2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications

A Blockchain-Based Offloading Approach in Fog Computing Environment

Wenda Tang[1,2], Xuan Zhao[1,2], Wajid Rafique[1,2], Wanchun Dou[1,2,*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, P. R. China
[2]The Department of Computer Science and Technology, Nanjing University, Nanjing, P. R. China
Email: {wdtang, xzhao, rafiqwajid}@smail.nju.edu.cn, douwc@nju.edu.cn

***Abstract*—With the development of mobile computing technology, there has higher demand for computation resource in mobile applications than before. Fog computing has emerged as a promising infrastructure to provide elastic resources at the proximity of mobile users. Mobile users can offload some computations from the mobile devices to the nearby Fog servers so as to release the workloads of mobile devices, and therefore improve mobile users' quality of experience. However, mobile users may mistakenly offload their computations to the nearby Fog servers which have been injected by some attackers, and therefore induce some privacy and security issues. As most of mobile devices have natural mobility feature, it is very necessary to check the veracity of a Fog server in a very short time before doing computation offloading. In view of this challenge, we bring blockchain technique into Fog environment so as to verify each Fog server's authenticity and propose a blockchain-based offloading approach in this paper. Concretely, the proposed approach constantly maintains a set of candidate authorized Fog servers by leveraging blockchain technology, and the offloading decision could be made in a real-time fashion. Extensive experimental results have demonstrated our method's feasibility and efficiency.**

***Index Terms*—Blockchain, Fog Computing, Computation Offloading**

I. Introduction

Mobile applications and mobile devices are developing rapidly in recent years. Also cloud computing has gained a momentum and is transforming the Internet computing infrastructure. Despite the high popularity of cloud computing in our daily life, some time-sensitive applications and services still cannot benefit from this popular computing paradigm due to its inherent problems, i.e., unacceptable latency and lack of mobility support.

According to [1], the computational tasks of mobile applications in the real world are usually not required to be processed immediately, but rather required to be processed in a certain urgent deadline. Also, taking smart vehicle as a specific mobile device, Kang et al. [2] argued that a single smart vehicle generally has limited computing and storage capability to support many resource hungry applications.

Taking these circumstances into consideration, a new computing paradigm called Edge Computing has been created to fill a vacancy in cloud computing architecture. Edge Computing paradigm provides context aware distributed computing and storage at the edge of the networks, and Fog Computing [3] is one of the most popular implementations of Edge Computing paradigm. Fog Computing deploys lightweight computing facility which usually called Fog servers (a.k.a Fog Computing Nodes) at the proximity of mobile users. Thus, mobile users can offload full or part of their smart devices' computational tasks to the Fog servers to release the workloads, and therefore prolong the battery life of their smart devices.

It should be mentioned that Fog servers are usually distributed outside so as to create one-hop wireless network at the proximity of mobile devices, which means that they are vulnerable to malfunction and intrusions. If there are some Fog servers which have been compromised by attackers, mobile users may mistakenly offload their computational tasks to the nearby Fog servers which have been injected by computer virus or Trojan horse program without authenticity checking. Therefore, these mobile devices of users who want to leverage Fog computing may face some privacy and security issues.

Also, due the natural mobility of mobile devices (especially for high-speed moving vehicles) and the limited service coverage of constant located Fog servers, the performance of required authenticity checking technique should be acceptable. Therefore, some authenticity checking techniques are need to address this issue properly.

As mentioned in recent work [4], to maximize computation offloading efficiency, each mobile device would collect available Fog servers by querying the information from centralized Computing Server when it wants to do computation offloading to the nearby candidate Fog servers. Therefore, the performance of querying is another problem because of the high-concurrent query requests from all the mobile devices in the whole city. It should be also mentioned that this kind of centralized architecture may involve some dependability issues because the fact that some malicious attackers would want to establish Distributed Denial-of-Service (DDoS) attack to further push great network pressure on the centralized proxy, which may induce the whole offloading system breakdown easily.

With this in mind, blockchain is considered as a feasible tool to cope with the problems above. Blockchain is initially known as one of the disruptive technologies in financial industry, which enables distributed nodes to trade with each other and maintain a consistent and tamper-proof ledger without a centralized bank [5]. Due to its natural high security and reli-

* Corresponding author: Wanchun Dou (e-mail: douwc@nju.edu.cn)

978-1-7281-1141-4/18/$31.00 ©2018 IEEE
DOI 10.1109/BDCloud.2018.00056

ability, blockchain has been widely studied in both academia and industry. Some practical applications have been applied in many non-financial scenarios, e.g., decentralized storage [6], decentralized trust management [7], trust-less medical data sharing [8], [9], etc.

Due to the decentralization nature of blockchain, information management can be conducted among distributed Fog servers, which we think can efficiently avoid the security problems of centralized implementation of architecture. Moreover, all Fog servers can work together and maintain a consistent true available service nodes database by leveraging blockchain technology. Also, with blockchain technology, it would be easy for the Fog computing operators to charge their offloading service. Even though a small portion of Fog servers may be compromised by attackers, the block generation speed of attackers is much slower than that of benign Fog servers [7].

The contributions of this paper are two-fold.

1) We proposed a new decentralized Fog server management approach in Fog computing environment based on the blockchain technology. It can enable all the Fog servers to participate in updating the trust information in a decentralized manner.
2) A secure computation offloading trading manner which under the umbrella of the blockchain technology between mobile users and Fog operator is investigated in this paper. It helps Fog operator to do liquidation for its offloading service.

The remainder of this paper is organized as follows. Section II introduces the background of Fog computing as well as blockchain technology, and reviews the existing offloading approaches. In Section III, we make key observation through a computation offloading example. In Section IV, we present details of our proposed approach in Fog computing environment. In Section V, experiments and performance results of our approach are presented, which are followed by summary and future work in Section VI.

II. Preliminary Knowledge

A. Fog Computing

Currently, Fog computing is an emerging paradigm to extend the Cloud service to the "ground" [10]. Sometimes the term "Fog" is used interchangeably with the term "Fog computing". Fog computing extends cloud computing by introducing an intermediate Fog layer between mobile devices and cloud. This accordingly leads to a three-layer Mobile-Fog-Cloud hierarchy [11], and the details could be found in Fig. 1. The intermediate Fog layer consists of geo-distributed Fog servers which locate near enough to the mobile users. These Fog servers aims provide compute, storage and communication resource in the close proximity of mobile users, and they can guarantee high quality of service to mobile users due to the local one-hop distance with high-rate wireless network connections. Fog computing not only reduces the backbone traffic to be sent to the cloud, but also improves the latency for delay-sensitive Internet of Things (IoT) applications by reducing the relatively long delay of remote cloud computing [12].

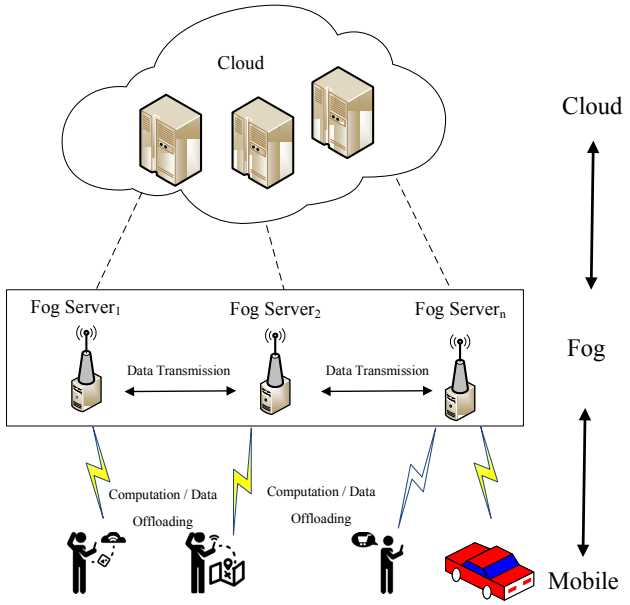


Fig. 1. Three-layer Mobile-Fog-Cloud hierarchy

In Fig. 1, Fog layer locates between Cloud layer and Mobile devices layer. Mobile devices not only can offload data traffic to Fog layer to enlarge their network transmission bandwidths, but also can offload computational tasks to the Fog layer to release their workloads. Any existing network components such as WiFi access point, Femtocell routers can easily change to Fog servers by upgrading their computing and storage resources and reusing the wireless interfaces. In this paper, we mainly focus on the computation aspect of Fog computing.

In summary, the idea of using Fog computing brings computational resource closer to the users, thus improving scalability from computation aspects and quality of experience from mobile user side.

B. Blockchain

Usually, Blockchain is considered as a series of techniques utilized in decentralized networks to achieve a consistent database among all network nodes. It is firstly proposed by Satoshi Nakamoto in order to abstract the core techniques of the well-known digital currency, i.e., the Bitcoin [13]. Decentralized network has its natural advantages in that there are no fixed center nodes in the networks, and all nodes in the network have relatively equal positions and keep the same copy of blockchain which can track all the records. Therefore, no one can change the data recored in the blockchain unless he has obtained strong enough capacity in confuse the crowds [7]. It is said that blockchain enables trust-less networks, because the parties can transact even though they do not trust each other. The absence of a trusted intermediary means faster reconciliation between transacting parties [14].

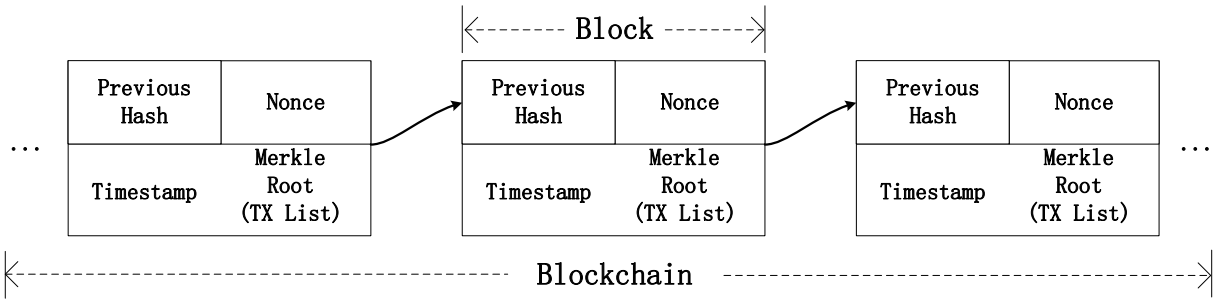


Fig. 2. Typical structure of blockchain

As shown in Fig. 2, a blockchain is an ordered, back-linked list of blocks of transactions. The blockchain can be stored as a flat file, or in a simple database. Each full node in the blockchain network stores the blockchain metadata using Google's LevelDB database. Blocks are linked "back," each referring to the previous block in the chain. The block is made of a header, containing metadata, followed by a long list of transactions that make up the bulk of its size. Each block in the blockchain contains a summary of all the transactions in the block using a merkle tree data structure. Merkle trees are binary trees containing cryptographic hashes [15], and it can efficiently summarize and verify the integrity of large sets of data.

C. Offloading in Fog Computing Environment

Currently, most researchers focus on designing variety of effective offloading approaches (e.g., mobile data offloading [16]–[18], computation offloading [12], [19], [20]) for the resource limited mobile devices.

Gao et al. [16] focused the problem of how to do mobile data offloading from cellular networks to WiFi networks to minimize the total transmission cost from the perspective of mobile users. Likewise, Wang et al. [17] proposed a pricing framework for cellular networks to offload mobile data traffic with the assistance of WiFi network. Specifically, the proposed framework can be utilized to motivate offloading service providers to participate in mobile data offloading, which is a new paradigm to alleviate cellular network congestion and to improve the level of user satisfaction as well. On the other hand, Wang et al. [18] studied the problem of how to offload the mobile cellular traffic by leveraging user-to-user local communications.

Computation offloading in Fog computing has received much attention in recent years due to the growing development of IoT systems. Shah-Mansouri et al. [12] studied the allocation of Fog computing resources to the IoT users in a hierarchical computing paradigm including Fog and remote cloud computing services. According to the evaluation results, the computation time of delay-sensitive IoT applications reduces significantly when utilizing the computing resources of Fog servers. Chang et al. [19] utilized queuing theory to bring a thorough study on the energy consumption and execution delay of the computation offloading process to help mobile devices to make the decision on whether to offload the tasks to the Fog servers nearby. Meng et al. [20] studied the hybrid computation offloading problem considering two types of computation offloading destinations: cloud computing servers and Fog computing servers. Their aim is to minimize the total energy consumption for computation offloading to different offloading destinations while completing the computational tasks within a given delay constraint.

III. A Motivating Example

In this section, we make key observation through a computation offloading example in Fog computing environment. Consider a smart tourist bus with passengers on it moving on the expressway. Suppose the expressway is almost covered throughout by Fog servers' service coverage, and each mobile device at most can offload its computational task to one Fog server. When it comes to make a offloading decision to utilize one of the forthcoming Fog servers, there are two important questions should be considered carefully: (1) Should the vehicle trust the forthcoming Fog servers? (2) Which Fog server should be selected to utilize and how much computation should be offloaded? These questions could be illustrated in Fig. 3.

Actually, the existing workload of each candidate Fog server should be taken consideration because of the fact that the higher workloads of Fog servers induce the longer completion time for the vehicle's computation task. In real life, one Fog server which embedded multi-core CPU could handle multiple requests in parallel. For simplicity, we suppose each Fog server maintains a task queue by collecting each offloading requests, and it handles each computation task in a First-Come-First-Served (FCFS) order or priority-based order. In fact, the Fog servers' actual workloads vary with time in that they could receive different computation offloading requests from different vehicles randomly. Since the workload of each Fog server is fluctuate, each vehicle should check the workloads of every forthcoming Fog servers so as to make better decision to do computation offloading.

Intuitively, to get the information of computation workloads of forthcoming Fog servers in the moving direction, there are two ways to get the information. First, directly communicate with each candidate Fog server so as to negotiate the offloading issue. Second, query the information from proxy which

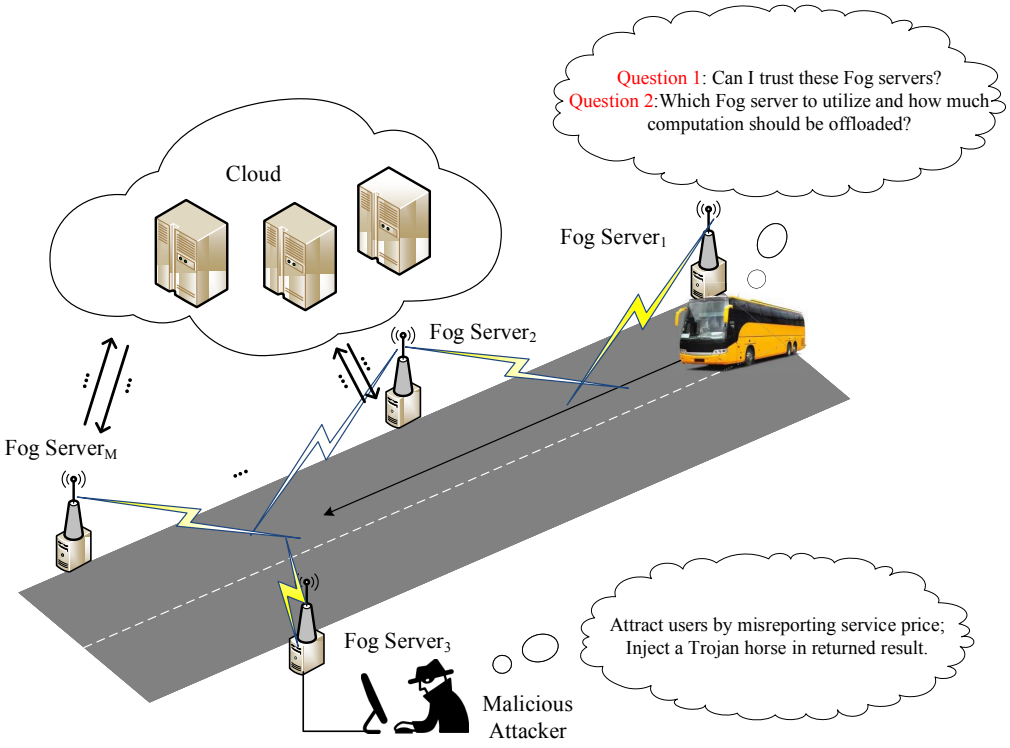


Fig. 3. Choose a Fog Server to offload computation

aggregates all the Fog servers' computation workloads before move into interested Fog server's service coverage.

For the first method, it should be mentioned that both the contemporary near field communication and traditional IP communication technologies are not impracticable to meet the moving vehicle's offloading requirement. For the near field communication (NFC) technology, this kind of communication method need the two electronic devices within 4cm (1.6in) of each other to establish communication, and therefore could not to be considered as the practical solution for high speed mobile smart vehicles. For the traditional IP communication technology, although this kind of communication method could be established between high speed mobile smart vehicles and Fog servers by leveraging IP protocol in advance, the fact that the time delay of communication establishment and also the network workload of target Fog server should not be ignored. Therefore, to communicate with each candidate Fog server to determine the optimal offloading decision is not feasible in a practical way.

For the second method, there is one computing server locating in the cloud layer, and it connects with all Fog servers with good network connectivity by wired network. Mobile users could establish network communication to computing server by cellular network. The computing server acts as proxy to help mobile users query Fog servers' computation workloads. Mobile users could query about his/her interested Fog servers' computation workloads from proxy. When computing server receives a query request from mobile user, it will establish parallel network connections to all requested Fog servers and aggregate the responses from them. To alleviate network pressure, all mobile users only query the information of limited available candidate Fog servers in their moving direction from the centralized proxy. Tang et al. [4] took this kind of method into consideration, and develop a offloading approach based on this.

For each task j arrives in vehicle i at time slot t, suppose the vehicle wants to know several Fog servers' computation workloads, and these Fog servers make up set $\mathcal{R}_j^i(t)$. Also, we use tw_k to denote the waiting time for Fog server k's availability for definiteness and without loss of generality.

As illustrated in Fig. 4, when a moving vehicle wants to know the information about the waiting time list for the forthcoming Fog servers, it should follow these five steps:

1) The vehicle sends the query request to the Computing Server about the $\mathcal{R}_j^i(t)$ Fog servers via cellular network;
2) Computing Server broadcasts the requests to all the $\mathcal{R}_j^i(t)$ Fog servers and waits for all the responses via wired network;
3) Each Fog server k which associated in $\mathcal{R}_j^i(t)$ reports its tw_k to Computing Server via wired network;
4) Computing Server aggregates all the responses and updates the Fog servers' status in its database;
5) Computing Server sends the aggregated response which contains all the query results to the vehicle via cellular network.

However, with more and more mobile devices in mobile

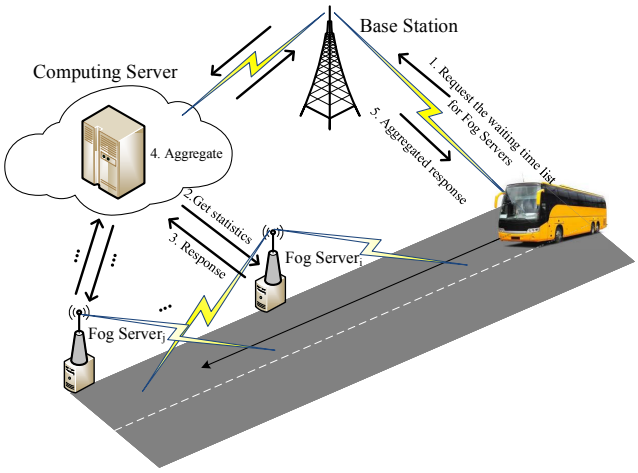


Fig. 4. Collect available Fog Servers by querying the information from Computing Server [4]

layer in Fig. 1, this kind of method may not feasible. The problem of this centralized design method is that the proxy may not able to response to all the query requests quickly enough, and this situation could be illustrated in Fig. 5, the computing server (namely proxy) responses to the query requests from all the vehicles and broadcasts the query requests to pull the updated workloads in Fog servers. Sometimes, the queried Fog server cannot response to the requests immediately due to the fact that it should answer the almost simultaneous requests from different vehicles one after the other. To make things worse, if the response packet from Fog server lost in the network, computing server would not receive the response in time. Computing server should adjust the maximum waiting period for its each broadcast according to the current network condition frequently. Furthermore, some attackers may prefer to establish DDoS attack to further push great network pressure on the centralized proxy, which may induce the whole offloading system breakdown easily.

To this end, all the questions above motivate us to design new management scheme for Fog servers and offloading approach in the Fog computing environment.

IV. Detailed Design of the Proposed Scheme

In this section, we explore how the blockchain can be used in Fog computing environment, and we focus on smart vehicles as the mobile devices in Fog computing environment without loss of generality.

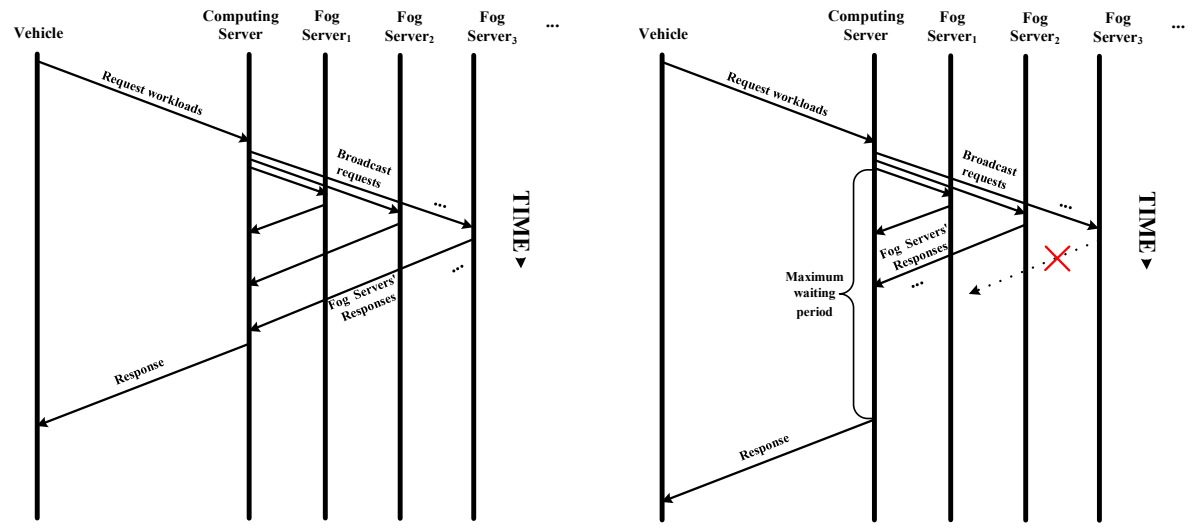
In fact, as mentioned above, with the rapid increase of smart mobile devices, the centralized management schemes for Fog servers may be unpractical. Considering the fact that the Fog servers are distributed among the city area, a small portion of Fog servers may be compromised during a short time period. Data stored in the Fog server may be added, deleted, or tampered by attackers.

In our opinion, the following advantages make blockchain a promising solution for offloading in Fog computing environment.

1) *Decentralization:* Blockchain enables peer-to-peer network on the distributed nodes to cooperate with each other and maintain a reliable database. Therefore, it is nearly impossible to make the entire system crash.
2) *Tamper-proofing:* Each Fog server's actual workload can be recorded in every transaction in the network. If the data stored in the compromised Fog server are tampered, the transaction will not be verified in the miners, and therefore the dirty data would not able to be recorded in the ledger.
3) *Consistency:* Blockchain enables distributed Fog servers to maintain a consistent database. All Fog servers provide the computation resource for the mobile devices and charge the service by generating corresponding transaction in the blockchain. Each transaction record could be tracked on every node in the whole network.

Since computation offloading service is virtual service, to pay for service via electronic coins comes the direct measure. Suppose vehicles pay for the service by using ***Fogcoin*** which can be exchanged in local currency. By using blockchain techniques, Fog servers and vehicles can work as the full-node client which stores the entire history of Fogcoin transaction (every transaction by every vehicle, ever) and every workload update transaction by every online Fog server. Fog servers use their private keys to signature the actual updated workloads, their geographical positions, and the updated information which will be propagated as the workload transaction to the whole peer-to-peer network. The propagation delay on the network could be enhanced by leveraging Bitcoin Clustering Based Ping Time protocol (BCBPT) [21]. On the other hand, vehicles use their private keys to signature the offloading transactions, which involves the actual payment of the offloading service. Since each vehicle stores the entire history of Fogcoin transaction and workloads transaction, it could easily determine which Fog server should be selected to utilize to offload its specific computation by using some offloading strategies.

Because of the fact that mining secures the whole system and enables the emergence of network-wide consensus without a central authority in blockchain, the operators should let some dedicated computing hosts to solve a mathematical problem concurrently to elect the temp centralized node. The first one who solve the mathematical problem will win the election and has the right to add the block which associates with all the transactions after the last block to the blockchain. Considering the fact that the geo-distributed Fog servers have limited computing power and their main tasks are helping mobile devices to release their computation workload by leveraging offloading technology. As illustrated in Fig. 6, different from the typical Bitcoin network, each nodes in our designed scheme let only limited mining nodes whose computing resource are plentiful to validate new transactions and record them on the global ledger, and these mining nodes could be the dedicated high



(a) Ideal query for workloads

(b) Delayed response of actual workloads

Fig. 5. Query for updated workloads of candidate Fog Servers

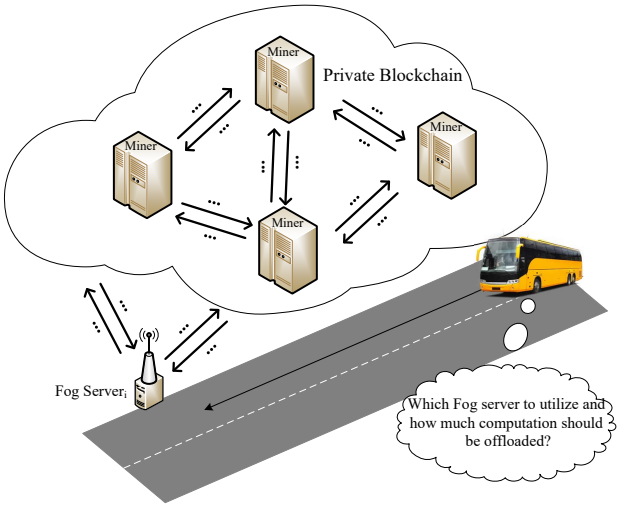


Fig. 6. Blockchain-based offloading environment

performance machines and protected by the service operators.

If one Fog server leaves the network due to power failure, earthquake, etc., its private key would be regenerated when it rejoins the network. The operator can track every ***Fogcoin*** transaction to do liquidation for its offloading service periodically.

As mentioned in section III, Tang et al. [4] argued that vehicles can collect available candidate Fog servers by querying workloads from Computing Server (centralized proxy) in their moving direction. However, the network delay from centralized Computing Server to vehicles are usually unacceptable. In this paper, we hold that each vehicle could easily know every Fog Server's actual workloads at any time by leveraging blockchain technology. Suppose that all Fog servers make up the full set $\mathcal{R}$, and each vehicle at most can offload its current task to one Fog server. Obviously, at a specific time slot t, the number of candidate Fog servers $\mathcal{R}^i(t)$ for vehicle i is quite small. It could be easily found that the lesser network distance between Fog server and mobile device would promote the higher performance of offloading. In common with [21], we argue that two nodes N_i and N_j are considered close to each other if $D_{i,j} \leq D_{thd}$, where $D_{i,j}$ is the distance between N_i and N_j measured by the round-trip latency in the network, and D_{thd} is the latency threshold.

Thus, we can have a two-stage cost effective offloading approach for each vehicle as illustrated in **Algorithm 1**. For each computation task in the moving vehicle, the vehicle could get the optimal Fog server to be utilized in its moving direction by running **Algorithm 1**.

Obviously, the detailed cost effective offloading strategy (**Stage II** in **Algorithm 1**) in this paper is the same as work [4]. Therefore, the effectiveness and feasibility of offloading strategy are also holds. Due to the length limitation of this paper, we omit the complete proof of validity for **Stage II**. Also, it should be also mentioned that the offloading strategy in **Stage II** in **Algorithm 1** could be replaced by other outstanding offloading strategies to achieve other offloading targets. This algorithm's worst-case running time is depend on the size of $\mathcal{R}_j^i(t)$, and the upper bound is $\mathcal{O}(\left|\mathcal{R}_j^i(t)\right|)$.

V. Performance Evaluation

In this section, performance evaluation is conducted to validate the effectiveness and feasibility of the **Stage I** in **Algorithm 1**. We compare it to the existing centralized approach work [4].

Algorithm 1 Two-Stage Cost Effective Offloading

Input: Time slot t, latency threshold D_{thd}, task j, vehicle's moving speed

Output: Optimal Fog server $\hat{k}$ and the corresponding offloading ratio α_j^i

Stage I: Candidate Fog Servers' Generation

Clustering candidate Fog servers with network distance less than D_{thd} as $\mathcal{R}^i_{j,network}(t)$.

Aggregate all the results as $\mathcal{R}^i_{j,geography}(t)$ from the R-tree when the approximate trajectory area is given as the input of the R-tree [4].

$\mathcal{R}^i_j(t) \leftarrow \mathcal{R}^i_{j,network}(t) \cap \mathcal{R}^i_{j,geography}(t)$

Filter out Fog servers whose service coverage cannot cover the transmission process of task j.

Stage II: Selection of the Optimal Fog Server

Maximize the utility function in [4] for each Fog server k in $\mathcal{R}^i_j(t)$ with corresponding optimal $\alpha^i_{j,k}$.

$\hat{k} \leftarrow \arg\max_k u^i_j(t,k)$ and $\alpha^i_j \leftarrow \alpha^i_{j,\hat{k}}$.

return $\hat{k}, \alpha^i_j$

A. Environment Setup

We consider a scenario where 10 Fog servers randomly locate in a 1000-meter road, and each Fog server's service coverage on the road randomly goes from [1, 100] meters. We use ns-3 as a discrete event simulator to evaluate the performance of our approach. In such a simulator, each event is associated with its execution time, and the simulation proceeds by executing events in the temporal order of simulation time. The average network transmission speed is 10 Mb/s in our simulation. Each Fog servers constitute one blockchain-enabled peer-to-peer network via wired cables. The environment context details are illustrated in TABLE I.

TABLE I
THE ENVIRONMENT CONTEXT

	Description
Hardware	Intel Core i5-7300M CPU @2.60GHz and 8.00GB Memory
Software	Windows 10 Home 1803

B. Task Setup

For each task j in vehicle i, we set its deadline constraint $t^{i,M}_j$ to be $\alpha t^{i,L}_j$, where $t^{i,L}_j$ is task j's *Local Execution Time* in vehicle i, and parameter α denotes the sensitivity of task's real-time feature. Actually, the larger α of a specific task can induce more candidate Fog servers to be utilized.

It should be mentioned the fact that both parameter α and D_{thd} would affect the total number of candidate Fog servers ($\left|\mathcal{R}^i_j(t)\right|$) during each time slot t. In practical deployment, D_{thd} could be set as 25ms [21]. To highlight the performance difference of two approaches in terms of network delay, we compare these two approaches under different total number of candidate Fog servers in our simulation. For convenience, TABLE II summarizes our environmental parameter settings in our simulation.

TABLE II
ENVIRONMENTAL PARAMETERS

Parameter	**Value**
The length of road (Meter)	1000
The number of Fog servers	10
The service coverage of each Fog cover (Meter)	[1, 100]
The network channel delay (Millisecond)	20
The package size of each request and response (byte)	100
The number of moving vehicles	[1,15]
The moving speed of vehicles (Km/Hr)	120
The latency threshold (Millisecond)	25
The response time of each Fog servers (Millisecond)	[40, 100]

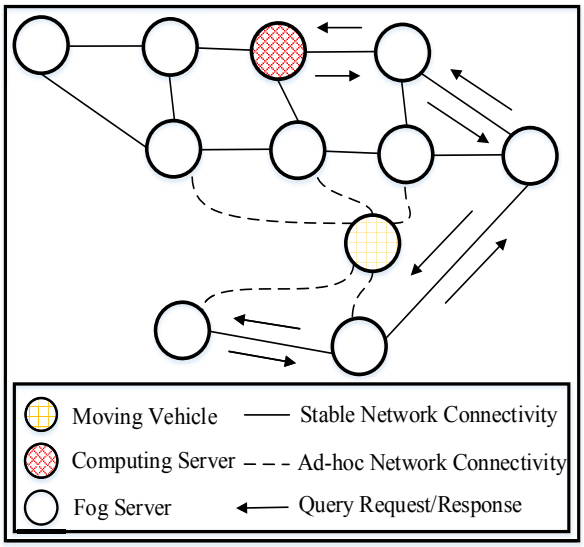


Fig. 7. Network topology of simulation environment

For simplicity, we consider the scenario that one vehicle want to query each candidate Fog server's workload under the network topology as mentioned in Fig. 7. The moving vehicle would like to collect some Fog servers' workloads so as to make comparison and make better offloading decision. For centralized style approach, the moving vehicle should ask computing server for all the candidate Fog servers in $\mathcal{R}^i_j(t)$. For decentralized style approach, the moving vehicle can track all the candidate Fog servers' workloads by querying the blockchain.

Fig. 8 shows the comparison results between these two approaches in simulation environment. For every number of available candidate Fog servers setting, we have ran 10000 trials to calculate both approaches' average query time performance and its [min, max] range (See errorbars in Fig. 8). It is hard not to notice that decentralized style approach has the less average query time delay under different number of candidate Fog servers. Also, with the increasing number of candidate Fog servers during the offloading decision time slot t, both approaches would spend more query time to collect the workloads of all Fog servers. However, in contrast to centralized style approach, the increasing trend of query time

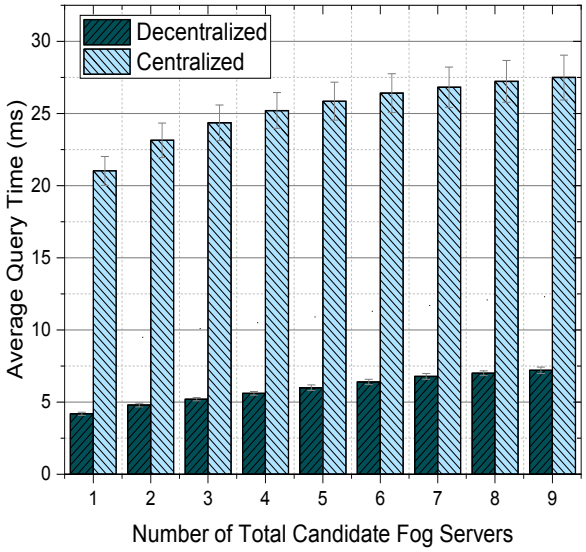


Fig. 8. Comparison results between decentralized and centralized methods in terms of average query time

delay for decentralized style approach is not obvious.

VI. Conclusion and Future Work

In this paper, we bring blockchain technology to Fog computing so as to verify each Fog server's authenticity and create a secure offloading environment. A blockchain-based offloading approach in Fog computing environment which aims to improve the query delay for candidate Fog servers as well as the offloading security was proposed. The simulation results confirm the proposed approach's efficiency and effectiveness. Also, we should admit that blockchain-based approach would have its natural limitation. If Fog server could handles multiple requests in parallel, then all those transactions in its own server and other servers need to be written into the one copy blockchain database in every server. This involves lots of synchronization overhead. Besides, the more transactions processed in the blockchain network, the faster the database size grows.

In our further work, we plan to find better ways to estimate execution time of task whatever it runs in remote Fog server or local vehicle, and therefore improve our approach's accuracy and performance. Besides, the time cost analysis of veracity checking for Fog servers would be considered and evaluated in future work. Moreover, we want to investigate the problem that how to collaborate multiple Fog servers to work together to finish computation tasks, and also investigate how to reduce synchronization overhead in blockchain when Fog server could handles multiple requests in parallel.

Acknowledgments

This work is supported in part by the National Science Foundation of China under Grant No. 61672276, the National Key Research and Development Program of China under Grant No. 2017YFB1400601, and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

References

[1] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.

[2] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. PP, no. 99, pp. 1–11, 2017.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct 2016.

[4] W. Tang, S. Li, W. Rafique, W. Dou, and S. Yu, "An offloading apporach in fog computing environment," in *2018 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2018*, 2018.

[5] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[6] C. Cai, X. Yuan, and C. Wang, "Towards trustworthy and private keyword search in encrypted decentralized storage," in *IEEE International Conference on Communications*, 2017, pp. 1–7.

[7] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2018.

[8] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

[9] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11 676–11 686, 2018.

[10] L. Gao, T. H. Luan, S. Yu, W. Zhou, and B. Liu, "Fogroute: Dtn-based data dissemination model in fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 225–235, 2017.

[11] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2015, pp. 685–695.

[12] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for iot systems: A computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, Aug 2018.

[13] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[14] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[15] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. O'Reilly Media, Inc., 2014.

[16] G. Gao, M. Xiao, J. Wu, K. Han, L. Huang, and Z. Zhao, "Opportunistic mobile data offloading with deadline constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 12, pp. 3584–3599, Dec 2017.

[17] K. Wang, F. C. M. Lau, L. Chen, and R. Schober, "Pricing mobile data offloading: A distributed market framework," *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 913–927, Feb 2016.

[18] X. Wang, M. Chen, Z. Han, D. O. Wu, and T. T. Kwon, "Toss: Traffic offloading by social network service-based opportunistic sharing in mobile social networks," in *INFOCOM, 2014 Proceedings IEEE*, 2014, pp. 2346–2354.

[19] Z. Chang, Z. Zhou, T. Ristaniemi, and Z. Niu, "Energy efficient optimization for computation offloading in fog computing system," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–6.

[20] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud and fog computing," *IEEE Access*, vol. 5, pp. 21 355–21 367, 2017.

[21] M. F. sallal, G. Owenson, and M. Adda, "Proximity awareness approach to enhance propagation delay on the bitcoin peer-to-peer network," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, June 2017, pp. 2411–2416.