Delay Based Wireless Scheduling and Server Assignment for Fog Computing Systems

Yuan Zhang[1](✉), Mingyang Xie[1], Qiang Guo[1], Wei Heng[1],
and Peng Du[2]

[1] National Mobile Communications Research Laboratory,
Southeast University, Nanjing 210096, China
{y.zhang,qguo,wheng}@seu.edu.cn, 1090492123@qq.com
[2] College of Automation and College of Artificial Intelligence,
Nanjing University of Posts and Telecommunications, Nanjing 210023, China
dupeng@njupt.edu.cn

Abstract. To further reduce the delay in fog computing systems, new resource allocation algorithms are needed. Firstly, we have derived the recursive expressions of the communication and computing delays in the fog computing system without assuming the knowledge of the statistics of user application arrival traffic. Based on these analytical formulas, an optimization problem of delay minimization is formulated directly, and then a novel wireless scheduling and server assignment algorithm is designed. The delay performance of the proposed algorithm is evaluated via simulation experiments. Under the considered simulation parameters, the proposed algorithm can achieve 13.5% less total delay, as compared to the traditional algorithm. The impact of the total number of subcarriers in the system and the average user application arrival rate on the percentage of delay reduction is evaluated. Therefore, compared with the queue length optimization based traditional resource allocation algorithms, the delay optimization based resource allocation algorithm proposed in this paper can further reduce delay.

Keywords: Fog computing · Resource allocation · Delay · Lyapunov

1 Introduction

Recent years have seen a trend of users needing to run computation-intensive applications. To meet this requirement, the idea of fog computing is introduced [1, 2]. That is, computing servers (also known as fog nodes) are located near users, then users' applications are offloaded to fog nodes to execute. In fog computing systems, the problem of resource allocation has two aspects. Firstly, how to schedule wireless resource among users? For example, as illustrated in Fig. 1, both U1 and U2 want to offload applications to F1. How to schedule wireless resource between U1 and U2? Secondly, how to assign servers to users? For example, as illustrated in Fig. 1, U3 can offload applications to F2, F3, or F4. How to assign servers to U3? This paper studies these two aspects of resource allocation and proposes wireless scheduling and server assignment algorithms for fog computing systems.

J. Zheng et al. (Eds.): ADHOCNETS 2019, LNICST 306, pp. 187–198, 2019.
https://doi.org/10.1007/978-3-030-37262-0_15

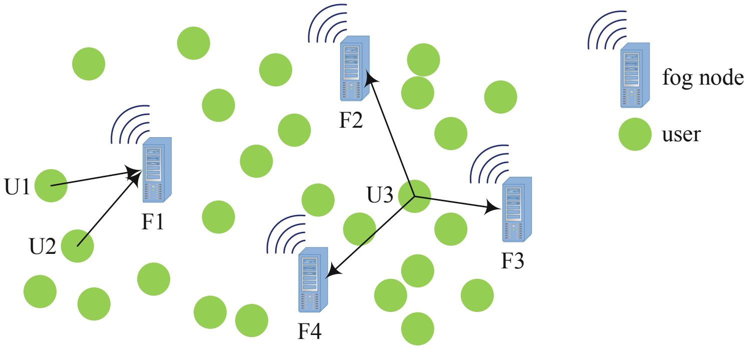

Fig. 1. Wireless scheduling and server assignment in fog computing systems.

There are many related studies in the literature (e.g., [3–15]). According to the assumptions of delay, there are three categories of resource allocation algorithms. For the first category of resource allocation algorithms (e.g., [3–7]), only the time of communicating data bits from user to fog node and the time of running application are considered. However, the time wasted in the user queues waiting to communicate or the time wasted in the fog node queues waiting to run is not considered. For the second category (e.g., [8–11]), in addition to communication time and running time, the delays of queueing are also included. However, this category assumes the queues can be modelled as M/M/1 or M/G/1 queues so that those formulas of delay in the queueing theory can be re-used. For the third category (e.g., [12–15]), the queueing delays are also included. For this category, the resource allocation algorithm is derived in the following manner. Firstly, according to Little's Law, the average delay and queue length can be considered equivalent; then, the queue length based Lyapunov function is introduced and the bound of the conditional drift of this Lyapunov function is estimated; finally, using the Lyapunov optimization framework established in [16, 17], the resource allocation algorithm is design to minimizes the drift.

In this work, we focus on the category of Lyapunov optimization technique based resource allocation algorithms. This category of algorithms does not need any assumptions about the statistics of traffic. Consequently, the formulas of delay in the queueing theory cannot be used. Thus, since there is no formula of delay, this category of resource allocation algorithms cannot directly attack the problem of delay minimization but have to address the problem of queue length stability as an alternative. Therefore, this work will extend the category of Lyapunov optimization technique based resource allocation by designing algorithms which can directly minimize the delay and at the same time does not need assumptions on the statistics of traffic. The work in [18] is our first step toward this direction which focused on the single access-point scenario. Compared with our previous work in [18], this work focuses on the multi-access-point scenario, in which in addition to the need to decide to schedule subcarriers, it is also necessary to decide which access point to transfer the computation application to. The contributions of this work are summarized as following. Firstly, the recursive expressions of queueing delays in fog computing systems are derived. During the derivations, no assumptions on the statistics of traffic is needed. Secondly, a resource allocation algorithm for fog computing systems is proposed which can minimize the total delay directly. Finally, simulation results are reported which show that the proposed delay based resource allocation algorithm provides better delay performance than the traditional queue length based Lyapunov allocation algorithm.

The organization of this paper is as follows. Section 2 derives the queueing and delay models. Section 3 proposes a resource allocation algorithm which minimizes delay directly. Section 4 reports the results of simulation experiments. Section 5 gives concluding remarks. The summary of the main notations used in this paper is provided in Table 1.

Table 1. Summary of notations.

Notation	Description
T	The duration of a slot
I	User number
E_i	The number of cycles needed by the application of user i
J	Fog node number
F_j	The number of cycles provided by fog node j per second
Ψ_i	The set of neighbor fog nodes of user i
Ω_{ij}	The set of competitor users of user i for fog node j
$U_i[n-1]$	The number of queued applications of the ith user sampled at the start of the nth slot
$X_i[n]$	The number of applications leaving the queue of the ith user during the nth slot
R	The number of subcarriers in the air interface
$R_{ij}[n]$	The number of subcarriers to transfer an application from the ith user to the jth fog node in the nth slot
$W_i[n]$	The virtual queue of the normalized communication delay of user i
ε_n	Smoothing coefficient
$x_{ij}[n]$	The number of applications from the ith user to the jth fog node which is decided at the end of the nth slot
α_{ij}	The value of E_i/F_jT
$S_j[n]$	The normalized value of the number of cycles needed by all the applications which are still in the jth computing queue at the end of the $(n+1)$th slot
$d_{ijh}[n]$	The normalized value of the computing delay of the hth application which comes from the ith user and is executed by the jth fog node
$D_{ij}[n]$	The value of $\sum_{h=1}^{x_{ij}[n]} d_{ijh}[n]$
$Z_{ij}[n]$	the virtual queue of the normalized computing delay of user i in fog node j
$L[n]$	The Lyapunov function
$\Delta[n]$	The conditional drift of the Lyapunov function

2 System Models

Consider a time-slotted fog computing system. Let T represent the duration of a slot. Let I and J represent the number of users and fog nodes, respectively. For any application of user i, it will need E_i cycles to execute. For each fog node j, it can provide F_j per second. Applications are offloaded to fog nodes to execute. Therefore, if an application of user i is transferred to the jth fog node to execute, it need E_i/F_j seconds to finish the execution.

Given a pair of fog node j and user i, if the fog node can receive the signal from user i with a SINR (i.e., the signal to interference plus noise ratio) exceeding a given threshold, user i is a *neighbor* of fog node j. Let Ψ_i denote the set of neighbor fog nodes of user i. Further, for any two users i and k, if there exists a fog node which is accessible by both i and k, we say user k is a *competitor user*. For any user i, let Ω_i be the set of all his competitor users.

2.1 Communication Delay Model

Firstly, we derive the equation describing the evolution of communication queues. For the ith user, let $U_i[n-1]$ denote the number of queued applications which is sampled at the start of nth slot. Then, let $X_i[n]$ denote the number of applications which is transferred to some fog node (i.e., depart the queue) during the nth slot. Finally, let $A_i[n]$ denote the number of applications which newly arrives to this queue during the nth slot. Although $X_i[n]$ is the number of applications which leave the ith communication queue during the nth slot, its value is actually decided at the start of the nth slot. The value of $X_i[n]$ should not be greater than the number of applications which are still staying in the ith communication queue when the decision is made, that is, at the start of the nth slot. Then we have:

$$0 \leq X_i[n] \leq U_i[n-1]. \tag{1}$$

Additionally, the value of $X_i[n]$ is constrained by the capability of wireless transmission resource. Let R represent the number of all possible subcarriers which can be used. Then for each fog node $j \in \Psi_i$, we have:

$$\sum\nolimits_{j \in \Psi_i} R_{i,j}[n] + \sum\nolimits_{k \in \Omega_i} \sum\nolimits_{h \in \Psi_k} R_{k,h}[n] \leq R, \tag{2}$$

where $R_{ij}[n]$ represents the number of subcarriers which are required to transfer a user i's application to the jth fog node in the nth slot. Hence, the recursive equation describing the communication queue of the ith user is:

$$U_i[n] = U_i[n-1] - X_i[n] + A_i[n], \tag{3}$$

where $X_i[n]$ satisfies the constraints in (1) and (2).

Next, we derive the recursive expression of the communication delay (including transmitting time and waiting time) of the ith user. Let $\Gamma_{\mathrm{tot},i}[n]$ be the total communication delay that has been experienced by all applications of the ith user until the $(n + 1)$th slot. Thus, we have that:

$$\Gamma_{\mathrm{tot},i}[n] = \sum\nolimits_{k=1}^{n} U_i[k]T. \tag{4}$$

Let $\Gamma_i[n]$ denote the time-average of $\Gamma_{\mathrm{tot},i}[n]$, that is:

$$\Gamma_i[n] = \frac{\Gamma_{\mathrm{tot},i}[n]}{n}. \tag{5}$$

In this paper, $\Gamma_i[n]$ is used to indicate length of the communication delay of the ith user. We further express $\Gamma_i[n]$ as a virtual queue:

$$\Gamma_i[n] = \Gamma_i[n-1] - \varepsilon_n \Gamma_i[n-1] + \varepsilon_n U_i[n]T, \tag{6}$$

with $\varepsilon_n = 1/n$. Let $W_i[n]$ be the value of $\Gamma_i[n]$ normalized to the slot length, that is, let $W_i[n] = \Gamma_i[n]/T$. Thus, we have that:

$$W_i[n] = W_i[n-1] - \varepsilon_n W_i[n-1] + \varepsilon_n U_i[n]. \tag{7}$$

2.2 Computing Delay Model

Firstly, we derive the equation describing the evolution of computing queues. Since there are J fog nodes, there are J computing queues to be modeled. At the end of the nth slot, there are $\Sigma_{1 \le i \le I} X_i[n]$ applications arriving to fog nodes. Let $x_{ij}[n]$ be the number of applications which are transferred from the ith user to the jth fog node. Thus, at the start of the $(n + 1)$th slot, there will be $\Sigma_{1 \le i \le I} x_{ij}[n]$ applications arriving to the jth fog node. These applications require $\Sigma_{1 \le i \le I} E_i x_{ij}[n]$ cycles. Obviously, $x_{ij}[n]$ must satisfy the following constrain:

$$\sum\nolimits_{j=1}^{J} x_{ij}[n] = X_i[n]. \tag{8}$$

Let $\Phi_j[n]$ be the cycle number of all the applications staying in the jth computing queue by the end of the $(n + 1)$th slot. Hence, the recursive equation of the computing queue is:

$$\Phi_j[n] = \left(\Phi_j[n-1] + \sum\nolimits_{i=1}^{I} E_i x_{ij}[n] - F_j T\right)^{+}, \tag{9}$$

where $(\cdot)^{+} = \max(\cdot, 0)$ and $x_{ij}[n]$ satisfies the constraint in (8). Similarly, let $S_j[n]$ be the value of $\Phi_j[n]$ normalized to the cycle number provided by fog node in one slot, that is, $S_j[n] = \Phi_j[n]/F_jT$. Thus, we have that:

$$S_j[n] = \left(S_j[n-1] + \sum\nolimits_{i=1}^{I} \alpha_{ij} x_{ij}[n] - 1\right)^{+}, \tag{10}$$

where $\alpha_{ij} = E_i/F_jT$.

Next, we derive the formula of computing delay (including execution time and waiting time) of user i. Let $Z_{\text{tot},ij}[n]$ be the normalized version of the total computing delay which is experienced by all applications of the ith user in the jth fog node until the $(n + 1)$th slot. Thus, we have that:

$$Z_{\text{tot},ij}[n] = \sum\nolimits_{k=1}^{n} \sum\nolimits_{h=1}^{x_{ij}[n]} d_{ijh}[k], \tag{11}$$

where $d_{ijh}[n]$ is the normalized version of the computing delay of the hth application and $1 \leq h \leq x_{ij}[n]$. The expression of $d_{ijh}[n]$ is derived as follows, which have three terms. For the first term, on the arrival of the hth application, if the queue is not null, it has to wait the applications queued before it to complete their executions. Therefore, $d_{ijh}[n]$ includes the term of $S_j[n - 1]$. For the second term, let $Bef_{ijh}[n] = \{(k, l)$: the lth application of user k is executed before the hth application of user i on fog node j in slot $n + 1\}$. Therefore, $d_{ijh}[n]$ includes the term of α_{kj} for each (k, l) in $Bef_{ijh}[n]$. For the third term, the running time of the application itself should also be considered. Thus, we have that:

$$d_{ijh}[n] = S_j[n-1] + \sum\nolimits_{(k,l) \in Bef_{ijh}[n]} \alpha_{kj} + \alpha_{ij}. \tag{12}$$

Let $Z_{ij}[n]$ denote the time-average of $Z_{\text{tot},ij}[n]$, that is:

$$Z_{ij}[n] = \frac{Z_{\text{tot},ij}[n]}{n}. \tag{13}$$

In this paper, we use $Z_{ij}[n]$ to indicate length of the computing delay which is experienced by the applications of the ith user in the jth fog node. We express $Z_{ij}[n]$ as a virtual queue:

$$Z_{ij}[n] = Z_{ij}[n-1] - \varepsilon_n Z_{ij}[n-1] + \varepsilon_n D_{ij}[n], \tag{14}$$

where

$$D_{ij}[n] = \sum\nolimits_{h=1}^{x_{ij}[n]} d_{ijh}[n]. \tag{15}$$

3 Algorithm Design

First of all, the Lyapunov function defined in this paper is:

$$L[n] = \sum\nolimits_{i=1}^{I} W_i[n]^2 + \sum\nolimits_{i=1}^{I} \sum\nolimits_{j=1}^{J} Z_{ij}[n]^2 \tag{16}$$

According to the Lyapunov optimization technique established in [16, 17], we need to estimate the value of the conditional drift $\Delta[n] = \mathrm{E}\{L[n] - L[n-1] | \mathbf{W}[n-1], \mathbf{Z}[n-1]\}$, where $\mathrm{E}\{\cdot\}$ is the expectation operation, $\mathbf{W}[n-1] = [W_1[n-1], \ldots, W_I[n-1]]$, and $\mathbf{Z}[n-1] = [\mathrm{Z}_{11}[n-1], \ldots, Z_{IJ}[n-1]]$. Substituting (16), we have $\Delta[n] = \mathrm{E}\{\Sigma_{1\le i \le I}\varepsilon_n^2 W_i[n-1]^2 + \Sigma_{1\le i\le I}\Sigma_{1\le j\le J}\varepsilon_n^2 Z_{ij}[n-1]^2 + \Sigma_{1\le i\le I}\varepsilon_n^2 U_i[n]^2 + \Sigma_{1\le i\le I}\Sigma_{1\le j\le J}\varepsilon_n^2 D_{ij}[n]^2 - \Sigma_{1\le i\le I} 2\varepsilon_n W_i[n-1]^2 - \Sigma_{1\le i\le I}\Sigma_{1\le j\le J} 2\varepsilon_n Z_{ij}[n-1]^2 + \Sigma_{1\le i\le I} 2\varepsilon_n(1-\varepsilon_n) W_i[n-1] U_i[n] + \Sigma_{1\le i\le I}\Sigma_{1\le j\le J} 2\varepsilon_n(1-\varepsilon_n) Z_{ij}[n-1] D_{ij}[n] | \mathbf{W}[n-1], \mathbf{Z}[n-1]\}$, where the first six terms can be upper bounded by a constant under the expectation operation. According to the Lyapunov optimization technique established in [16, 17], this expression can be minimized by an algorithm which obtains the values of $\mathbf{W}[n-1]$ and $\mathbf{Z}[n-1]$ and chooses $X_i[n]$ and $x_{ij}[n]$ to minimize $\Sigma_{1\le i\le I} W_i[n-1] U_i[n] + \Sigma_{1\le i\le I}\Sigma_{1\le j\le J} Z_{ij}[n-1] D_{ij}[n]$. Further, substituting (3), the objective can be written as $\Sigma_{1\le i\le I} W_i[n-1] U_i[n-1] + \Sigma_{1\le i\le I} W_i[n-1] A_i[n] - \Sigma_{1\le i\le I} W_i[n-1] X_i[n] + \Sigma_{1\le i\le I}\Sigma_{1\le j\le J} Z_{ij}[n-1] D_{ij}[n]$, where the first two terms can also be upper bounded by a constant. Thus, this expression can be minimized by the algorithm that minimizes $-\Sigma_{1\le i\le I} W_i[n-1] X_i[n] + \Sigma_{1\le i\le I}\Sigma_{1\le j\le J} Z_{ij}[n-1] D_{ij}[n]$. Substituting (15), the final form of the programming to be solved for each slot n is:

$$\begin{aligned} \min_{\{X_i[n],\, x_{ij}[n]\}} \; & -\sum_{i=1}^{I} W_i[n-1] X_i[n] + \sum_{i=1}^{I}\sum_{j=1}^{J}\left(Z_{ij}[n-1] \sum_{h=1}^{x_{ij}[n]} d_{ijh}[n] \right) \\ \text{s.t.} \quad & X_i[n] \le U_i[n-1] \\ & R_{ij}[n] X_i[n] + \sum\nolimits_{k\in\Omega_{ij}} R_{kj}[n] X_k[n] \le R,\ j \in \Psi_i \\ & \sum\nolimits_{j=1}^{J} x_{ij}[n] = X_i[n] \end{aligned} \tag{17}$$

where $d_{ijh}[n]$ is provided in (12) and $X_i[n]$ and $x_{ij}[n]$ are integers.

Before describing the algorithm, the concept of feasible user is needed to be introduced. Specifically, for the ith user, if the following judging criteria are true, one more application can be allowed to be transferred from this user to some fog node. For the first criteria, according to the constraint in (1), if $X_i[n] < U_i[n-1]$, then one more application can be allowed to be transferred from the ith user to some fog node; otherwise, if $X_i[n] = U_i[n-1]$, then no application is allowed to be transferred from the ith user to some fog node. For the second criteria, if the constraint in (2) holds with equality, then no application is allowed to be transferred from the ith user to some fog node. Thus, the set of feasible user is define as:

$$C[n] = \{i : X_i[n] < U_i[n-1] \text{ and } \sum\nolimits_{j \in \Psi_i} R_{i,j}[n] + \sum\nolimits_{k \in \Omega_i} \sum\nolimits_{h \in \Psi_k} R_{k,h}[n] \leq R - 1\} \tag{18}$$

Then, the proposed resource allocation algorithm works as following. Initially, we have $X_i[n] = 0$, $x_{ij}[n] = 0$, $TW_i = (1 - \varepsilon_n)W_i[n-1]$, $TZ_{ij} = (1 - \varepsilon_n)Z_{ij}[n-1]$, $TS_j = S_j[n-1]$ for each i and j. The steps of the proposed algorithm are as follows.

Step 1: Determine the value of the feasible user set $C[n]$. If $C[n]$ is null, the algorithm halts.
Step 2 (Wireless Scheduling): Select the user $i^* = \arg\max TW_i$ over all feasible users in $C[n]$. Update $X_{i^*}[n] \leftarrow X_{i^*}[n] + 1$ and $TW_{i^*} \leftarrow TW_{i^*} + \varepsilon_n$.
Step 3 (Server Assignment): Determine the fog node $j^* = \arg\min TZ_{i^*j}$ over all fog node $j \in \Psi_{i^*}$. Update $x_{i^*j^*}[n] \leftarrow x_{i^*j^*}[n] + 1$, $TS_{j^*} \leftarrow TS_{j^*} + \alpha_{i^*j^*}$, and $TZ_{i^*j^*} \leftarrow TZ_{i^*j^*} + \varepsilon_n TS_{j^*}$. Go to Step 1.

3.1 The Traditional Queue Length Based Algorithm

For convenience, the traditional queue length based Lapunov resource allocation algorithm is outlined in this subsection. The queue length based Lyapunov function is defined as follows:

$$L[n] = \sum\nolimits_{i=1}^{I} U_i[n]^2 + \sum\nolimits_{j=1}^{J} S_j[n]^2 \tag{19}$$

We need to estimate the bound of the conditional drift of this Lyapunov function, which can be written as $\Delta[n] = \mathrm{E}\{L[n] - L[n-1] | \mathbf{U}[n-1], \mathbf{S}[n-1]\}$, where $\mathbf{U}[n-1] = [U_1[n-1], \ldots, U_I[n-1]]$, and $\mathbf{S}[n-1] = [\mathrm{S}_1[n-1], \ldots, S_J[n-1]]$. After similar derivations [16, 17], the optimization problem to be solved by the traditional queue length based algorithm is:

$$\begin{aligned} \min_{\{X_i[n],\, x_{ij}[n]\}} \; & -\sum_{i=1}^{I} U_i[n-1]X_i[n] + \sum_{i=1}^{I}\sum_{j=1}^{J}\left(S_j[n-1]\sum_{h=1}^{x_{ij}[n]} \alpha_{ij}\right) \\ \text{s.t.} \quad & X_i[n] \leq U_i[n-1] \\ & R_{ij}[n]X_i[n] + \sum\nolimits_{k \in \Omega_{ij}} R_{kj}[n]X_k[n] \leq R,\; j \in \Psi_i \\ & \sum\nolimits_{j=1}^{J} x_{ij}[n] = X_i[n] \end{aligned} \tag{20}$$

where $X_i[n]$ and $x_{ij}[n]$ are integers. Since this optimization problem is similar to the one in (17), we can use the similar procedure to address this problem. Initially, set $X_i[n] = 0$, $x_{ij}[n] = 0$, $TU_i = U_i[n-1]$, and $TS_j = S_j[n-1]$. Then steps of the traditional queue length based algorithm are as follows.

Step 1: Calculate the value of the feasible user set $C[n]$. If $C[n]$ is null, the algorithm halts.

Step 2 (Wireless Scheduling): Select the user $i^* = \arg\max TU_i$ over all $i \in C[n]$. Update $X_{i^*}[n] \leftarrow X_{i^*}[n] + 1$ and $TU_{i^*} \leftarrow TU_{i^*} - 1$.

Step 3 (Server Assignment): Determine the fog node $j^* = \arg\min TS_j$ over all fog node $j \in \Psi_{i^*}$. Update $x_{i^*j^*}[n] \leftarrow x_{i^*j^*}[n] + 1$ and $TS_{j^*} \leftarrow TS_{j^*} + \alpha_{i^*j^*}$. Go to Step 1.

4 Performance Evaluation

Consider a time-slotted fog computing system. Assume there are $J = 4$ fog nodes. We set the geographical locations of fog nodes as (400, 400), (400, 800), (800, 400) and (800, 800) in meter. The default value of F_j is 2×10^9. Set $I = 25$ users. We set the geographic locations of user to be evenly distributed within 1200 × 1200 in meter. For each user i, the applications arrive according to a Poisson distribution with the average inter-arrival time of T_i. The default value of T_i is 1.5 slots. The value of R_{ij} is set to be 2 and the value of E_i is set to be 3×10^6 for each i. Let d_{ij} represents the distance between user i and fog node j. If $d_{ij} < 1.1 \times (\max_{1 \le k \le I} (\min_{1 \le h \le J} d_{kh}))$, user i is a neighbor of fog node j. Then the set Ψ_i and Ω_{ij} can be determined for each user i and fog node j. Selected simulation results are reported as follows. The performance considered in this paper is the total delay which is the sum of the average communication delay and computing delay. Two different algorithms are considered in simulations: the first is the proposed delay based resource allocation algorithm, the second is the traditional queue length based resource allocation algorithm. The outline of the traditional queue length based scheduling algorithm can be found Sect. 3.1. Given the parameter configuration, the simulation experiment is repeated 100 times and then averaged as the final result.

Figure 2 shows the normalized value of the total delay (i.e., measured in slot time) with different R (i.e., the total number of subcarriers in the system) for different resource allocation algorithms. The curve with square represents the proposed delay based resource allocation algorithm and the curve with triangle represents the traditional queue length based resource allocation algorithm. We can observe that, as the value of R increases, the value of the delay increases. Specifically, for the proposed delay based resource allocation algorithm, when R increases from 20 to 26, the delay decreases from 1.95 to 1.50 slots. Further, it can be observed that, the proposed delay based resource allocation algorithm has better delay performance than that of the traditional queue length based one. Specifically, when $R = 22$, the delay of the queue length based resource allocation algorithm is 1.63 slots, while the delay of the proposed delay based resource allocation algorithm is 1.83 slots, with a drop of 12.45%. The main reason is that the proposed delay based resource allocation algorithm minimizes delay directly, while the traditional queue length based one does not. Therefore, the proposed delay based resource allocation algorithm has better delay performance that the traditional queue length based one.

Figure 3 shows the normalized value of the total delay (i.e., measured in slot time) for different values of $1/T_i$ (i.e., the average arrival rate of applications). In this experiment, we set R = 25. For the curves in the figure, we can observe that, with the increase of $1/T_i$ (i.e., with the decrease of T_i), the delay also increase. Specifically, for the delay based resource allocation algorithm, when the value of $1/T_i$ increases from 1/1.8 to 1/1.2 (i.e., T_i decreases from 1.8 to 1.2), the delay increases from 1.42 to 2.17 slots. Further, it can be observed that, the proposed delay based resource allocation algorithm can provide better delay performance than the traditional queue length based resource allocation algorithm. Specifically, when the value of $1/T_i$ is 1/1.2 (i.e., the value of T_i is 1.2), the delay of the traditional queue length based resource allocation algorithm is 2.46 slots, while the delay of the proposed delay based resource allocation algorithm is 2.17 slots, with a drop of 13.5%. The reason is also that the proposed delay based resource allocation algorithm minimizes delay directly, while the traditional queue length based resource allocation algorithm does not.

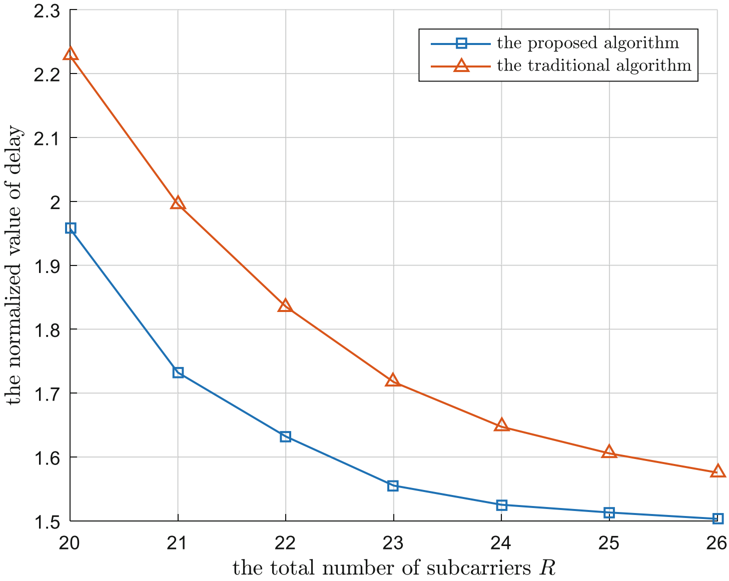

Fig. 2. Impact of the total number of subcarriers.

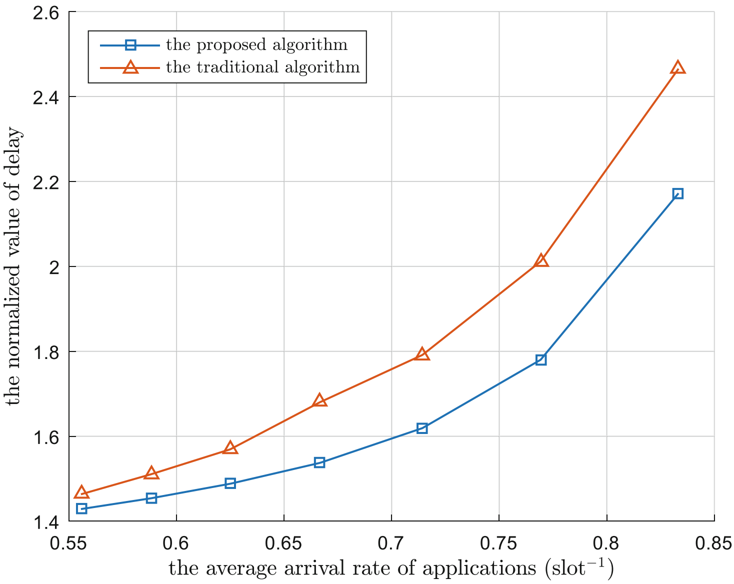

Fig. 3. Impact of the average arrival rate of applications.

5 Conclusions

In this work, the recursive expressions of the communication and computing delays in fog computing systems were derived in which the assumptions on the statistics of traffic is not needed at all. Using the framework of Lyapunov optimization, a novel delay based wireless scheduling and server assignment algorithm was proposed to stabilize the virtual queues of communication and computing delays. Simulation results were reported which showed that the average delay of the proposed delay based resource allocation algorithm can be 13.5% lower as compared to the traditional queue length based one.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 61571111 and No. 61771132) and the NFSC Incubation Project of NUPT (No. NY219106).

References

1. Chiang, M., Zhang, T.: Fog and IoT: an overview of research opportunities. IEEE Internet Things J. **3**(6), 854–864 (2016)
2. Aazam, M., Zeadally, S., Harras, K.: Fog computing architecture, evaluation, and future research directions. IEEE Commun. Mag. **56**(5), 46–52 (2018)
3. Bittencourt, L., Diaz-Montes, J., Buyya, R., Rana, O., Parashar, M.: Mobility-aware application scheduling in fog computing. IEEE Cloud Comput. **4**(2), 26–35 (2017)

4. Yang, Y., Wang, K., Zhang, G., Chen, X., Luo, X., Zhou, M.: MEETS: maximal energy efficient task scheduling in homogeneous fog networks. IEEE Internet Things J. **5**(5), 4076–4087 (2018)
5. Jiang, Y., Tsang, D.: Delay-aware task offloading in shared fog networks. IEEE Internet Things J. **5**(6), 4945–4956 (2018)
6. Rahman, S., Peng, M., Zhang, K., Chen, S.: Radio resource allocation for achieving ultra-low latency in fog radio access networks. IEEE Access **6**, 17442–17454 (2018)
7. Alameddine, H., Sharafeddine, S., Sebbah, S., Ayoubi, S., Assi, C.: Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. IEEE J. Sel. Areas Commun. **37**(3), 668–682 (2019)
8. Deng, R., Lu, R., Lai, C., Luan, T., Liang, H.: Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. IEEE Internet Things J. **3**(6), 1171–1181 (2016)
9. Zeng, D., Gu, L., Guo, S., Cheng, Z., Yu, S.: Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. IEEE Trans. Comput. **65**(12), 3702–3712 (2016)
10. Misra, S., Saha, N.: Detour: dynamic task offloading in software-defined fog for IoT applications. IEEE J. Sel. Areas Commun. **37**(5), 1159–1166 (2019)
11. Josilo, S., Dan, G.: Decentralized algorithm for randomized task allocation in fog computing systems. IEEE/ACM Trans. Netw. **27**(1), 85–97 (2019)
12. Zhao, S., Yang, Y., Shao, Z., Yang, X., Qian, H., Wang, C.: FEMOS: fog-enabled multitier operations scheduling in dynamic wireless networks. IEEE Internet Things J. **5**(2), 1169–1183 (2018)
13. Yang, Y., Zhao, S., Zhang, W., Chen, Y., Luo, X., Wang, J.: DEBTS: delay energy balanced task scheduling in homogeneous fog networks. IEEE Internet Things J. **5**(3), 2094–2106 (2018)
14. Deng, Y., Chen, Z., Zhang, D., Zhao, M.: Workload scheduling toward worst-case delay and optimal utility for single-hop fog-IoT architecture. IET Commun. **12**(17), 2164–2173 (2018)
15. Li, L., Guan, Q., Jin, L., Guo, M.: Resource allocation and task offloading for heterogeneous real-time tasks with uncertain duration time in a fog queueing system. IEEE Access **7**, 9912–9925 (2019)
16. Tassiulas, L., Ephremides, A.: Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Trans. Autom. Control **37**(12), 1936–1948 (1992)
17. Neely, M.: Stochastic Network Optimization with Application to Communication and Queueing Systems. Morgan & Claypool, San Rafael (2010)
18. Zhang, Y., Du, P., Wang, J., Ba, T., Ding, R., Xin, N.: Resource scheduling for delay minimization in multi-server cellular edge computing systems. IEEE Access **7**, 86265–86273 (2019)