A Novel Collaborative Task Offloading Scheme for Secure and Sustainable Mobile Cloudlet Networks

Ning Yang*, Xiaochen Fan†‡, Deepak Puthal†, Xiangjian He†‡, Priyadarsi Nanda†, Shiping Guo*
* School of Automation, Northwestern Polytechnical University, Xi'an, 710072, China
† School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
‡Corresponding authors: Xiaochen Fan (e-mail: Xiaochen.Fan@uts.edu.au) and Xiangjian He (e-mail:Xiangjian.He@uts.edu.au)

***Abstract*—With the advancement of wireless networking technologies and communication infrastructures, mobile cloud computing has emerged as a pervasive paradigm to execute computing tasks for capacity-limited mobile devices. More specifically, at the network edge, the resource-rich and trusted cloudlet system can provide in-proximity computing services by executing the workloads for nearby devices. Nevertheless, there are chances for malicious users to generate DDoS (Distributed Denial-of-Service) flooding tasks to overwhelm cloudlet servers and block computing services from legitimate users. Load balancing is one of the most effective methods to solve DDoS task in distributed networks, however, the existing solutions require overall load information of cloudlet networks, making it costly in both communication and computation. To achieve more efficient and low-cost load balancing, we propose CTOM, a novel Collaborative Task Offloading scheMe to avoid DDoS attacks for secure and sustainable mobile cloudlet networks. The proposed solution is based on the balls-and-bins theory and it can balance the task loads and only require extremely limited information. Extensive simulations and evaluation demonstrate that, the proposed CTOM outperforms the conventional random and proportional allocation schemes in reducing the task gaps among mobile cloudlets by 65% and 55% respectively. Thus CTOM reduces the overloaded cloudlets smoothly and handles the potential DDoS attacks in mobile cloudlet networks.**

***Keywords*-load balancing; mobile cloudlet network; task allocation; DDoS attacks**

I. Introduction

In recent years, with the pervasive proliferation of mobile devices and the advance in networking technologies, mobile users are free to enjoy various powerful and functional applications, such as Augmented Reality, Virtual Reality and Face Recognition [1]. While these mobile applications are more and more demanding in computation and resources, the capacity of smart devices is still constrained. Such that, most mobile users constantly face with the problems of resource-exhaustion or energy-drain. To tackle this issue, cloud computing has been proposed and pervasively used for processing resource-intensive tasks [2]. However, due to the long distance between the central servers and mobile users, there are some inevitable limitations in cloud computing, such as network latency, signal loss, link noise and transmission delays [3]. To provide more accessible and distributed computing services to mobile users, an alternative cloud computing paradigm has been proposed, *i.e.*, the so called 'cloudlet' [4].

A cloudlet is a trusted, resource-rich cluster of servers that are integrated with wireless access points (APs), by which it is accessible and connected to nearby mobile users [5]. By providing seamless access with low-latency and high-bandwidth, cloudlets can execute computation tasks for mobile users almost in real time, and thereby significantly improve the performance of cloud computing [4], [6], [7]. Recent studies [8]–[11] have focused on mobile cloudlets, which utilize the multitude of near-user vehicular networks to achieve more efficient task offloading and processing. There have been numerous applications including computation offloading [13], [25], path planning [14], energy charging [15] based on cloudlet infrastructures.

Despite the rapid development of cloudlets in vehicular networks [13], [14], [16], the security issues emerge as mobile cloudlets are generally open and accessible to any nearby users. Meanwhile, the potential attackers can easily exploit this vulnerability to launch DDoS attacks against mobile cloudlets. A typical DDoS attack deploys multiple attacking entities to disrupt normal traffic on targeted servers, by overwhelming the targets with flooding traffic flows [17], [18]. Thus, it is essential for service providers to address the concerns of potential DDoS attacks. However, in mobile cloudlet networks, it is not practical to apply typical DDoS detection techniques [19], due to the distributed nature of networks and dynamic nature of task flows [20]. As mobile cloudlets travel around various metropolitan areas with different population density, it is impossible to centrally control the amount of user task flow to any single cloudlet. Fortunately, the potential DDoS attacks can be smoothly avoided and handled through balanced task offloading, as most of tasks can be concurrently processed by multiple servers among all the mobile cloudlets. Therefore, the average task response time is reduced even if there exit DDoS tasks from malicious users.

Meanwhile, how to achieve load balancing in mobile cloudlet networks remains a challenge. There are some studies aiming to address the load balancing issues in static cloudlet systems, either by strategic cloudlet placement [5], [21] or by cloudlet-oriented task redistribution [6], [22]. However, these methods are not applicable in mobile network scenario, where the cloudlets are enhanced with random mobility and the network is intermittently connected. Moreover, some previous studies [12], [25] only focused on unbalanced offloading

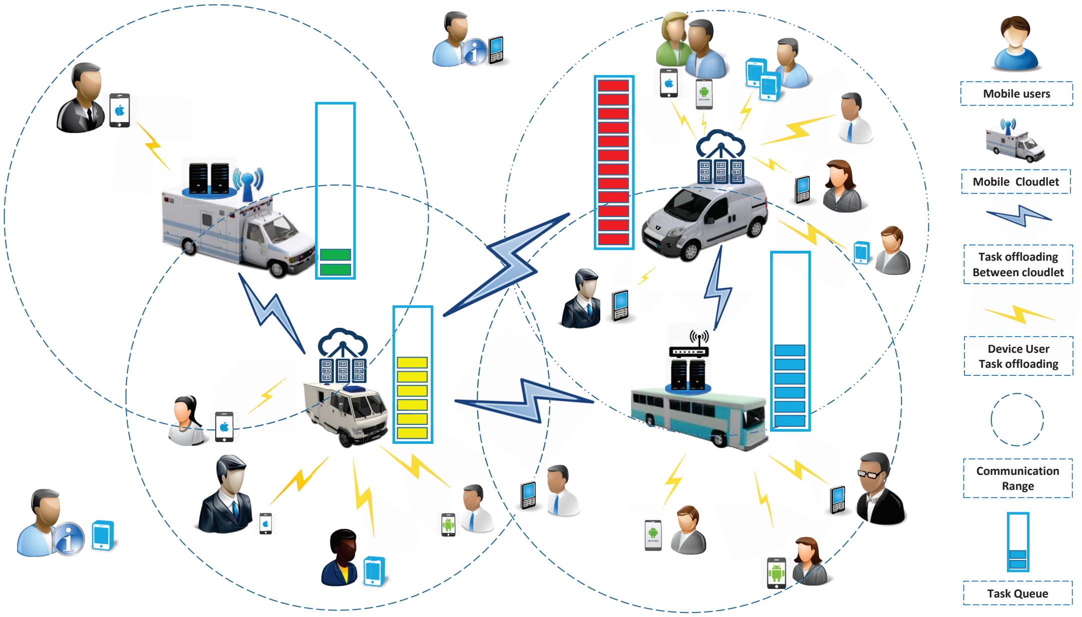


Fig. 1. Task offloading scenario in mobile cloudlet networks.

problems of cloudlets without considering any security issues. Indeed, it is quite daunting to achieve load balancing among mobile cloudlets as they are purely distributed. Even worse, for each cloudlet, the load information of its neighbors constantly change, which makes it more costly to collect the overall load information. Accordingly, two challenges need to be carefully addressed.

First, to address the potential DDoS attacks, the load balancing should be achieved through collaborative task offloading. As the mobility of cloudlets can neither be centrally controlled nor predicted, it is hard to constantly redirect an exact amount of task flow from one cloudlet to another. Fortunately, it is possible for encountering cloudlets to collaboratively offload tasks to each other with shared load information, thus handling the possible attack tasks on overwhelmed cloudlets.

Second, the balanced task allocation in securing cloudlet networks should be low-cost and light-weight in communication and computation respectively. It is impractical to query global load information in mobile cloudlet networks. Even if it can be achieved, the accumulative communication cost on the overall network would be extremely high. Moreover, with transmission delays, the out-sync load information may lead to wrong task offloading decision to already overloaded cloudlets.

In this paper, to deal with the aforementioned challenges, we propose CTOM, a novel Collaborative Task Offloading scheMe for secure and sustainable mobile cloudlet networks. CTOM leverages the balls-into-bins theory [23] to fit the distributed task allocation scenario in mobile cloudlet networks. Based on the ‘two-choice’ [24] paradigm, by querying load information only from two randomly selected neighbors, cloudlets can process well-balanced task offloading. Accumulatively, every long task queue in a cloudlet network will be significantly reduced with high probability. In this way, the potential DDoS attacks that aim at overwhelming targeted cloudlets can be smoothly handled and even avoided.

We summarize the contributions of this paper as follows:

1) We propose a novel collaborative task offloading mechanism for secure and sustainable mobile cloudlet networks, where the cloudlets are enhanced with mobility and intermittently connected. To the best of our knowledge, this is the first work focusing on collaborations among mobile cloudlets for secure and sustainable load balancing.
2) Inspired by the balls-and-bins probability theory, we propose a novel solution for secure and sustainable task allocation in distributed mobile cloudlet networks. By comparing the task load of only two neighbors, a mobile cloudlet can process balanced task offloading at low communication cost.
3) In order to validate and demonstrate the effectiveness of our idea, extensive simulation and trace-based evaluation have been conducted. The simulation results show that the proposed CTOM algorithm has achieved exceedingly balanced results in mobile cloudlet task allocation and performed closely to the optimal allocation. The potential DDoS attacks on overwhelmed cloudlets are processed and filtered out through the collaboration of mobile cloudlets.

The rest of this paper is organized as follows. We review the brief background and related work in Section II. In Section III and Section IV, we introduce the system model of mobile cloudlet networks and load balancing problem respectively. Then we present CTOM algorithm in details in Section V and theoretically analyze it in Section VI. We further evaluate the CTOM’s performance with extensive simulation and trace-driven evaluation in Section VII. The conclusion is in Section VIII.

II. Related Work

In this section, we first present the background of distributed systems and cloudlet networks. Then we review the recent literatures in load balancing in mobile cloudlet networks.

A. Cloudlet Networks

In recent years, as a centralized computing paradigm, cloud computing systems have been widely implemented to process tasks and backup data for mobile users [26]. More recently, Satyanarayanan [4] proposed 'cloudlet', an ubiquitous facility that acts as "data center in a box" to provide distributed computing services. The cloudlet is in the middle of a three tiered hierarchy, *i.e.*, mobile devices, cloudlets and the central cloud. With cloudlets, rather than requesting services to distant central cloud, mobile users can leverage in-proximity servers in cloudlets for executing resource-intensive and energy-consuming tasks. As the communication from the local cloudlet to surrounding users is usually within one hop access, cloudlets are capable of providing low latency and high bandwidth network connectivity. Tasks such as real-time face recognition, object recognition and high-resolution augmented reality [1] can executed with fast response time in cloudlet networks [27]. Furthermore, mobility-enhanced cloudlet systems are also proposed with the emergence of mobile edge computing, where cloudlet-integrated vehicles travelling in metropolitan areas to collect and process tasks from mobile users [7], [16], [28].

B. DDoS in Cloud Networks

DDoS attacks in cloud networks are becoming one of the major security concerns of service providers. The malicious DDoS attacks can destroy the availability of cloud computing and prevent the legitimate use of computing services [17]. Researches in cyber security community have designed various defense mechanisms and solutions against DDoS attacks [20], which can be categorized as attack prevention, attack detection and attach mitigation and recovery [19]. The attack prevention methods filtered or dropped the suspected attacker's requests, through techniques such as challenge response [29], hidden servers or hidden ports [30] and restrictive access [31]. In attack detection, the possible attack signs on the servers are detected and monitored in terms of performance metrics for further prevention actions. The attack detection methods can be classified into anomaly detection [32], source and spoof trace [33], filter-based selection [34] and strategic resource allocation [35], [36]. In this paper, we leverage strategic resource allocation method to balance the task load between overloaded and underloaded cloudlets. In this way, the attacking DDoS tasks are quickly processed and filtered out from in cloudlet networks while overall performance stays sustainable and reliable.

C. Load Balancing in Mobile Cloudlet Networks

Researchers have proposed a variety of game-theoretic approaches to solve the load balancing problem for distributed systems, including static load balancing [37], dynamic load balancing [38], cooperative load balancing [39], noncooperative load balancing [40], selfish load balancing [41] and randomized load balancing [24]. In mobile cloudlets networks, each cloudlet randomly travels in different and their locations are not fixed at all. Considering the different population density in each area, the amount of incoming task flow on each cloudlet would fluctuate heavily. Such that, the load balancing problem emerges, where the cloudlets frequently appearing the areas with high user-density areas are overloaded, while the rest of cloudlets at sparsely populated areas are at underloaded or idle states. The computing resource is not fully utilized in the above networks and the overall average task response time is also dragged down.

Several existing studies proposed different methods to solve the load balancing problem for statistic cloudlets. The first approach is strategic cloudlet placement. Xu *et al.* [5] proposed a placement strategy for capacitated cloudlets in a wireless metropolitan area network. Their solution is to minimize the cloudlet accessing delay and average task response time for device users. Jia *et al.* [6] further formulated an optimal task redirection problem in static cloudlet systems. They devised a load balancing algorithm to minimize the task response time. However, in our scenario, the cloudlets are enhanced with mobility, so the connectivity in the network is intermittent. With task flows from edge devices to cloudlet continuously changing, the above solutions become incompetent. Moreover, Zhang *et al.* [7] developed an optimal offloading algorithm for mobile users considering both user mobility pattern and cloudlet admission control. In [21], Jia *et al.* further associated the cloudlet placement problem with task assignment. They proposed a heaviest-AP first algorithm and a density-based clustering algorithm to balance the workload among cloudlets. Different from the above works, in this paper, we focus on the cooperation of mobile cloudlets and propose a low-cost and light-weight scheme for balanced task offloading. We explore the possibility of collaborative task offloading for load balancing, with the concerns of DDoS attacks in mobile cloudlet networks.

III. System model

In this section, we introduce the system model in the following aspects: network model, cloudlet model, communication model and task offloading model and attack models.

Network Model: We start the network model with a set of mobile cloudlets deployed in a metropolitan area. We assume that K mobile cloudlets $\mathrm{C} = \{\mathrm{c}_1,\mathrm{c}_2,...,\mathrm{c}_K\}$ are integrated with vehicular access points(APs), where they communicate with each other via network connection [6]. It is also assumed that the user's applications are dynamically partitioned into offloadable and executable computing tasks that can be processed by any of the k cloudlets. As depicted in Fig. 1, while users can offload computation tasks to any nearby cloudlets, the cloudlets can locally process incoming tasks or transfer current tasks to their neighbours in the network.

Cloudlet Model: According to [6], for each mobile cloudlet $i \in \{1, 2, ..., K\}$, we model it aa an $M/M/n$ queue. Each cloudlet i has s_i server(s) with the service rate μ_i. Also, we adopt random walk to model the mobility of cloudlets, as they randomly travel in the metropolitan areas. For any cloudlet i, the number of incoming task offloading from nearby user change constantly. Based on that, Poisson Process is adopted to model the incoming user tasks [6]. The task arrival rate (from mobile users) at cloudlet i is λ_i. Also, to store the arrived tasks pending for execution, each mobile cloudlet holds a FIFO task queue $Q_i = \{q_1, q_2, .., q_n\}$, where the queueing length is $||Q_i||$.

Communication Model: Similar to [6], we assume that the mobile cloudlets in this model are also integrated with wireless access points, which provides for one-hop, low-latency and high-bandwidth wireless access for task offloading. Only when the distance d_{ij} between cloudlets i and j is within the inter-contact range R, a communication can be established between them [7]. The inter-meeting time of cloudlets c_i and c_j is denoted as $t_{i,j}$. Referring to [42] and [43], $t_{i,j}$ would follow an exponential distribution with a pairwise rate α_{ij}, i.e., $f\left(t\right) = \frac{1}{\alpha_{i,j}} e^{-\frac{1}{\alpha_{i,j}} \cdot t}, t \geq 0, t \geq 0$. Between any two time interval t_a and t_b, the encountering probability of cloudlets c_i and c_j is computed as followed:

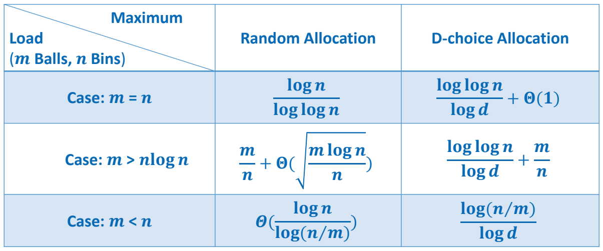
$$P_{i,j}\left(t_a, t_b\right) = e^{-\frac{1}{\alpha_{i,j}} \cdot t_a} - e^{-\frac{1}{\alpha_{i,j}} \cdot t_b} \tag{1}$$

Satyanarayanan *et al.* [1] conducted several task offloading experiments in cloudlet networks that connected by WiFi, where the execution time of offloaded task is approximately $10^{-4} \sim 10^{-2}$ seconds for applications such as augmented reality and face recognition. Adding to the round-trip time (RTT) of wireless transmission (hundreds of milliseconds), we consider the time interval set in this model is reasonably long enough for the inter-contact time (including execution time and RTT). In another word, the task execution results can be sent back to the corresponding mobile users within the same time interval [8].

Task Offloading Model: In this model, a 'task' refers to an application phase that involves executable codes and offloadable data that can be processed by any mobile cloudlet [7]. Such that, the total number of tasks generated from different user's application would fluctuate constantly. We address above considerations by sampling Poisson Process [6] to determine the actual number of tasks at cloudlet i. We denote λ_i as arriving task rate at cloudlet i. We also adopt the percent imbalance metric η and the statistical moment φ from [44] to evaluate the overall load balancing of task allocation. The above metrics are calculated as follows:

$$\eta = \left(\frac{L_{max}}{\bar{L}} - 1\right) \times 100\%, \quad \varphi = \frac{\frac{1}{n}\sum\limits_{i=1}^{n}\left(L_i - \bar{L}\right)^3}{\left(\frac{1}{n}\sum\limits_{i=1}^{n}\left(L_i - \bar{L}\right)^2\right)^{3/2}} \tag{2}$$

where L_{max} and $\bar{L}$ are the maximum and average load respectively. The percent imbalance metric measures the severity of load imbalance, while the skewness provides a detailed description of load distribution [44].



Maximum Load (m Balls, n Bins)	Random Allocation	D-choice Allocation
Case: $m = n$	$\frac{\log n}{\log\log n}$	$\frac{\log\log n}{\log d} + \Theta(1)$
Case: $m > n\log n$	$\frac{m}{n} + \Theta(\sqrt{\frac{m\log n}{n}})$	$\frac{\log\log n}{\log d} + \frac{m}{n}$
Case: $m < n$	$\Theta(\frac{\log n}{\log(n/m)})$	$\frac{\log(n/m)}{\log d}$

Fig. 2. Theoretical results of maximum load in balls-and-bins problem

Attack Model: In the DDoS attack model, the attackers control a group of compromise mobile devices as a botnet, then they launch malicious task flooding to nearby cloudlets. The DDoS attack tasks can exhaust the computing resources and bandwidth on mobile cloudlets, such that the targeted cloudlets will not be able to respond to any arrived or incoming legitimate tasks [19]. In reality, the DDoS attacks could result in service degradation, bottleneck, system failure and further financial loss for cloudlet networks and service providers. In proposed task offloading model, the incoming user tasks at cloudlet i is randomly sampled from Poisson process with arrival rate λ_i. Base on that, we assume that the potential DDoS attacks can be revealed by the sampled arrival rates that have extremely high values. Note that the our main goal is to smoothly handle and avoid the potential DDoS tasks on the cloudlets for sustainable network performance, not to detect or trace any potential DDoS attack.

IV. Problem Formulation

The load balancing problem in a mobile cloudlet network can be formulated as follows.

Given a mobile cloudlet network G with a set of cloudlet $\mathrm{C} = \{\mathrm{c}_1, \mathrm{c}_2, ..., \mathrm{c}_K\}$, where each cloudlet $\mathrm{c_i}$ holds a FIFO task queue in $Q = \{q_1, ...q_k\}$ to store the received tasks. Meanwhile, cloudlet i has n_i servers with a service rate of μ_i and the task arrival rate at the cloudlet $\mathrm{c_i}$ is λ_i. Our main objective is to achieve balanced task offloading and handle potential DDoS attacks with the following constraints:

1) Due to the mobility of cloudlets, the network is intermittently connected.
2) The differential densities in different areas results in fluctuant task load at each cloudlet.
3) Because of the distributed network, the task offloading can only be processed with limited information.
4) For cloudlet $\mathrm{c_i}$, the total outgoing tasks should be no greater than the number of arrived tasks.
5) Every cloudlet aims to minimize its current task load by offloading its tasks to other cloudlets at each time interval.

6) The mobile cloudlets in the network will cooperatively accept tasks from each other.
7) The DDoS attack task are potentially exist, especially in cloudlets that have extremely heavy task arrival rate.

Above all, we investigate the constraints when offloading tasks among mobile cloudlets collaboratively in wireless metropolitan area networks. The aim is to solve the following problems with concerns of DDoS attack tasks:

1) Basic Load Balancing Problem: In particular, we aim to minimize the overall variance of task queues in mobile cloudlets to achieve balanced task distribution, which can be defined as:

$$Minimize \sum_{i \in C} \|Q_i - E\left[Q\right]\|, \mu_i \cdot n_i \geq \lambda_i, i \in C, \quad (3)$$

where the incoming task flow is no greater than the total service rate at each cloudlet $\mathrm{c_i}$.

2) Gap Minimization and Balance Metric Evaluation: The task load gap between the maximum queue and the average queue is also worth evaluating. Note that the maximum load $L_{\max}$ and the average load $\bar{L}$ both count for the imbalance metric and statistical skewness in Section 2. The evaluation of load gap can be described as:

$$Minimize \max_{i \in C} \|Q_i\| - E_{i \in C}[Q_i], \mu_i \cdot n_i \geq \lambda_i, i \in C, \quad (4)$$

where the incoming task flow is no greater than the total service rate at each cloudlet $\mathrm{c_i}$.

3) Requirements for the Load Balancing Algorithm Design: We aim to propose an efficient task offloading algorithm for mobile cloudlets. Such that, each cloudlet can have a relatively equal share of the total tasks. Meanwhile, there are three basic requirements for designing such an algorithm in order to solve the above load balancing problem, *i.e.*,

- The algorithm should be designed to achieve *dynamic* load balancing among mobile cloudlets, which means that the balanced offloading is processed at each time interval.
- The proposed algorithm should be *highly efficient* in regard to cloudlet communication. There should be as few interactions as possible among cloudlets so as to achieve low communication overhead.
- The algorithm should be *computationally smart*. Task allocation should be processed under simple operations with collected load information.
- The algorithm should achieve *dynamic resource provisioning*. The under provisioning resources should be exploited to efficiently process and filter out the attack tasks.

Next, we illustrate the solution of a collaborative task load balancing in order to achieve these objectives.

V. Propose solution and Algorithm Design

In this work, we adopt the balls-and-bins theory and design a novel collaborative task offloading mechanism, *i.e.*, CTOM, to improve the sustainability of cloudlet utilization under potential DDoS attacks. Before describing the details of CTOM algorithm, we first briefly introduce the balls-and-bins theory.

A. The balls-and-bins theory

The balls-and-bins model is a classic probability model for randomized allocation process. Suppose that n balls are to be thrown into n bins, with each ball choosing a bin independently and uniformly at random. Then, the *maximum load*, *i.e.*, the largest number of balls in any bin, can be approximated as [24]:

$$\frac{\log n}{\log \log n}. \quad (5)$$

Now assuming that for each ball, it is placed into the fullest bin, among $d \geq 2$ bins chosen independently and uniformly, which is called *d-choice paradigm*. In this case, the maximum load is

$$\frac{\log \log n}{\log d} + \Theta(1). \quad (6)$$

The extension of the maximum load problem in balls-and-bins model is further considered, where m balls are sequentially placed into n bins with $m \gg n \log n$. In this case, for random allocation, the number of balls in the fullest bin is

$$\frac{m}{n} + \sqrt{\frac{m \log n}{n}}. \quad (7)$$

While for d-choice, if $m \gg n \log n$ then the maximum load is

$$\frac{m}{n} + \Theta(\sqrt{\frac{m \log n}{n}}). \quad (8)$$

In this work, the d-choice paradigm is applied to mobile cloudlet network model, where tasks and mobile cloudlets are considered as balls and bins respectively. We further conclude the theoretical maximum load of random allocation and d-choice allocation in Fig. 2, and we provide the theoretical analysis in Section VI.

B. Algorithm Design

We now illustrate the detailed design in the following subsections.

1) Overview: In designing the algorithm, we leverages two properties of d-choice paradigm with theoretical guarantees. The first is the power of random choices. Indeed, if we simply apply two random choices (*i.e.*, $d = 2$), it can still yield a larger reduction on the maximum load than just having one choice. Any additional choice beyond two will also decrease the maximum load by just a constant factor. The second is the randomness of selecting d possible offloading targets. The opportunistic encounter of mobile cloudlets leads to intermittent connectivity of the network. Such that, for each cloudlet, its neighboring cloudlets change along with the time interval randomly and independently.

There are some basic assumptions in algorithm design. First, we mainly focus on the collaboration among mobile cloudlets in task offloading. For each cloudlet $\mathrm{c_i}$, the incoming tasks from users follow Poisson process with a constant task arrival rate. Second, we assume that the tasks in the network are of the same size, so that the final allocation results can be measured precisely. Third, at each cloudlet $\mathrm{c_i}$, the arrived tasks are stored in the task queue Q_i. Fourth, the time interval is long enough for an inter-contact communication (including execution time and RTT).

Algorithm 1 The CTOM Algorithm

```
Input:
    Mobile Cloudlet C, Time Interval T, Contact Range $R$
    User Task Flow $\lambda_i$, Number of Servers S, Service Rate $\mu_i$
Output:
    Task Queue Q, Imbalance Metric and Statistical Moments
Minimize $\min \sum_{i \in C} \|Q_i - E[Q]\|$ using the $d$-choice method.
Initialize cloudlet's location (X,Y)
for Interval $t = [1 : T]$ do
    Mobile cloudlets perform random walk in a metropolitan area
    Update each cloudlet's location at the current time interval
    Update cloudlet's load information with $\lambda_i$, $q_i$, $\mu_i$
    for Each cloudlet $i = [1 : k]$ do
        Add user's task offloading into $q_i$
        Calculate encounters of cloudlets based on Eq. 1
                Update neighboring list $l(i)$
                if $||l(i)|| \geq d$
                Select $d$ neighbors randomly and independently
                else if $||l(i)|| \leq d$
                do $d \leftarrow d/2$ until $||l(i)|| \geq d$
                end if
                Select $d$ neighbors randomly and independently
                $s \leftarrow$ the first selected neighbor in d
                    for $v = 2$ to $d$ do
                        if $q_s > q_v$ then $s \leftarrow v$
                        end if
                    end for
                if $q_i > q_s$ then
                    $P \leftarrow 1 - q_s/q_i$
                    $q_s \leftarrow l_s + W(i) * P$
                    $q_i \leftarrow l_i - W(i) * P$
                end if
    end for
end for
return Q, $\eta$, $\varphi$
```

2) Algorithm Description: The detailed description for C-TOM in Algorithm 1 is elaborated as follows.

Basic inputs and outputs: The basic inputs include the main parameters of the system model. We input a set of cloudlet C, time interval T, the inter-contact range R for cloudlets. For each cloudlet $\mathrm{c_i}$, we have the user's task offloading rate as λ_i, the number of servers s_i and the service rate μ_i. For the simplicity of load balancing evaluation, the final outputs of the algorithm include a set of task load queues Q, and the unbalanced metric of the network, denoted as η, and the statistical skewness, denoted as φ.

Exploring opportunistic encounters: In the initializing step, CTOM algorithm first randomly generates each mobile cloudlet's initial location. As a new time interval begins, all the cloudlets will perform random walk, then the algorithm will update cloudlets' current locations and task loads. From each cloudlet $\mathrm{c_i}$, according to 1, the algorithm will firstly check whether there are new mobile cloudlets falling into its communication range. Then, it takes a record in neighboring cloudlets list $l(i)$ and calculate the number of neighbors. If the number of neighbors is greater than d, the algorithm will apply the d-choice paradigm; otherwise, the algorithm will assign $d/2$ to d until the value of d is smaller than the number of current neighbors.

Task offloading paradigm: The proposed CTOM will randomly select d neighboring mobile cloudlets from the current encountering list, and iteratively compare their task load to sort for the least task queue. For greedy algorithm, it will select all neighbors for comparison, with cost of higher computation complexity. Also, the algorithm will check whether the selected neighbor is appropriate for taking over the task held by the current cloudlet, by comparing their task load. The proportional algorithm [41] continues to compute the offloading probability based on the proportion of the task load between the current cloudlet and the selected cloudlet. When the task allocation process finishes, the current time interval ends and a new time interval begins. At last, the imbalance metric together with statistical moment will be calculated.

VI. Method Validation

In this section, we present the claims made in the proposed load balancing algorithm and provide proofs. First, we give out the definitions and notations as follows.

We consider a *finite* task offloading process, where there are m tasks and n mobile cloudlets. Initially, the mobile cloudlets are all idle and each of the tasks is allowed to be offloaded into one of d ($d \geq 2$) neighbouring cloudlets chosen independently and uniformly at random. The arrived tasks at each cloudlet are stored by FIFO. We denote the above task allocation process as a (m, n, d)-problem. In our proof, to make the exposition more clear, we first prove the case when $m = n$, and then we can shift the proof to $m > n$ case.

TABLE I
Notations and definitions

Notations	*Definitions*
$l_j^c(t)$	the load of cloudlet j, *i.e.*, the number of tasks in cloudlet j at time t, resulting from the proposed CTOM algorithm
$N_k^c(t)$	the number cloudlets that with the load of k at time t
$N_{\geq k}^c(t)$	the number of cloudlets that have the load larger than or equal to k at time t, *i.e.*, $N_{\geq k}^c(t) = \sum_{i \geq k} N_i^c(t)$
H_t^c	the length of the task queue t, which equals to the number of tasks at time t in a cloudlet
$M_k^c(t)$	the number of tasks that have a height of k at time t
$M_{\geq k}^c(t)$	the number of tasks with the height larger than or equal to k at time t, *i.e.*, $M_{\geq k}^c(t) = \sum_{i \geq k} M_i^c(t)$

Our proposed algorithm CTOM assigns a task j from its current cloudlet to the cloudlet with lowest load among its d

randomly selected neighbors. Next, we prove the upper bound of tasks in the fullest cloudlet under CTOM algorithm.

Claim 1: Suppose there are n tasks to be allocated to n cloudlets. For each cloudlet, it allocates the task to the least loaded neighbor out of d selected neighbors. Then the upper bound, *i.e.*, the total number of tasks in the fullest cloudlet is at most $\ln\ln n/\ln d$ with a high probability. We list the definitions of variables used in our proof in Table I).

Proof: The basic intuition of the proof is as follows.

Let $p_i = M_{\geq i}/n$. For each cloudlet, it offloads the current task independently and $N^c_{\geq k} \leq M^c_{\geq k}$, then we roughly have $p_{i+1} \leq p_i^d$ (d is the number of offloading choices), which shows the decrease in p_i is doubly exponential, as long as $M_{\geq i} < n/2$. Obviously, $M_{\geq i+1}$ is based on the condition that $M_{\geq i}$.

We consider the task allocation process is finite and denote a binomial and distributed random variable by $B(n,p)$. Then we start with a standard lemma as follows.

Lemma 1: Let $X_1, X_2, ..., X_n$ be a sequence of random variables with arbitrary values. Let $Y_1, Y_2, ..., Y_n$ be a sequence of binary random variables, with $Y_i = Y_i(X_1, ..., X_i)$. If

$$\Pr(Y_i = 1|X_1, ..., X_{i-1}) \leq p,$$

then we have

$$\Pr(\sum Y_i \geq k) \leq \Pr(B(n,p) \geq k).$$

Similarly, if

$$\Pr(Y_i = 1|X_1, ..., X_{i-1}) \geq p,$$

we have

$$\Pr(\sum Y_i \leq k) \leq \Pr(B(n,p) \leq k).$$

As the d choices are independent for each task, we have $\Pr(H_t \geq i+1|N_{\geq i}(t-1)) \leq \frac{(N_{\geq i}(t-1))^d}{n^d}$.

We use θ_i to denote the event of $N_{\geq i}(n) \leq \alpha_i$ (α_i will be illustrated in the following steps), which implies that $N_{\geq i}(t) \leq \alpha_i$ for $t = 1, 2, ..., n$).

For $i \geq 1$, we consider Y_t ($t = 2, ..., n$) as the serial binary variables, where $Y_t = 1 \iff h_t \geq i+1$ and $\nu_{\geq i}(t-1) \leq \beta_i$.

That is to say, $Y_t = 1$ if the height of the task t is greater than $i+1$, even the number of cloudlets that have more than i tasks is less than α_i.

We use γ_j to denote the choices available for the j_{th} ball. Then, we have

$$\Pr(Y_t = 1|\gamma_1, ..., \gamma_{t-1}) \leq \frac{\alpha_i^d}{n^d} \overset{\text{def}}{=\!=} p_i.$$

Now we apply Lemma 1 to conclude that

$$\Pr(\sum \mathrm{Y_t} \geq k) \leq \Pr(B(n,p_i) \geq k).$$

Also, when conditioned on θ_i, we have $M_{\geq i+1} = \sum Y_t$. Such that,

$$\Pr(\sum M_{\geq i+1} \geq k|\theta_i) = \Pr(\sum Y_t \geq k|\theta_i) \leq \frac{Pr(\sum \mathrm{Y_t} \geq k)}{\Pr(\theta_i)}.$$

By combining the above two formulas, we can obtain

$$\Pr(\sum N_{\geq i+1} \geq k|\theta_i) \leq \Pr(\sum M_{\geq i+1} \geq k|\theta_i) \leq \frac{\Pr(B(n,p_i) \geq k)}{\Pr(\theta_i)}.$$

According to [45] (see Appendix A), the large deviations in the binomial distribution can be bounded as follows

$$\Pr(B(n,p_i) \geq ep_in) \leq e^{-p_in}.$$

Therefore, we can set

$$\alpha_i = \begin{cases} n, & i = 1, 2, ..., 5; \\ \frac{n}{2e}, & i = 6; \\ \frac{e\alpha_{i-1}^d}{n^{d-1}}, & i > 6. \end{cases}$$

As $\theta_{\geq 6} = \{N_6 \leq n/(2e)\}$ still holds, for $i \geq 6$,

$$\Pr(\neg\theta_{i+1}|\theta_i) \leq \frac{1}{n^2\Pr(\theta_i)},$$

with $p_in \geq 2\ln n$. Since

$$\Pr(\neg\theta_{i+1}) \leq \Pr(\neg\theta_{i+1}|\theta_i)\Pr(\theta_i) + \Pr(\neg\theta_i),$$

we have

$$\Pr(\neg\theta_{i+1}) \leq \frac{1}{n^2} + \Pr(\neg\theta_i).$$

Let i^* be the smallest i such that $\alpha_{i^*}^d/n^d \leq 2\ln n/n$. While

$$\alpha_{i+6} = \frac{ne^{(d^i-1)/(d-1)}}{(2e)^{d^i}} \leq \frac{n}{2^{d^i}}$$

, we have $i^* \leq \ln\ln n/\ln d + \mathrm{O}(1)$.

As above,

$$\Pr(N_{\geq i^*+1} \geq 6\ln n|\theta_{i^*}) \leq \frac{\Pr(B(n, 2\ln n/n) \geq 6\ln n)}{\Pr(\theta_{i^*})} \leq \frac{1}{n^2\Pr(\theta_{i^*})}. \tag{9}$$

Thus, we have

$$\Pr(N_{\geq i^*+1} \geq 6\ln n) \leq \frac{1}{n^2} + \Pr(\neg\theta_{i^*})$$

. Finally,

$$\Pr(M_{\geq i^*+2}|N_{\geq i^*+1} \leq 6\ln n) \leq \frac{\Pr(B(n, 6\ln n/n)^d \geq 1)}{\Pr(N_{\geq i^*+1} \leq 6\ln n)} \leq \frac{n(6\ln n/n)^d}{\Pr(N_{\geq i^*+1} \leq 6\ln n)}. \tag{10}$$

Based on the Markov inequality [46], we can obtain

$$\Pr(M_{\geq i^*+2} \geq 1) \leq \frac{n(6\ln n)^d}{n^{d-1}} + \Pr(N_{\geq i^*+1} \geq 6\ln n).$$

By combining the above three formulas, we have

$$\Pr(N_{\geq i^*+2} \geq 1) \leq \frac{n(6\ln n)^d}{n^{d-1}} + \frac{i^*+1}{n^2} = o(1). \qquad (11)$$

Note that $i^* \leq \ln\ln n/\ln d + \mathrm{O}(1)$. Then the above proof shows that, the maximum load achieved by the proposed CTOM is no more than i^*+2 with a high probability, where $i^*+2 = \frac{\ln\ln n}{\ln d} + \mathrm{O}(1)$.

For the case $m > n$, *i.e.*, (m, n, d)-problem, if we consider θ_i be the event that $N_{\geq i}(m) \leq \alpha_i$ and also define $p_i = \alpha_i^d/n^d$. Following the proof for $m = n$ case, we can derive that

$$\Pr(\sum N_{\geq i+1} \geq k|\theta_i) \leq \frac{\Pr(B(m, p_i) \geq k)}{\Pr(\theta_i)}.$$

We suppose that $\alpha_x = n^2/(2em)$ for special values of x while θ_x also holds, *i.e.*,

$$\Pr(N_x \geq \frac{n^2}{2em}) = o(1).$$

Then we can have

$$\alpha_{i+x} = \frac{n}{2^{d^i}}(\frac{me}{n})^{(d^i-1)/(d-1)-d^i} \leq \frac{n}{2^{d^i}}.$$

By continuing as the proof of $m = n$ case, we can obtain that

$$\Pr(M \geq x + \ln\ln n/\ln d + 2) = o(1).$$

Above all, we show that for $m.n.d$-problem, the maximum task queue in any cloudlet is no more then

$$(1 + o(1))\ln\ln n/\ln d + \mathrm{O}(m/n). \qquad (12)$$

To this end, we have proved the upper bound of task load under CTOM. ■

Claim 2: The communication cost of the proposed CTOM (applying 2-choice paradigm) is no more than twice the random allocation on a ρ-round (infinite) (m, n, d)-problem.

Proof: For a (m, n, d)-problem, we denote the average communication cost of our CTOM, the random allocation and the greedy allocation as $C_C(m,n)$, $C_R(m,n)$ and $C_G(m,n)$ respectively.

Under the scheme of random allocation, a mobile cloudlet queries the load information from a randomly selected neighbor within the contact range at each interval (round). Thus, we have

$$C_R(m,n) \leq \rho n.$$

For the case of greedy allocation, a mobile cloudlet queries the global load information from its neighbors, which results in a high communication cost as

$$C_G(m,n) \leq \rho(n-1)^2.$$

In our CTOM, when applying the 2-choice paradigm, a cloudlet only queries two randomly selected neighbors. However, there are chances that only one or no cloudlet is within the communication range of the current cloudlet. Such that, no query process happens. So, we have

$$C_C(m,n) \leq \rho \cdot 2n.$$

Above all, the communication cost under different task allocation scheme are ranked as

$$C_R(m,n) < C_C(m,n) < C_G(m,n),$$

where $C_R(m,n) \leq 2C_C(m,n)$. ■

Claim 3: The proposed collaborative load balancing scheme smoothly handle the potential DDoS attacks on cloudlets.

Proof: Our solution to DDoS attack can be categorized as a DDoS aware resource allocation strategy, by which the overloaded cloudlets collaborate with underloaded and idle cloudlets for computing resource sharing [19]. Based on balls-and-bins theory, our solution resolves the potential DDoS attacks that aim at overwhelming cloudlets with the guaranteed upper bound of task offloading 12 as proved in *Claim 1*. ■

VII. Performance Evaluation

The performance evaluation of proposed scheme is twofold. First, we evaluate the proposed CTOM in a simulated network scenario, where cloudlet encounters are generated from random walk simulations. Second, we apply the proposed algorithm to a real-world trace for further evaluations.

A. Simulation Study

1) Basic setups: We run the simulation in a $10km^2$ region, which is of the similar scale of a city's central area. Here, we set the number of mobile cloudlets as 100 and the communication range as 20 metres. The total number of time slots is 600. According to [6], for each cloudlet i, we set the service rate μ_i by sampling normal distribution $\mathcal{N}(2,1) > 0$, and we set the number of its servers by sampling the Poisson distribution with a mean of 2. For tasks arriving at cloudlet i, we set task arrival rate λ_i by sampling the Normal distribution $\mathcal{N}(4,2) > 0$. and we consider the extreme distribution as potential attack tasks.

Under our CTOM scheme, during each time interval, a cloudlet first randomly chooses 2 neighbors in its contact range. After querying and comparing their load states, the cloudlet offloads a task to the neighbor with less task load, where the computing complexity in each time interval is $O(1)$. Similar to [43], we compare the performance of the proposed scheme with three benchmarks, *i.e.*, random allocation, proportional allocation [23] and greedy allocation.

In the random allocation, a mobile cloudlet offloads tasks by randomly selecting another mobile cloudlet in its contact range. Conversely, the greedy allocation method first queries all load information from its neighbors, and and compares their task loads then allocates tasks to the optimal cloudlet (with a computing complexity of $O(n)$). As for the proportional allocation, the chance for tasks to be offloaded to a randomly selected cloudlet depends on a probability parameter, which is calculated with task load information.

The simulation programs are all written in MATLAB codes. We run the programs in a Dell laptop with Intel Core i5 processor and 8 GB RAM. In general, each simulation program is executed for 100 times, and we take the average results as the final performance.

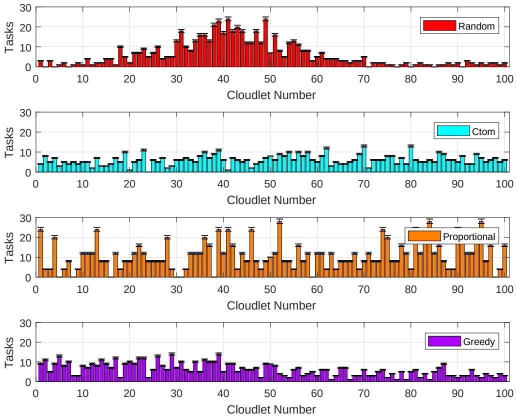


Fig. 3. Task allocation result

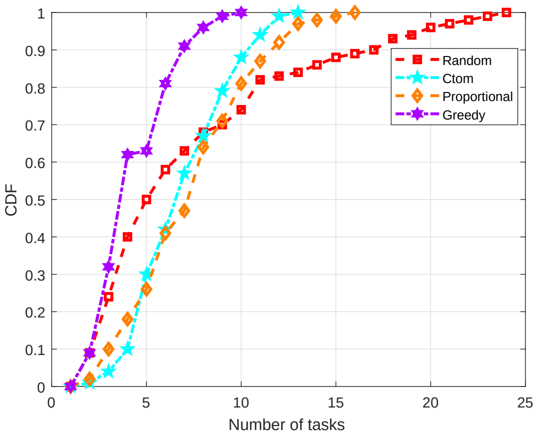


Fig. 4. The distribution of task loads obtained with the four schemes.

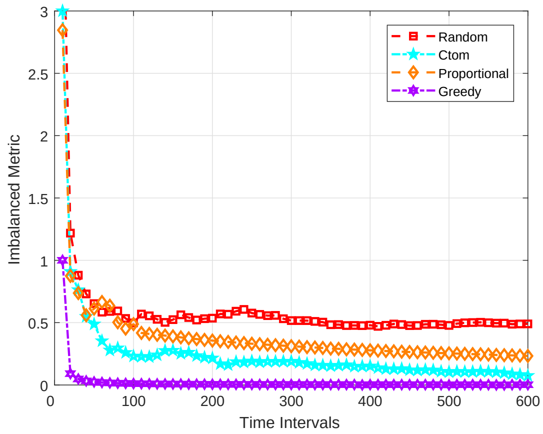


Fig. 5. The imbalance metrics obtained with the four schemes.

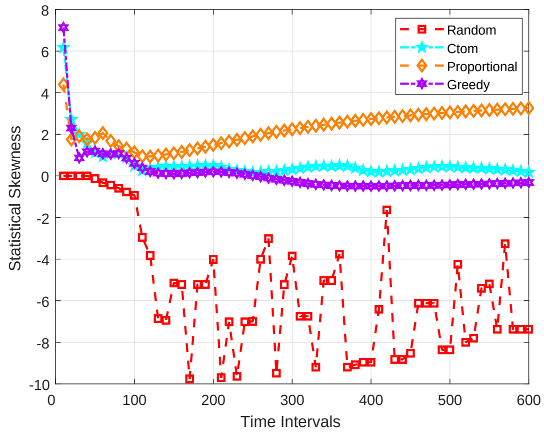
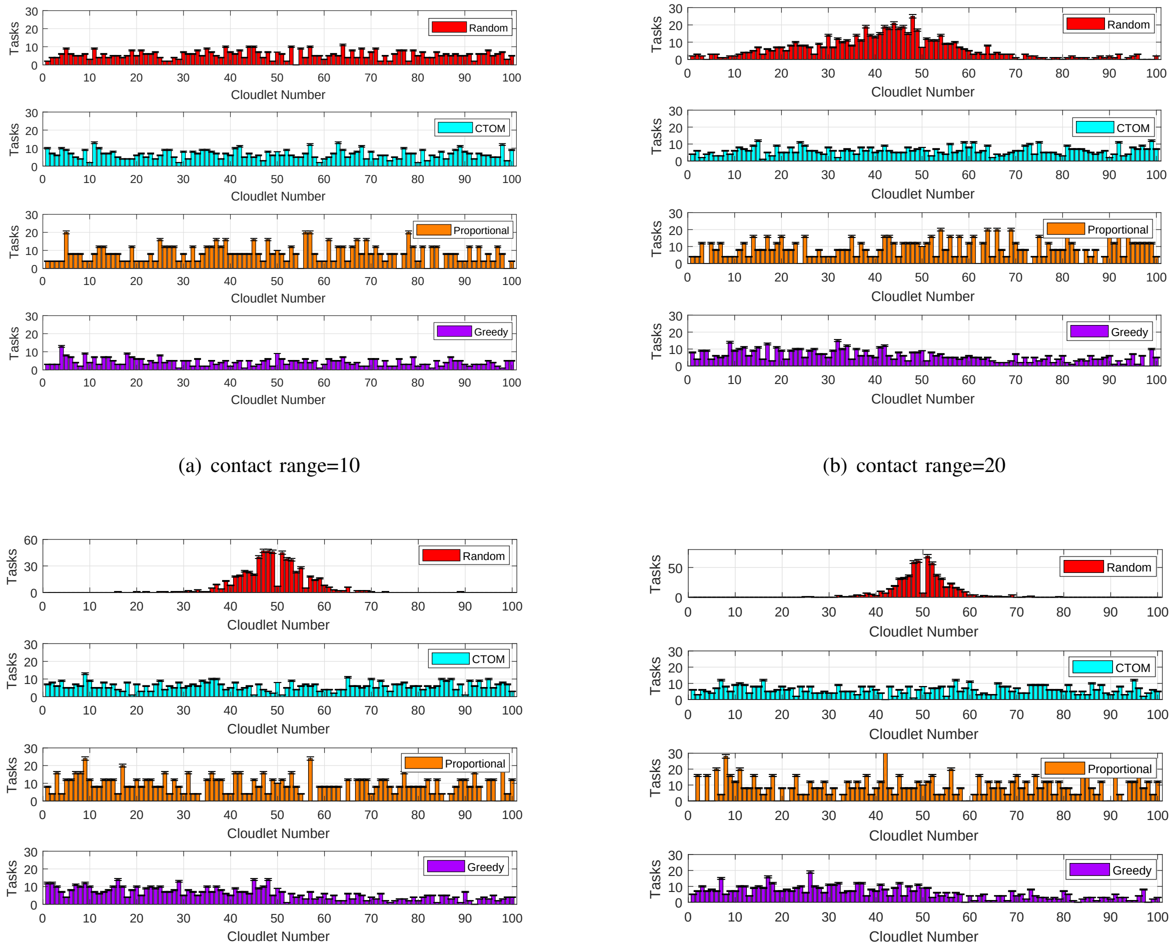


Fig. 6. The statistical skewness metrics obtained with the four schemes.

2) Overall Performance: Fig. 3 plots the overall task allocation results of mobile cloudlets obtained with our CTOM scheme and the three benchmark methods, *i.e.*, random allocation, proportional allocation and greedy allocation. Since the cloudlet's servers keep processing tasks, the overall allocation shows the remaining tasks at each mobile cloudlet. In random allocation, an adjacent group of cloudlets (ID 18 to 60) are overloaded with potential DDoS tasks, where most of their task loads are more than 10 and up to 24. Such that, the legitimate tasks can not be processed normally. Meanwhile, the mobile cloudlets at edge area are loaded with much fewer tasks (average less than 5) or even at idle state. Similarly, the task allocation obtained by the proportional allocation is also extremely unbalanced, where the distribution of overwhelmed (with over 25 remaining tasks) cloudlets is more sparse. In contrast, under CTOM and the greedy allocation, mobile cloudlets are equally allocated with tasks (around or below 10). In this way, the potential DDoS attack tasks will be effectively processed and filtered out from the cloudlet network.

Fig. 4 demonstrates the task allocation performance of the four methods in cumulative distribution. Under the schemes of random allocation and proportional allocation, about 30% mobile cloudlets are allocated with more than 10 tasks, which will affect the overall task response time. Meanwhile, our CTOM performs closely to the greedy method in balanced task offloading, where nearly 90% cloudlets are with task load under 10 and 55% cloudlets are offloaded with 5 to 10 tasks.

We further evaluate the task offloading performance using the imbalance metric [44], and the imbalance percentage and statistical skewness are calculated as in 2. The lower imbalance metric means the better balance performance in task allocation, *i.e.*, lower ratio of maximum and average task loads. From Fig. 5, it is obvious that the greedy algorithm achieves the best performance in terms of imbalance metric, which converges to almost 0. The imbalance metrics of our proposed CTOM and the proportional allocation scheme converge to 0.1 and 0.25 respectively. The random allocation scheme performs worst with imbalance metric 0.5. Meanwhile, Fig. 6 shows the statistics of the skewness obtained by the four schemes, where a positive or negative skewness indicates that the quantities of the mobile cloudlets having a higher or lower task load than average respectively. In Fig. 6, we can observe that the greedy



(a) contact range=10

(b) contact range=20

(c) contact range=30

(d) contact range=50

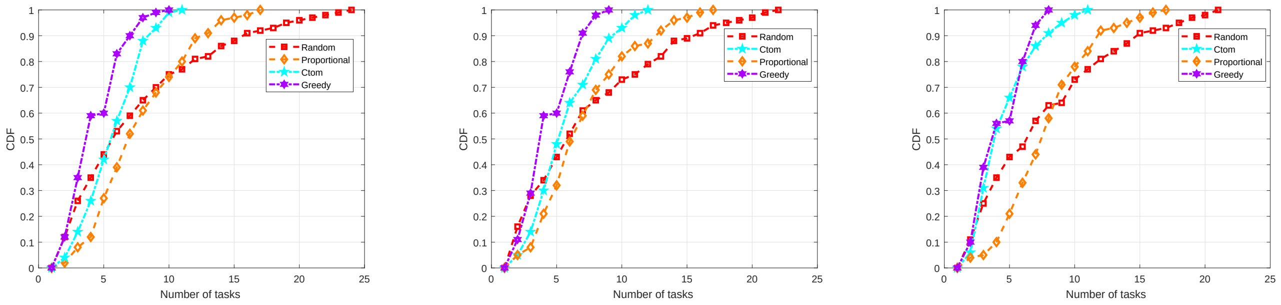
Fig. 7. Load analysis with the inter-contact range r ranging from 10-50.

allocation and our CTOM have both achieved the ultimate skewness values at about 0, which means that there are few cloudlets with an unbalanced load. As a contrast, the proportional method has a skewness of 2, and the random allocation's skewness fluctuates violently in negative range (from -10 to 0), which means there exist many mobile cloudlets with much lower task load than the average. For proportional allocation, the skewness varies from about 4 to 1, revealing that there are also many overloaded mobile cloudlets.

3) Analysis on parameters: We further evaluate the influence of the d in d-choice as well as the value of the inter-contact range on the load allocation performance. Firstly, we show the task offloading results with a contact range from 10 meters to 50 meters in Figure. 7. It is quite obvious that when the contact range increases, the tasks in random allocation are more centralized at a few mobile cloudlets, resulting in an unbalanced task distribution. Also, the proportional method performs poorly for all contact ranges, where a great number of cloudlets are overloaded (with up to 30 tasks) or at idle. The performance of CTOM and the greedy method are sustainable, where the overall task allocation is balanced and well distributed (most of cloudlets are with around 10 tasks). Secondly, we investigate the number of choice d. In Fig. 8, we plot the CDF of task allocation results with different values of d. From Figs. 8(a) to 8(c), the CDF lines of all methods pull back (maximum load decreases) as d increased. With greater values of d, in each time interval, one mobile cloudlet may have more options to offload its tasks, which results in a sustainable task allocation. The above simulation results demonstrate that, the proposed CTOM can achieve balanced task allocation, in this way, the overall tasks can be processed concurrently. Such that, CTOM improves the utilization efficiency of mobile cloudlets and shortens the task response time, thus handling the potential DDoS attack tasks smoothly.

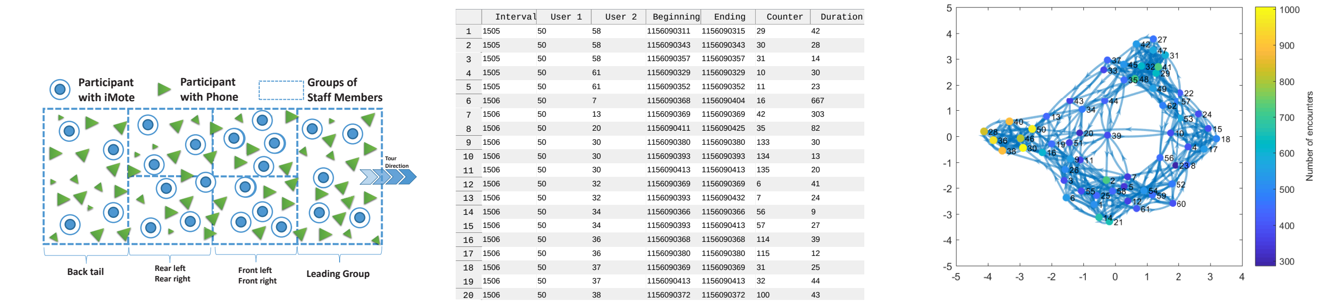
B. Trace-driven Evaluation

We further explore the balanced task allocation in a trace-driven study. The mobility dataset we used is called RollerNet [42]. The RollerNet data set was collected in a 15000



(a) d=4 (b) d=8 (c) d=16

Fig. 8. Load analysis with the number of choices d ranging from 4-16



	Interval	User 1	User 2	Beginning	Ending	Counter	Duration
1	1505	50	58	1156090311	1156090315	29	42
2	1505	50	58	1156090343	1156090343	30	28
3	1505	50	58	1156090357	1156090357	31	14
4	1505	50	61	1156090329	1156090329	10	30
5	1505	50	61	1156090352	1156090352	11	23
6	1506	50	7	1156090368	1156090404	16	667
7	1506	50	13	1156090369	1156090369	42	303
8	1506	50	20	1156090411	1156090425	35	82
9	1506	50	30	1156090380	1156090380	133	30
10	1506	50	30	1156090393	1156090393	134	13
11	1506	50	30	1156090413	1156090413	135	20
12	1506	50	32	1156090369	1156090369	6	41
13	1506	50	32	1156090393	1156090432	7	24
14	1506	50	34	1156090366	1156090366	56	9
15	1506	50	34	1156090393	1156090413	57	27
16	1506	50	36	1156090368	1156090368	114	39
17	1506	50	36	1156090380	1156090380	115	12
18	1506	50	37	1156090369	1156090369	31	25
19	1506	50	37	1156090413	1156090413	32	44
20	1506	50	38	1156090372	1156090372	100	43

(a) Diagram of device deployment in roller tour (b) The encounter dataset of iMote (c) The node-relation graph in iMote trace dataset

Fig. 9. iMote dataset illustration

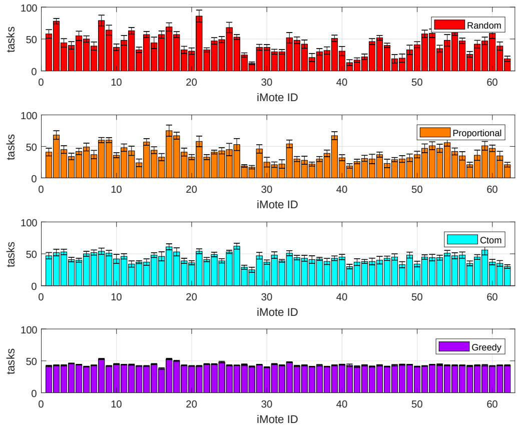


Fig. 10. Task load results in trace-driven evaluation

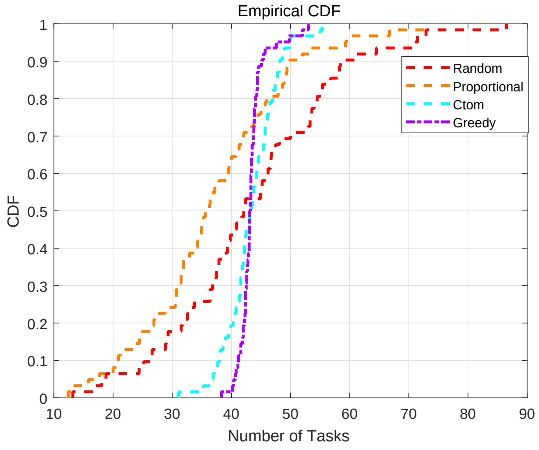


Fig. 11. Load distribution in trace-driven evaluation

people participated rollerblading tour in Paris, France. The rollerblading tour lasted for three hours and travelled 20 miles, covering the major metropolitan area in the city of Paris.

1) Basic setups: Our real-world evaluation is based on the real-world trace dataset for mobility-enhanced cloudlets, named as RollerNet, which includes the traces of opportunist sightings by wireless networking nodes called iMotes. The iMotes were distributed to a group of people to collect any opportunistic sighting of other mobile devices (including the other iMotes) via Bluetooth. We drew a sample diagram of iMote deployment as depicted in Fig. 9(a), where totally 62 skaters are equipped with iMotes and they were divided into 6 groups at different regions in the roller crowd. In this evaluation, we consider each iMote as a mobile cloudlet that can remotely execute computing tasks for mobile users. For cloudlet i with a service rate of μ_i, we assign the service rate by sampling the normal distribution $N(6, 2) > 0$. The number of servers at cloudlet i is sampled from Poisson distribution with a mean of 3. The task arrival rate λ_i follows a normal distribution $0 < N(18, 6) < s_i \cdot \mu_i$, where s_i is

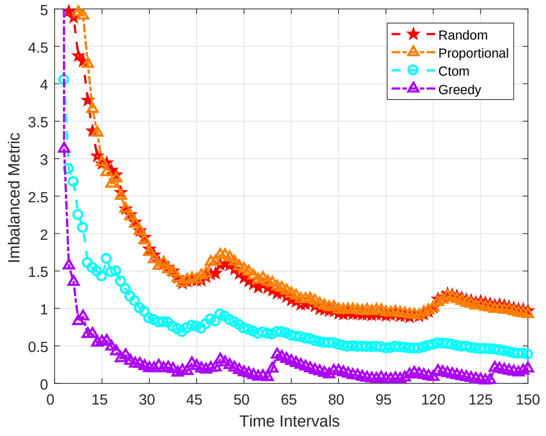


Fig. 12. The imbalance metric of different schemes

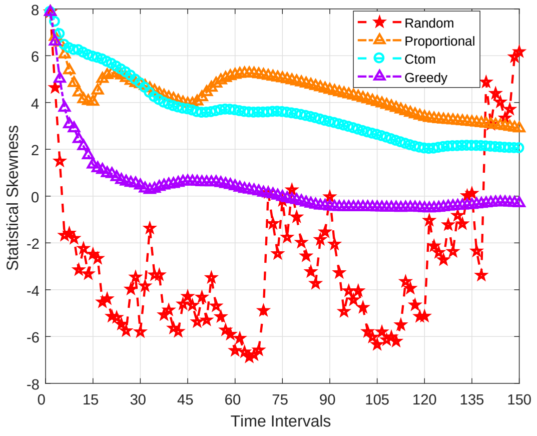


Fig. 13. The statistical skewness of different schemes

the number of servers at mobile cloudlet i. We assume that there are potential DDoS attackers in this rollerblading tour and the attack tasks are revealed by the extreme task arrival rate. All the settings are derived according to [6]. Meanwhile, our evaluation is based on real-world trace dataset and the cloudlets are enhanced with mobility.

We conduct a twofold pre-processing on the RollerNet dataset. First, we unify the timing of user encounter records. By setting a common starting time based on the earliest record, we convert the duration of all encounters into serial time slots by minutes. Based on the unified encounter records, we find that the total inter-contact time is $1567 - 1417 = 150$. Such that, we set the total time interval for task offloading as 150. Second, we plot an encounter graph to depict the frequency of communications among all the iMote skaters in Fig. 9(c), and we find that the iMote carriers can be roughly divided into three groups based on their communication frequency,*i.e.*, active group (with 800-1000 contacts), common group (with 500-800 contacts) and passive group (with 300-500 contacts). The above division consist with the formation of iMote skaters: skater association, staff and a set of friends.

2) Evaluation performance: Fig. 10 shows the task allocation results in bar graph obtained on RollerNet. As revealed from this figure, the performance of our CTOM method is comparable to that of the greedy allocation, where most of the mobile cloudlets are offloaded with around 50 tasks. Meanwhile, in random and proportional allocations, the allocation results are unbalanced with task loads fluctuating severely among different cloudlets (up to 80 and down to 10). In this case, the extremely overwhelmed iMotes can be viewed as attacked cloudlets, whose computing resources have been consumed by DDoS attack tasks.

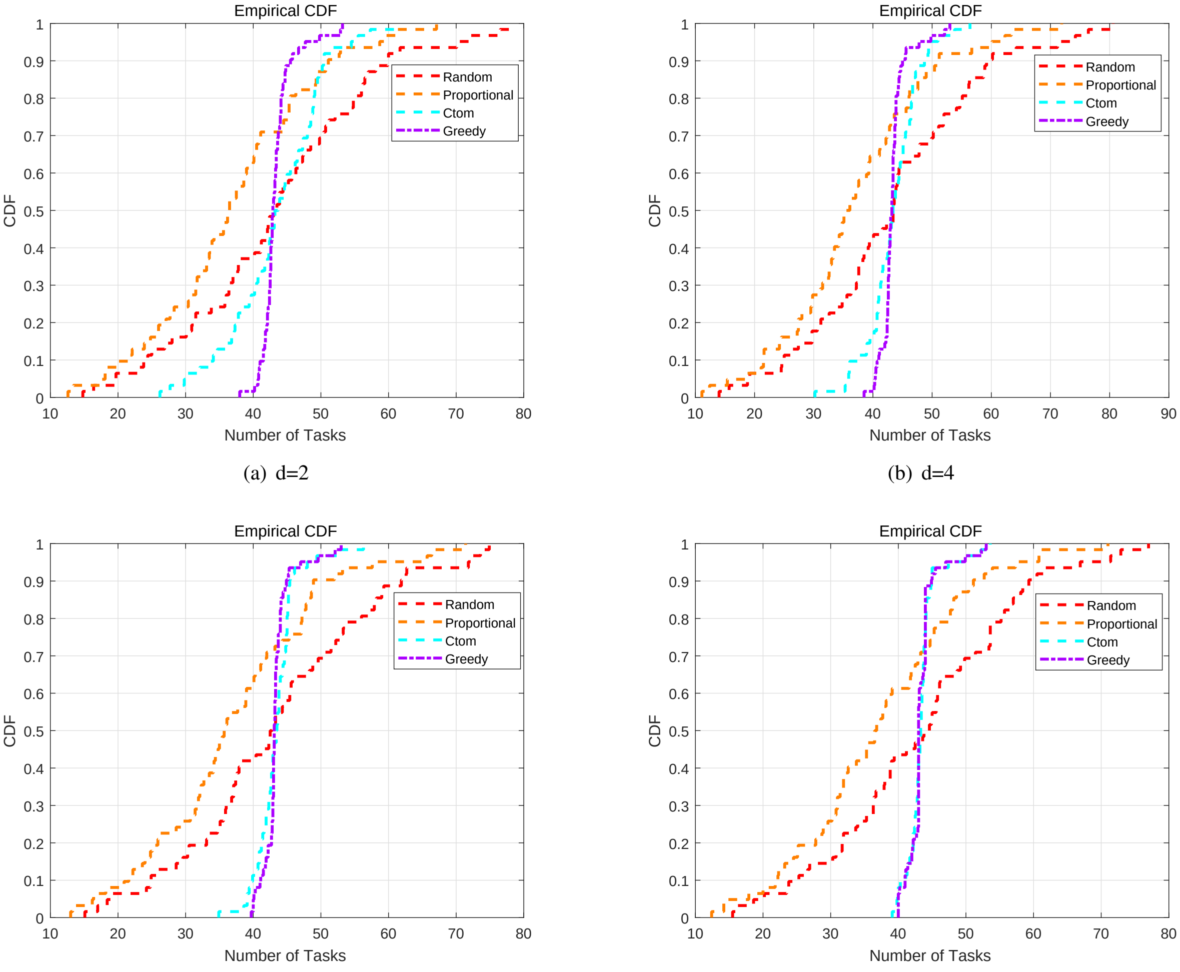
Fig. 11 illustrates the cumulative distribution of the task allocation. In random allocation scheme, more than 30% mobile cloudlets have more than 50 tasks and about 30% others are with less than 30 tasks. This unbalance can result in longer average task response time. Meanwhile, under CTOM scheme, around 95% of cloudlets are allocated with 30-50 tasks, which means that the cloudlets are collaboratively processing tasks and no cloudlets are overwhelmed by DDoS attack. As the CDF line of greedy algorithm is the most centralised, it means that the task loads at different cloudlet only vary within a small range (around 40 to 50).

We also evaluate the percent imbalance metric and statistical skewness. In Figure. 12, still the greedy algorithm achieved the best performance with 0.2 imbalance value, followed by our CTOM with converged results of 0.5. Interestingly, random and proportional allocation perform similarly with imbalance metric at about 1, showing that both of them are not applicable enough for trace-driven mobile cloudlet scenario. In Figure. 13, the skewness obtained with the random allocation scheme fluctuates violently between positive and negative values, which implies that the task loads are continuously unbalanced throughout the process of allocation. While the greedy allocation scheme has achieved the best performance with a skewness of 0, there are overloaded mobile cloudlets under our CTOM scheme and the proportional method, which are revealed by their statistical skewness values of 2 and 3 respectively. We further evaluate the influence of d on trace-driven task allocation. Interestingly, with d increasing from 2 to 16, the proposed CTOM performs more and more close to optimal results by greedy method, with most of mobile cloudlets offloaded with 40 to 50 tasks.

The above simulation and evaluation results validate the effectiveness of proposed algorithm in balancing the tasks among mobile cloudlets. Under CTOM, the number of overwhelmed cloudlets are significantly reduced and the task distribution is well-balanced. In summary, CTOM can effectively tame the potential DDoS attacks and is capable of achieving secure and sustainable task offloading.

VIII. Conclusion

In this paper, we have addressed the DDoS attack problem in mobile cloudlet networks with load balancing. By leveraging balls-and-bins theory, we have devised CTOM, a novel collaborative task offloading scheme for secure and



(a) d=2

(b) d=4

(c) d=8

(d) d=16

Fig. 14. Load analysis in trace-driven with d ranging from 2-16.

sustainable mobile cloudlet networks. The proposed solution can effectively reduce every long task queue in task allocation process and query only limited load information from cloudlets. The simulation and trace-driven evaluation results have demonstrated that, CTOM outperforms the conventional and proportional allocation schemes by 65% and 55% on maximum task gaps respectively. In this way, the potential DDoS attacks that aiming at overwhelming cloudlets are smoothly handled and the computing services are guaranteed for legitimate users.

ACKNOWLEDGMENT

Ning Yang and Xiaochen Fan have equal contributions to this work.

REFERENCES

[1] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[2] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: elastic execution between mobile device and cloud,” in *Proceedings of the sixth conference on Computer systems*. ACM, 2011, pp. 301–314.

[3] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *IEEE pervasive Computing*, vol. 8, no. 4, 2009.

[5] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, “Efficient algorithms for capacitated cloudlet placements,” *IEEE Transactions on Parallel and Distributed Systems*, pp. 2866–2880, 2016.

[6] M. Jia, W. Liang, Z. Xu, and M. Huang, “Cloudlet load balancing in wireless metropolitan area networks,” in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, vol. 27, no. 10. IEEE, 2016, pp. 1–9.

[7] Y. Zhang, D. Niyato, and P. Wang, “Offloading in mobile cloudlet systems with intermittent connectivity,” *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, 2015.

[8] Y. Liu, M. J. Lee, and Y. Zheng, “Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2398–2410, 2016.

[9] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, “Vehicular fog computing: A viewpoint of vehicles as the infrastructures,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[10] H. Cao and J. Cai, “Distributed multi-user computation offloading for cloudlet based mobile cloud computing: A game-theoretic machine learning approach,” *IEEE Transactions on Vehicular Technology*, 2017.

[11] S. Jeong, O. Simeone, and J. Kang, “Mobile edge computing via a uav-

mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, 2017.

[12] D. Yao, L. Gui, F. Hou, F. Sun, D. Mo, and H. Shan, "Load balancing oriented computation offloading in mobile cloudlet," in *Vehicular Technology Conference (VTC-Fall), 2017 IEEE 86th*. IEEE, 2017, pp. 1–6.

[13] H. Cao and J. Cai, "Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 752–764, 2018.

[14] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2018.

[15] Y. Sui, X. Wang, M. Pengt, and N. An, "Optimizing mobility and energy charging for mobile cloudlet," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[16] X. Sun and N. Ansari, "Green cloudlet network: A distributed green mobile cloud network," *IEEE Network*, vol. 31, no. 1, pp. 64–70, 2017.

[17] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2016.

[18] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[19] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, "Ddos attacks in cloud computing: Issues, taxonomy, and future directions," *Computer Communications*, vol. 107, pp. 30–48, 2017.

[20] O. Osanaiye, K.-K. R. Choo, and M. Dlodlo, "Distributed denial of service (ddos) resilience in cloud: review and conceptual cloud ddos mitigation framework," *Journal of Network and Computer Applications*, vol. 67, pp. 147–165, 2016.

[21] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, 2015.

[22] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu, and G. Xu, "A heuristic clustering-based task deployment approach for load balancing using bayes theorem in cloud environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 305–316, 2016.

[23] B. Vöcking, "How asymmetry helps load balancing," *Journal of the ACM (JACM)*, vol. 50, no. 4, pp. 568–589, 2003.

[24] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094–1104, 2001.

[25] F. Xiaochen, H. Xiangjian, P. Deepak, C. Shiping, X. Chaocan, N. Priyadarsi, and R. Xunpeng, "Ctom: Collaborative task offloading mechanism for mobile cloudlet networks," in *Communications (ICC), 2018 IEEE International Conference on*. USA, 2018.

[26] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future generation computer systems*, vol. 29, no. 1, pp. 84–106, 2013.

[27] M. Satyanarayanan, "A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets," *GetMobile: Mobile Computing and Communications*, vol. 18, no. 4, pp. 19–23, 2015.

[28] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1060–1068.

[29] S. Alharbi, P. Rodriguez, R. Maharaja, P. Iyer, N. Subaschandrabose, and Z. Ye, "Secure the internet of things with challenge response authentication in fog computing," in *Performance Computing and Communications Conference (IPCCC), 2017 IEEE 36th International*. IEEE, 2017, pp. 1–2.

[30] S. Venkatesan, M. Albanese, K. Amin, S. Jajodia, and M. Wright, "A moving target defense approach to mitigate ddos attacks against proxy-based architectures," in *Communications and Network Security (CNS), 2016 IEEE Conference on*. IEEE, 2016, pp. 198–206.

[31] Z. A. Baig, S. M. Sait, and F. Binbeshr, "Controlled access to cloud resources for mitigating economic denial of sustainability (edos) attacks," *Computer Networks*, vol. 97, pp. 31–47, 2016.

[32] S. Lee, J. Kim, S. Shin, P. Porras, and V. Yegneswaran, "Athena: A framework for scalable anomaly detection in software-defined networks," in *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*. IEEE, 2017, pp. 249–260.

[33] J. Mirkovic, E. Kline, and P. Reiher, "Resect: Self-learning traffic filters for ip spoofing defense," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 2017, pp. 474–485.

[34] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for ddos detection in cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 130, 2016.

[35] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat ddos attacks in clouds?" *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2245–2254, 2014.

[36] M. Moshref, M. Yu, R. Govindan, and A. Vahdat, "Dream: dynamic resource allocation for software-defined measurement," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 419–430, 2015.

[37] D. Grosu and A. T. Chronopoulos, "Noncooperative load balancing in distributed systems," *Journal of parallel and distributed computing*, vol. 65, no. 9, pp. 1022–1034, 2005.

[38] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," *Journal of parallel and distributed computing*, vol. 7, no. 2, pp. 279–301, 1989.

[39] D. Grosu, A. T. Chronopoulos, and M.-Y. Leung, "Cooperative load balancing in distributed systems," *Concurrency and Computation: Practice and Experience*, vol. 20, no. 16, pp. 1953–1976, 2008.

[40] S. Penmatsa and A. T. Chronopoulos, "Game-theoretic static load balancing for distributed systems," *Journal of Parallel and Distributed Computing*, vol. 71, no. 4, pp. 537–555, 2011.

[41] P. Berenbrink, T. Friedetzky, L. A. Goldberg, P. W. Goldberg, Z. Hu, and R. Martin, "Distributed selfish load balancing," *SIAM Journal on Computing*, vol. 37, no. 4, pp. 1163–1181, 2007.

[42] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. De Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 1116–1124.

[43] Q. Li, P. Yang, X. Fan, S. Tang, C. Xiang, D. Guo, and F. Li, "Taming the big to small: efficient selfish task allocation in mobile crowdsourcing systems," *Concurrency and Computation: Practice and Experience*, 2017.

[44] O. Pearce, T. Gamblin, B. R. De Supinski, M. Schulz, and N. M. Amato, "Quantifying the effectiveness of load balance algorithms," in *Proceedings of the 26th ACM international conference on Supercomputing*. ACM, 2012, pp. 185–194.

[45] N. Alon and J. H. Spencer, *The probabilistic method*. John Wiley & Sons, 2004.

[46] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.