sensors

MDPI

Article

Optimal Design of Hierarchical Cloud-Fog&Edge Computing Networks with Caching

Xiaoqian Fan [1], Haina Zheng [1,2,*], Ruihong Jiang [1,2] and Jinyu Zhang [1]

[1] School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China; xqfan1995@bjtu.edu.cn (X.F.); rhjiang@bjtu.edu.cn (R.J.); zjy@bjtu.edu.cn (J.Z.)
[2] State Key Lab of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: hnzheng@bjtu.edu.cn

Received: 10 February 2020; Accepted: 9 March 2020; Published: 12 March 2020

Abstract: This paper investigates the optimal design of a hierarchical cloud-fog&edge computing (FEC) network, which consists of three tiers, i.e., the cloud tier, the fog&edge tier, and the device tier. The device in the device tier processes its task via three computing modes, i.e., cache-assisted computing mode, cloud-assisted computing mode, and joint device-fog&edge computing mode. Specifically, the task corresponds to being completed via the content caching in the FEC tier, the computation offloading to the cloud tier, and the joint computing in the fog&edge and device tier, respectively. For such a system, an energy minimization problem is formulated by jointly optimizing the computing mode selection, the local computing ratio, the computation frequency, and the transmit power, while guaranteeing multiple system constraints, including the task completion deadline time, the achievable computation capability, and the achievable transmit power threshold. Since the problem is a mixed integer nonlinear programming problem, which is hard to solve with known standard methods, it is decomposed into three subproblems, and the optimal solution to each subproblem is derived. Then, an efficient optimal caching, cloud, and joint computing (CCJ) algorithm to solve the primary problem is proposed. Simulation results show that the system performance achieved by our proposed optimal design outperforms that achieved by the benchmark schemes. Moreover, the smaller the achievable transmit power threshold of the device, the more energy is saved. Besides, with the increment of the data size of the task, the lesser is the local computing ratio.

Keywords: fog&edge computing; cloud computing; content caching; computation offloading; energy minimization

1. Introduction

1.1. Background

With the rapid development of wireless communications technologies and the wide deployment of mass smart devices, a large number of emerging applications [1], such as artificial intelligence (AI) [2], augmented reality (AR) [3], and virtual reality (VR) [4], have been arising in Internet of Things (IoT) networks, which put forward higher requirements for computation capability and transmit power to the smart device in the IoT network [5]. As we all know, most smart devices usually have limited communication, computation, storage, and energy resources, which is a huge challenge to complete such computation-intensive and delay-sensitive applications [6–10].

To solve these problems, fog&edge computing (FEC) is regarded as a potential solution via providing computing service to smart devices on the edge of the network, which meets the requirements of the smart device for processing the computation-intensive and delay-sensitive tasks in real time [11–13]. The FEC has two advantages: (i) Compared with local computing by the device

itself [14], FEC enables the limited computation capabilities of the smart devices. (ii) Compared with cloud computing [15], FEC reduces the delay caused by long distances and traffic congestion for offloading to the cloud server.

Apart from computation offloading in the FEC, content caching is another promising technology to solve the limited computation capability of the device and reduce the transmission delay [16]. Significantly, an important part of the delay is caused via the redundant transmission and computation of a few popular files, such as the files for rendering scenes in the typical VR application scenario [17]. Therefore, caching popular content in the fog&edge tier is an effective way to avoid duplicate transmission and computation.

To face the challenge of completing the computation-intensive and delay-sensitive applications, integrating three such technologies, i.e., cloud computing, fog&edge computing, and content caching, into a single network system could bring strong performance improvement, which is of great significance.

1.2. Related Work

Over the past few years, a large number of research works has investigated cloud computing [18–20], fog&edge computing [21–29], and content caching [30–32]. However, most of these works involved these three technologies separately.

Then, the combination of two technologies began to be studied; see, e.g., [13,21,22,33–41]. Specifically, in [33], the authors investigated an energy efficiency maximization problem in a cloud-assisted FEC system. In [34], the authors proposed a cloud-assisted mobile edge computing (MEC) framework designed to guarantee user service quality with minimal system cost. In [35], the authors investigated a heterogeneous cloud-MEC two-tier offloading framework, and an computing offloading scheme was designed to minimize overall energy consumption. In [36], the authors proposed an integration framework of the cloud, MEC, and IoT to solve the scalability problem of MEC and designed a selective offloading scheme to achieve the minimum energy consumption of mobile devices while meeting the delay requirements. In [37], the authors investigated the optimal workload allocation problem in a fog-cloud computing system toward the minimal power consumption with constrained service delay. However, non of the above works involved content caching.

On the other hand, in [21], the authors studied joint service caching and task offloading for MEC-enabled dense cellular networks. In [22], the authors proposed a collaborative offloading scheme to cache the popular computation results to reduce the task execution delay. In [38], the authors investigated an optimization problem that considered offloading decisions, computing resources, and content caching. An alternative direction algorithm based on a multiplier was proposed to solve the maximize revenue problem. In [39], the authors proposed a joint caching and offloading mechanism to minimize the average total energy minimization problem. In [40,41], the authors investigated an energy minimization problem for a cache-aided FEC system. However, none of the above works involved cloud computing.

Recently, a few works began to study these three technologies in a single system to further improve system performance, i.e., in [42], the offloading and caching strategy was studied for a cloud-assisted FEC system to minimize delay, where however, it was not the aim to reduce the energy consumption. In [43], the offloading and caching decision was investigated for a hybrid cloud/edge computing system to minimize energy consumption, where however, only the binary decision was considered.

1.3. Motivation and Contributions

As mentioned above, there exist a few works that have studied these three technologies together, and to the best of our knowledge, no work has been done on the optimal design of a hierarchical cloud-FEC network with caching to minimize the energy consumption. Therefore, to explore the benefits of cloud computing, fog&edge computing, and content caching, we study the optimal design of a hierarchical cloud-FEC network with caching to minimize the energy consumption.

The main contributions of our work are summarized as follows.

- A three-tier network framework is considered, and correspondingly, we propose three computing modes to process the computation task of the device, i.e., cache-assisted computing mode, cloud-assisted computing mode, and joint device-fog&edge computing mode. Specifically, the task corresponds to being completed via the content caching in the FEC tier, the computation offloading to the cloud tier, and the joint computing in the fog&edge and device tier, respectively.
- For such a system, an energy minimization problem is formulated by jointly optimizing the computing mode selection, the local computing ratio, the computation frequency, and the transmit power, while guaranteeing multiple system constraints, including the task completion deadline time, the achievable computation capability, and the achievable transmit power threshold.
- Since the problem is a mixed integer nonlinear programming problem, which is hard to solve with known standard methods, it is decomposed into three subproblems, and the optimal solution to each subproblem is derived. Then, an efficient optimal caching, cloud, and joint computing (CCJ) algorithm to solve the primary problem is proposed.
- Simulation results show that the system performance achieved by our proposed optimal design outperforms that achieved by the benchmark schemes. Moreover, the smaller the achievable transmit power threshold of the device, the more energy is saved. Besides, with the increment of the data size of the task, the lesser is the local computing ratio.

The rest of this paper is organized as follows. Section 2 describes the system model, and the optimization problem is formulated. In Section 3, the closed-form and semi-closed-form solutions to the three subproblems are derived, and an efficient algorithm, i.e., the CCJ algorithm, is presented. Section 4 provides some simulation results, and finally, Section 5 summarizes this paper.

2. System Model and Problem Formulation

2.1. System Model

Consider a hierarchical cloud-FEC network as shown in Figure 1, which consists of three tiers, i.e., the cloud tier, the fog&edge tier, and the device tier. Specifically, in the device tier, the smart device generates K different computation tasks, which follow the uniform distribution of $\mathcal{K}$, where $\mathcal{K}$ denotes the set of task types with $\mathcal{K} \triangleq \{1, ..., K\}$. Each task is computed locally or offloaded. In the FEC tier, the FEC server is deployed at the base station (BS), which has the content caching and the computation capability to process the offloaded tasks. Besides, the BS is connected to the cloud server in the cloud tier via optical fiber. For any task k, for example, face recognition is a typical application scenario, which usually consists of five main computing components, including image acquisition, face detection, preprocessing, feature extraction, and classification. Image acquisition components can be executed on devices to support the user interface, but other complex computing components, such as signal processing and the machine learning (ML) algorithm, can be offloaded to the fog&edge computing or cloud computing to execute. Some of the components are cached in the FEC server in advance, which could reduce computing delay and energy consumption of the device.

We define task k as $\tau_k \triangleq \{a_k, C_k, d_k, T_k^{\max}\}$, where $a_k \in \{0, 1\}$ is the caching indicator. When $a_k = 1$, it indicates that the kth task has been cached in the FEC tier, and when $a_k = 0$, it indicates that the kth task has not been cached. C_k is the number of central processing unit (CPU) cycles required for computing one bit of the kth task; d_k is the data size of the kth task; and $T_k^{\max}$ is the completion deadline time of the kth task.

For the FEC tier, due to limited caching space, we assume that the FEC server only caches several of the most popular files. The popularity of the files follows a Zipf distribution. Therefore, the popularity of the kth task is described as:

$$z_k = \frac{\frac{1}{k^{\mu}}}{\sum_{k=1}^{K} \frac{1}{k^{\mu}}}, \tag{1}$$

where μ is the shape parameter and is regarded as constant [44,45]. For our considered system, denote Z as the caching threshold according to the popularity. When $z_k \geq Z$, the kth task is cached; otherwise, the task is not cached.

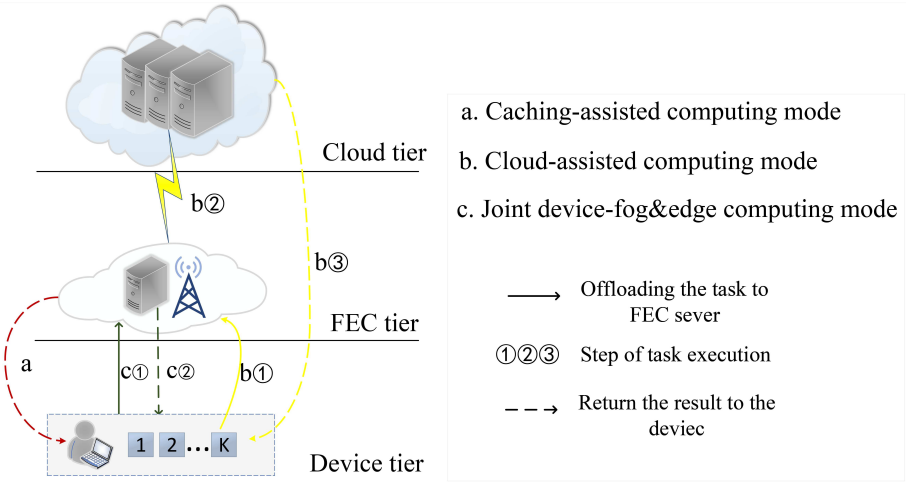


Figure 1. Illustration of the hierarchical cloud-fog&edge computing network.

For each task, the transmit protocol is shown in Figure 2. When the task is cached, it is processed in the caching computing mode.

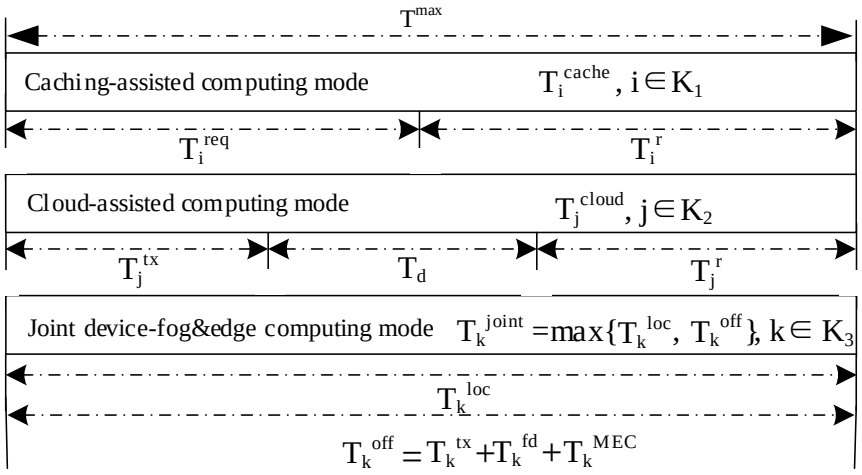


Figure 2. Illustration of the transmit protocol for a given time block.

2.1.1. Caching-Assisted Computing Mode

For the caching-assisted computing mode, the delay includes two parts. One is the task request time T_k^{req}, which is too small to be ignored. The other is the result feedback time T_k^{r}, which depends on the data size of the results. The delay of task k in the caching-assisted computing mode is given by:

$$T_k^{\text{cache}} = T_k^{\text{req}} + T_k^{\text{r}} \approx \frac{\delta d_k}{B \log_2(1 + \frac{p^{\text{FEC}}|h|^2}{\sigma^2})}. \tag{2}$$

where δ is the data ratio of the results. $|h|^2$, p^{FEC}, f^{FEC}, B, and σ^2 are the channel coefficient between the device and the FEC tier, the transmit power and the computing capability of the FEC server, and the system bandwidth and the noise power, respectively. The energy consumption of device for task k in the caching-assisted computing mode is given by:

$$E_k^{\text{cache}} = p^{\text{c}} T_k^{\text{cache}}, \tag{3}$$

where p^{c} is the circuit power of the device for waiting.

When the task is not cached, it is processed in the joint device-fog&edge computing mode or the cloud-assisted computing mode. Furthermore, we define $\gamma_k \in \{0, 1\}$ as the uncached task execution decision, where $\gamma_k = 1$ indicates that the cloud-assisted computing mode is selected; otherwise, the joint device-fog&edge computing mode is selected.

2.1.2. Cloud-Assisted Computing Mode

Consider a cloud tier with a strong enough computation capability, so the execute time in the cloud tier can be neglected. For the cloud-assisted computing mode, the delay includes three parts. One is the task transmission time between the device and the FEC tier T_k^{ts}. One is the task transmission time between the FEC tier and the cloud tier T_d, which depends on the distance between the FEC tier and the cloud tier and is regarded as a constant in this work. The other one is the result feedback time T_k^{r}. Therefore, the total delay of task k is given by:

$$T_k^{\text{cloud}} = T_k^{\text{tx}} + T_d + T_k^{\text{r}} = \frac{d_k}{B \log_2(1 + \frac{p_k^{\text{tx}}|h|^2}{\sigma^2})} + T_d + \frac{\delta d_k}{B \log_2(1 + \frac{p^{\text{FEC}}|h|^2}{\sigma^2})}. \tag{4}$$

Meanwhile, the energy consumption of the device for task k is given by:

$$E_k^{\text{cloud}} = p_k^{\text{tx}} T_k^{\text{tx}} + p^{\text{c}} T_k^{\text{cloud}}, \tag{5}$$

where p_k^{tx} is the transmit power of the device for task k.

2.1.3. Joint Device-Fog&Edge Computing Mode

For joint device-fog&edge computing mode, the device portions each task into two parts. One part executes by local computing. The other one executes by offloading to the FEC tier for computing.

- Local execution

  According to most existing related works, to achieve minimal energy consumption, an identical CPU frequency should be adopted for each CPU cycle. Thus, we denote f_k^{loc} as the average computation frequency of the device for each bit of the kth task. Therefore, the execution time of task k is given by:

$$T_k^{\text{loc}} = \frac{\beta_k C_k d_k}{f_k^{\text{loc}}}, \tag{6}$$

where $\beta_k \in [0,1]$ is the ratio of task k for local execution at the device and $(1-\beta_k)$ represents the offloading ratio of task k for FEC execution.

The energy consumption of the device for task k is given by:

$$E_k^{\text{loc}} = \kappa f_k^{\text{loc3}} T_k^{\text{loc}} = \kappa f_k^{\text{loc2}} \beta_k C_k d_k, \tag{7}$$

where κ is the effective switched capacitor depending on the chip architecture.

- FEC execution

The FEC execution delay includes three parts. The first one is task offloading time T_k^{tx}. The last one is FEC execution time T_k^{FEC}. The other is the result feedback time T_k^{fd}. Thus, the delay of FEC execution for task k is given by:

$$\begin{aligned} T_k^{\text{off}} &= T_k^{\text{tx}} + T_k^{\text{FEC}} + T_k^{\text{fd}} \\ &= \frac{(1-\beta_k)d_k}{B\log_2(1+\frac{p_k^{\text{tx}}|h|^2}{\sigma^2})} + \frac{(1-\beta_k)C_k d_k}{f^{\text{FEC}}} + \frac{\delta(1-\beta_k)d_k}{B\log_2(1+\frac{p^{\text{FEC}}|h|^2}{\sigma^2})}. \end{aligned} \tag{8}$$

The energy consumption of the device in FEC execution for task k is given by:

$$E_k^{\text{off}} = p_k^{\text{tx}} T_k^{\text{tx}} + p^{\text{c}} T_k^{\text{off}}. \tag{9}$$

As a result, the total delay of task k in the joint device-fog&edge computing mode is:

$$T_k^{\text{joint}} = \max\left\{T_k^{\text{loc}}, T_k^{\text{off}}\right\}, \tag{10}$$

and the energy consumption of the device for task k in this mode is given by:

$$E_k^{\text{joint}} = E_k^{\text{loc}} + E_k^{\text{off}}. \tag{11}$$

Denote $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{K}_3$ as the set in the caching-assisted computing, cloud-assisted computing, and joint device-fog&edge computing mode, respectively. $|\mathcal{K}_1|$, $|\mathcal{K}_2|$, and $|\mathcal{K}_3|$ are the element number of $\mathcal{K}_1$, $\mathcal{K}_2$, and $\mathcal{K}_3$, respectively. Then, the average energy consumption of the device is given by:

$$E^{\text{ave}} = \frac{\sum_k^{\mathcal{K}_1} E_k^{\text{cache}} + \gamma_k \sum_k^{\mathcal{K}_2,\mathcal{K}_3} E_k^{\text{cloud}} + (1-\gamma_k)\sum_k^{\mathcal{K}_2,\mathcal{K}_3} E_k^{\text{joint}}}{|\mathcal{K}_1| + |\mathcal{K}_2| + |\mathcal{K}_3|}, \mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_3 = \mathcal{K}. \tag{12}$$

2.2. Problem Formulation

Our goal is to minimize the average energy consumption of the device in the hierarchical cloud-FEC system. Mathematically, the average energy minimization problem is formulated as:

$$\mathbf{P}_0: \min_{\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{p}^{\mathrm{tx}},\boldsymbol{f}^{\mathrm{loc}}} \quad E^{\mathrm{ave}} \tag{13}$$

$$\text{s.t. } T_i^{\mathrm{cache}} \le T_i^{\max}, \forall i \in \mathcal{K}_1, \tag{13a}$$

$$T_j^{\mathrm{cloud}} \le T_j^{\max}, \forall j \in \mathcal{K}_2, \tag{13b}$$

$$T_k^{\mathrm{joint}} \le T_k^{\max}, \forall k \in \mathcal{K}_3, \tag{13c}$$

$$0 \le f_k^{\mathrm{loc}} \le f^{\max}, \forall k \in \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \tag{13d}$$

$$0 \le p_k^{\mathrm{tx}} \le p^{\max}, \forall k \in \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3, \tag{13e}$$

$$\beta_k \in [0,1], \forall k \in \mathcal{K}_3, \tag{13f}$$

$$\gamma_k \in \{0,1\}, \forall k \in \mathcal{K}_2, \mathcal{K}_3, \tag{13g}$$

where $\boldsymbol{\beta} \triangleq [\beta_1, \beta_2, ..., \beta_K]^T$, $\boldsymbol{\gamma} \triangleq [\gamma_1, \gamma_2, ..., \gamma_K]^T$, $\boldsymbol{p}^{\mathrm{tx}} \triangleq [p_1^{\mathrm{tx}}, p_2^{\mathrm{tx}}, ..., p_K^{\mathrm{tx}}]^T$, and $\boldsymbol{f}^{\mathrm{loc}} \triangleq [f_1^{\mathrm{loc}}, f_2^{\mathrm{loc}}, ..., f_K^{\mathrm{loc}}]^T$ denote the local computing ratio, the computing mode selection, the transmit power, and the computation frequency of the device, respectively. $f^{\max}$ and $p^{\max}$ denote the maximal achievable computation frequency and transmit power of the device, respectively. Constraints (13a), (13b), and (13c) mean that the delay in the three computing modes cannot exceed the completion deadline time, respectively. Constraints (13d) and (13e) represent the computation capability constraint and transmit power of the device, respectively.

3. Optimal Solution Approach

In this section, in order to solve Problem $\mathbf{P}_0$, we shall first decompose it into three subproblems. Then, by respectively solving them, the optimal solution to Problem $\mathbf{P}_0$ is derived.

3.1. Optimization of the Caching-Assisted Computing Mode

As mentioned above, when $z_i \ge Z$, $i \in \mathcal{K}_1$ and the caching-assisted computing mode is employed. In this mode, the optimal energy consumption is:

$$E_i^{\mathrm{cache}} = p^c T_i^{\mathrm{r}} = \frac{p^c \delta d_i}{B \log_2(1 + \frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2})}. \tag{14}$$

Proposition 1. *When p^{FEC} has its maximal achievable threshold, E_i^{cache} achieves the optimal value.*

Proof of Proposition 1. The larger p^{FEC} is, the smaller T_i^{r} is, and the smaller E_i^{cache} is. Therefore, when p^{FEC} is with its maximal achievable threshold, the energy consumption of the device reaches its minimum value. Therefore, the optimal E_i^{cache} can be obtained. Thus, Proposition 1 is proven. □

3.2. Optimization of the Cloud-Assisted Computing Mode

When $z_j < Z$ and $\gamma_j = 1$, $j \in \mathcal{K}_2$ and the cloud-assisted computing mode is employed. In this mode, the optimal problem is expressed as:

$$\mathbf{P}_1: \min_{p_j^{\mathrm{tx}}} E_j^{\mathrm{cloud}} \tag{15}$$

$$\text{s.t. } T_j^{\mathrm{cloud}} \le T_j^{\max}, \tag{15a}$$

$$0 \le p_j^{\mathrm{tx}} \le p^{\max}. \tag{15b}$$

By expanding the expressions of the variables of Problem $\mathbf{P}_1$, it is equivalently rewritten as:

$$\mathbf{P}_{1\_A}: \min_{p_j^{\mathrm{tx}}} \; p^{\mathrm{c}}\left(\frac{d_j}{B\log_2\left(1+\frac{p_j^{\mathrm{tx}}|h|^2}{\sigma^2}\right)}+T_d+\frac{\delta d_j}{B\log_2\left(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2}\right)}\right)+p_j^{\mathrm{tx}}\frac{d_j}{B\log_2\left(1+\frac{p_j^{\mathrm{tx}}|h|^2}{\sigma^2}\right)} \tag{16}$$

$$\text{s.t.} \;\; \frac{d_j}{B\log_2\left(1+\frac{p_j^{\mathrm{tx}}|h|^2}{\sigma^2}\right)}+T_d+\frac{\delta d_k}{B\log_2\left(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2}\right)}\leq T_j^{\max}, \tag{16a}$$

$$0\leq p_j^{\mathrm{tx}}\leq p^{\max}. \tag{16b}$$

Lemma 1. *Problem $\boldsymbol{P}_{1\_A}$ is a convex optimization problem.*

Proof of Lemma 1. Denote $f(p_j^{\mathrm{tx}}) = p^{\mathrm{c}}\left(\frac{d_j}{B\log_2\left(1+\frac{p_j^{\mathrm{tx}}|h|^2}{\sigma^2}\right)}+T_d+\frac{\delta d_j}{B\log_2\left(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2}\right)}\right)$. The second order derivative of $f(p_j^{\mathrm{tx}})$ is always larger than zero, so the objective function of Problem $\mathbf{P}_{1\_A}$ is convex. The first constraint is rewritten as $g(p_j^{\mathrm{tx}})\leq 0$, i.e., $\frac{d_j}{B\log_2\left(1+\frac{p_j^{\mathrm{tx}}|h|^2}{\sigma^2}\right)}+\frac{\delta d_j}{B\log_2\left(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2}\right)}+T_{\mathrm{d}}-T_j^{\max}\leq 0$, and its second order derivative is also always larger than zero. Therefore, the first constraint is also convex. Therefore, Problem $\mathbf{P}_{1\_A}$ is a convex optimization problem. Lemma 1 is proven. □

With Lemma 1 and the derivative of $g(p_j^{\mathrm{tx}})=0$, the optimal solution to Problem $\mathbf{P}_{1\_A}$ is that $p_j^{\mathrm{tx}*}=\min\left\{\frac{(Np^{\mathrm{c}}-1)\mathcal{W}(0,exp(-1)(Np^{\mathrm{c}}-1))-1}{N}, p^{\max}\right\}$, where $N=\frac{|h|^2}{\sigma^2}$.

3.3. Optimization of the Joint Device-Fog&Edge Computing Mode

When $z_k < Z$ and $\gamma_k = 0, k\in\mathcal{K}_3$ and the joint device-fog&edge computing mode is employed. The optimization problem can be expressed as:

$$\mathbf{P}_2: \min_{\beta_k, p_k^{\mathrm{tx}}, f_k^{\mathrm{loc}}} \; E_k^{\mathrm{loc}}+E_k^{\mathrm{off}} \tag{17}$$

$$\text{s.t.} \;\; T_k^{\mathrm{joint}}\leq T_k^{\max}, \tag{17a}$$

$$0\leq f_k^{\mathrm{loc}}\leq f^{\max}, \tag{17b}$$

$$0\leq p_k^{\mathrm{tx}}\leq p^{\max}, \tag{17c}$$

$$\beta_k\in[0,1]. \tag{17d}$$

We design an alternating iteration method to solve Problem $\mathbf{P}_2$. Firstly, we fix β_k, and the primal Problem $\mathbf{P}_2$ becomes a sub-problem in terms of p_k^{tx} and f_k^{loc}. Then, we substitute the optimal values of p_k^{tx} and f_k^{loc} into Problem $\mathbf{P}_2$, and Problem $\mathbf{P}_2$ is reformulated as a subproblem in terms of β_k. Hence, Problem $\mathbf{P}_2$ is divided into two sub-problems as follows, i.e., Problem $\mathbf{P}_3$ w.r.t. the transmit power p_k^{tx} and computation frequency f_k^{loc} and Problem $\mathbf{P}_4$ w.r.t. the local computing ratio β_k.

Let $\beta_k^{(0)}$ be the feasible point to Problem $\mathbf{P}_2$. Problem $\mathbf{P}_2$ is re-expressed as:

$$\mathbf{P}_3: \min_{p_k^{\mathrm{tx}}, f_k^{\mathrm{loc}}} \; E_k^{\mathrm{loc}}+E_k^{\mathrm{off}} \tag{18}$$

$$\text{s.t.} \; T_k^{\mathrm{loc}}\leq T_k^{\max}, \tag{18a}$$

$$T_k^{\mathrm{off}}\leq T_k^{\max}, \tag{18b}$$

$$0\leq f_k^{\mathrm{loc}}\leq f^{\max}, \tag{18c}$$

$$0\leq p_k^{\mathrm{tx}}\leq p^{\max}. \tag{18d}$$

To solve Problem $\mathbf{P}_3$, we expand the expressions of the variables of Problem $\mathbf{P}_3$ to be:

$$\mathbf{P}_{3\_A}: \min_{p_k^{\mathrm{tx}}, f_k^{\mathrm{loc}}} \kappa f_k^{\mathrm{loc2}} \beta_k^{(0)} C_k d_k + \frac{p_k^{\mathrm{tx}}(1-\beta_k^{(0)})d_k}{B\log_2(1+\frac{p_k^{\mathrm{tx}}|h|^2}{\sigma^2})} + p^{\mathrm{c}}\Big(\frac{(1-\beta_k^{(0)})d_k}{B\log_2(1+\frac{p_k^{\mathrm{tx}}|h|^2}{\sigma^2})} + \frac{(1-\beta_k^{(0)})C_k d_k}{f^{\mathrm{FEC}}} + \frac{\delta(1-\beta_k^{(0)})d_k}{B\log_2(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2})}\Big) \tag{19}$$

$$\text{s.t. } \frac{\beta_k^{(0)} C_k d_k}{f_k^{\mathrm{loc}}} \le T_k^{\max}, \tag{19a}$$

$$\frac{(1-\beta_k^{(0)})d_k}{B\log_2(1+\frac{p_k^{\mathrm{tx}}|h|^2}{\sigma^2})} + \frac{(1-\beta_k^{(0)})C_k d_k}{f^{\mathrm{FEC}}} + \frac{\delta(1-\beta_k^{(0)})d_k}{B\log_2(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2})} \le T_k^{\max}, \tag{19b}$$

$$0 \le f_k^{\mathrm{loc}} \le f^{\max}, \tag{19c}$$

$$0 \le p_k^{\mathrm{tx}} \le p^{\max}. \tag{19d}$$

Lemma 2. *Problem $\boldsymbol{P}_{3\_A}$ is a convex optimization problem.*

Proof of Lemma 2. Let the objective function as $\beta_k^{(0)} h_1(f_k^{\mathrm{loc}}) + (1-\beta_k^{(0)}) h_2(p_k^{\mathrm{tx}})$, where $h_1(f_k^{\mathrm{loc}}) = \kappa f_k^{\mathrm{loc2}} C_k d_k$ and $h_2(p_k^{\mathrm{tx}}) = p_k^{\mathrm{tx}} \frac{d_k}{B\log_2(1+\frac{p_k^{\mathrm{tx}}|h|^2}{\sigma^2})} + p^{\mathrm{c}}(\frac{d_k}{B\log_2(1+\frac{p_k^{\mathrm{tx}}|h|^2}{\sigma^2})} + \frac{C_k d_k}{f^{\mathrm{FEC}}} + \frac{\delta d_k}{B\log_2(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2})})$. The second order derivatives of $h_1(f_k^{\mathrm{loc}})$ and $h_2(p_k^{\mathrm{tx}})$ are respectively given by:

$$\frac{\partial^2 h_1(f_k^{\mathrm{loc}})}{\partial f_k^{\mathrm{loc2}}} > 0, \frac{\partial^2 h_2(p_k^{\mathrm{tx}})}{\partial p_k^{\mathrm{tx2}}} > 0, \tag{20}$$

which means that the objective function is convex. The first constraint is re-written as $\beta_k^{(0)} h_3(f_k^{\mathrm{loc}}) + (1-\beta_k^{(0)})\{h_4(p_k^{\mathrm{tx}}) + \frac{d_k}{B\log_2(1+\frac{p_k^{\mathrm{tx}}|h|^2}{\sigma^2})} + \frac{C_k d_k}{f^{\mathrm{FEC}}} + \frac{\delta d_k}{B\log_2(1+\frac{p^{\mathrm{FEC}}|h|^2}{\sigma^2})} - T_k^{\max}\} \le 0$. The second order derivatives of $h_3(f_k^{\mathrm{loc}})$ and $h_4(p_k^{\mathrm{tx}})$ are respectively derived as:

$$\frac{\partial^2 h_3(f_k^{\mathrm{loc}})}{\partial f_k^{\mathrm{loc2}}} > 0, \frac{\partial^2 h_4(p_k^{\mathrm{tx}})}{\partial p_k^{\mathrm{tx2}}} > 0. \tag{21}$$

Therefore, the constraint is also convex. Lemma 2 is proven. □

Lemma 2 indicates that Problem $\mathbf{P}_3$ is a joint convex optimization problem w.r.t. f_k^{loc} and p_k^{tx}, which can be solved by using some standard convex optimization tools, such as CVX. According to Proposition 1, when $p^{\mathrm{FEC}} = p^{\max}$ and $f^{\mathrm{FEC}} = f^{\max}$, the optimal E_k^{joint} can be achieved.

By substituting the optimal solution of f_k^{loc} and p_k^{tx} into Problem $\mathbf{P}_3$ to get $E_k^{\mathrm{loc}(0)}$ and $E_k^{\mathrm{off}(0)}$, we have:

$$\mathbf{P}_4: \min_{\beta_k} \beta_k E_k^{\mathrm{loc}(0)} + (1-\beta_k) E_k^{\mathrm{off}(0)} \tag{22}$$

$$\text{s.t. } T_k^{\mathrm{joint}} \le T_k^{\max}, \tag{22a}$$

$$\beta_k \in [0,1]. \tag{22b}$$

Since the objective function of Problem $\mathbf{P}_4$ is linear w.r.t. β_k, it is solved by several well-studied method.

Algorithm 1 Optimal caching, cloud, and joint computing (CCJ) algorithm.

Initialize $C_i, d_i, p^{\mathrm{c}}, \mathcal{K}_1 = 0, \mathcal{K}_2 = 0, \mathcal{K}_3 = 0$, and other known parameters;
for $i = 1 : |\mathcal{K}|$ **do**
  if $a_k = 1$ **then**
    Calculate E_k^{cache} according to (14);
    $k \rightarrow \mathcal{K}_1$;
  else
    Calculate E_k^{cloud} according to (15);
    Calculate E_k^{joint} according to (17);
    if $E_k^{\mathrm{cloud}} < E_k^{\mathrm{joint}}$ **then**
      $\gamma_k = 1$;
      $k \rightarrow \mathcal{K}_2$;
    else
      $\gamma_k = 0$;
      $k \rightarrow \mathcal{K}_3$;
    end if
  end if
end for
Calculate $\sum_i^{\mathcal{K}_1} E_i^{\mathrm{cache}}, \sum_j^{\mathcal{K}_2} E_j^{\mathrm{cloud}}, \sum_i^{\mathcal{K}_3} E_k^{\mathrm{joint}}$
Calculate E^{ave} according to (13);

With the closed-form or well-structured solutions to the cloud-assisted computing mode and the joint device-fog&edge computing mode in Sections 3.2 and 3.3, the minimal energy consumption (i.e., E_j^{cloud} and E_k^{joint}) can be calculated. Therefore, for uncached task k, $\forall k \in \mathcal{K}_2, \mathcal{K}_3$, the computing mode selection can be determined by:

$$\gamma_k = \begin{cases} 0, & \text{if} \quad E_k^{\mathrm{cloud}} > E_k^{\mathrm{joint}}, \\ 1, & \text{otherwise.} \end{cases}$$

In order to show our proposed algorithm clearly, i.e, the optimal caching, cloud, and joint computing (CCJ) algorithm, we summarize it as shown in Algorithm 1. It is able to converge to the global optimal solution with low computational complexity.

4. Simulation Results

4.1. Simulation Setup

In this section, we present some numerical results to discuss the performance of the hierarchical cloud-FEC system. We considered a centralized FEC network covered by a 200 m $\times$ 200 m area, where the BS was connected to the cloud server via optical fiber. In the device tier, the number of tasks requested by the device was $K = 10$. The input data size of the task was randomly distributed within $[100, 1000]$ MB, and the data ratio of the result δ was 0.1. The corresponding number of required CPU cycles was distributed within $[0.2, 1]$ G-cycles. The maximum achievable transmit power of the device was set as $p^{\max}$ = 0.1 W. The circuit power of the device was p^{c} = 0.01 W. In the FEC tier, the maximum achievable transmit power and computation capability of the FEC server was p^{FEC} = 1 W and $f^{\mathrm{FEC}} = 5$ G-cycles, respectively. In the cloud tier, since the calculation delay in the cloud server was ignored, we set T_d = 0.2 s, which was the transmission delay regarding the distance between the FEC tier and the cloud tier. In terms of communication, the system bandwidth was set as B = 3 MHz, and the white Gaussian noise was set to be $\sigma^2 = 10^{-8}$ W [29]. In addition, the channel gain was

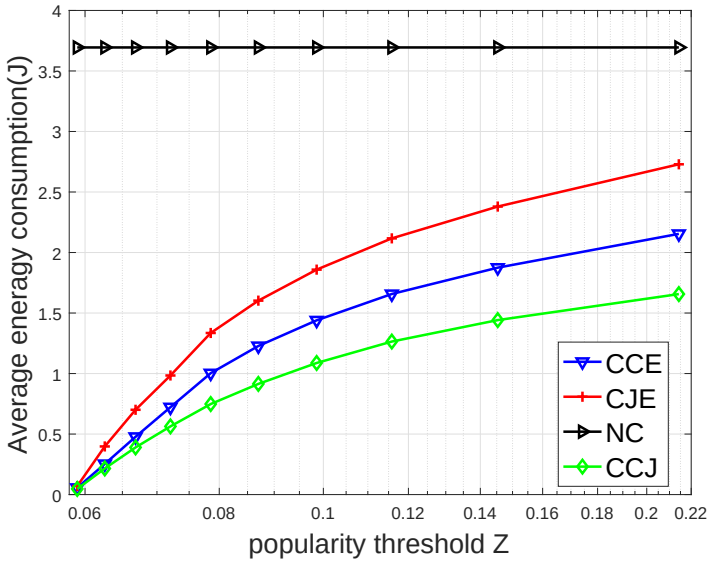
modeled by $h = 127 + 30 \times \log d$ with independent Rayleigh fading, where d is the distance between the device and the FEC server. According to the realistic measurements in [46], we set the effective switched capacitor $\kappa = 10^{-26}$. In the caching policy, we set the shape parameter μ = 0.56 and the caching threshold Z = 0.16. In this paper, all experiments were implemented in MathWorks MATLAB R2016b on a laptop equipped with a 12.00 GHz Corei5-3337U CPU and 128 GB random access memory. Every point in the figures was the result averaged over 10^4 independent channel realizations.

We compared our proposed algorithm with three different benchmark schemes as follows:

- No caching (NC) scheme: This scheme supposed that the FEC system did not have a cache function. Therefore, the task could only be executed through cloud computing mode or joint computing mode.
- Caching and joint execution (CJE) scheme: This scheme used our proposed cache policy. For the uncached task, it could be processed by the joint computing mode, that is $\gamma = 0$.
- Caching and cloud execution (CCE) scheme: This scheme used our proposed cache policy. For the uncached task, it could be processed by the cloud computing mode, that is $\gamma = 1$.

4.2. Experimental Results

Figure 3 compares the average energy consumption versus different caching popularity thresholds. It is seen that with the increment of Z, the average energy consumption of the device increased. The reason is that the larger the Z, the more the task was cached and processed in the FEC tier. The energy consumption was mainly caused by the circuit consumption of the device during waiting for the FEC server to execute the task and return the results to the device.



.

Figure 3. Average energy consumption on four schemes versus the caching popularity threshold

Figure 4 shows the average energy consumption versus different maximal achievable transmit powers of the device. It is seen that with $p^{\max}$ increasing, the average energy consumption of the device decreased and finally tended to be stable. The reason was that the higher the transmit power of the device, the faster the transmission rate, and the less the transmission time, the lower the energy consumption of the device. When $p^{\max}$ was relatively small, the optimal solution of the transmit power was on the boundary, i.e., $p^{\max}$. When $p^{\max}$ reached a certain value, the optimal solution of the transmit power shall not change.

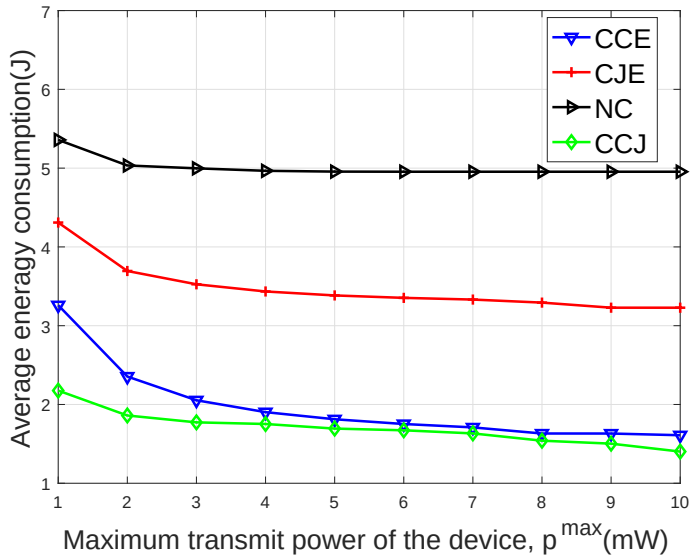


Figure 4. Average energy consumption of the four schemes versus the maximum achievable transmit power of device $p^{\max}$.

Figure 5 compares the average energy consumption versus different system bandwidths. It is seen that with the bandwidth increasing, the average energy consumption decreased. The reason may be that the larger the system bandwidth, the larger the transmission rate, which resulted in less delay and lower energy consumption.

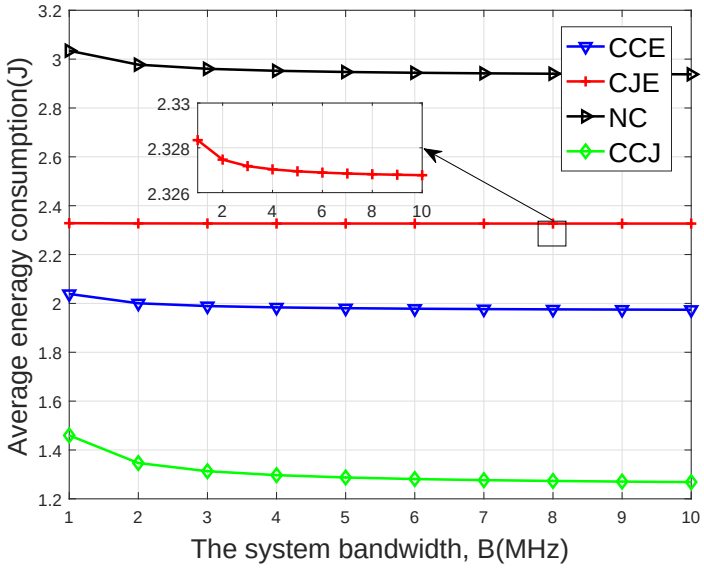


Figure 5. Average energy consumption on the four schemes versus the system bandwidth B.

Figure 6 compares the average energy consumption versus different task data sizes. It is seen that with the data size increasing, the average energy consumption increased. The reason may be that the larger the data size of the task, the larger the transmission delay and the calculation delay, which led to the greater energy consumption of the device.

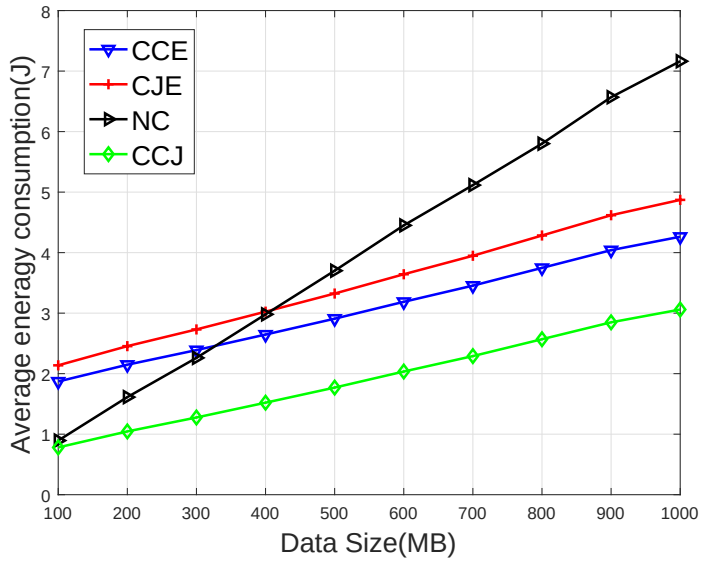


Figure 6. Average energy consumption on the four schemes versus the data size d_k.

Figure 7 compares the local computing ratio versus different task data sizes. It is seen that with the increment of the data size of the task, the local computing ratio decreased. The reason was that when the data size of the task was small, it was computed locally with less energy consumption compared with offloading. When the data size of the task was large, the computation capacity of the device was not enough to support the calculation, and more parts of the task should be offloaded to the FEC tier for computing.

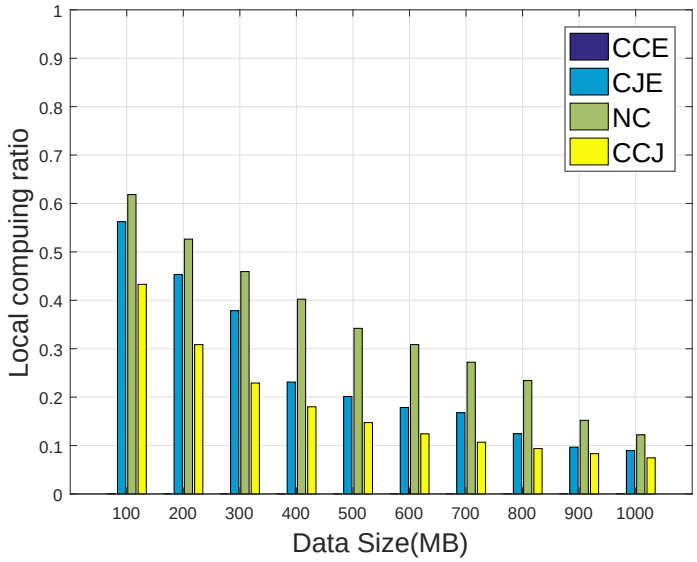


Figure 7. Average energy consumption on the four schemes versus the data size d_k.

5. Conclusions

This paper studied the optimal design of a hierarchical cloud-FEC network with caching. For such a system, an energy minimization problem was formulated by jointly optimizing the computing mode selection, the local computing ratio, the computation frequency, and the transmit power of the device, while guaranteeing multiple system constraints, including the task completion deadline time, the

achievable computation capability, and the achievable transmit power threshold of the device.Since the problem was a mixed integer nonlinear programming problem, which was hard to solve, it was decomposed into three subproblems, and the optimal solution for each subproblem was derived. Then, an efficient CCJ algorithm to solve the primary problem was designed. Simulation results showed that the system performance achieved by our proposed optimal design outperformed that achieved by the benchmark schemes. Specifically, compared with the NC scheme, the energy consumption reduced by our proposed optimal design by about 56%. Compared with the CJE scheme, the energy consumption reduced by our proposed optimal design by about 44%. Compared with the CCE scheme, the energy consumption reduced by our proposed optimal design by about 5%. Moreover, the smaller the achievable transmit power threshold of the device, the more energy was saved. Besides, with the increment of the data size of the task, the lesser was the local computing ratio.

Author Contributions: X.F. and H.Z. had an equal contribution to this work on the system modeling and methodology; R.J. and J.Z. contributed to the review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fundamental Research Funds for the Central Universities (No. 2019YJS035 and No. 2018YJS197) and in part by the General Program of the National Natural Science Foundation of China (No. 61071077).

Acknowledgments: We could like to thank all the reviewers for their constructive comments and helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
FEC	Fog&edge computing
AI	Artificial intelligence
ML	Machine learning
AR	Augmented reality
VR	Virtual reality
CPU	Central processing unit
BS	Base station
CCJ	Caching cloud joint
NC	No caching
CCE	Caching cloud execution
CJE	Caching joint execution

References

1. Qin, M.; Chen, L.; Zhao, N.; Chen, Y.; Yu, F.R.; Wei, G. Power-constrained edge computing with maximum processing capacity for IoT networks. *IEEE Internet Things J.* **2018**, *6*, 4330–4343. [CrossRef]
2. Wang, L.; Jiao, L.; Li, J.; Gedeon, J. Moera: Mobility-agnostic online resource allocation for edge computing. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1843–1856. [CrossRef]
3. Dong, Y.; Guo, S.; Liu, J.; Yang, Y. Energy-efficient fair cooperation fog computing in mobile edge networks for smart city. *IEEE Internet Things J.* **2019**, *6*, 7543–7554. [CrossRef]
4. Mehrabi, A.; Siekkinen, M.; Ylä-Jääski, A. Edge computing assisted adaptive mobile video streaming. *IEEE Trans. Mob. Comput.* **2018**, *18*, 787–800. [CrossRef]
5. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update*; 2016–2021 White Paper; Cisco: San Jose, CA, USA, 2017.
6. Wang, T.; Lu, Y.; Cao, Z.; Lei, S.; Zheng, X.; Liu, A.; Xie, M. When Sensor-Cloud Meets Mobile Edge Computing. *Sensors* **2019**, *19*, 5324. [CrossRef]
7. Zheng, H.; Xiong, K.; Fan, P.; Zhou, L.; Zhong, Z. SWIPT-aware fog information processing: Local computing vs. fog offloading. *Sensors* **2018**, *18*, 3291. [CrossRef]

8. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K. A survey on mobile edge computing: the communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
9. Jeong, H.J. Lightweight Offloading System for Mobile Edge Computing. In Proceedings of the IEEE PerCom Workshops, Kyoto, Japan, 11–15 March 2019; pp. 451–452.
10. Xiong, K.; Chen, C.; Qu, G.; Fan, P.; Letaief, K.B. Group cooperation with optimal resource allocation in wireless powered communication networks. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 3840–3853. [CrossRef]
11. Cui, T.; Hu, Y.; Shen, B.; Chen, Q. Task Offloading Based on Lyapunov Optimization for MEC-Assisted Vehicular Platooning Networks. *Sensors* **2019**, *19*, 4974. [CrossRef]
12. Wang, P.; Yao, C.; Zheng, Z.; Sun, G.; Song, L. Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems. *IEEE Internet Things J.* **2019**, *6*, 2872–2884. [CrossRef]
13. Ren, J.; Yu, G.; Yu, G.; He, Y.; Li, G.Y. Collaborative cloud and edge computing for latency minimization. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5031–5044. [CrossRef]
14. Neto, J.L.D.; Yu, S.Y.; Macedo, D.F.; Nogueira, J.M.S.; Langar, R.; Secci, S. ULOOF: A user level online offloading framework for mobile edge computing. *IEEE Trans. Mob. Comput.* **2018**, *17*, 2660–2674. [CrossRef]
15. Mian, G.; Li, L.; Guan, Q. Energy-efficient and delay-guaranteed workload allocation in IoT-edge-cloud computing systems. *IEEE Access* **2019**, *7*, 3336–3347.
16. Wei, H.; Luo, H.; Sun, Y. Mobility-Aware Service Caching in Mobile Edge Computing for Internet of Things. *Sensors* **2020**, *20*, 610. [CrossRef] [PubMed]
17. Liu, X.; Sun, C.; Zhang, X. Context-aware caching with social behavior in MEC-enabled wireless cellular networks. In Proceedings of the IEEE PerCom Workshops, Kyoto, Japan, 11–15 March 2019; pp. 1004–1008.
18. Zhou, B.; Dastjerdi, A.V.; Calheiros, R.N.; Srirama, S.N.; Buyya, R. Mcloud: A context-aware offloading framework for heterogeneous mobile cloud. *IEEE Trans. Serv. Comput.* **2017**, *10*, 797–810. [CrossRef]
19. Mahmoodi, S.E.; Uma, R.N.; Subbalakshmi, K.P. Optimal joint scheduling and cloud offloading for mobile applications. *IEEE Trans. Cloud Comput.* **2019**, *7*, 301–313. [CrossRef]
20. Misra, S.; Wolfinger, B.E.; Achuthananda, M.P.; Chakraborty, T.; Das, S.N.; Das, S. Auction-Based Optimal Task Offloading in Mobile Cloud Computing. *IEEE Syst. J.* **2019**, *13*, 2978–2985. [CrossRef]
21. Xu, J.; Chen, L.; Zhou, P. Joint service caching and task offloading for mobile edge computing in dense networks. In Proceedings of the IEEE INFOCOM, Honolulu, HI, USA, 16–19 April 2018; pp. 207–215.
22. Yu, S.; Langar, R.; Fu, X.; Wang, L.; Han, Z. Computation offloading with data caching enhancement for mobile edge computing. *IEEE Trans. Veh. Technol.* **2018**, *67*, 11098–11112. [CrossRef]
23. Hu, G.; Jia, Y.; Chen, Z. Multi-user computation offloading with d2d for mobile edge computing. In Proceedings of the IEEE GLOBECOM, Abu Dhabi, UAE, 9–13 December 2018; pp. 1–6.
24. Wang, Y.; Sheng, M.; Wang, X.; Wang, L.; Li, J. Mobile-edge computing: partial computation offloading using dynamic voltage scaling. *IEEE Trans. Commun.* **2016**, *64*, 4268–4282. [CrossRef]
25. Guo, H.; Liu, J.; Zhang, J. Computation offloading for multi-access mobile edge computing in ultra-dense networks. *IEEE Internet Things J.* **2018**, *56*, 14–19. [CrossRef]
26. Guo, H.; Liu, J. Collaborative Mobile-Edge Computation Offloading for IoT over Fiber-Wireless Networks. *IEEE Network* **2018**, *32*, 12–18. [CrossRef]
27. Rodrigues, T.G.; Suto, K.; Nishiyama, H.; Kato, N. Hybrid Method for Minimizing Service Delay in Edge Cloud Computing Through VM Migration and Transmission Power Control. *IEEE Trans. Comput.* **2017**, *66*, 810–819. [CrossRef]
28. Liu, M.; Liu, Y. Price-based distributed offloading for mobile-edge computing with computation capacity constraints. *IEEE Commun. Lett.* **2017**, *7*, 420–423. [CrossRef]
29. Hao, Y.; Chen, M.; Hu, L.; Hossain, M.S.; Ghoneim, A. Energy efficient task caching and offloading for mobile edge computing. *IEEE Access* **2018**, *6*, 11365–11373. [CrossRef]
30. Hou, T.; Feng, G.; Qin, S.; Jiang, W. Proactive Content Caching by Exploiting Transfer Learning for Mobile Edge Computing. In Proceedings of the IEEE Globecom, Singapore, 4–8 December 2017; pp. 1–6.
31. Jia, G.; Han, G.; Du, J.; Chan, S. A maximum cache value policy in hybrid memory-based edge computing for mobile devices. *IEEE Internet Things J.* **2018**, *6*, 4401–4410. [CrossRef]
32. Ale, L.; Zhang, N.; Wu, H.; Chen, D.; Han, T. Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network. *IEEE Internet Things J.* **2019**, *6*, 5520–5530. [CrossRef]
33. Tao, X.; Ota, K.; Dong, M.; Qi, H.; Li, K. Performance guaranteed computation offloading for mobile-edge cloud computing. *IEEE Commun. Lett.* **2017**, *6*, 774–777. [CrossRef]

34. Ma, X.; Zhang, S.; Yang, P.; Lin, C.; Shen, X.S. Cost-Efficient Resource Provisioning in Cloud Assisted Mobile Edge Computing. In Proceedings of the IEEE Globecom, Singapore, 4–8 December 2017; pp. 1–6.
35. Dai, Y.; Xu, D.; Maharjan, S.; Zhang, Y. Joint computation offloading and user association in multi-task mobile edge computing. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12313–12325. [CrossRef]
36. Lyu, X.; Tian, H.; Jiang, L.; Vinel, A.; Maharjan, S.; Gjessing, S.; Zhang, Y. Selective offloading in mobile edge computing for the green internet of things. *IEEE Netw.* **2018**, *32*, 54–60. [CrossRef]
37. Deng, R.; Lu, R.; Lai, C.; Luan, T.H.; Liang, H. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet Things J.* **2016**, *3*, 1171–1181. [CrossRef]
38. Wang, C.; Liang, C.; Chen, Q.; Tang, L. Joint computation offloading, resource allocation and content caching in cellular networks with mobile edge computing. In Proceedings of the IEEE ICC, Paris, France, 21–25 May 2017; pp. 1–6.
39. Cui, Y.; He, W.; Ni, C.; Guo, C.; Liu, Z. Energy-efficient resource allocation for cache-assisted mobile edge computing. In Proceedings of the IEEE LCN, Singapore, 9–12 October 2017; pp. 640–648.
40. Pietro, D.; Strinati, E.C. An optimal low-complexity policy for cache-aided computation offloading. *IEEE Access* **2019**, *7*, 182499–182514. [CrossRef]
41. Liu, P.; Xu, G.; Yang, K.; Wang, K.; Meng, X. Jointly optimized energy-minimal resource allocation in cache-enhanced mobile edge computing systems. *IEEE Access* **2019**, *7*, 3336–3347. [CrossRef]
42. Zhang, J.; Hu, X.; Ning, Z.; Ngai, E.; Zhou, L.; Wei, J.; Cheng, J.; Hu, B.; Leung, V.C.M. Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching. *IEEE Internet Things J.* **2018**, *6*, 4283–4294. [CrossRef]
43. Yang, X.; Fei, Z.; Zheng, J.; Zhang, N.; Anpalagan, A. Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing. *IEEE Trans. Veh. Technol.* **2019**, *68*, 11018–11030. [CrossRef]
44. Wang, C.; Liang, C.; Yu, F.R.; Chen, Q.; Tang, L. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Trans. Wireless Commun.* **2017**, *16*, 4924–4938. [CrossRef]
45. Breslau, L.; Cao, P.; Fan, L.; Phillips, G.; Shenker, S. Web caching and Zipf-like distributions: Evidence and implications. In Proceedings of the IEEE INFOCOM, New York, NY, USA, 21–25 March 1999; pp. 126–134.
46. Rappaport, T.S. *Wireless Communications: Principles and Practice*; Prentice-Hall: Upper Saddle River, NJ, USA, 1996; Volumn 2.