Hindawi
Journal of Sensors
Volume 2018, Article ID 1419843, 14 pages
https://doi.org/10.1155/2018/1419843

Research Article

BP Network Control for Resource Allocation and QoS Ensurance in UAV Cloud

Ang Gao,[1] **Yansu Hu**,[2] **Lixin Li**,[1] **and Xu Li**[1]

[1]*School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China*
[2]*School of Electronics and Control, Chang'an University, Xi'an 710064, China*

Correspondence should be addressed to Ang Gao; gaoang@nwpu.edu.cn

Received 3 November 2017; Accepted 24 January 2018; Published 10 April 2018

Academic Editor: Raymond Swartz

Unmanned aerial vehicle (UAV) cloud can greatly enhance the intelligence of unmanned systems by dynamically unloading the compute-intensive applications to cloud. For the uncertain nature of UAV missions and fast-changing environment, different UAV applications may have different quality of service (QoS) requirements. This paper proposes a mixed QoS ensurance and energy-balanced (MQEB) architecture for UAV cloud from a view of control theory, which can support both hard and soft QoS ensurance with the consideration of energy saving. The hard and soft QoS requirements are decoupled by being normalized into a two-level cascaded feedback loop. The former is time slot loop (TS-Loop) to enforce the absolute QoS ensurance for real-time applications, and the latter is contention window loop (CW-Loop) to enforce the plastic QoS ensurance for non-real-time applications. Finally, the back propagating (BP) neuron network is used for parameters' self-tuning and controller design. The hardware experiments demonstrate the feasibility of MQEB. In heavy load, MQEB has greater throughput and better energy efficiency, and in light load, MQBE has lower total power consumption.

1. Introduction

In the networked unmanned aerial vehicle (UAV) system, UAVs, also known as drones, share the information and cooperate with each other by a decentralized wireless network, which enhances the performance and mission effectiveness. Nevertheless, UAVs' intelligence is still limited by the aerial platform carriage. Cloud robotics, as the combination of cloud and robotics [1, 2], provides a novel conception that breaks the limitations of high complexity of UAV cluster and low intelligence of individual UAV. By means of paralleled computing, storage, and communication, the compute-intensive applications on UAVs can be unloaded to cloud dynamically. This operation mode brings out the new conception of UAV cloud [3].

As shown in Figure 1, compute-intensive applications like SLAM (simultaneous localization and mapping) are divided into a series of task flows and then are dispatched to the cloud hosts for fast processing. In the process of task flow transmission, as an outgoing transmission passes down through the stack, each layer repacks the actual data with its own header information and then passes down to the lower layer. The unloaded tasks are finally encapsulated into MAC frames at the datalink. In wireless ad hoc network (WANET), UAVs share the same link and contend for communication channel in carrier sense multiple access/collision avoidance (CSMA/CA) mode.

Generally, UAVs differ in processing capacity, mission environment, and task emergency, and thus the corresponding transmission flows may be the mixture of real-time and non-real-time traffics. They have different quality of service (QoS) requirements in terms of bandwidth, delay, losing rate, and so on. For real-time traffic, QoS metrics must be within the "hard" of "inelastic" QoS constraints, while for non-real-time traffic, a preferred QoS metric can be some flexible, that is, "soft" or "elastic" QoS [4].

Initially, real-time flows should be treated preferentially. It is better to arrange more resources to them at the datalink and network. Unfortunately, as UAVs' movement intensifying the wireless link's uncertainty and traffic's

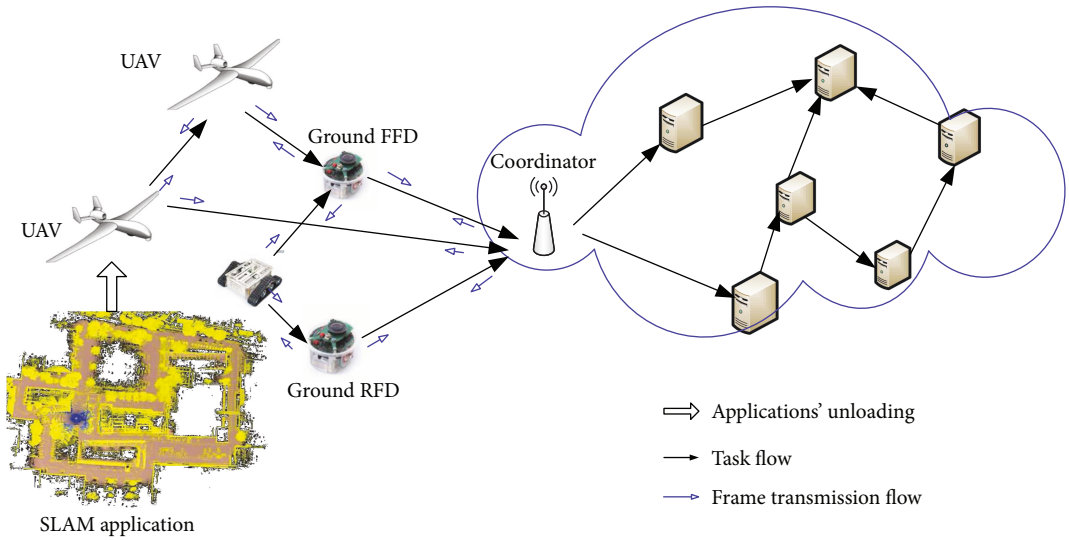


Figure 1: The schematic of UAV cloud.

unpredictability, it is very hard to arrange precise resource that just meet the QoS constraints. The overprovision for one class of traffic will deteriorate the QoS of other traffics because the total channel resource is limited, and the short provision for real-time traffic may induce the failure of the UAV application.

To support native massive machine-to-machine communications in high-density wireless networks [5, 6], contention resolution protocols, for example, contention resolution diversity slotted ALOHA (CRDSA) and its variants [7], have contributed to a drastic throughput improvement comparing with CSMA/CA [8]. The principle is each packet contains the header information about the location of the replicas within the frame, which can be used for successive interference cancellation (SIC) and retransmission avoidance. Nevertheless, QoS support is still out of consideration in contention resolution approaches.

To the best of our knowledge, for both contention-based and contention-free MAC protocols, there are improvements of QoS support with multiqueueing architecture. The methods of QS-MAC [9], PRIN-MAC [10], AS-MAC [11], RF-MAC [12], CACC [13] and DARS [14] are typical contention-based protocols with priority queues. QoS is controlled by assigning different queues with different resources or different CSMA/CA parameters, such as contention window size [9], interframe space [10], active time [11], transmission power [12], TCP congestion window [13] and data rate [14].

Contention-free approach, for example, frequency division multiple access (FDMA), that assigns channel to only one node at any given frequency can also provide high throughput. However, it is a centralized approach that needs base stations for frequency synchronization and dedicated control channel to exchange control messages, which makes FDMA is unfeasible for aerial drones and adds a high communication overhead [15]. Some researches have also proposed the multichannel MAC in vehicular network. Time slots on subchannel can be arranged according to the priority of transmission [16, 17]. Other resources like data rate and power allocation can be adaptive with both time-varying channel and user's quality requirement [18].

Actually, these approaches have the common feature of QoS metric feedback. But most of them focus on the architecture realization and performance evaluation without theoretically analyzing the system stability. Feedback-based Diff-MAC (FD-MAC) [19] has solved the converged problem by a less-step controller to provide proportional delay-differentiated (PDD) QoS support. But it has the following problems. First, in a self-autonomous unmanned system, duty cycling is always implemented for energy saving. So energy efficiency must be considered when QoS support is provided. Second, FD-MAC only controls the relative QoS, which is not suitable for hard QoS constraints of real-time applications. Third, the control model should not be a linear system due to the load burst from the upper layer and the network topology. It is inconvincible to use the linear model for system identification.

For resource allocation in cloud and wireless communication, most researches commonly formulate the issue into optimization problems with the subjects of resource limitations, for example, energy consumption [18, 20, 21], bandwidth [22], financial cost [23], processing time [24], utility function [25], secrecy outage probability [26], and caching [27] The assumption of these approaches is that the resource requirements of different clients or traffic should be explicit, while the summation of total resource is beyond the limitation. In the circumstance of UAV cloud, these approaches are not suitable anymore because the model is different. It exists a random process in the model that breaks the interconnection between preferred QoS metrics and resource quantization. Transmission traffic always changes with the UAV mission and environment. It is impossible to predict the mount of resource to meet the QoS requirements.

This paper, from a renewed view of control theory and back propagating (BP) neural network, develops an all-in-one mixed QoS ensurance and energy-balanced (MQEB) architecture that can support the common mixture of soft and hard QoS traffic. For the energy-saving feature of

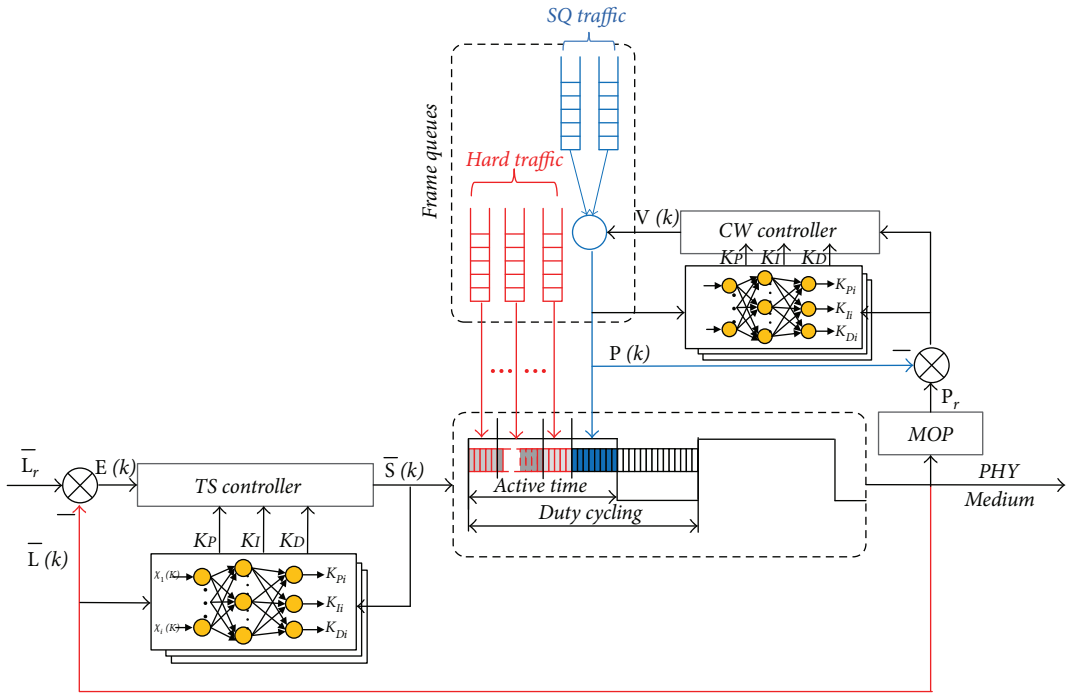


Figure 2: Dynamic time slot and accessing provability assignment for mixed QoS ensurance.

self-autonomous system, the energy consumption is also under consideration. The hard and soft QoS constraints are decoupled by normalized into a two-level cascaded feedback loop. The former is time slot adaptive loop (TS-Loop) to enforce the hard QoS ensurance for real-time application, and the latter is contention window loop (CW-Loop) to enforce the soft QoS ensurance for non-real-time traffic. The hardware experiments demonstrate the feasibility of the MQEB architecture. Comparing with FD-MAC, MQEB-MAC has a new feature of absolute QoS ensurance and further develops following two advantages. In the condition of heavy load, MQEB has a great throughput and a better energy efficiency; and in light load, MQEB has lower total power consumption.

The rest of the paper is organized as follows. Section 2 presents the initial ideas of MQEB. The hard QoS ensurance with time slot isolation and BP self-tuning control is formulated in Section 3, and soft QoS ensurance with accessing probability optimization is described in Section 4. In Section 5, the hardware experiments show the performance evaluations of MQEB, including dynamic performance, energy efficiency and power consumption, static performance, and robust verification.

2. Overview of Mixed QoS Ensurance and Energy-Balanced (MQEB) Architecture

Figure 2 shows the architecture of MQEB. The basic idea is dynamic time slot isolation for hard QoS (HQ) and accessing probability optimization for soft QoS (SQ). Both hard and soft QoS can be ensured in a manner of a two-cascaded feedback control architecture. TS-Loop ensures the HQ with the dynamic time slot isolation, and CW-Loop balances the QoS satisfaction of SQ and energy saving by multiobjective optimization.

The symbol definitions are summarized in Table 1 and the detailed concepts in MQEB are as follows.

2.1. Dynamic Time Slot Isolation. In IEEE 802.11 distributed coordination function (DCF), the node which has buffered MAC frames starts carrier sense with a random back-off in the range of $[0, W_{\min}]$. Once the collision is detected, the node performs another longer random back-off in the range of $W_\tau = \min(2^\tau W_{\min}, W_{\max})$. This is known as binary exponential back-off (BEB). The more intensive that node attempts to transmit frames, the higher probability collision and retransmission occur, which will not only cause the unnecessary energy wasting but also induce a higher frame delay. As shown in Figure 3, there are S time slots in total in a duty cycling. For hard QoS traffic, different types of traffic contend for channel medium in the class-specified time slots $\bar{s}_i$, which is actually a kind of resource reservation mechanism avoiding the across-class contention.

2.2. Energy Balancing. For the inherent feature of energy saving in self-autonomous unmanned system, the nodes always work periodically and go dormant to enlarge the lifetime. Therefore, the time slots tend to be used for dormancy to save energy. In this paper, MQEB trades off the energy-saving and soft QoS satisfaction by multiobjective optimization (MOP) to get the optimized accessing probability as well as the number of slots $\tilde{s}$ for soft QoS traffic.

Table 1: Nomenclature.

I	Types of HQ in the network
J	Types of SQ in the network
M	Number of WSN nodes in the network
S	Total number of time slots in a duty cycling
$\bar{s}_i$	Number of time slots assigned for ith HQ
$\tilde{s}$	Number of time slots assigned for all SQ
$\bar{L}_i$	Preferred average delay of HQ
$\tilde{L}_j$	Preferred average delay of SQ
ℓ_i	Actual average delay of traffic i
λ_m	PHY transmission rate of node m
$n_{j,m}$	Queueing length of traffic j on node m
$\rho_{j,m}$	Accessing probability of the medium contention
v_j	Scale factor for the initial up boundary of back-off time
$\mathfrak{E}_{\text{len}}$	Mathematical expectation of frame length
$W_{\min}, W_{\max}$	Minimum and maximum of initial up boundary of CW
τ	Time that medium is consecutively sensed to be busy

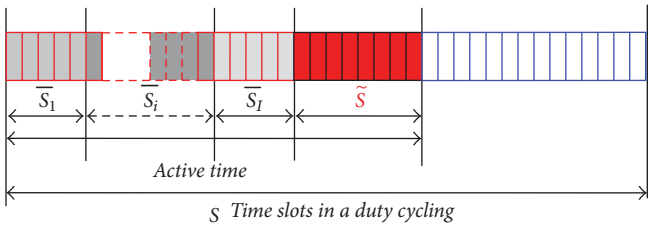


Figure 3: Dynamic time slot mechanism.

2.3. Accessing Probability Optimization. The remaining time slots $(S - \sum \bar{s}_i)$ can be arranged to SQ. They contend for channel medium in the same time duration with $\tilde{s}$ slots. Their QoS constraints are elastic and different with the traffic types, so the accessing probability of them should be precisely selected to balance the energy and preferred QoS. Therefore, the arrangement of the accessing probability is a network utility maximization (NUM) model, which takes the utility function to quantify the "satisfaction" of traffic's QoS [22, 25, 28].

3. Time Slot Isolation Adaption for HQ Ensurance

Supposing there are $N = I + J$ types of traffic, that is, I types of HQ and J types of SQ (best effort can be treated as a special SQ), and there are M nodes in the network. The constraints for HQ can be expressed as follows:

$$\ell_i \le \bar{L}_i, \quad i = 1, \dots, I. \tag{1}$$

The average QoS metrics, such as average node-to-node delay, throughput, and frame loss rate, can be denoted by $\ell_i(k)$, and $\bar{L}_i$ denotes the corresponding QoS constraints. In general, we mainly take the node-to-node delay as the QoS metric, so the smaller value of ℓ, the better QoS for delay.

3.1. Feedback for HQ Time Slot Assignment. HQ should be priority ensured. The paper proposed the mechanism of time slot isolation for HQ. HQ frames of the same class are transmitted in specified time slots. Let $\bar{s}_i(k)$, $i = 1, \dots, I$, denote the number of slots reserved for the ith HQ. In the duration of these slots, all the nodes only transmit frames of the ith HQ and the transmissions of other HQ are blocked. As shown in Figure 2, time slot isolation and dynamic adaption actually make up a feedback control loop (TS-Loop for short). The vector $\bar{\mathbf{S}}(k) = [\bar{s}_1(k), \bar{s}_2(k), \dots, \bar{s}_I(k)]$ acts as the output of the controller, which can be adjusted according to the deviation of the actual traffic delay $\bar{\mathbf{L}}(k)$ to the preferred QoS metrics $\bar{\mathbf{L}}_r$, that is,

$$\begin{aligned} \mathbf{E}(k) &= \bar{\mathbf{L}}_r - \bar{\mathbf{L}}(k), \\ \bar{\mathbf{L}}(k) &= [\ell_1(k), \ell_2(k), \dots, \ell_1(k)], \\ \bar{\mathbf{L}}_r &= \left[\bar{L}_i, \dots, \bar{L}_I\right]. \end{aligned} \tag{2}$$

TS controller operates by responding to the deviation $\mathbf{E}(k)$, that is, adjusting the set of time slots $\bar{\mathbf{S}}(k)$ distributed to every type of HQ. Thus, the delay $\ell_i = \bar{L}_i$ is sustained. Despite the uncertainty of medium accessing, the inherent self-stabilization of feedback mechanism liberates us from calculating the time slot assignment for every traffic precisely. For the coherent nonlinearity approximating features, the paper develops a BP neural network-based self-tuning PID controller for TS-Loop.

3.2. BP-Based Self-Adaptive Control. This paper takes a 3-layered BP neural network for PID parameters' self-tuning for two reasons. First, theoretically, a 3-layered neural network is able to learn any function [29], and the computation amount would increase exponentially as more hidden layers are added; Second, F-test is used to prove that under 5% significance level, the 3-layered and more than 3-layered neural networks have no significant difference.

There are $(4, q, 3)$ neurons in the input layer, the hidden layer and the output layer, respectively. PID parameters of every type are trained independently. So there are actually an I-set of neural networks as shown in Figure 2.

The classic PID controller of (3) can be rewritten as (4)

$$\begin{aligned} \bar{s}_i(k) = \bar{s}_i(k-1) &+ K_{P_i}(e_i(k) - e_i(k-1)) + K_{I_i} e_i(k) \\ &+ K_{D_i}(e_i(k) - 2e_i(k-1) + e_i(k-2)), \end{aligned} \tag{3}$$

$$\bar{s}_i(k) = h[\bar{s}_i(k-1), K_{Pi}, K_{Ii}, K_{Di}, e_i(k), e_i(k-1), e_i(k-2)], \tag{4}$$

where $h(\cdot)$ is a nonlinear function related to all the variables.

Take the ith HQ traffic for example. The learning process is composed of forward propagation and back propagation. The details are as follows [29]:

(1) *Forward propagation: compute the network output.*

The output of input layer is as follows:

$$O_j^{(1)} = x_j(l), \quad j = 1, \dots, 4. \tag{5}$$

$$\begin{aligned} x_1 &= e_i(k) - e_i(k-1), \\ x_2 &= e_i(k), \\ x_3 &= e_i(k) - 2e_i(k-1) + e_i(k-2), \\ x_4 &= \bar{s}_i(k-1). \end{aligned} \tag{6}$$

The input and output of hidden layer are as follows:

$$\begin{aligned} \mathrm{net}_l^{(2)}(k) &= \sum_{j=1}^{4} \omega_{l,j}^{(2)} O_j^{(1)}, \\ O_l^{(2)}(k) &= f\left(\mathrm{net}_l^{(2)}(k)\right), \\ & \qquad l = 1, 2, \dots, q, \end{aligned} \tag{7}$$

where $\omega_{l,j}^{(2)}$ is the weight factor from the jth neuron of the input layer to the lth neuron of the hidden layer. The superscript (1), (2), and (3) represent the input layer, hidden layer, and output layer. $f(\cdot)$ is the activating function, $f(\cdot) = \tanh(\cdot)$.

The input and output of output layer are as follows:

$$\begin{aligned} \mathrm{net}_m^{(3)}(k) &= \sum_{l=1}^{q} \omega_{m,l}^{(3)} O_l^{(2)}, \\ O_m^{(3)}(k) &= g\left(\mathrm{net}_m^{(3)}(k)\right), \\ & \qquad m = 1, 2, 3, \end{aligned} \tag{8}$$

$$\begin{aligned} O_1^{(3)}(k) &= K_{Pi}, \\ O_2^{(3)}(k) &= K_{Ii}, \\ O_1^{(3)}(k) &= K_{Di}; \end{aligned} \tag{9}$$

where $g(\cdot) = 1/2[1 - \tanh(\cdot)]$.

(2) *Backward propagation: adapt the weight factors.*

Suppose the performance index function is

$$J = \frac{1}{2}\left[\bar{L}_i - \ell_i(k+1)\right] = \frac{1}{2}e^2(k+1). \tag{10}$$

According to gradient descent method, the learning process of weight is as follows:

$$\Delta\omega_{m,l}^{(3)}(k+1) = -\eta\frac{\partial J}{\partial \omega_{m,l}^{(3)}} + \gamma\Delta\omega_{m,l}^{(3)}, \tag{11}$$

where $\eta > 0$ is the learning rate and $0 < \gamma < 1$ is the inertia factor to accelerate the convergence speed. Since there are

$$\begin{aligned} \frac{\partial \bar{s}_i(k)}{\partial Q_1^{(3)}(k)} &= e_i(k) - e_i(k-1), \\ \frac{\partial \bar{s}_i(k)}{\partial Q_2^{(3)}(k)} &= e_i(k), \\ \frac{\partial \bar{s}_i(k)}{\partial Q_2^{(3)}(k)} &= e_i(k) - 2e_i(k-1) + e_i(k-2). \end{aligned} \tag{12}$$

The weight of output layer is

$$\begin{aligned} \Delta\omega_{m,l}^{(3)}(k+1) &= \eta\delta_m^{(3)} O_l^{(2)}(k) + \gamma\Delta\omega_{m,l}^{(3)}(k), \\ \delta_m^{(3)} &= e(k+1)\mathrm{sgn}\left(\frac{\partial \ell_i(k+1)}{\partial \bar{s}_i(k)}\right)\frac{\bar{s}_i(k)}{O_m^3(k)}\, g\left[net_m^{(3)}(k)\right]. \end{aligned} \tag{13}$$

The weight of hidden layer is

$$\begin{aligned} \Delta\omega_{l,j}^{(2)}(k+1) &= \eta\delta_l^{(2)} O_j^{(1)}(k) + \gamma\Delta\omega_{l,j}^{(2)}(k), \\ \delta_l^{(2)} &= f\left[\mathrm{net}_l^{(2)}(k)\right]\sum_{m=1}^{3}\delta_m^{(3)}\omega_{m,l}^{(3)}(k). \end{aligned} \tag{14}$$

The BP neural network calculates the self-tuning parameters K_{Pi}, K_{Ii}, K_{Di} based on pretrained weights, which is simultaneous trained by back propagating the derivation to weights. The detailed algorithm of BP for self-tuning PID control is shown in Algorithm 1. All parameters in the BP will be settled down, and we can feed the neural network with raw data and use its output for further work. Therefore, the TS controller is able to handle a system with dynamic traffic arriving rate by adjusting the time slots $\bar{s}_i(k)$.

4. Accessing Probability Adaption for SQ Ensurance

The remained time slots can be assigned to SQ. On the one hand, constraints for SQ are noncompulsory; for example, "the actual delay is better to be smaller than $\tilde{L}$," which implies that it is better to deploy more time slots for a smaller delay. On the other hand, in self-autonomous unmanned system, the nodes always work periodically to save energy. So more dormancy time is also expected. To solve this contradiction, the nodes should balance the energy efficiency and the QoS requirements.

4.1. Utility Function. The quantized satisfaction with given value of QoS metrics can be described by the utility function $U_j(\cdot)$, a nonincreasing function with respect to the give metrics $\ell_j, j = 1, \dots, J$.

$\tilde{L}_j$ denotes the preferred average delay of each type of SQ. The value of utility functions is normalized in the range of [0,1]. The marginal utility can be expressed as $u(\ell) = U'(\ell)$. It is not straightforward to obtain the exact expression of the function for traffic. However, the characteristic of the utility function can be summarized as follows [28]:

(1) $k : k = 0$, initialize weight factor for each layer $\omega_{l,j}^{(2)}, \omega_{m,l}^{(3)}$; learning rate η and inertial factor γ.
(2) Sample the actual ith HQ delay $\ell_i(k), \bar{s}_i(k)$ and get $e_i(k) = \bar{L}_i - \ell_i(k)$.
(3) Normalize $\bar{L}_i(k), \ell_i(k), \bar{s}_i(k-1)$ and construct the training set for BP as (6).
(4) According to (5), (7), and (8), calculate the forward propagation of BP.
(5) The output of output layer is the adapted values of K_{Pi}, K_{Ii}, K_{Di}.
(6) Calculate the PID controller output $\bar{s}_i(k)$ as (4), which will be used for control loop.
(7) Calculate the performance index $J(k)$ as (10).
(8) If $J(k) \le J_{\text{threshold}}$ or $k > K_{\max}$, stop the iteration; else, continue the loop.
(9) According to (13), refresh the weight factor $\omega_{m,l}^{(3)}$ for the output layer.
(10) According to (14), refresh the weight factor $\omega_{l,j}^{(2)}$ for the hidden layer.
(11) $k : k = k + 1$, go to step 2.

Algorithm 1: Algorithm of BP for self-tuning PID control.

$$\begin{cases} u(\ell) < 0, u'(\ell) \ge 0, \ell \in \left(0, \tilde{L}\right] \\ u(\ell) < 0, u'(\ell) \ge 0, \ell \in \left(\tilde{L}, \infty\right) \\ u(0) \approx 1, u(\infty) \approx 0 \end{cases} \tag{15}$$

Without a loss of generality, the paper assumes the unified utility functions as sets of universal antisigmoid functions [28], which have different characteristics with different parameters.

$$U_j(\ell) = \frac{1}{1 + e^{C_j\left(\ell_j - \tilde{L}_j\right)}}. \tag{16}$$

The parameter $\tilde{L}_j$ is the inflexion of $U_j(\ell)$. When the average delay is smaller than $\tilde{L}_j$, the utility function is concave, which implies that the traffic expects a shorter delay strongly. While the average delay is larger than $\tilde{L}_j$, the utility function is convex, which implies that the traffic expects a shorter delay not so strong.

The parameter C_j is used to adjust the slope of the utility curve around $\tilde{L}_j$, which reflects the sensitiveness to the preferred delay. The larger C_j, the steeper the slope of the utility curve around $\tilde{L}_j$, so that the traffic requires a lower delay ℓ more strongly, and verse versa. Figure 4 shows the utility functions changing with C_j. As best effort (BE) is a special SQ with preferred delay $\tilde{L} = 0$ ($\tilde{L}_j = 10$ ms for normal SQ), SQ and BE can be uniformly analyzed and modeled in the following.

4.2. Accessing Portability and Equivalent Transmission Rate. It should be noticed that during the remaining slots, all kinds of SQ frames contend for channel in the same time span of $\tilde{s}$ slots, where $\tilde{s} \le S - \sum \bar{s}_i$, $i = 1, \ldots, I$. Let $\rho_{j,m}$ denote the accessing probability of the jth SQ on the node m, the *equivalent time slot* assigned to class j is $\rho_{j,m} \cdot \tilde{s}$, and

$$\forall m, \sum_{j=1}^{J} \rho_{j,m} = 1. \tag{17}$$

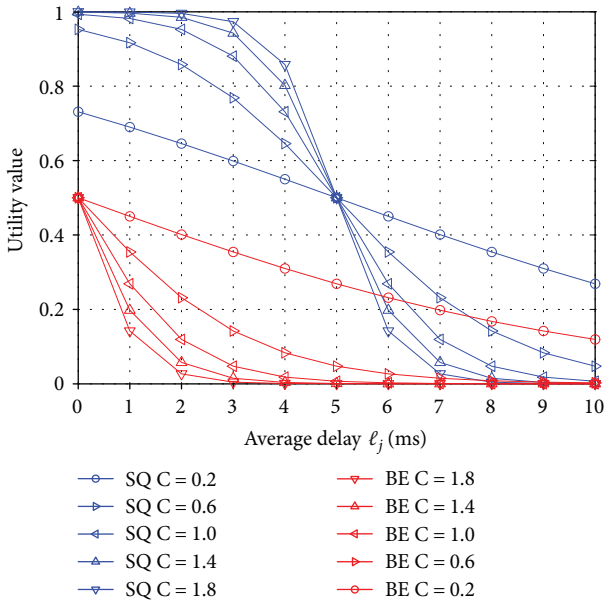


Figure 4: Relationship between utility function and parameter.

Generally, the average delay can hardly be formalized into a closed-form expression because it is a multivariate random process related to the transition power, channel fading, intensity of traffic load, and the BEB parameters. For every node, supposing PHY transmission ratio is λ_m, $m = 1, \ldots, M$, the *equivalent transmission ratio* for traffic j is

$$\lambda_m \frac{\rho_{j,m} \cdot \tilde{s}}{S}, \tag{18}$$

and the average delay for class j in node m (including queueing delay and transmission delay) is

$$\ell_{j,m} = \frac{n_{j,m} \cdot E_{\text{length}}}{\lambda_m \left(\rho_{j,m} \cdot \tilde{s}/S\right)} B_{j,m} = \frac{n_{j,m} \cdot E_{\text{length}} \cdot S}{\rho_{j,m} \cdot \tilde{s} \cdot \lambda_m} B_{j,m}. \tag{19}$$

E_{length} is the average frame length, and $n_{j,m}$ is the queueing length of traffic class j on the node m, which can be

sampled by the node itself. $B_{j,m}$ is a slow changing parameter related to the practical network condition. Q factor is defined as follows:

$$Q_{j,m} = \frac{n_{j,m} E_{\text{length}} S B_{j,m}}{\lambda_j}, \tag{20}$$

and submit (19) into (16), the utility function $U(\ell)$ can be rewriten as follows:

$$U_{j,m}\left(\rho_j, \tilde{s}\right) = \frac{1}{1 + e^{C_j\left(\left(Q_j/\rho_j \tilde{s}\right) - \bar{L}_j\right)}}, \tag{21}$$

$U_{j,m}(\rho_j, \tilde{s}) > 0$ holds for all ρ_j. Because MQEB is actually a topology-transparency approach, and every UAV node can operate independently, so in the following discussion, the node number m is omitted.

4.3. Multiobjective Optimization. Accessing probability assignment can be modeled as an optimal problem. The optimization objective is to maximize the summation of utility functions of all SQ on the node, while minimize the time slots $\tilde{s}$ assigned for it. The multiobjective optimization problem (MOP) is

$$F(\mathfrak{R}) = \begin{cases} \max \sum U_{j=1}^{J}\left(\rho_j, \tilde{s}\right), \\ \min \tilde{s}, \end{cases} \tag{22}$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{j=1}^{j} \rho_{j,m}, \\ & 0 \le \rho_{j,m} \le 1, \\ & 0 < \tilde{s} \le S - \sum_{j=1}^{J} \bar{s}_j, \\ & j = 1, \ldots, J. \end{aligned} \tag{23}$$

It can be equivalent to a single-object optimization problem (SOP) by the multiply-divide method as follows:

$$\min H(\mathfrak{R}) = \frac{\tilde{s}}{\sum U_{j=1}^{J}\left(\rho_j, \tilde{s}\right)}, \tag{24}$$

and particle swarm optimization (PSO) is used to solve the SOP. Supposing D is the swarm volume, $\mathfrak{R}_d \in R^{J+1}$. Let $\mathfrak{R}_d = (\rho_j, \ldots, \rho_J, \tilde{s})$, $d = 1, 2, \ldots, D$, be the location of particle d. $v_d \in R^{J+1}$ is the corresponding speed. The symbol $\circ$ is Hadamard product operator. The detailed algorithm of solving SOP by PSO is shown in Algorithm 2.

The output of PSO is $\mathfrak{R} = (P_r, \tilde{s})$, where $P_r = [\rho_1, \rho_2, \ldots, \rho_J]$ is a feasible solution of MOP in (22) and indicates the accessing probability ρ_j for every SQ and the time slots $\tilde{s}$ for them. However, the channel contention is a random process that can hardly be controlled directly. In order to provide an exact accessing probability control for every SQ, a linear-differ binary exponential back-off (LD-BEB) scheme is introduced. The accessing probability is controlled by adjusting the initial up boundary of back-off time in CW-Loop dynamically.

The initial back-off time is randomly chosen in the range of $[0, v_j(k) W_{\min} - 1]$, $1 \le v \le W_{\min}/W_{\max}$. 1]. Once the initial back-off time is set, the frame starts contention for medium accessing. If a collision is detected, the back-off time is reset in the range of $[0, W_\tau - 1]$.

$$[0, W_\tau - 1], W_\tau = \min(v_i(k) \cdot 2^\tau W_{\min}, W_{\max}), \tag{25}$$

where $v_j(k)$ is a scale factor that regulates the initial up boundary of back-off time for the jth SQ. Generally, a smaller v_j will have a high accessing probability ρ_j, because it tends to try to access medium more impatiently and vice versa. As shown in Figure 2, the vector $\mathbf{V}(k) = [v_1(k), v_2(k), \ldots, v_J(k)]$ acts as the output of CW controller, which can be adapted by operating to the deviation of P_r to the actually accessing probability $P(k)$. The controller is also a BP-based self-tuning PID controller that is similar to that in Section 3.2.

5. Experiment and Results

5.1. Hardware Configuration. UAVs take ZigBit™ 900 as the communication module, which is a 784/868/915 MHz IEEE 802.15.4 OEM product with A*tmel*$^{\text{R}}$ AVR2025 software package. The module contains an ATmega1281V microcontroller and an AT86RF212 RF transceiver in it. A*tmel*$^{\text{R}}$ AVR2025 is a developer's kit with fundamental MAC API for hardware operation. Thanks to these API callback functions, once a frame is transmitted, a callback function will generate a software interruption. Therefore, the node-to-node delay can be measured by sender node.

20 UAV nodes with ZigBit 900 are uniformly distributed in the radius of 100 meters. Every node randomly sends data packets to the other 19 nodes. At MAC, packets are encapsulated into MAC frames, and frame traffic is generated by continuously sending packets. The interval of frames obeys the normal distribution with the average of $-\bar{T}_t/\log(1 - G)$. $G(0 < G < 1)$ is the offered traffic, which is normalized by transmission data rate; that is, $G = T_t/R$. $\bar{T}_t$ (bit) is the average MAC frame length and R (bps) is the data rate. The frame length follows Pareto distribution with the shape parameter of 1.1 and average of $105 * 8$ bit.

Simultaneous with traffic generation, every node runs MQEB independently. The detailed algorithm of MQBE is shown in Algorithm 3. Ad hoc on-demand distance vector (AODV) routing protocol is realized at the application layer, and a specific thread processes the routing maintenance and routing discovery. In the experiments, the transmitted power is set to be 1 mW, $W_{\min} = 2^3$, $W_{\max} = 2^8$, and $\tau_{\max} = 3$.

5.2. Dynamic Performance. The dynamic performance mainly concerns the QoS metrics changing with time. Supposing that there are 4 types of frame traffic, two are HQ (HQ 1 and HQ 2) carrying real-time applications, and the other two are SQ (SQ 1 and SQ 2) carrying non-real-time applications.

In the consideration of typical working condition of UAV cloud, the paper develops two groups of comparison

(1) $k : k = 0$, initialize the location vector $\mathfrak{R}_d^{(0)}$ for d swarms and their speed $v_d^{(0)}$, $p_d^{(0)} \doteq \mathfrak{R}_d^{(0)}$, $d = 1, 2, \ldots, D$. Let $g^{(0)} = \arg \min_{\mathfrak{R} \in \{\mathfrak{R}_1^{(0)}, \ldots, \mathfrak{R}(0)_d\}} H(\mathfrak{R})$.

(2) For $d = 1, 2, \ldots, D$, generate two random vectors $\mathbf{r}_d{}^{(k)}$, $\mathbf{t}_d^{(k)} \in R^{J+1}$ in the range of [0,1] with uniform distribution.
$v_d^{(k+1)} = \omega v_d^{(k)} + c_1 \mathbf{r}_d^{(k)} \circ (p_d^{(k)} - \mathfrak{R}_d^{(k)}) + c_2 \mathbf{t}_d^{(k)} \circ (g_d^{(k)} - \mathfrak{R}_d^{(k)})$,
$\mathfrak{R}_d^{(k+1)} = \mathfrak{R}_d^{(k)} + v_d^{(k+1)}$.

(3) For $d = 1, 2, \ldots, D$, if $H(\mathfrak{R}_d^{(k+1)}) < H(p_d^{(k)})$, $p_d^{(k+1)} = \mathfrak{R}_d{}^{(k+1)}$; else, $p_d^{(k+1)} = p_d{}^{(k)}$.

(4) If $\exists d$ make $H(\mathfrak{R}_d^{(k+1)}) < H(g_d^{(k)})$, let $d^* = \arg \min_d H(\mathfrak{R}_d^{(k+1)})$, $g_d^{(k+1)} = \mathfrak{R}_{d^*}{}^{(k+1)}$; else, $g_d^{(k+1)} = g_d{}^{(k)}$.

(5) If $H(g_d^{(k+1)}) \le H_{\text{threshold}}$ or $k > K_{\max}$, stop the iteration; else, continue the loop.

(6) $k : k = k + 1$, go to step 2.

Algorithm 2: Algorithm of solving SOP by PSO.

(1) $k : k = 0$, initialize $\bar{L}_i$, $\tilde{L}_j$, $(i = 1, 2, \ldots, I)$, $(j = 1, 2, \ldots, J)$.
(2) Start TS-LOOP.
(3) For $i = 1, 2, \ldots, I$.
(4) Sample the actually ith HQ delay $\ell_i(k)$.
(5) According Algorithm 1, get the K_{Pi}i¢§i¢§K_{Ii}i¢§i¢§K_{Di} and the number of time slots $\bar{s}_i(k)$.
(6) Distribute $\bar{s}_i(k)$ slots for ith HQ.
(7) End For.
(8) End TS-LOOP.
(9) Start CW-LOOP.
(10) Sample the actual jth SQ delay $\ell_j(k)$, the accessing probability $\rho_j(k)$, and queueing length $n_j(k)$.
(11) Construct $\mathrm{P}(k)$ and SOP $H(\mathfrak{R})(k)$ as (24).
(12) Solve SOP according to Algorithm 2, and get the solution $\mathfrak{R}(k) = (P_r(k), \tilde{s})$.
(13) Distribute $\tilde{s}$ slots for all SQ.
(14) For $j = 1, 2, \ldots, J$.
(15) Sample the actual accessing probability $\rho_j(k)$ and scale factor $v_j(k-1)$ as inputs. According to Algorithm 1, get K_{Pj}, K_{Ij}, K_{Dj} and the scale factors $v_j(k)$.
(16) Adjust scale factor $v_j(k)$ for jth SQ.
(17) End For.
(18) End CW-LOOP.
(19) $k : k = k + 1$, go to step 1.

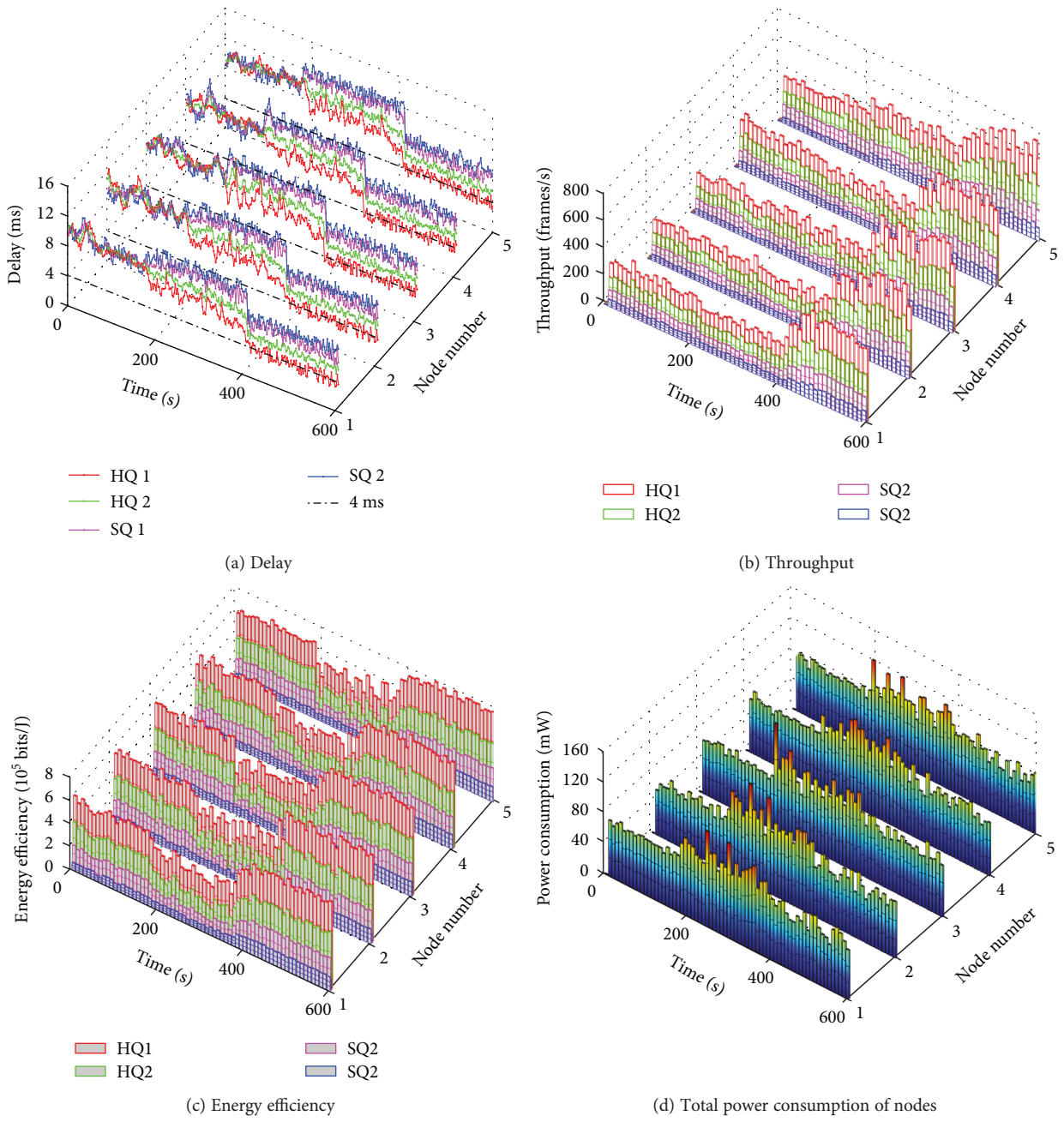
Algorithm 3: Algorithm of MQEB.

experiments. The former is strong time condition (STC), in which the preferred QoS metrics are better than the general case. The latter is weak time condition (WTC), which means the actual performance has been oversupplied.

MQEB is actually a double-looped control model. The paper takes a novel off-and-then-on model to test the dynamic performance, which is an equivalence of a step-function signal. The experiments last 600 s both in STC and WTC. 0~200 s, no controller acts in the loop and, thus, it is the original IEEE 802.15.4 MAC; 200~400 s, the controller which dynamically adjusts the contention windows size (FD-MAC) [19] starts to operate; 400~600 s, MQEB takes over to control the transmission. The consequences are shown in Figures 5 and 6. For clarity, only 5 nodes out of 20 are shown in the experiment results.

(1) Figures 5(a) and 6(a) show the average delay in STC and WTC. In the first 200 s (802.15.4 MAC), there is no significant difference among the traffic. During 200–400 s, FD-MAC operates to distinguish the average delay of different priorities. But the delay of HQ is larger than STC preset value and smaller than WTC preset value. FD-MAC can guarantee the delay rations of different classes be constants, also known as PDD. It is effective for SQ but not feasible to HQ; after 400 s (MQEB-MAC), the average delay is still distinguished and delay of HQ is also converged to the preset preferred value.

(2) Figures 5(b) and 6(b) show the corresponding throughput in STC and WTC. In the first 400 s, the throughput nearly has no change, which is identical to the results in our previous work [19]. However, when MQEB works on the system, a new feature emerges after 400 s. Throughput gets larger as delay decreases in STC and vice versa in WTC. It is very



(a) Delay

(b) Throughput

(c) Energy efficiency

(d) Total power consumption of nodes

FIGURE 5: STC: $\bar{L}_1 = 4$ ms, $\bar{L}_2 = 5$ ms; $\tilde{L}_1 = 7$ ms, $\tilde{L}_2 = 8$ ms.

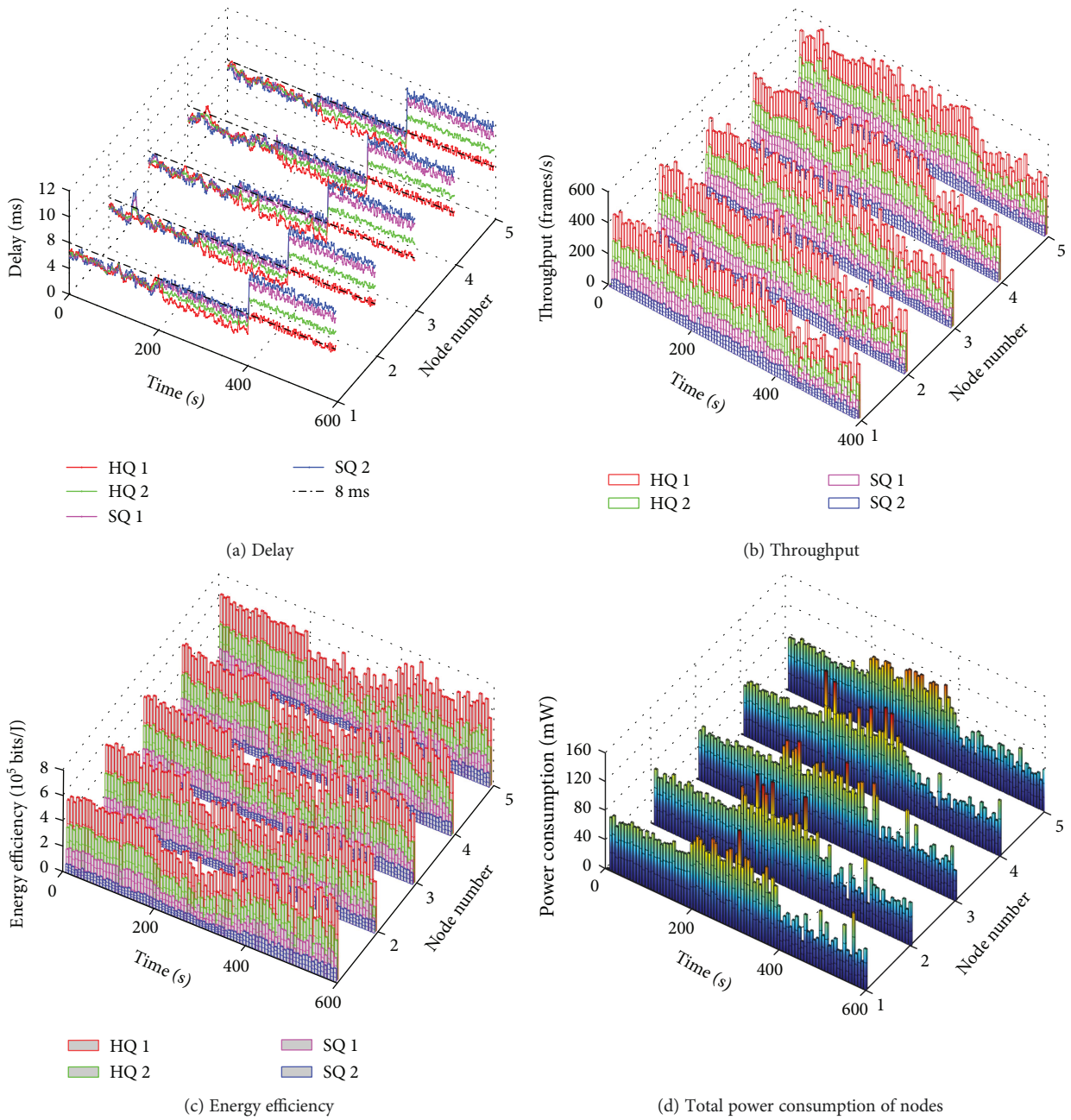
clear in Figure 5(b) that throughputs of HQ and SQ are greatly improved compared with the original 802.15.4 MAC and FD-MAC.

5.3. Energy Efficiency and Power Consumption. Energy efficiency and power consumption are two important factors for WANET. Energy efficiency is used to evaluate the efficiency of data transmission to the cost of energy [30]; the larger the better. In the experiment, valid data is the total length in bits of frame sent successfully, and energy is in unit of Joule. So the energy efficiency is measured by bits/Joule (bits/J), which is also known as bits/second/Watt.

Power consumption indicates the lifetime of UAV node; the smaller power consumption, the longer working time. Power consumption is in unit of milliwatt (mW).

$$\begin{aligned}\text{Efficiency} &= \frac{\sum_i^N \text{len}_i}{\text{energy}} = \frac{\sum_i^N \text{len}_i/t}{\text{energy}/t} \\ &\approx \frac{\mathfrak{E}_{\text{len}} N/t}{\text{power}} = \frac{\mathfrak{E}_{\text{len}} \text{throughput}}{\text{power}}.\end{aligned} \tag{26}$$

Supposing during the interval t, there are N frames with length of len_i are sent successfully by a node, the energy efficiency can be estimated as (28).



(a) Delay

(b) Throughput

(c) Energy efficiency

(d) Total power consumption of nodes

FIGURE 6: WTC: $\bar{L}_1 = 8$ ms, $\bar{L}_2 = 9$ ms; $\tilde{L}_1 = 9$ ms, $\tilde{L}_2 = 10$ ms.

(1) Figures 5(c) and 6(c) show the energy efficiency in STC and WTC. During 200–400 s (FD-MAC), whatever STC or WTC, the energy efficiency is smaller than that in the first 200 s (802.15.4 MAC). In channel contention, energy is consumed once nodes transmit no matter whether or not data is correctly received. FD-MAC making small CW size for high priority increases the collision probability, which will induce unwanted retransmitting and energy wasting.

(2) Comparing with 200–400 s (FD-MAC), the energy efficiency during 400–600 s (MQEB-MAC) enhances greatly in STC (Figure 5(c)), while remains nearly same in WTC (Figure 6(c)). The phenomenon can be explained by (24). In STC, the enlarged throughput (Figure 5(b)) goes further than the power wasting caused by collision. But in WTC, low throughput (Figure 6(b)) offsets the power saving.

(3) Figures 5(d) and 6(d) show the total power consumption on every nodes. During 400–600 s (MQEB-MAC), the total power consumption is different in STC and WTC. In STC, the total power consumption changes little compared with that of 200–400 s (Figure 5(d)) but decreased greatly in WTC (Figure 6(d)). The reason is MQEB can

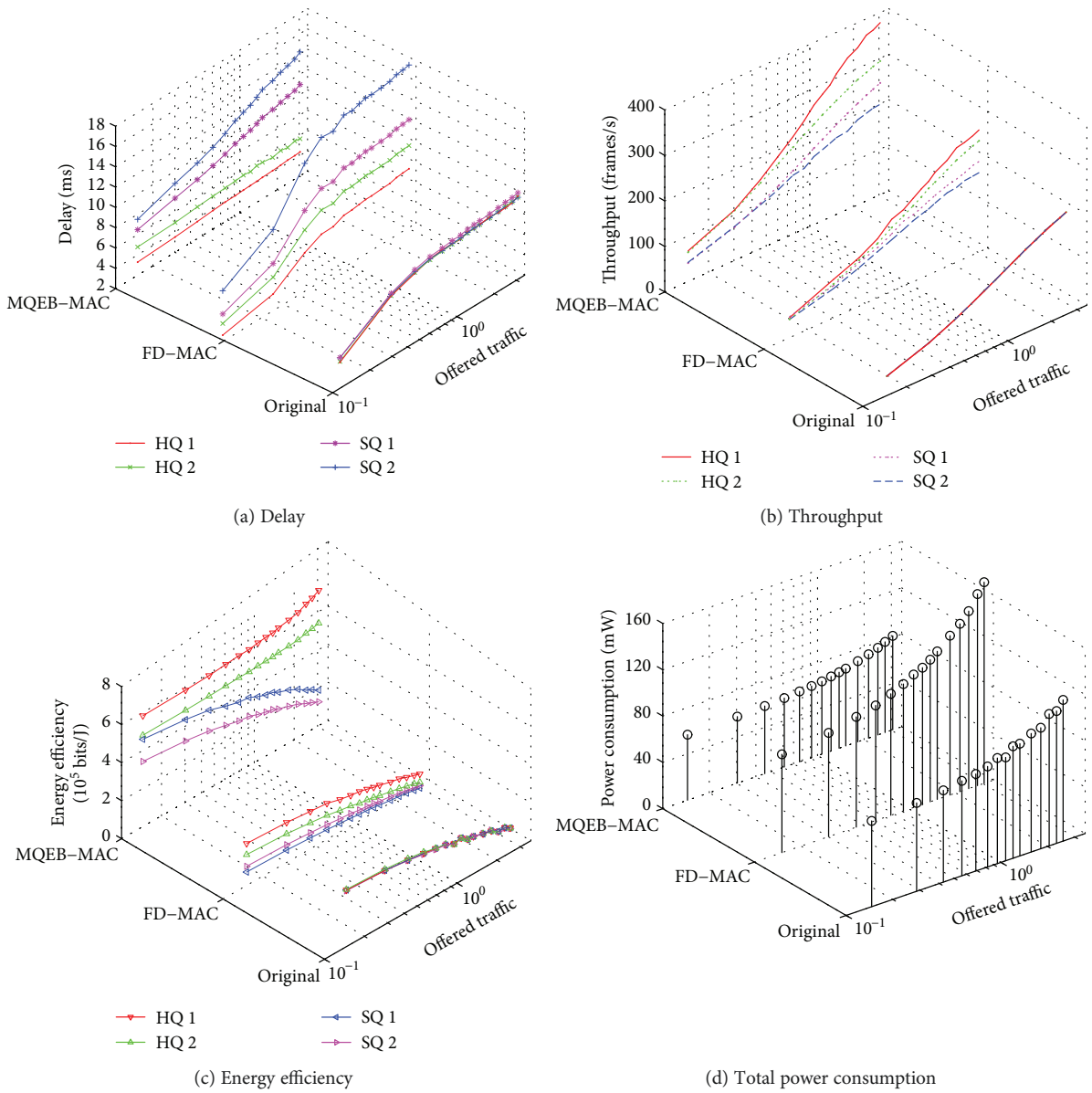


Figure 7: Comparison of MQEB in STC with offered traffic.

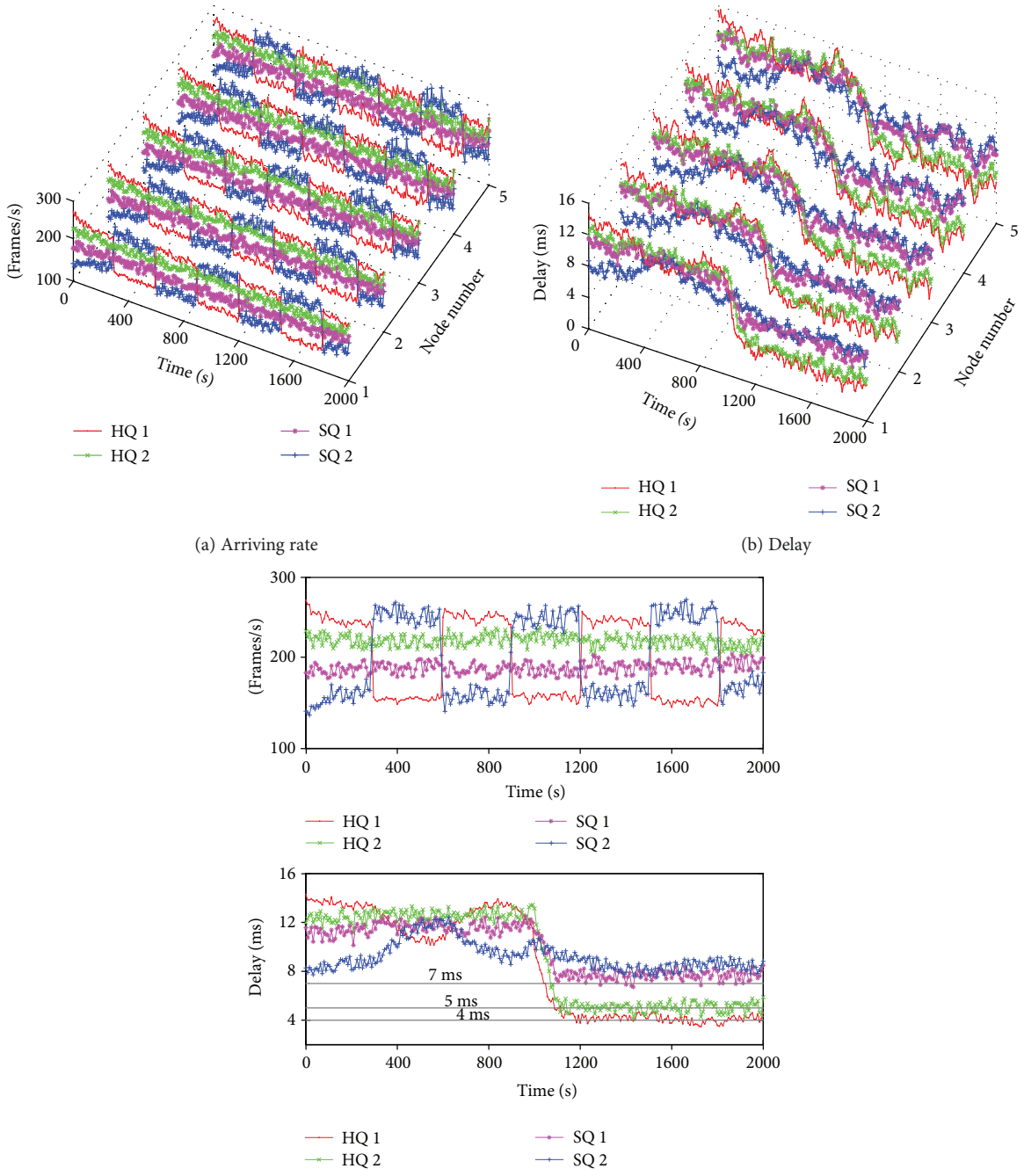
reduce the unnecessary active time to avoid circuit power consumption when the QoS performance is overprovided.

(4) It is worth mentioning that MQEB has a better energy efficiency in STC, because time slot isolation avoids the across-classes collision and retransmission energy wasting. Meanwhile, MQEB also has a lower power consumption and a longer lifetime in WTC because it can reduce the unnecessary active time.

5.4. Static Performance. MQEB performs a better throughput but sacrifices the lifetime in STC (Figures 5(b) and 5(d)) compared with that in WTC. Actually, STC is more common in UAV cloud. So the paper analyzes the static performance of MQEB in STC, which concerns the QoS metrics changing with the offered traffic. The experiment results are shown in Figure 7. x-axis presents different control approaches, which are original IEEE 802.15.4, FD-MAC and MQEB-MAC, respectively. The offered traffic is increased along with y-axis (logarithm scale). z-axis (vertical axis) represents the delay (Figure 7(a)), throughput (Figure 7(b)), energy efficiency (Figure 7(c)), and total power consumption (Figure 7(d)). The experiments indicate the comprehensive superiority of MQEB.

(1) In Figure 7(a), comparing with original 802.15.4 MAC, both FD-MAC and MQEB-MAC can support differentiated delay QoS. But MQEB-MAC can further control the delay fixed to the preset value of 4 ms for HQ 1 and 5 ms for HQ 2.

(2) In Figure 7(b), MQEB-MAC could not only support the differentiated service in delay but also greatly



(a) Arriving rate

(b) Delay

(c) Traffic load and delay on node 1

Figure 8: The robust verification when traffic burst in STC

enhance the throughput (both HQ and SQ). It is because in STC, active time is tend to increase with offered traffic. The more working time obviously enlarges the throughput.

(3) In Figure 7(c), the energy efficiency of MQEB-MAC is better than that of FD-MAC. Especially for HQ, the energy efficiency even increases with offered traffic. It is because throughput goes much faster than power consumption. This shows the advantage of time slot isolation again can void the transmission failure as well as enlarge the throughput of HQ. By measurement, the power wasting of one frame collision (about 10 mW) is at least ten times of (about 1 mW) circuit power consumption.

(4) In Figure 7(d), FD-MAC can support differentiated QoS but sacrifice the total power consumption. The reason is mainly the frame collision and retransmission that have been discussed before. For MQEB-MAC, the total power consumption is nearly the same (a litter smaller) with original 802.15.4 MAC.

The possible reason is that the saved energy of collision avoidance by time slot isolation just compensates the energy cost of enlarged working time.

5.5. Robust Verification. The above experiments concern the effectiveness of MQEB without considering the traffic burst. Robust verification is enforced when the traffic arriving rate varies dynamically. Figures 8(a) and 8(b) are the arriving rate and average delay changing with time. Figure 8(c) is the condition of node 1 for clarity. The arriving rates (traffic load) of HQ 1 and SQ 2 are changing in turns with the interval of 300 s. The maximum and minimum load are approximately 250 frames/s and 50 frames/s. The other two flows, that is, HQ 2 and SQ 1, are constant near 230 frames/s and 170 frames/s.

Considering the analysis in Section 5.4, we only discuss STC for simplification. The preferred QoS metrics are identical to that of STC in Figure 5, that is, $\bar{L}_1 = 4$ ms, $\bar{L}_2 = 5$ ms and $\tilde{L}_1 = 7$ ms, $\tilde{L}_2 = 8$ ms. MQEB controller starts at 1000 s.

(1) In the first 1000 s (original 802.15.4 MAC), the delay of all types varies along with the load and the delays of HQ 1 and HQ 2 are larger than the preferred value. When the arriving rate of HQ 1 bursts at 600 s, the delay is even larger than that of SQ 1 and SQ 2, which is not a desired phenomenon.

(2) After 1000 s, MQEB works on the system. The delays of HQ 1 and HQ 2 converge to the preferred value, which benefits from the BP-based self-tuning feedback control.

(3) Although the delays of SQ 1 and SQ 2 get some improvement, they could not always converge to the preset value, and there exist a little jitter with the load fluctuation. This is because the QoS ensurance of SQ is actually a discount of preferred QoS metric with balance of queueing length as well as energy saving.

(4) Because of load changing as a gating function, both the up edge and down edge are step impulses, which are the most serious conditions for controller. So the experiment shows that BP-based self-tuning controller in MQEB is robust stability.

6. Conclusion

To ensure the performance of compute-intensive applications unloaded to cloud, the paper proposes a BP-based self-tuning PID controller for mixed QoS ensurance architecture, which can support both real-time and non-real-time traffic. By time slot isolation for hard QoS and accessing probability optimization for soft QoS, the back propagating (BP) neuron network-based PID control is used for parameters' self-tuning. The hardware experiments demonstrate the feasibility of MQEB-MAC. Comparing with FD-MAC, MQEB has new feature of hard QoS ensurance and soft QoS equilibrium with energy saving. It has further developed two advantages. In the condition of heavy load (STC), MQEB has greater throughput and better energy efficiency; and in light load (WTC), MQEB has lower power consumption.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (nos. 61203233 and 661601365), China Postdoctoral Science Foundation (no. 2017M623243), Natural Science Basic Research Plan in Shaanxi Province of China (nos. 2016JM6062 and 2016JQ6017), Shanghai Aerospace Science and Technology Innovation Fund (no. SAST2016034), and China Fundamental Research Fund for the Central Universities (nos. 3102017zy029 and 300102328105).

References

[1] G. Hu, W. P. Tay, and Y. Wen, "Cloud robotics: architecture, challenges and applications," *IEEE Network*, vol. 26, no. 3, pp. 21–28, 2012.

[2] K. Kamei, S. Nishio, N. Hagita, and M. Sato, "Cloud networked robotics," *IEEE Network*, vol. 26, no. 3, pp. 28–34, 2012.

[3] F. Luo, C. Jiang, S. Yu, J. Wang, Y. Li, and Y. Ren, "Stability of cloud-based UAV systems supporting big data acquisition and processing," *IEEE Transactions on Cloud Computing*, no. 99, p. 1, 2017.

[4] L. Tan, Z. Zhu, W. Zhang, and G. Chen, "An optimal solution to resource allocation among soft QoS traffic in wireless network," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 2642–2657, 2013.

[5] B. Li, X. Li, R. Zhang, W. Tang, and S. Li, "Joint power allocation and adaptive random network coding in wireless multicast networks," *IEEE Transactions on Communications*, no. 99, p. 1, 2017.

[6] B. D. Winter and R. A. Swartz, "Wireless structural control using multi-step TDMA communication patterning bandwidth allocation," *Structural Control and Health Monitoring*, vol. 24, no. 12, article e2025, 2017.

[7] A. Mengali, R. De Gaudenzi, and P.-D. Arapoglou, "Enhancing the physical layer of contention resolution diversity slotted ALOHA," *IEEE Transactions on Communications*, vol. 65, no. 10, pp. 4295–4308, 2017.

[8] R. De Gaudenzi, O. Del Rio Herrero, G. Gallinaro, S. Cioni, and P.-D. Arapoglou, "Random access schemes for satellite networks, from VSAT to M2M: a survey," *International Journal of Satellite Communications and Networking*, vol. 36, no. 1, pp. 66–107, 2018.

[9] N. Saxena, A. Roy, and J. Shin, "Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks," *Computer Networks*, vol. 52, no. 13, pp. 2532–2542, 2008.

[10] A. K. Subramanian and I. Paramasivam, "PRIN: a priority-based energy efficient MAC protocol for wireless sensor

networks varying the sample inter-arrival time," *Wireless Personal Communications*, vol. 92, no. 3, pp. 863–881, 2017.

[11] B. Jang, J. B. Lim, and M. L. Sichitiu, "An asynchronous scheduled mac protocol for wireless sensor networks," *Computer Networks*, vol. 57, no. 1, pp. 85–98, 2013.

[12] M. Y. Naderi, P. Nintanavongsa, and K. R. Chowdhury, "RF-MAC: a medium access control protocol for re-chargeable sensor networks powered by wireless energy harvesting," *IEEE Transactions on Wireless Communications*, vol. 13, no. 7, pp. 3926–3937, 2014.

[13] B. J. Chang and S. P. Chen, "Cross-layer-based adaptive congestion and contention controls for accessing cloud services in 5G IEEE 802.11 family wireless networks," *Computer Communications*, vol. 106, pp. 33–45, 2017.

[14] Y. Özen, N. Bandrmal, and C. Baylmç, "Two tiered service differentiation mechanism for wireless multimedia sensor network MAC layers," in *2015 23nd Signal Processing and Communications Applications Conference (SIU)*, pp. 2318–2321, Malatya, Turkey, 2015.

[15] M. Hadded, P. Muhlethaler, A. Laouiti, R. Zagrouba, and L. A. Saidane, "TDMA-based MAC protocols for vehicular ad hoc networks: a survey, qualitative analysis, and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2461–2492, 2015.

[16] H. A. Omar, W. Zhuang, and L. Li, "VeMAC: a TDMA-based MAC protocol for reliable broadcast in VANETs," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1724–1736, 2013.

[17] S. Khan, M. Alam, N. Müllner, and M. Fränzle, "Poster: a hybrid MAC scheme for emergency systems in urban VANETs environment," in *2016 IEEE Vehicular Networking Conference (VNC)*, pp. 1-2, Columbus, OH, USA, 2016.

[18] Y. Guo, Q. Yang, and K. S. Kwak, "Quality-oriented rate control and resource allocation in time-varying OFDMA networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2324–2338, 2017.

[19] A. Gao and Y. Hu, "A feedback approach for QoS-enhanced MAC in wireless sensor network," *Journal of Sensors*, vol. 2016, Article ID 8365217, 12 pages, 2016.

[20] J. Du, C. Jiang, Y. Qian, Z. Han, and Y. Ren, "Resource allocation with video traffic prediction in cloud-based space systems," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 820–830, 2016.

[21] N. Mastronarde, J. Modares, C. Wu, and J. Chakareski, "Reinforcement learning for energy-efficient delay-sensitive CSMA/CA scheduling," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, Washington, DC, USA, 2016.

[22] L. Tan, Z. Zhu, F. Ge, and N. Xiong, "Utility maximization resource allocation in wireless networks: methods and algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1018–1034, 2015.

[23] K. Gai, M. Qiu, and H. Zhao, "Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing," *IEEE Transactions on Cloud Computing*, no. 99, p. 1, 2016.

[24] A. Verma and S. Kaushal, "A hybrid multi-objective particle swarm optimization for scientific workflow scheduling," *Parallel Computing*, vol. 62, pp. 1–19, 2017.

[25] C. Wang and W.-H. Kuo, "A utility-based resource allocation scheme for IEEE 802.11 WLANs via a machine-learning approach," *Wireless Networks*, vol. 20, no. 7, pp. 1743–1758, 2014.

[26] A. Benfarah, N. Laurenti, and S. Tomasin, "Resource allocation for downlink of 5G systems with OFDMA under secrecy outage constraints," in *2016 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, Washington, DC, USA, 2017.

[27] W. Chen and H. V. Poor, "Joint pushing and caching with a finite receiver buffer: optimal policies and throughput analysis," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, Malaysia, 2016.

[28] L. Chen, B. Wang, X. Chen, X. Zhang, and D. Yang, "Utility-based resource allocation for mixed traffic in wireless networks," in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 91–96, Shanghai, China, 2011.

[29] K. I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.

[30] Z. Yan, M. Peng, and C. Wang, "Economical energy efficiency: an advanced performance metric for 5G systems," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 32–37, 2017.