0\}$. In the belief-MDP problem, TESLA τ is introduced to approximate the probability of the arrival phase. With same cost function and optimization objective, the optimal policy structure for the MDP $\{(S_n, Q_n, W_n), n \ge 0\}$ should give hints on studying the derived belief-MDP problem.

For this MDP $\{(S_n, Q_n, W_n), n \ge 0\}$, to distinguish some of the notations from Section II, the system state is expressed as $\tilde{i} = (S, Q, W)$. The action taken based on the system state at time slot n, under a certain policy $\tilde{\pi}$, is given by:

$$W_{n+1} = \tilde{A}_{\tilde{\pi}}(\tilde{i}_n). \tag{23}$$

The cost function consists of the same three parts as the belief-MDP cost function given by Eq.(5). The transition probabilities under different Q values can be calculated similarly as the belief-MDP problem. The optimality equation still holds in this case and there exists a stationary optimal policy.

With the same factors considered in the cost function between (S, Q, W) formulation with IBP arrival and (Q, W) with memoryless traffic model [19], the following conjecture is given that the optimal policy of MDP (S, Q, W) is also a two-threshold based policy.

***Conjecture** 1:* The optimal policy of MDP $\{(S_n, Q_n, W_n),$ $n \ge 0\}$ is a state-based two-threshold policy. Concretely, for each $S \in \{\text{ON, OFF}\}$, there exist two thresholds $0^- \le \theta^{\{S\}}_{sleep} \le \theta^{\{S\}}_{active}$ such that for $W \in \{active, sleep\}$,

$$\tilde{A}_{\tilde{\pi}^*}(S, Q, W) = active, \text{for } Q \ge \theta^{\{S\}}_{active}, \tag{24}$$

$$\tilde{A}_{\tilde{\pi}^*}(S, Q, W) = sleep, \text{for } Q \le \theta^{\{S\}}_{sleep}, \tag{25}$$

$$\tilde{A}_{\tilde{\pi}^*}(S, Q, W) = W, \text{for } \theta^{\{S\}}_{sleep} < Q < \theta^{\{S\}}_{active}. \tag{26}$$

When a *sleep* threshold is 0^-, it indicates that the server would never switch to *sleep* mode from *active* mode. Once the server turns into *active* mode, it will never sleep.

***Remark** 1:* This conjecture follows the following insights. With a given arrival phase S and a certain server mode W, the decision at each state only depends on the current queue length. Therefore, the optimal policy has to be a threshold-based policy, where with different queue length, the server would take corresponding actions.

Why two-threshold, yet not three-threshold? The key factor that forms a two-threshold policy is that there are only two possible actions in the action space Φ and the switching costs considered here are constant. If there are more possible actions considered, the structure for the optimal policy has to be more complex. Moreover, if the switching cost for a certain switch, e.g. from *active* to *sleep* mode, is varying with queue length instead of a constant, the threshold structure might be much more complicated, and there might be more thresholds. Yet with constant switching costs, intuitively, the server would choose to turn into *active* mode to avoid huge delay penalty, which might go to infinity when the queue length is sufficiently large. With increasing queue length, there is no reason for the server to turn into *sleep* from *active* mode. As a result, the optimal policy should be a two-threshold policy.

Why two-threshold, yet not one-threshold? Imagine the case of zero switching cost, only delay and operating energy consumption are considered. The decision of each time slot will only depend on the current queue length, but no longer be influenced by the current operating mode of the server. As a result, the optimal policy should be a one-threshold policy instead of a two-threshold policy. However, we consider nonzero switching cost, which leads to two-threshold optimal policy eventually.

In fact, the N-policy mentioned in [10] with bursty traffic model is also a two-threshold policy with *active* threshold N and *sleep* threshold 0 for both arrival phases. Since the variation of *active* and *sleep* thresholds gives the system more flexibility in sleeping mechanism design, the state-based two-threshold policy should outperform N-policy when other system parameters are fixed.

More structural results can be given as follows.

***Theorem** 2:* If the buffer size is infinite and Conjecture 1 holds, then the optimal *sleep* thresholds $\theta^{\{S\}}_{sleep}, \forall S \in \{\text{ON, OFF}\}$ have the property $\min\{\theta^{\{\text{ON}\}}_{sleep}, \theta^{\{\text{OFF}\}}_{sleep}\} \le 0$.

Proof: Assume $\forall S \in \{\text{ON, OFF}\}, \theta^{\{S\}}_{sleep} > 0$. When the buffer size B is infinite, obviously all the states with $Q < \min\{\theta^{\{\text{ON}\}}_{sleep}, \theta^{\{\text{OFF}\}}_{sleep}\}$ are transient states. In this case, by reducing all the *active* and *sleep* thresholds by 1, the total system cost could be reduced by the delay penalty of one job. That is, replacing (S, Q, W) with $(S, Q-1, W)$, the

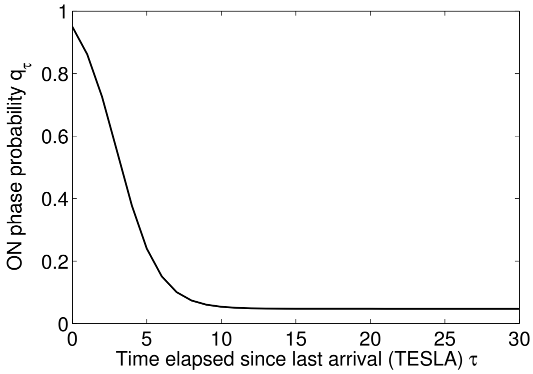


Fig. 2. The ON arrival phase probability evolution in sequence $\{q_\tau\}$.

holding cost of one job could be saved, which contradicts the optimality of this policy. Therefore, $\forall S \in \{\text{ON, OFF}\}$, $\min\{\theta_{sleep}^{\{\text{ON}\}}, \theta_{sleep}^{\{\text{OFF}\}}\} \leq 0$ holds. ■

Theorem 2 indicates that with infinite buffer size, the optimal policy for at least one arrival phase is either non-sleeping policy (*sleep* threshold is 0^-, once the server turns into *active*, it never turns to *sleep*), or an N-policy mentioned in [10], under which the server will go to sleep when the queue is empty (the *sleep* threshold is 0) and turn into *active* mode when N jobs have been accumulated in the queue.

***Remark** 2:* In the belief-MDP formulation, TESLA τ is brought up to approximate the arrival phase at each time slot. With $\tau = 0$, the arrival phase has to be ON, while with sufficiently large τ, the arrival phase is more likely to be OFF. Therefore, the optimal policy for each possible τ should be in between the optimal policies for arrival phases ON and OFF. In addition, the insights of the two-threshold policy given above still hold for the belief-MDP problem. Thus, based on Conjecture 1, the optimal policy for the unobservable-arrival-phase case should also be a two-threshold based policy, which can be described as below.

The optimal policy for the belief-MDP $\{(\tau_n, Q_n, W_n), n \geq 0\}$ is a TESLA-based two-threshold policy. For each $\tau \in \{0, 1, 2, ...\}$, there exist two thresholds $0^- \leq \theta_{sleep}^{\{\tau\}} \leq \theta_{active}^{\{\tau\}}$ such that for $W \in \Phi$,

$$A_{\pi^*}(\tau, Q, W) = active, \text{for } Q \geq \theta_{active}^{\{\tau\}}, \tag{27}$$

$$A_{\pi^*}(\tau, Q, W) = sleep, \text{for } Q \leq \theta_{sleep}^{\{\tau\}}, \tag{28}$$

$$A_{\pi^*}(\tau, Q, W) = W, \text{for } \theta_{sleep}^{\{\tau\}} < Q < \theta_{active}^{\{\tau\}}. \tag{29}$$

The *active* and *sleep* thresholds vary with τ.

V. Numerical and Simulation Results

In this section, we first give numerical results of the optimal policy based on the given belief-MDP formulation, under different system parameters. System performance based on simulations are then analyzed. We compare the performance of the optimal policy with the queue-based policy in the prior work [19] and the (S, Q, W) formulation with known arrival phases explained in Section IV-B.

Value iteration algorithm for MDP is applied to get our numerical results. The optimal policy and the value function are updated every iteration, until the value function converges. The details of the algorithm has been given in [19]. To apply the algorithm, we have to restrict the maximum recorded TESLA as M and set a finite buffer size to get a finite state space. To show an intact optimal policy, the buffer size B is set to be 80 to make sure the maximum *active* threshold in our simulation would not exceed the buffer size. The mean durations of the ON phase and the OFF phase, α^{-1} and β^{-1}, are set to be 20 time slots and 40 time slots, respectively. The average interval of job arrivals λ^{-1} is 2 time slots. The average job size is l = 1MB and the steady data rate x is 143Mbps. The length of one time slot Δ is assumed to be 0.1s. As a result, the mean service time μ^{-1} is 1.2 time slots. Therefore, the traffic load of the system is assumed to be $\lambda'/\mu = 0.2$.

Based on [19], the energy consumption for an active server in a time slot E_{active} is 25J. According to [16], the switching cost only incurs when the server wakes up (setup power), $E_{\text{sw}}^{\text{a}\to\text{s}}$ is set as 0, while $E_{\text{sw}}^{\text{s}\to\text{a}}$ is set to be 200J based on [19]. The weight between energy and delay penalty ω is 2 and the discount factor γ is 0.99, which is chosen to be close to 1, similar to the discount factor taken in [19]. The larger the discount factor is, the more future system cost would be taken into account. The value of the discount factor does not effect the structure of the optimal policy.

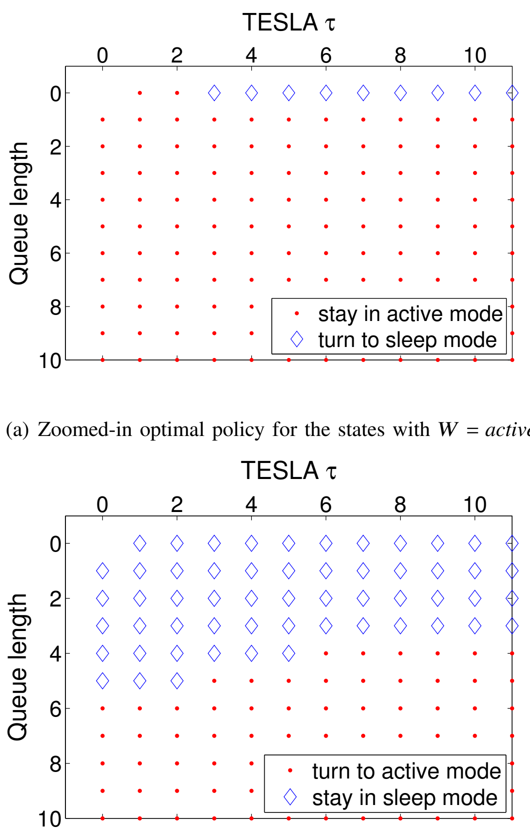
As for the maximum recorded TESLA M, the evolution of the ON phase probability $\{q_\tau\}$ calculated from the parameters above is given in Fig.2. The curve decreases sharply at first and converges to 0.05 when k is larger than 10. Therefore, M is set as 30 to guarantee the convergence of $\{q_\tau\}$.

All the numerical and simulation results are divided into three parts. In the first subsection, the numerical results of the optimal policy under the proposed POMDP formulation are given. After that, how the optimal policies change with system parameters and the comparison between the optimal policies under other formulations are discussed. At last, we focus on analyzing the influence of the sleeping mechanism on the system performance, including the total system cost, energy consumption per bit and average delay, etc.

A. Optimal Policy Structure

Fig.3 shows the optimal policy of belief-MDP formulation under default parameters. To get a clearer vision of the structure, we only give the zoomed-in version of the optimal policy in Fig.3, which shows how the decision changes with TESLA τ and Q in details. The decisions for larger τ and Q that are not included in the figure follow the same decisions with the same Q with largest τ, and the same τ with largest Q, respectively. It is shown that when the current server mode is *active*, the server would choose to sleep only if $Q = 0$ and $\tau \geq 4$, and if the server is in *sleep* mode at the current time slot, the server would wake up only if $Q \geq 6$ and $\tau \leq 2$, or $Q \geq 5$ and $3 \leq \tau \leq 5$, or $Q \geq 4$ and $\tau \geq 6$. Note that these decision thresholds depend on the parameters of the arrival process.

It is obvious that the optimal policy displayed in Fig.3 follows the TESLA-based two-threshold structure mentioned



(a) Zoomed-in optimal policy for the states with W = *active*.

(b) Zoomed-in optimal policy for the states with W = *sleep*.

Fig. 3. Zoomed-in optimal policy for (τ, Q, W) belief-MDP formulation.

in Remark 2. In this case, the *sleep* thresholds for different τ can be expressed as:

$$\begin{cases} \theta_{sleep}^{\{\tau\}} = 0^-, & \tau \leq 2, \\ \theta_{sleep}^{\{\tau\}} = 0, & \tau \geq 3. \end{cases} \tag{30}$$

Note that states with $Q = 0, \tau = 0$ do not exist, the *sleep* threshold cannot be determined clearly from the optimal policy. It is only known that $\theta_{sleep}^{\{\tau\}} \leq 0$. The *active* thresholds are given as follows:

$$\begin{cases} \theta_{active}^{\{\tau\}} = 6, & \tau \leq 2, \\ \theta_{active}^{\{\tau\}} = 5, & 3 \leq \tau \leq 5, \\ \theta_{active}^{\{\tau\}} = 4, & \tau \geq 6. \end{cases} \tag{31}$$

Notice that both the *active* and *sleep* thresholds are monotone with τ, because the probability of being in ON phase is monotone decreasing with τ.

From Fig.3, it is found that the optimal policy can be explained as a **wait-and-see** policy regardless of the current operation mode. Even an arrival happens at some queue length (for $Q > 0$, $\tau = 0$; for $Q = 0$, $\tau = 1$ instead), the server prefers waiting for several time slots to see if there is any arrival. If there is no arrival, the server would finally choose to switch its operation mode. In short, the server would hold the operation mode for a certain period before it switches to the other mode. For instance, in Fig.3(a), when the queue is empty and an arrival just took place one time slot ago (state $(\tau = 1, Q = 0, active)$), instead of turning to *sleep* mode at once, the server would hold the *active* mode for two more time slots. If there is still no arrival in these two time slots, the server would finally choose to sleep at state $(\tau = 3, Q = 0, active)$. In this way, the server avoids to switch too frequently, so that switching cost can be saved. In [16], this period is called *close-down* time, which is a random variable. However, the length of the waiting period in our scheme is a constant corresponding to a certain Q.

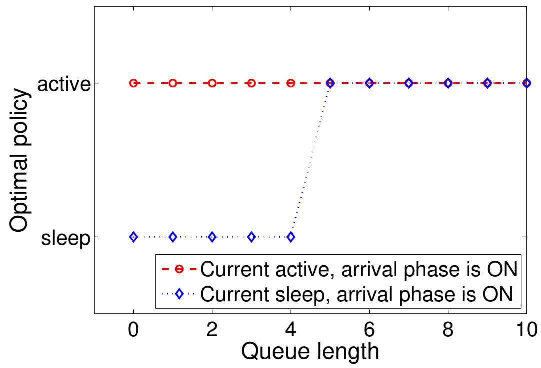
Similarly, in Fig.3(b), at state $(\tau = 0, Q = 5, sleep)$, even there is a job arrival at the current time slot, the server would not turn to *active* mode immediately, but wait for another 3 time slots. If there is still no arrival in this period, it finally choose to be active at $(\tau = 3, Q = 5, sleep)$. At state $(\tau = 0, Q = 5, sleep)$, the server would hesitate either to aggregate more jobs to serve together (in order to save switching cost), or to turn *active* (in order to avoid large delay penalty since the OFF period might begin soon). With 3 more time slots with no arrivals, the server cannot afford to wait for another job arrival anymore, thus turn to *active* at $(\tau = 3, Q = 5, sleep)$ at last.

With massive numerical tests with different system parameters, it is confirmed that this property always holds. Firstly, it is found that the *sleep* thresholds for $\tau = M$ are always equal to or larger than those for $\tau = 0$, i.e., the *sleep* thresholds increase monotonely with τ. This is because there is no traffic arrival in the OFF phase, and the server is always more willing to sleep and save energy. When $\theta_{sleep}^{\{0\}} \neq \theta_{sleep}^{\{M\}}$, there would be a non-zero waiting period for $\theta_{sleep}^{\{0\}} < Q \leq \theta_{sleep}^{\{M\}}$.

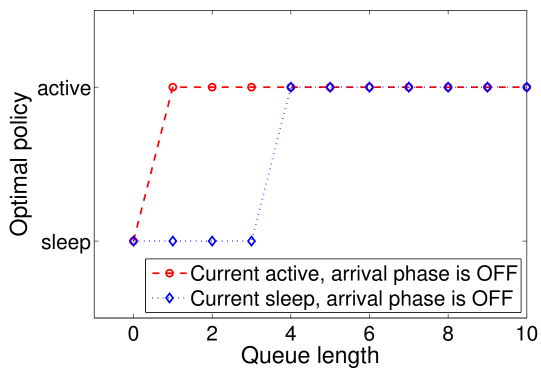
When the *active* thresholds decrease monotonely with τ, there would be a non-zero waiting period for $\theta_{active}^{\{M\}} \leq Q < \theta_{active}^{\{0\}}$, which is the case of Fig.3(b). Furthermore, even if the *active* thresholds increase monotonely with τ, we claim that the policy still has the wait-and-see property with zero waiting period for all possible Q. In this situation, there might be states with certain $Q = x$ and $W = sleep$, where the optimal decisions switches from *active* to *sleep* mode with increasing τ. It is found that all the states $(\tau > 0, x, sleep)$ are transient states. Therefore, when the queue length is x, the server would not hesitate and turn to *active* mode, which means the waiting period is zero. The explanation of transient states is given in Appendix A.

Further, we would like to compare the optimal policy under the belief-MDP (τ, Q, W) state and the optimal policy proposed in [19], which gives the best mapping from tuples of the form (Q, W) to actions. This (Q, W) formulation is sufficient for Bernoulli traffic because of its memoryless property.

In the discrete time (Q, W) formulation, we consider the arrival process as memoryless Bernoulli arrivals with the average interarrival time $(\lambda')^{-1}$, which means this system state has no information of traffic burstiness, but only the average arrival rate. The results show that in (Q, W) formulation, the server would turn to *sleep* from *active* mode only if $Q = 0$ and switch to *active* mode from *sleep* mode only if $Q \geq 6$, under the same system parameters introduced at the beginning of this section. The optimal policy is a two-threshold policy with



(a) Zoomed-in optimal policy for the states with arrival phase S = ON.



(b) Zoomed-in optimal policy for the states with arrival phase S = OFF.

Fig. 4. Zoomed-in optimal policy for (S, Q, W) MDP formulation with observable arrival phases.

active threshold 6 and *sleep* threshold 0. Comparing this policy and the optimal policy of the belief-MDP formulation shown in Fig.3, when $\tau \leq 2$, the server would not sleep because of traffic burstiness. With TESLA, the server knows within 2 time slots, there would be traffic arrivals with a large chance. Thus, staying at *active* state is a better choice to save switching cost. However, if the server is currently asleep, in the belief-MDP formulation, the server would turn to *active* mode earlier than the (Q, W) formulation, e.g., when τ is larger than 2 and $Q = 5$. This is because when $t \geq 2$, the arrival state might be OFF with larger probability. If the server still stays asleep, the system might wait for a longer time until the next job arrives and the server turns to *active* with $Q = 5$. Thus, the delay penalty will be quite large and have a bad impact on the total cost. As a result, in this belief-MDP formulation, the server switches to *active* earlier from sleeping than the (Q, W) formulation.

To further give some insights of the threshold values listed in Eqs.(30-31), Fig.4 shows the optimal policy for the (S, Q, W) formulation where the arrival phase is observable, with the same system parameters as in Fig.3. Fig.4(a) displays the optimal policy for the states with S = ON. Further, the dashed line with circles gives the corresponding actions for the states with $W = active$. Based on Conjecture 1, it is obvious that the *sleep* threshold $\theta^{\{ON\}}_{sleep} = 0^-$. Meanwhile, the dotted line with diamonds in Fig.4(a) shows how the policy varies with Q when S = ON and $W = sleep$, from which we can tell $\theta^{\{ON\}}_{active} = 5$. Similarly, it can be observed that $\theta^{\{OFF\}}_{sleep} = 0$ and $\theta^{\{OFF\}}_{active} = 4$ from Fig.4(b).

Compared the optimal policies for (S, Q, W) and (Q, W) formulation, all of the thresholds for (S, Q, W) case are correspondingly equal to or smaller than the thresholds in (Q, W) formulation. It means that with sufficient burstiness information, the server prefers staying *active* without sleeping when the arrival phase is ON, and turning to *active* mode from sleeping earlier in both arrival phases.

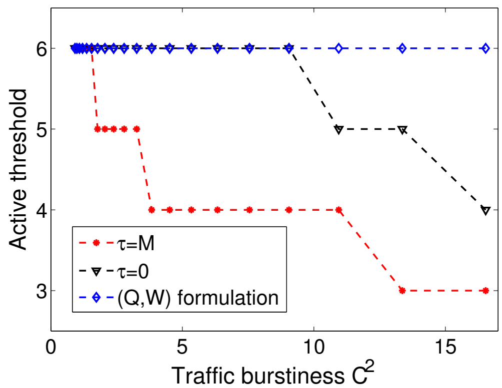
Furthermore, focusing on the *active* and *sleep* thresholds of Fig.3 and Fig.4, the *sleep* thresholds of belief-MDP formulation transits from $\theta^{\{ON\}}_{sleep} = 0^-$ to $\theta^{\{OFF\}}_{sleep} = 0$, with increasing TESLA τ. However, the values of *active* thresholds for the belief-MDP formulation given in Eq.(31) could be larger, nut not exactly the same with the *active* thresholds in (S, Q, W) formulation. With smaller τ, $\theta^{\{\tau\}}_{active}$ is closer to the *active* threshold of the ON phase, while with larger τ, $\theta^{\{\tau\}}_{active}$ is closer to the *active* threshold of the OFF phase.

In summary, with information of traffic burstiness in the system state, the server prefers staying *active* for a long period before sleeping to avoid frequent switches, and turning to *active* mode from sleeping earlier to avoid large delay penalty. The (S, Q, W) formulation has sufficient burstiness information, while the (Q, W) formulation has no burstiness information. The belief-MDP formulation with TESLA only has partial information of traffic burstiness. Therefore, the values of the *active* and *sleep* thresholds for the belief-MDP sit in between the *active* and *sleep* thresholds of (S, Q, W) and (Q, W) formulations, respectively.

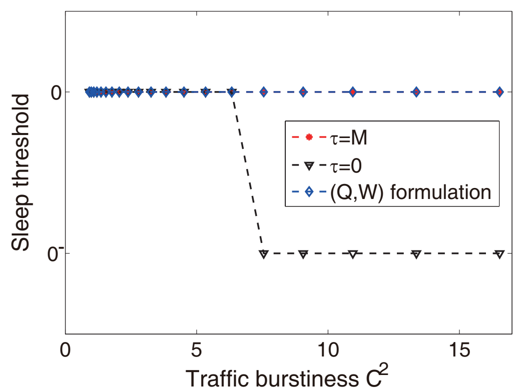
B. *Influence of System Parameters on Thresholds*

Through numerical results, we try to find out how the *active* and *sleep* thresholds for the belief-MDP problem change with system parameters. Since the thresholds are monotone with TESLA τ. The thresholds of the boundary TESLA's $\tau = 0$ and $\tau = M$ are analyzed. The influence of traffic burstiness, switching cost and weight between energy and delay are studied.

1) Traffic Burstiness: Fig.5 displays how the *active* and *sleep* thresholds change with the burstiness of traffic. The thresholds for the (Q, W) formulation are also given in the figure as a reference. By adjusting arrival rate at the ON phase λ and expected duration for the OFF phase β^{-1} at the same time, the traffic burstiness can be adjusted. It is shown that when C^2 is close to 0, the *active* and *sleep* thresholds for the belief-MDP are the same as the reference thresholds, since the traffic arrivals are closer to memoryless Bernoulli arrivals. Yet, as the burstiness goes up, both the *active* and *sleep* thresholds for both $\tau = 0$ and $\tau = M$ cases would decrease. When C^2 is sufficiently large, the *sleep* threshold when TESLA $\tau = 0$ becomes 0^-, which means the server would not turn into *sleep* mode even if the queue is empty. The server is more willing to begin serving tasks at a lower queue length, with larger burstiness.

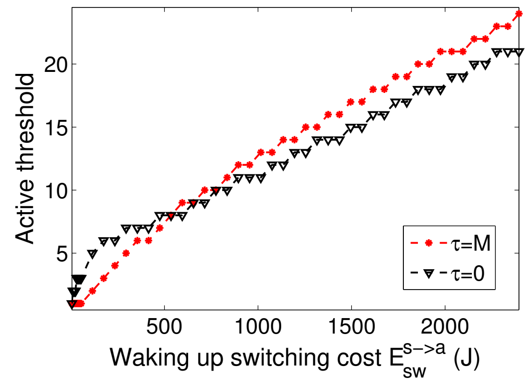


(a) The *active* thresholds of the optimal policies.

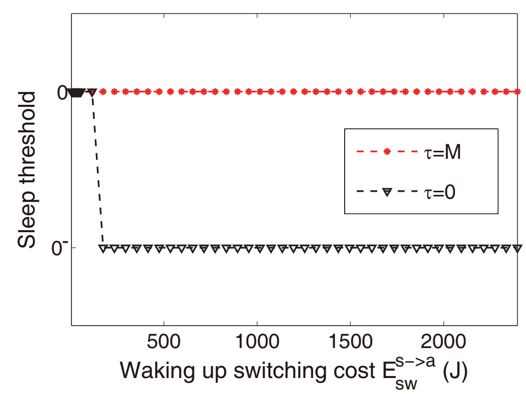


(b) The *sleep* thresholds of the optimal policies.

Fig. 5. The thresholds of optimal policies changing with traffic burstiness in (S,Q,W) and (Q,W) formulations.



(a) The *active* thresholds of the optimal policies.



(b) The *sleep* thresholds of the optimal policies.

Fig. 6. The thresholds of optimal policies changing with switching cost $E_{\text{sw}}^{\text{s}\to\text{a}}$ in (S,Q,W) formulation.

2) Switching Cost: By numerical methods, the thresholds of the optimal policies with different "waking up" switching cost $E_{\text{sw}}^{\text{s}\to\text{a}}$ are shown in Fig.6. From Fig.6(a), the *active* thresholds become higher with a larger switching cost, because the server would like to aggregate more jobs and serve together, avoiding frequent switches between *active* and *sleep*.

In Fig.6(b), we find that *sleep* thresholds become 0^- for $\tau = 0$ with $\theta_{sleep}^{\{M\}} = 0$ if $E_{\text{sw}}^{\text{s}\to\text{a}}$ is sufficiently large. In this case, even if the queue is empty, the server would hold the *active* mode for several time slots before going to sleep, i.e. the waiting period is nonzero. That is because with increasing switching cost, the server starts to become more "hesitating" about switching to *sleep* mode when the queue is empty.

Further, from Fig.6(a), with small switching cost (under around 500J), the *active* thresholds decrease with τ, which indicates under these parameters, the system follows wait-and-see property with $W = active$. In contrast, with larger switching cost, the *active* thresholds increase with τ. Based on the explanation of transient states with $W = sleep$ in Appendix A, these cases do not violate the wait-and-see property.

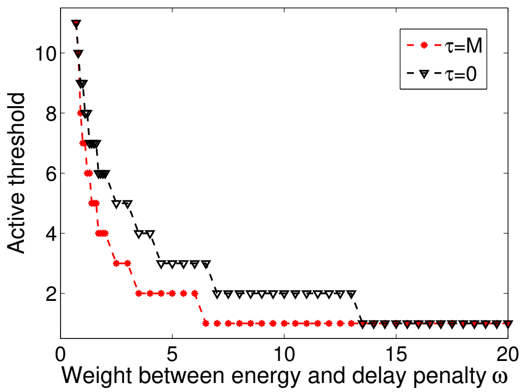
In addition, when the switching cost is sufficiently large (e.g. 10000J), the *sleep* thresholds for $\tau = 0$ and $\tau = M$ are both 0^-. Once the server start to serve tasks, it would never choose to *sleep*.

3) Weight of Delay Penalty: If adjusting the weight between energy consumption and delay penalty, ω, the thresholds are given in Fig.7. Note that the lower ω is, the more delay-tolerant the system would be. Both the *active* and *sleep* thresholds become lower with larger ω, as the delay penalty is large, and the system would want to serve the jobs in the queue as quickly as possible. Especially, the *active* thresholds in Fig.7(a) would decrease severely when ω is very small.

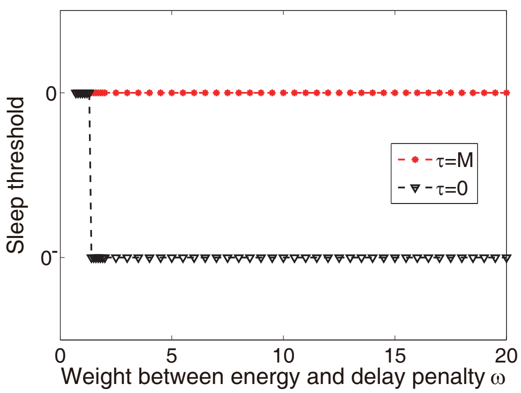
To summarize, a larger switching cost $E_{\text{sw}}^{\text{s}\to\text{a}}$ leads to higher *active* thresholds, while increasing the traffic burstiness C^2, the weight between energy and delay ω (i.e. decreasing delay tolerance) causes the *active* and *sleep* thresholds to decrease.

C. Influence of System Parameters on System Performance

Except for the optimal policies, we also compare the system performance of the optimal policies under (τ,Q,W), (S,Q,W) and (Q,W) formulations. A sequence of random arrivals following IBP are generated. These three optimal policies under different scenarios are applied at each time slot to decide whether the server would be in the *active* or *sleep* mode in the next time slot. To further explain the scenarios involved, the optimal policy of (Q,W) formulation demonstrates how the server works assuming the traffic arrivals are memoryless. In this case, the system has no information of burstiness at all. However, (S,Q,W) formulation demonstrates the system



(a) The *active* thresholds of the optimal policies.



(b) The *sleep* thresholds of the optimal policies.

Fig. 7. The thresholds of optimal policies changing with the weight between energy and delay, ω, in (S, Q, W) formulation.

has sufficient bursiness information, i.e., the arrival phase is observable. The (τ, Q, W) formulation represents the scenario that the system only has partial information on burstiness. The simulation runs for 200000 time slots. To eliminate the impact of traffic randomness, the simulation is run for at least 15 times to get the average of system performance results. The total cost, energy consumption per bit and average delay are evaluated.

1) Traffic Burstiness: In Fig.8, we give the comparison of the accumulated cost among optimal policies under the following three formulations, the (τ, Q, W) belief-MDP formulation, (S, Q, W) formulation with sufficient information on burstiness and (Q, W) formulation with no information on burstiness. On each curve, the total cost varies with the SCV of the interarrival time C^2, while the average interarrival time $(\lambda')^{-1}$ remains 6 time slots. Obviously, the total system cost decreases with traffic burstiness, which means combining traffic burstiness with sleeping mechanism would bring performance enhancement. Note that even if the system has no information on arrival phases, i.e. in (Q, W) formulation, the total cost still decreases with traffic burstiness. The curve for (τ, Q, W) formulation always lies between (S, Q, W) formulation and (Q, W) formulation. Since (S, Q, W) formulation has sufficient information of traffic burstiness, it actually provides the lower bound of the system cost performance for the belief-MDP. Similarly, the

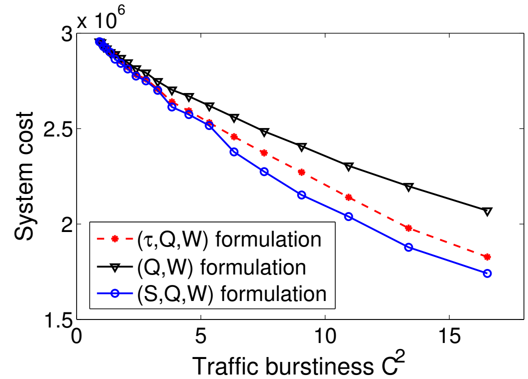


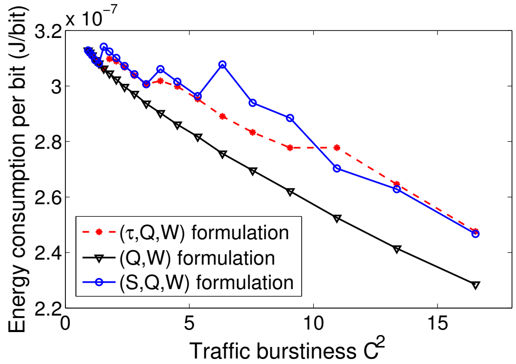
Fig. 8. The comparison of total system cost under different traffic burstiness, among three optimal policies, which are based on (τ, Q, W) POMDP formulation, known-arrival-phase (S, Q, W) formulation and (Q, W) formulation without bursty information (from [19]).

(Q, W) state considers no information of burstiness, it provides the upper bound of the system cost performance for the belief-MDP problem.

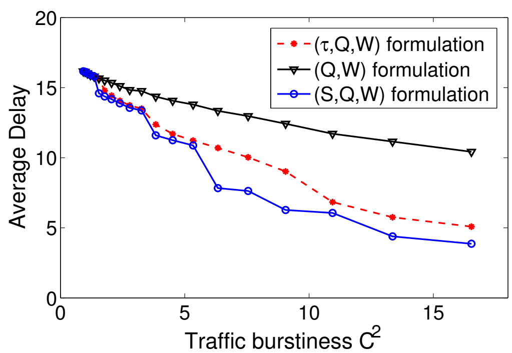
Figs.9(a) and 9(b) display the total energy consumption per bit and average delay for each job under three formulations same as Fig.8. For the (Q, W) formulation, with larger burstiness, the average delay and energy consumed per bit decrease. However, if having information about the arrival phases, the energy and average delay are not simply monotone decreasing with the traffic burstiness C^2 anymore. Compared to the *active* and *sleep* thresholds shown in Fig.5, once the thresholds change, there would be a jump in the curves of energy and average delay. Once the thresholds become lower with larger C^2, the system would consume more energy to exchange for lower average delay and lower system cost eventually, which shows the tradeoff between delay and energy. The reason is that with larger burstiness, the *active* and *sleep* thresholds decrease, so that the server switches more frequently and stays at *active* mode for longer time (corresponding to the energy performance) and the server starts to serve at a lower queue length (corresponding to the lower average delay).

Focusing on the belief-MDP formulation, different from system cost, the energy and delay performance curves do not lay between the other two formulations all the time. Comparing to the case that the server never goes to sleep, if the thresholds do not change, the larger the traffic burstiness is, the more energy would be saved. At the largest burstiness in the figure (around $C^2 = 15$), with the optimal sleeping policy given by (τ, Q, W) belief-MDP, the system would save up to 70% energy.

2) Switching Cost: Fig.10 gives the delay and energy performance varying with waking up switching cost $E_{\rm sw}^{\rm s\to a}$ in three different formulations. With increasing switching cost, the delay performance gets worse because the *active* thresholds get higher and more jobs wait in the queue until the server switches to *active* mode. Because of the higher switching cost, the energy consumption per bit also goes up. That is, increasing switching cost incurs both energy and delay performance deterioration.

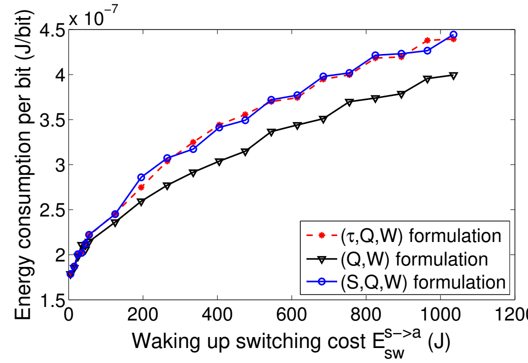


(a) Energy consumption per bit.

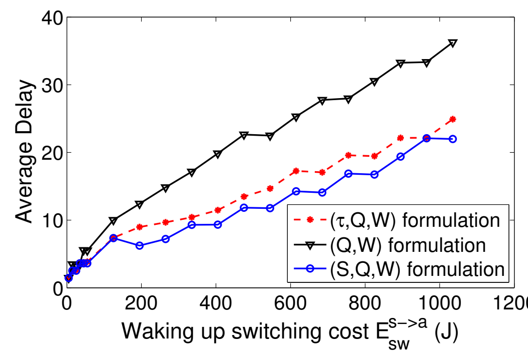


(b) Average delay.

Fig. 9. Energy consumption per bit and delay performance changing with burstiness under three optimal policies, which are based on (τ, Q, W) POMDP formulation, (S, Q, W) formulation and (Q, W) formulation.



(a) Energy consumption per bit.



(b) Average delay.

Fig. 10. The comparison of energy consumption per bit and average delay with waking up switching cost $E_{sw}^{s\to a}$, among three optimal policies, which are based on (τ, Q, W) POMDP formulation, (S, Q, W) formulation and (Q, W) formulation.

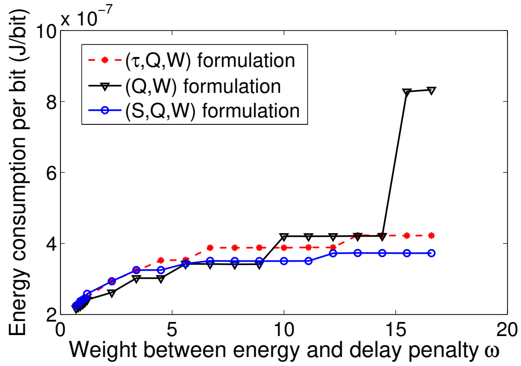
3) Weight of Delay Penalty: The influence of the weight ω is studied in Fig.11. With default traffic burstiness, the system energy consumption per bit and average delay under different ω are evaluated in Fig.11(a) and Fig.11(b), respectively. With larger ω, the delay penalty plays a more important role in the system cost, the average delay for each job decreases in Fig.11(b). Thus, in exchange for the delay performance improvement, the energy consumption per bit goes up as shown in Fig.11(a). Comparing with Fig.7, it can be found that both the energy and delay performance stay steady if the *active* and *sleep* thresholds do not change. With smaller thresholds, the energy consumption per bit is higher, while the average delay becomes lower. The simulation results for the belief-MDP formulation basically stay in between the other two curves.

Delay performance for all these three formulations shown in Fig.11(b) converges to the same level with sufficiently large ω, after that, further increasing ω would not bring delay performance gain anymore. The thresholds will not change either. It indicates there exists a best average delay for each group of system parameters. In this situation, whenever there is any job in the queue, the server is guaranteed in the *active* mode, so that all the jobs can be served as soon as possible. E.g., for the (S, Q, W) formulation shown in Fig.7, when ω is very large, the server must be in *active* mode when there is any job in the queue, since both the *active* thresholds are 1. Only in the OFF arrival phase with no job arrivals, the server would turn to *sleep* mode if the queue becomes empty. Therefore, the average delay in this case should be exactly the same with a single server system without sleeping mechanisms. However, even in this case, there will be energy saving because the queue could be empty and the server could go to sleep. The differences between the energy performance of the three formulations are caused by different sleeping mechanisms. Here in the belief-MDP formulation, compared to the server with no sleeping mechanism, there is still 49% energy saving with the same average delay.

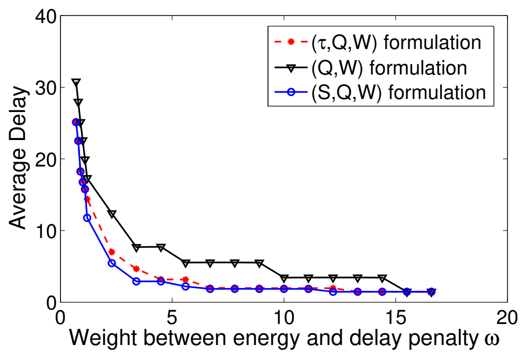
To conclude, from all the performance analysis given in this section, it is found that (Q, W) formulation provides the upper bound of system cost to the (τ, Q, W) belief-MDP formulation, while (S, Q, W) formulation gives the lower bound. However, the relationship of these three formulations in the perspectives of energy consumption per bit and average delay is not that simple, since there exists tradeoff between energy and delay. Furthermore, increasing switching cost causes both energy and delay performance deterioration.

VI. Conclusions

In this paper, we consider a single server queue with bursty traffic,which is modeled as IBP arrivals, and formulate the operation decision problem into a partially observable Markov decision process (POMDP), where the arrival phases of IBP are partially observed by the Time Elapsed Since the Last Arrival (TESLA). The optimal policy is proven to be hysteretic



(a) Energy consumption per bit.



(b) Average delay.

Fig. 11. The comparison of energy consumption per bit and average delay with different weight ω, among three optimal policies, which are based on (τ, Q, W) POMDP formulation, (S, Q, W) formulation and (Q, W) formulation.

and conjectured as a TESLA-based two-threshold policy. The *active* and *sleep* thresholds of queue length would change with TESLA.

By numerical results, we find the optimal policy is a wait-and-see policy. The server would wait for several time slots to see if there are extra arrivals before switching. The *active* thresholds increase with larger switching cost, while all the thresholds decrease with higher traffic burstiness, or lower delay tolerance.

System performance including total cost, energy consumption per bit and average delay are analyzed. The system with no bursty information provides the upper bound of system cost, while the system with observable arrival phases provides the lower bound. The total cost decreases with traffic burstiness, which indicates applying sleeping mechanisms to a server with bursty traffic would bring performance enhancement.

As for the next step of this work, we would like to consider other bursty traffic models, such as MMBP (Markov Modulated Bernoulli Process) to extend our results. Moreover, considering multiple servers with overlapping traffic load is another important direction which would reveal some insights in traffic offloading.

Appendix A
Explanation of Transient States

If the optimal policy of the belief-MDP is a TESLA-based two-threshold policy and the *active* thresholds change monotonely with TESLA, states $(\tau > y, Q, sleep)$, where

$$y = \arg\min_{\tau'}\{\tilde{A}_{\tilde{\pi}^*}(\tau', Q, sleep) = active\},$$

are transient states.

Intuitively, this says for a certain Q, if the server is currently asleep and would not turn to *active* mode until some $\tau = y$, the states with $\tau > y$ are transient states.

Let's take state $(\tau = 6, Q = 4, sleep)$ in Fig.3(b) as an example. From Eq.11, this state can only be obtained from $(\tau = 5, Q = 4, sleep)$. Based on Theorem 1, once the decision for $(\tau = 5, Q = 4, sleep)$ is *active*, the decision for $(\tau = 5, Q = 4, active)$ must be the same, which means $(\tau = 6, Q = 4, sleep)$ is a transient state. Similarly, the states $(\tau > 6, Q = 4, sleep)$ are also transient states.

Acknowledgment

This work is sponsored in part by the Nature Science Foundation of China (No. 91638204, No. 61571265, No. 61621091), and Hitachi R&D Headquarter.

References

[1] B. Leng, B. Krishnamachari, X. Guo and Z. Niu, "Optimal operation of a green server with bursty traffic," *IEEE GLOBECOM*, Washington D.C., USA, Dec. 2016.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong and J. C. Zhang, "What will 5G be?" *IEEE J. Select. Areas Commun.*, vol. 32, no. 6, pp. 1065-1082, Jun. 2014.

[3] B. Rengarajan, G. Rizzo, M. A. Marsan, "Bounds on QoS-constrained energy savings in cellular access networks with sleep modes," *International Teletraffic Congress (ITC)*, San Francisco, USA, Sept. 2011.

[4] E. Oh, B. Krishnamachari, X. Liu and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 56-61, Jun. 2011.

[5] K. Son, H. Kim, Y. Yi, and B. Krishnamachari, "Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks," *IEEE J. Select. Areas Commun.*, vol. 29, no. 8, Sept. 2011.

[6] J. Wu, S. Zhou, and Z. Niu, "Traffic-aware base station sleeping control and power matching for energy delay tradeoffs in green cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 8, pp. 4196-4209, Aug. 2013.

[7] Z. Niu, X. Guo, S. Zhou and P. R. Kumar, "Characterizing energy-delay tradeoff in hyper-cellular networks with base station sleeping control," *IEEE J. Select. Areas Commun.*, vol. 33, no. 4, pp. 641-650, Apr. 2015.

[8] Z. Niu, "TANGO: traffic-aware network planning and green operation", *IEEE Wireless Commun.*, vol. 18, pp. 25-29, 2011.

[9] Z. Niu, S. Zhou, S. Zhou, X. Zhong, and J. Wang, "Energy efficiency and resource optimized hyper-cellular mobile communication system architecture and its technical challenges," *Sci. China Inf. Sci.*, vol. 42, no. 10, pp. 1191-1202, 2012.

[10] J. Wu, Y. Bao, G. Miao, S. Zhou and Z. Niu, "Base station sleeping control and power matching for energy-delay tradeoffs with bursty traffic," *IEEE Trans. Vehicular Tech.*, vol. 65, no. 5, pp. 3657-3675, May 2016.

[11] C. R. Baugh, J. Huang, R. Schwartz, and D. Trinkwon, "Traffic model for 802.16 TG3 MAC/PHY simulations," IEEE 802.16 Broadband Wireless Access Working Group, Technical Report, Mar. 2001.

[12] S. Andreev, A. Anisimov, Y. Koucheryavy, and A. Turlikov, "Practical traffic generation model for wireless networks," *the 4th ERCIM eMobility Workshop*, Lulea, Sweden, May 2010.

[13] S. Ahmad, M. Liu, T. Javidi, Q. Zhao and B. Krishnamachari, "Optimality of myopic sensing in multichannel opportunistic access," *IEEE Trans. on Information Theory*, vol. 55, no. 9, pp. 4040-4050, Sept. 2009.

[14] Q. Zhao, L. Tong, A. Swami and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE J. Select. Areas Commun.*, vol. 25, no. 3, pp. 589-600, Apr. 2007.

[15] W. Jiang, J. Tang and B. Krishnamachari, "Optimal power allocation policy over two identical Gilbert-Elliott channels," *IEEE International Conference on Communications (ICC)*, Budapest, Hungary, Jun. 2013.

[16] X. Guo, Z. Niu, S. Zhou and P. R. Kumar, "Delay-constrained energy-optimal base station sleeping control," *IEEE J. on Select. Areas Commun.*, vol. 34, no. 5, pp. 1073-1085, May 2016.

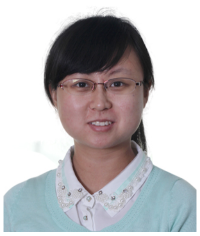
[17] D. Park, H.G. Perros, "Approximate analysis of discrete-time tandem queueing networks with customer loss," *IEEE GLOBECOM*, 1992.

[18] S. K. Hipp and U. D. Holzbaur, "Decision processes with monotone hysteretic policies", in *Operations Research*, vol. 36, no. 4, pp. 585-588, 1988.

[19] I. Kamitsos, L. Andrew, H. Kim and M. Chiang, "Optimal sleep patterns for serving delay-tolerant jobs", *the 1st International Conference on Energy-Efficient Computing and Networking*, Apr. 2010.

[20] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons Inc., 2005.

Bingjie Leng received her Bachelor degree in electronic engineering from Tsinghua University, Beijing, China, in 2014. Currently, she is a Master student in engineering at Tsinghua University, Beijing, China. Her research interests include green wireless communications, infrastructure sharing, and mobile big data analysis.



Xueying Guo received her Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2017. She is currently a postdoctoral researcher in the Computer Science Department at the University of California, Davis, CA, USA. From October 2013 to October 2014, she was a Visiting Scholar at Computer Engineering and System Group, Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA. Her research interests include wireless networking, edge cloud system, and reinforcement learning.

Xi Zheng received her B.S. degree in Physics from Tsinghua University, Beijing, China, in 2014. Currently, she is a Ph.D. student in Electronic Engineering at Tsinghua University, Beijing, China. Her research interests include resource management in wireless networks and green wireless communications.

Bhaskar Krishnamachari received his B.E. in Electrical Engineering at The Cooper Union, New York, in 1998, and his M.S. and Ph.D. degrees from Cornell University in 1999 and 2002 respectively. He is a Professor in the Department of Electrical Engineering at the University of Southern California's Viterbi School of Engineering. His primary research interest is in the design, theoretical analysis and experimental evaluation of algorithms and protocols for next-generation wireless networks including low power wireless sensor networks. He has co-authored more than 200 publications on this topic, including best paper awards at Mobicom (2010), IPSN (2004, 2010), and MSWiM (2006), that have been collectively cited more than 20000 times per Google Scholar.

Zhisheng Niu received the degree from Beijing Jiaotong University, China, in 1985, and the M.E. and D.E. degrees from the Toyohashi University of Technology, Japan, in 1989 and 1992, respectively. From 1992 to 1994, he was with Fujitsu Laboratories Ltd., Japan. In 1994, he joined with Tsinghua University, Beijing, China, where he is currently a Professor with the Department of Electronic Engineering. His major research interests include queueing theory, traffic engineering, mobile Internet, radio resource management of wireless networks, and green communication and networks.

He was an active volunteer for various academic societies, including the Director of Conference Publications from 2010 to 2011 and the Director of Asia-Pacific Board of the IEEE Communication Society from 2008 to 2009, a Membership Development Coordinator of the IEEE Region 10 from 2009 to 2010, the Councilor of the IEICE Japan from 2009 to 2011, and a Council Member of the Chinese Institute of Electronics from 2006 to 2011. He was a Distinguished Lecturer from 2012 to 2015, the Chair of the Emerging Technology Committee of the IEEE Communication Society from 2014 to 2015, a Distinguished Lecturer of the IEEE Vehicular Technologies Society from 2014 to 2016, a member of the Fellow Nomination Committee of the IEICE Communication Society from 2013 to 2014, a standing committee member of the Chinese Institute of Communications from 2012 to 2016, and an Associate Editor-in-chief of the IEEE/CIC joint publication China Communications.

Dr. Niu received the Outstanding Young Researcher Award from the Natural Science Foundation of China in 2009 and the Best Paper Award from the IEEE Communication Society Asia-Pacific Board in 2013. He was a co-recipient of the best paper awards from the 13th, 15th, and 19th Asia-Pacific Conference on Communication in 2007, 2009, and 2013, respectively, the International Conference on Wireless Communications and Signal Processing, and the Best Student Paper Award from the 25th International Teletraffic Congress. He is a fellow of the IEICE.