

Mobile Intercloud System with Blockchain

Yik Him Ho, Ziv Cheng, Peter Man Fai Ho, and Henry C. B. Chan

Abstract—In this paper, we present a mobile intercloud system with blockchain by combining cloud computing, mobile computing and blockchain-based computing. Each mobile terminal is associated with a virtual terminal for performing mobile cloud computing and other functions. Virtual terminals, as well as data, programs and applications, can be moved across clouds. Furthermore, blockchains can be set up among virtual terminals for sharing data and files in a distributed manner (i.e., without using common servers). A prototype system focusing on the blockchain component has been developed. Performance analysis for the blockchain component is presented, which provides valuable insights into the development of the mobile intercloud system.

Index Terms—mobile cloud computing, intercloud, blockchain

I. INTRODUCTION

IN recent years, there has been considerable interest in three computing technologies, namely: cloud computing, mobile computing and blockchain-based computing. It is of great interest to combine these three technologies to provide a more powerful service. The aim of this paper is to present a framework to realize this goal (i.e., a mobile intercloud system with blockchain). This framework may provide the basis for further research.

According to [1], cloud computing allows the sharing of computing resources, the use of computing services dynamically, service provision using a utility-based scheme, service access through various means, and allocation of resources, based on user demand. Furthermore, cloud computing is provided through three service models, namely: infrastructure as a service, platform as a service and software as a service, and deployed through private clouds, community clouds, public clouds and hybrid clouds [1]. As the number of clouds increases, there is a need to interconnect them [2] [3]. In general, an intercloud system can be created based on three approaches, namely Internet-like, new framework, and overlay. In this paper, we adopt the Internet-like approach based on the IEEE P2302 framework. In essence, clouds can be interconnected to form an intercloud system, similar to how networks can be interconnected to form the Internet.

Mobile computing involves the use of computing through mobile devices. With the advances that have been made in cloud computing and mobile computing, mobile cloud computing has come to the forefront as a research interest (i.e., to extend cloud computing to mobile devices). According to [4], the service models for mobile cloud computing

can be categorized as follows: receiving service from a cloud, functioning as a service provider, and acting as a service broker. Currently, mobile cloud computing focuses on supporting computation offloading, such as for multimedia applications [5] [6]). Due to the limited resources (i.e., processing speed and battery) of mobile devices, it is desirable to offload certain applications to the cloud to enhance system efficiency and to save energy. Most previous works focus on supporting mobile cloud computing over homogeneous clouds (i.e., for intra-cloud operation). Extending intercloud to a mobile environment (i.e., mobile intercloud) provides new research challenges. For instance, in a mobile intercloud system, in addition to mobile terminals, data and applications can also be moved across different clouds (e.g., by means of various handoff processes). Furthermore, clouds operating over different platforms (i.e., heterogeneous clouds) can work in a collaborative manner under a mobile environment.

The concept of blockchain was first introduced as the core technology for supporting Bitcoin [7] [8], one of the earliest cryptocurrencies. In general, blockchain can be viewed as a distributed ledger technology to allow transactions to be processed in a distributed manner with consensus [9]. In recent years, other distributed ledger technologies have been proposed, such as:

- Hyperledger (<https://www.hyperledger.org>)
- Ethereum (<https://www.ethereum.org>)
- Corda (<https://www.corda.net>)

Through an open-source initiative, the aim of hyperledger is to create blockchain platforms for different industries, facilitating industry-based distributed transactions. The key projects include: Burrow, Fabric, Indy, Iroha and Sawtooth (see the aforementioned Hyperledger website for details). Ethereum aims to provide a distributed system for running applications called smart contracts. A special programming language called Solidity is available for developing smart contracts. Corda seeks to develop a distributed ledger system for financial applications using blockchain-inspired technologies. The core elements are state objects, transactions and transaction protocols (see the aforementioned Corda website for details). In this paper, we adopt Ethereum for the mobile intercloud system.

As mentioned above, this paper presents a mobile intercloud system with blockchain by integrating cloud computing, mobile computing and blockchain-based computing. The rest of the paper is organized as follows. Section II gives an overview of the system architecture and protocol. Section III presents a prototype application. Section IV discusses the performance analysis related to the prototype application, with a focus on the blockchain component. Section V concludes the paper.

II. SYSTEM ARCHITECTURE AND PROTOCOL

Fig. 1 shows the system architecture of a mobile intercloud system with blockchain. There are two subsystems, namely

Manuscript received January 8, 2018, revised February 11, 2018; This work was supported by The Hong Kong Polytechnic University.

Yik Him Ho is with the Department of Computing, The Hong Kong Polytechnic University (email: cshho@comp.polyu.edu.hk).

Ziv Cheng is with the Department of Computing, The Hong Kong Polytechnic University (email: 16002231g@connect.polyu.hk).

Peter Man Fai Ho was with the Department of Computing, The Hong Kong Polytechnic University (email: peter-man-fai.ho@polyu.edu.hk).

Henry C. B. Chan is with the Department of Computing, The Hong Kong Polytechnic University (email: henry.chan.comp@polyu.edu.hk).

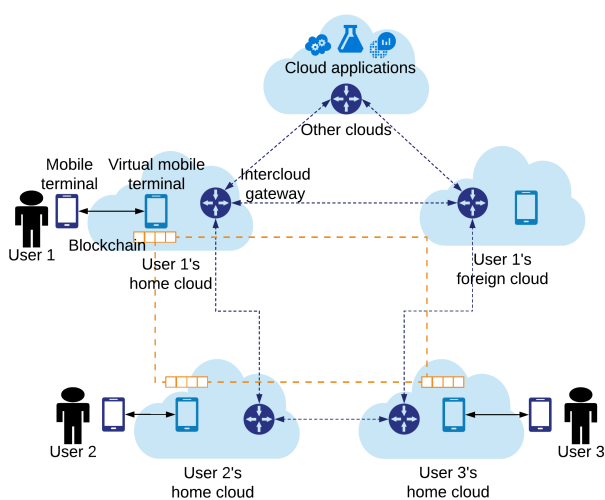


Fig. 1. System architecture

the intercloud subsystem and the blockchain subsystem. Heterogeneous clouds can communicate with each other to deliver different cloud services to users, such as intercloud applications, object storage, and intercloud object transfer. Blockchains can be formed for users to share files in a distributed manner under a server-less environment. In other words, no centralized or common servers are used to store the shared data/files.

A. System Components

The system comprises the following key components:

1) *Mobile terminal*: Every user in the system has his/her own mobile terminal (e.g., smartphones or notebook computers) with data, programs and applications. Typically, a mobile terminal has limited processing power and data storage.

2) *Virtual mobile terminal*: To support mobile cloud computing and other functions, each mobile terminal can be associated with a virtual mobile terminal (or virtual terminal) in a cloud. The virtual terminal (i.e., a virtual machine) has the same environment and settings as the mobile terminal, but it has more powerful resources (e.g., central processing unit (CPU), memory and storage). A mobile user can use the virtual terminal to store more data and run applications. For example, to save power, a physical mobile terminal can offload certain applications to its virtual terminal for processing. The virtual terminal may also run the applications with the assistance of other virtual machines.

3) *Cloud service provider (CSP)*: A CSP offers various cloud computing services, such as storage and platform service to users. In general, a mobile network operator (MNO) can also provide these cloud computing services to mobile users (i.e., serving as a CSP) in addition to offering mobile communications (e.g., voice and data) services (i.e., to extend the existing mobile communications services).

4) *Home cloud*: Inspired by mobile IP, a mobile terminal is associated with a home cloud provided by a CSP or MNO. The primary virtual terminal with user data, programs and applications are set up in the home cloud. As the mobile terminal moves to another cloud (e.g., another city), the corresponding virtual terminal together with certain data,

programs and applications may also be transferred or cloned to that cloud.

5) *Foreign cloud*: When a mobile user travels to another country or city that is far away from the home cloud, he/she can join a foreign cloud and move/clone the virtual terminal to the foreign cloud. The user can also migrate data objects and applications to the foreign cloud (i.e., from the home cloud) so that the communications latency can be reduced. The operational or protocol details will be explained in the following section (see also [10]). Note that in an intercloud environment, the foreign cloud and home cloud may run under different cloud platforms and communicate through intercloud gateways.

6) *Other clouds*: Other clouds may be involved to provide services. For example, these clouds may work in conjunction with the home cloud and foreign cloud to provide specialized cloud services.

7) *Intercloud gateway*: Following the IEEE P2302 draft standard, the mobile intercloud system seeks to support collaboration and communications between heterogeneous clouds (i.e., clouds using different systems). To allow heterogeneous clouds to communicate with each other, intercloud gateways are employed. Basically, they can communicate using an XML-based intercloud communications protocol. For example, CSPs and MNOs in the foreign area may run their cloud services on different cloud platforms. With the intercloud gateways, even though the foreign cloud is not running on the same platform as the home cloud, inter-cloud operations (such as communications and migrations) can still be achieved.

8) *Blockchain*: Private blockchains can be formed in the system for sharing data and files. Note that because mobile terminals have limited computing resources, the blockchains are formed among the virtual terminals (i.e., virtual machines) in the clouds. In other words, all communications and mining processes are carried out by virtual terminals instead of mobile terminals. The use of blockchains not only enables server-less computing for sharing data and files with version management functions, but also ensures data integrity and traceability. A prototype and use case example is presented in Section III.

B. Mobile Intercloud Protocol

The mobile intercloud system is inspired by the mobile internet protocol (i.e., mobile IP [11]) to enable communications and object transfer across heterogeneous clouds. To minimize network latency and access delay, it is beneficial for users to access a local cloud (i.e., as close to the user as possible). For example, when a user moves to a foreign city, the network latency will increase, and thus the user experience is affected. Therefore, we present a mobile intercloud protocol for migrating objects from one cloud to another cloud (e.g., from the home cloud to the foreign cloud). Fig. 2 explains the basic operation of the protocol.

1) *Roaming*: Suppose that a mobile user enters a new location/area. The mobile terminal can find the nearest CSPs and MNOs through a roaming protocol. In a way, this resembles how mobile users can join a WiFi network. For example, after entering a new city, the mobile user should join a foreign cloud for better service (e.g., to shorten access

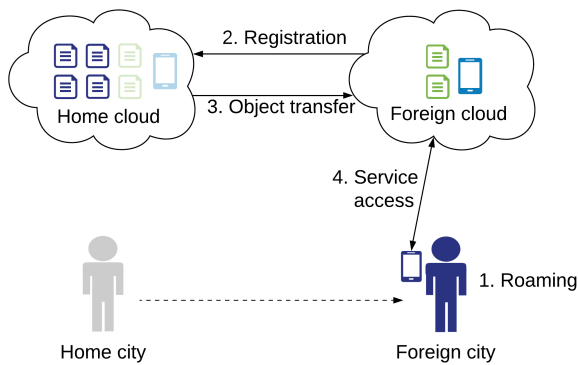


Fig. 2. Mobile intercloud protocol

delay). This is like the roaming cellular service provided by mobile network operators. Indeed, CSPs and MNOs can form an alliance to provide this type of cloud roaming service. Through this service, users can access the home cloud via the foreign cloud when they enter a foreign city.

2) *Registration*: After joining a foreign cloud, a mobile user registers with the foreign cloud and informs the home cloud accordingly. Subsequently, the foreign cloud and home cloud can work together to serve the mobile user. For example, the mobile terminal can offload certain applications to the foreign cloud, and certain data can be transferred from the home cloud to the foreign cloud for processing.

3) *Object transfer*: When a mobile terminal enters a foreign city and joins a foreign cloud, the virtual terminal (i.e., in the home cloud) can also be transferred to the foreign cloud. Alternatively, part of the virtual terminal can be cloned in the foreign cloud (e.g., the essential components). Furthermore, the user can also transfer data, programs and applications, so that user experience and system efficiency can be enhanced. Note that the object transfer can be conducted by the intercloud communications protocol (see [10] for details).

4) *Services and Blockchain*: After the registration phase and object transfer (if required), the mobile user can access services through the foreign cloud. The services can be provided by the foreign cloud (e.g., after transferring the objects) in addition to the home cloud and other clouds. Basically, the foreign cloud functions as a service agent (i.e., to provide required services to the mobile user). If a service can be provided by the foreign cloud, the mobile user can access the service directly. If not, the mobile user needs to access the service from the home cloud through the foreign cloud. In some cases, the foreign cloud, home cloud and possibly other clouds need to work together to provide the required service to the mobile user. Depending on the user requirements, frequently used applications and data may be transferred to the foreign cloud to minimize access delay and to provide better service, as explained in [10]. Furthermore, blockchains may be set up to facilitate data/file sharing among other users (i.e., through the virtual terminals).

5) *De-registration*: When the mobile user leaves the city (e.g., moving to another area/city), he/she needs to deregister from the foreign cloud and inform the home cloud. In some cases, the virtual terminal should be transferred back to

the home cloud for data management or synchronization purposes.

III. PROTOTYPE

In this section, we present a prototype system/application focusing on the blockchain component based on Ethereum. This can be viewed as an extension of the previous work in [10]. Assume that there are three mobile users. Each has a mobile terminal associated with his/her home cloud. As shown in Fig. 3a, user 1 can view the files or data objects through the mobile terminal. If he/she visits a foreign cloud, these data objects or files can be transferred to the foreign cloud following certain transfer rules (see [10]). Furthermore, programs and applications can also be transferred to enhance system efficiency. Here, we assume that the three users may share data or files through a blockchain. Note that the users may be supported through different cloud service providers running different cloud platforms (i.e., it is difficult to set up a central or common data/file server). The blockchain approach provides an effective and secure way for them to share data or files in a server-less environment over a mobile intercloud system. In the example, we assume that user 1 created a text file called `setup.txt` in Fig. 3b. User 1 opened the file located on his home cloud using an Android mobile application. He/she then updated the content of the file and shared the file to user 2 and user 3 through a blockchain (i.e., by pressing the “Publish” button as shown in Fig. 3c). User 1’s virtual terminal then processed the request and submitted it to the system. Subsequently, the file was published through a blockchain among the users. The users could see the file in the blockchain. Furthermore, due to the nature of the blockchain, they could check all of the historical files recorded in the blockchain (e.g., previous versions for version management) as shown in Fig. 3d. A user could also download a historical version of the file from the blockchain to his/her cloud storage, as shown in Fig. 3e. After downloading a file, he/she could change the file and then publish it through the blockchain (i.e., adding a new file to the blockchain). The users could generate and manage different versions of files in the blockchain over the mobile intercloud system. For future work, we are also studying different ways to manage the files, such as assigning different editable rights and designing a voting mechanism for agreeing on changes.

IV. PERFORMANCE ANALYSIS

In this section, we conduct tests on the blockchain component with the following assumptions.

- 1) The network latency is negligible.
- 2) The virtual machines/terminals are based on Windows Server 2016 OS.
- 3) The virtual terminals are well-connected.
- 4) Static routes are used.
- 5) All virtual terminals participate in the block mining.

A. Number of virtual terminals

In this test, we aim to study the effect of the number of virtual terminals on the average transaction time, which is defined as the average time duration between when a message is submitted and when it is published to, or included

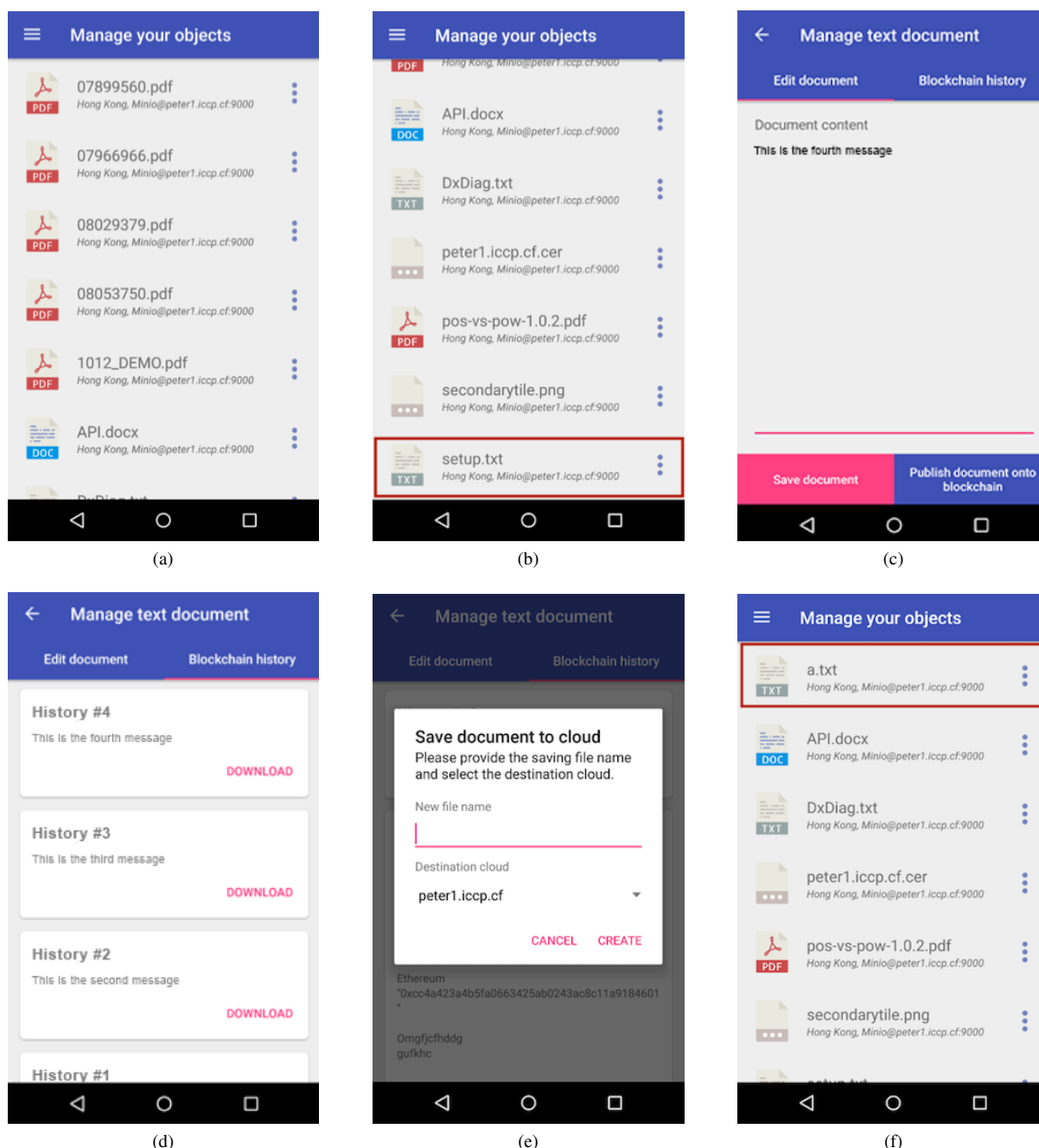


Fig. 3. Screenshots of the prototype application: (a) Listing objects/files stored in the user’s cloud storage. (b) Selecting a file. (c) Editing and processing the file. (d) Viewing the historical files. (e) Downloading the file from the blockchain to the user’s cloud storage. (f) Showing the new file in the users cloud storage.

in, the blockchain. At the beginning of each test, we reset the virtual machines/terminals to measure the transaction time. To minimize the effect of other software, each virtual terminal had the same software configuration. As shown in Fig. 4, the results show that the average transaction time is between 19 and 29 seconds. It shows that the average transaction is not sensitive to the change in the number of virtual terminals (i.e., it is scalable).

B. CPU

In the previous test, it was found that the average transaction time stays relatively constant. In this test, we analyze the effect of computing processing power on the transaction time. In the previous test, each virtual terminal had only

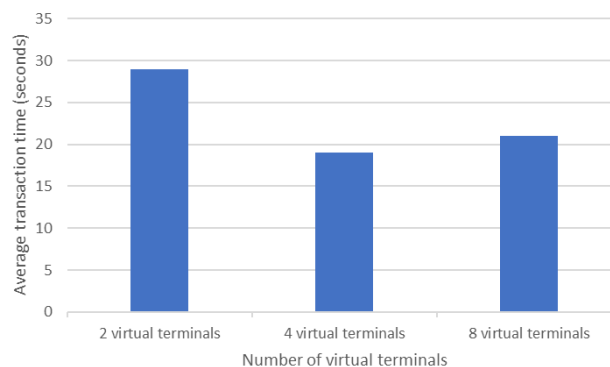


Fig. 4. Average transaction time performance for different numbers of virtual terminals

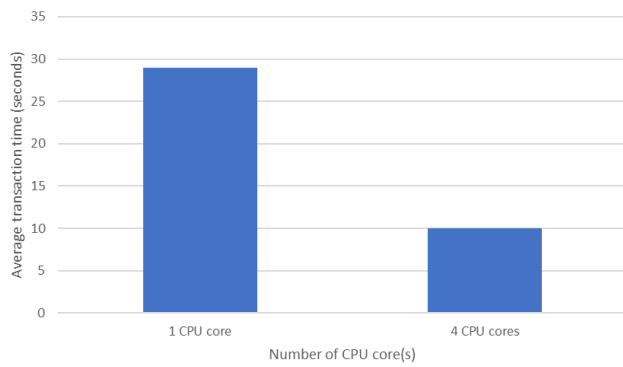


Fig. 5. Average transaction time performance for different CPU configurations

one CPU core. In this test, we created two virtual terminals, each configured with four CPU cores. We repeated the aforementioned tests and measured the average transaction time. As shown in Fig. 5, the results show that with four CPU cores, the average transaction time is reduced significantly. It can be seen that the average transaction time decreases from 29 seconds to 10 seconds.

C. Distribution of Transaction Time

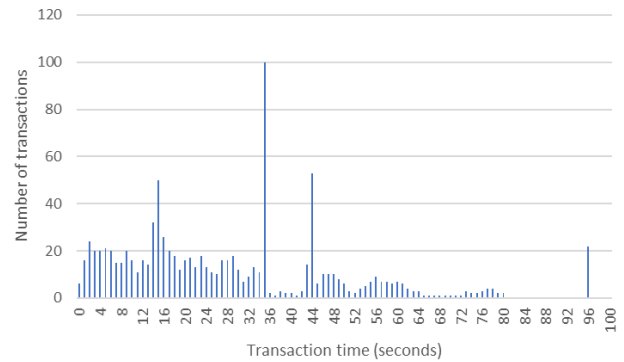
In summary, Fig. 6 and Fig. 7 show the distribution and cumulative distribution of transaction time for the previous tests. It can further be seen that with more CPU cores, the performance is greatly improved. In most cases, more than 50% of the transactions were processed within 30 seconds. Almost all of the transactions were processed within one minute.

D. Content Length

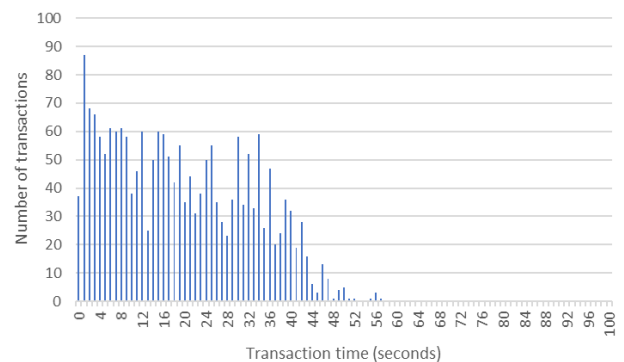
In the previous tests, each message had very few characters (i.e., less than 100). In this test, we seek to analyze the effect of content length on transaction time. We basically repeated the previous tests. However, we allowed 30 minutes idle time for the blockchain to clear the transaction pool before starting a new measurement. Instead of finding the average transaction time, we also found the minimum and maximum transaction times. As shown in Fig. 8, in general, the average transaction time increases when the content length increases. In particular, when the number of characters in a transaction is more than about 5,000, the increase becomes more significant. For example, the average transaction time is about 500 seconds when the number of characters in a message is 5,000. However, when the number of characters in a message becomes 7,000, the average transaction time increases to about 1,000 seconds. The results also show that the maximum transaction time increases even more dramatically as the number of characters increases. For example, when there are 9,000 characters in a message, the maximum transaction time can be close to 5,000 seconds.

E. Block Generation and Difficulty Value

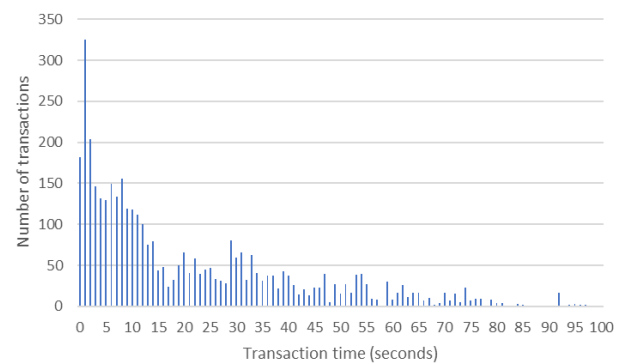
Last but not least, Fig. 9 and Fig. 10 show some examples of the blocks generated per 15-minute time interval and the difficulty value for some blocks. The transactions are



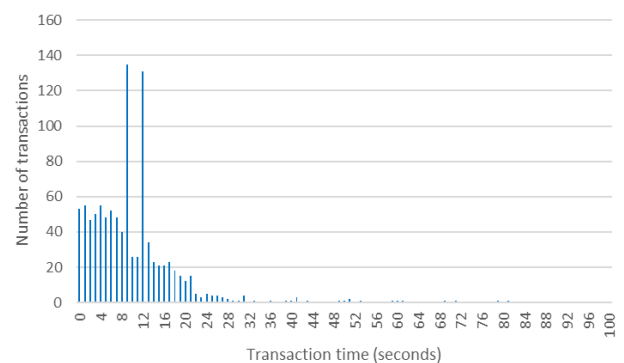
(a)



(b)



(c)



(d)

Fig. 6. Distribution of transaction time for different configurations (a) 1 CPU core, 2 virtual terminals (b) 1 CPU core, 4 virtual terminals (c) 1 CPU core, 8 virtual terminals (d) 4 CPU cores, 2 virtual terminals

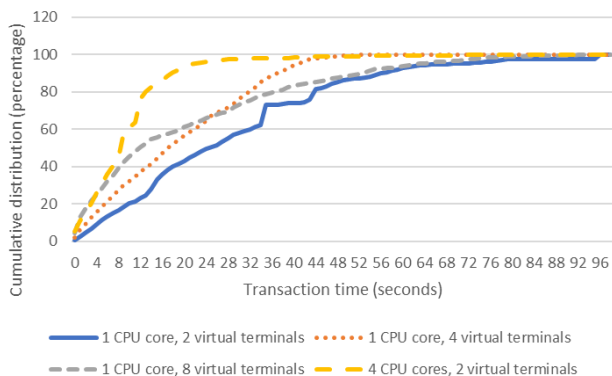


Fig. 7. Cumulative distribution of transaction time

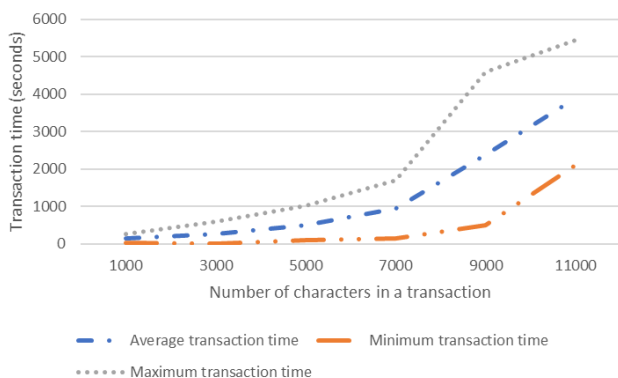


Fig. 8. Transaction time for different content length

published to or included in a block. Note that in Ethereum, “difficulty” is a parameter value used to ensure that blocks are generated at a relatively constant rate, irrespective of processing power. It can be seen that the block generation rate is relatively stable, in accordance with the Ethereum specification. As shown in Fig. 10, the difficulty values are adjusted to maintain the stable or relatively constant block generation rate.

V. CONCLUSION

In conclusion, we have presented a mobile intercloud system with blockchain. The proposed system seeks to integrate cloud computing, mobile computing and blockchain-based computing (e.g., based on Ethereum) to provide innovative

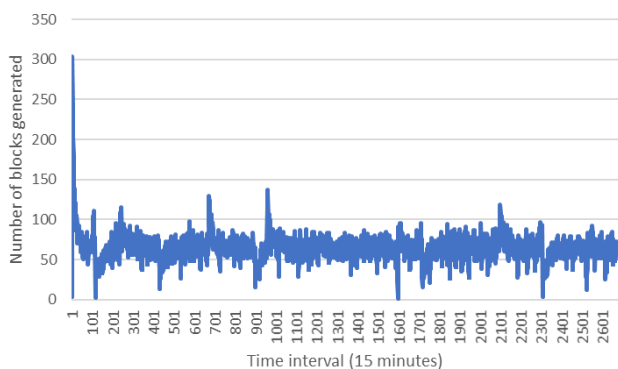


Fig. 9. Number of blocks generated

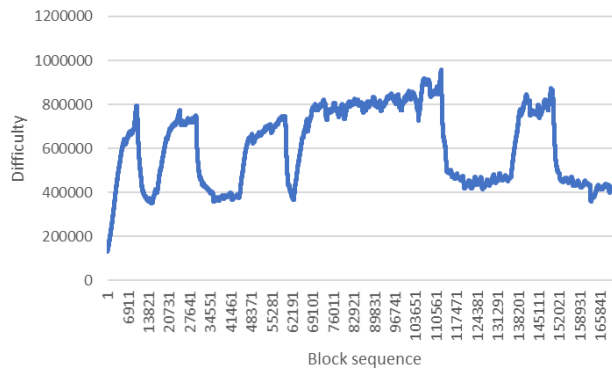


Fig. 10. Difficulty values

mobile services. In particular, data and files can be shared through blockchains. To study the blockchain component, a basic prototype system has been developed. Performance analysis for the blockchain component has been conducted. Further work will be conducted to enhance the system and to extend the prototype.

REFERENCES

- [1] P. Mell and T. Grance, “The nist definition of cloud computing,” 2011.
- [2] A. N. Toosi, R. N. Calheiros, and R. Buyya, “Interconnected cloud computing environments: Challenges, taxonomy, and survey,” *ACM Comput. Surv.*, vol. 47, no. 1, pp. 7:1–7:47, May 2014.
- [3] C. K. Law, W. Xie, Z. Xu, Y. Dou, C. T. Yu, H. C. B. Chan, and D. W. K. Kwong, “System and protocols for secure intercloud communications,” in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, Dec 2016, pp. 399–404.
- [4] D. Huang, T. Xing, and H. Wu, “Mobile cloud computing service models: a user-centric approach,” *IEEE Network*, vol. 27, no. 5, pp. 6–11, September 2013.
- [5] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, “Computation offloading for service workflow in mobile cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3317–3329, Dec 2015.
- [6] S. Yang, D. Kwon, H. Yi, Y. Cho, Y. Kwon, and Y. Paek, “Techniques to minimize state transfer costs for dynamic execution offloading in mobile cloud computing,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 11, pp. 2648–2660, Nov 2014.
- [7] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ, USA: Princeton University Press, 2016.
- [8] 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, thirdquarter 2016.
- [9] T. Aste, P. Tasca, and T. D. Matteo, “Blockchain technologies: The foreseeable impact on society and industry,” *Computer*, vol. 50, no. 9, pp. 18–28, 2017.
- [10] Y. H. Ho, P. M. F. Ho, and H. C. B. Chan, “Mobile intercloud system and objects transfer,” in *2017 IEEE Global Communications Conference (GLOBECOM)*, Dec 2017.
- [11] C. E. Perkins, “Mobile ip,” *IEEE Communications Magazine*, vol. 35, no. 5, pp. 84–99, May 1997.