A Hierarchical Game Framework for Resource Management in Fog Computing

Huaqing Zhang, Yanru Zhang, Yunan Gu, Dusit Niyato, and Zhu Han

The authors propose a three-layer hierarchical game framework to solve the challenges in fog computing networks. In the proposed framework, they apply the Stackelberg sub-game for the interaction between DSOs and ADSSs, moral hazard modeling for the interaction between DSOs and FNs, and the student project allocation matching sub-game for the interaction between FNs and ADSSs.

ABSTRACT

Supporting real-time and mobile data services, fog computing has been considered as a promising technology to overcome long and unpredicted delay in cloud computing. However, as resources in FNs are owned by independent users or infrastructure providers, the ADSSs cannot connect and access data services from the FNs directly, but can only request data service from the DSOs in the cloud. Accordingly, in fog computing, the DSOs are required to communicate with FNs and allocate resources from the FNs to the ADSSs. The DSOs provide virtualized data services to the ADSSs, and the FNs, motivated by the DSOs, provide data services in the physical network. Nevertheless, with fog computing added as the intermediate layer between the cloud and users, there are challenges such as the resource allocation in the virtualized network between the DSOs and ADSSs, the asymmetric information problem between DSOs and ADSSs, and the resource matching from the FNs to the ADSSs in the physical network. In this article, we propose a three-layer hierarchical game framework to solve the challenges in fog computing networks. In the proposed framework, we apply the Stackelberg sub-game for the interaction between DSOs and ADSSs, moral hazard modeling for the interaction between DSOs and FNs, and the student project allocation matching sub-game for the interaction between FNs and ADSSs. The purpose is to obtain stable and optimal utilities for each DSO, FN, and ADSS in a distributed fashion.

INTRODUCTION

Ever since the digital revolution half a century ago, the scale of digital data has grown exponentially. Nowadays, with the high demand of data storage and computing requests, various data services and applications have been proposed to facilitate businesses and our daily lives. However, the traditional rigid deployment of data centers by data service operators (DSOs) is unable to fulfill the requirements of various data services and applications. To improve the flexibility and efficiency of resource allocation, the concept of cloud computing is advocated, where all the resources can be organized as a sharing pool, and authorized data service subscribers (ADSSs) can access the resource pool on demand.

Nevertheless, for some emerging data services and applications, such as vehicle-to-vehicle communication, augmented reality, and smart grid, not only the volume of resources, but the service delay and delay jitter determine the quality of service (QoS) [1]. Moreover, most resources in cloud are physically located far from ADSSs, failing to support the requirements of mobility and real-time interactions during the data services. Accordingly, in order to improve QoS for ADSSs, it is necessary to pull the computing resources closer to ADSSs [2].

In 2014, the idea of fog computing was first proposed by Cisco [3]. Fog computing, similar to cloudlet edge computing proposed by other companies, is composed of geo-distributed fog nodes (FNs), which can be any fixed or mobile collaborative devices with built-in data storage, computing, and communication devices. Benefiting from small scale, low cost, and mobility, the FNs located around ADSSs are able to offload data traffic from the cloud, reduce the communication cost in the networks, and provide real-time, location-aware data services [4].

For the system architecture shown in Fig. 1, there are multiple DSOs serving multiple ADSSs at the same time. In order to improve QoS of the ADSSs, fog computing is introduced in addition to cloud computing. However, as the resources in FNs are owned by independent users or infrastructure providers (InPs), the ADSSs cannot connect to and access data services from the FNs directly. Currently, the ADSSs can only request data service from the data service operators (DSOs) in the cloud, such as Amazon S3, Google Cloud, and IBM Cloud. Therefore, the DSOs are required to communicate with the FNs and allocate resources from the FNs to the ADSSs. Accordingly, in fog computing, the DSOs provide virtualized data services to the ADSSs, and the FNs, after the communication with the DSOs, provide data services in the physical network [5].

With the introduction of fog computing, the resource allocation problem becomes complicated and challenging since there are multiple distributed and autonomous entities in the network. In order to solve the problem, various optimization methods have been adopted in the literature. In [6], the joint radio and computing resource allocation in fog computing was studied by solving the formulated optimization problem in a dis-

Digital Object Identifier: 10.1109/MCOM.2017.1600896

Huaqing Zhang and Zhu Han are with the University of Houston; Yanru Zhang is with the Chinese University of Hong Kong; Yunan Gu is with Huawei; Dusit Niyato is with Nanyang Technological University.

tributed fashion. Furthermore, being aware of the ADSSs' locations with fog computing, dynamic adaptation of computing resources was proposed by [7].

However, the fog computing architecture considered in prior work is based on a single DSO scenario, which simplifies the system architecture and lacks generality. Following the sequential decision making behaviors for the DSOs, FNs, and ADSSs, we propose a three-layer hierarchical game framework with the following three sub-games:

- We first introduce the Stackelberg sub-game for the interaction between the DSOs and the ADSSs to solve the virtualized resource allocation problem. The key problem is the pricing mechanism.
- Then, according to the amount of requested virtualized resources, in order to motivate the FNs to offer the optimal amount of virtualized resources, the moral hazard model in contract theory is utilized to model the interaction between the DSOs and the FNs. The key problem is the incentive mechanism design between the DSOs and the FNs.
- Based on the physical resources offered and virtualized resources provided, the student project allocation matching sub-game from matching theory is adopted to achieve a stable resource allocation solution. The key problem is resource matching in a distributed way that is combinatorial in nature.

The rest of this work is organized as follows. In the following section, we discuss the challenges in fog computing. Based on the challenges, we propose a hierarchical game framework to model the three-layer architecture, where the interactions between different parties, which are the FNs and ADSSs, the DSOs and ADSSs, and the DSOs and ADSSs are then analyzed. Finally, the article is concluded.

Resource Allocation Challenges in Fog Computing

In cloud computing, only DSOs and ADSSs exist in the network. Thus, the resource allocation problem between the DSOs and ADSSs is a straight two-layer structure, where there is a market for all DSOs to compete for ADSSs. In [8], the authors introduced an in-depth game theoretic study of the market and provided pricing strategies for DSOs to achieve Nash equilibrium solutions. However, when the intermediate fog layer is introduced, the relation among DSOs, FNs, and ADSSs becomes complicated. In this section, based on the general system shown in Fig. 1, we classify and discuss challenges of resource allocation in fog computing.

The Interactions between DSOs and ADSSs

The interaction between DSOs and ADSSs in fog computing is similar to the interaction in cloud computing. When there is one DSO serving ADSSs in the network, the DSO is able to adjust its price to motivate all ADSSs to purchase its virtualized resources. In fact, there is a trade-off when the DSO sets the price. On one hand, if the DSO sets a high price, the DSO is able to receive high revenue from the unit amount of resource,

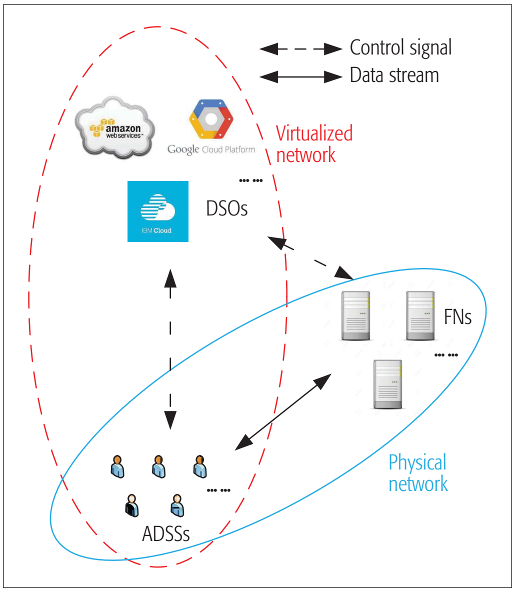

Figure 1. System architecture.

As most of the FNs are deployed and maintained by private users or independent InPs, the FNs will not directly provide services to ADSSs to help relieve the computation load at DSOs. Therefore, the DSOs need to motivate the FNs to help serve the ADSSs by paying a certain amount of monetary rewards to the FNs.

while the ADSSs may choose to purchase less resources considering the high unit price. On the other hand, if the DSO sets low prices, although a large amount of resources are purchased by ADSSs, the revenue gained from the ADSSs may decrease.

When there are multiple DSOs serving at the same time, competition among DSOs exists in the fog computing market. Thus, there is also a trade-off when DSOs compete for the ADSSs. On one hand, if one DSO increases its price, the revenue from serving one single ADSS may increase, while the ADSS may refuse its service and switch to other DSOs with lower prices. On the other hand, if one DSO reduces the price, more ADSSs will be attracted, but the revenue received from each ADSS decreases. Therefore, it is important for each DSO to set an optimal service price to achieve the highest total revenue.

The Interactions between DSOs and FNs

As most of the FNs are deployed and maintained by private users or independent InPs, the FNs will not directly provide services to ADSSs to help relieve the computation load at DSOs. Therefore, the DSOs need to motivate the FNs to help serve the ADSSs by paying a certain amount of monetary rewards to the FNs. The DSO aims to maximize its revenue by purchasing the exact amount of computing resources needed by the FNs. On one hand, if not enough resources are purchased, even though less payment is required, such insufficient resources will cause poor QoS to ADSSs and result in poor revenue in the future. On the other hand, if physical resources are over-supplied, even though high QoS will attract more ADSSs, the high payment will decrease the DSO's revenue. Thus, it is challenging for the DSO to determine an optimal amount of resources to purchase from each FN. In order to maximize the revenue while minimizing the payment, each DSO needs to design an efficient incentive mechanism so that the DSO's objective revenue maximization can be achieved, and the FNs still have the incentive to participate in such an activity.

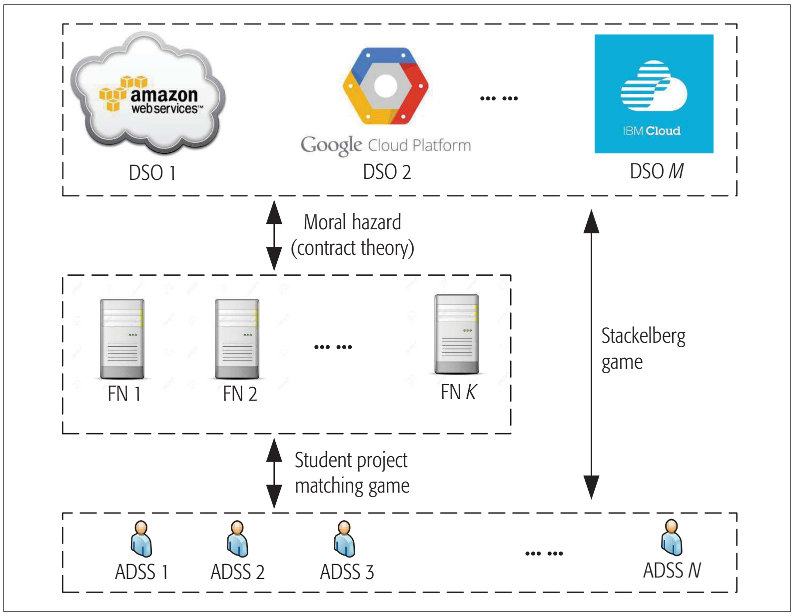

Figure 2. Hierarchical game framework.

The Interactions between FNs and ADSSs

When the ADSSs have determined the amount of virtualized resources needed to purchase, and the FNs have been motivated to supply physical resources, the next step is to efficiently map the physical resources offered by the FNs to the virtualized resources required by the ADSSs. Since the FNs compete with each other for serving ADSSs, and the ADSSs compete with each other for better services. The ultimate goal is to find an optimal matching between the physical resources offered by the FNs and the virtualized resources requested by the ADSSs so that the network can achieve maximum efficiency. Given the large numbers of ADSSs and FNs, the optimal mapping will be hard to solve, or even unsolvable using traditional centralized methods. It is essential to find a sub-optimal stable mapping by a distributed method so that the computation complexity can be reduced, and none of the FNs or ADSSs can deviate from its current mapping to achieve a higher revenue.

The Hierarchical Game Framework

Considering the aforementioned challenges of resource allocation in fog computing, each DSO, FN, or ADSS can be regarded as a rational individual that aims to maximize its own revenue based on the behaviors of all other DSOs, FNs, and ADSSs. Therefore, game theory can be adopted as a suitable mathematical tool to analyze the competition and coordination among all DSOs, FNs, and ADSSs [9, 10].

In fog computing, all DSOs, FNs, and ADSSs form a three-layer architecture and make decisions sequentially. Thus, we model the network as a hierarchical game as shown in Fig. 2, which consists of three sub-games corresponding to the interactions between DSOs and ADSSs, between DSOs and FNs, and between FNs and ADSSs in a sequential manner.

In the following, adopting the backward induction, we sequentially analyze the interactions between different parties, which are the FNs and ADSSs, the DSOs and FNs, and the DSOs and ADSSs, so as to obtain optimal strategies for the DSOs, FNs, and ADSSs with stable and optimal payoffs.

Student Project Allocation Matching Game Analysis for the Interactions between FNs and ADSSs

In this step, we suppose the initial values of the offered resource from each FN and requested resources from each ADSS are given. We consider the resource to be composed of both computation/storage resources and radio resources from different FNs, which are combined as a resource pair and mapped to requested ADSSs.

The mapping between the resource pairs from FNs and ADSSs can be suitably modeled as a student project allocation problem [11], which is a many-to-many stable matching problem and can be described as follows. In many departments of universities, students are required to undertake a series of projects from classes or research by lecturers. The students have preferences among the offered projects. Considering the purpose of training students, for different projects, the lecturers may have preferences over (student, project) pairs according to their suitability. In addition, not only each lecturer but also each project has restrictions on the maximum number of students to accommodate, which are called their capacities. In order to find a stable matching between the students and projects, the SPA-(S,P) algorithm can be utilized.

In this article, we assume the DSOs, resource pairs, and ADSSs act as lecturers, projects, and students, respectively. The preference list of each ADSS is built based on the total revenues gained from the data transmission minus the penalty of service latency as well as the payment to the DSO for the services. On the other hand, the preference list of the DSOs is the mandatory revenue collected from the ADSS minus the cost of service delay using a certain resource pair. If the ADSS requires high QoS, the ADSSs are willing to offer high payment for better resources. At the same time, the DSOs give higher priorities to those ADSSs that offer higher prices by allocating better resources for them. Thus, each DSO establishes its preference list as the ratio of price collected from an ADSS over its service delay.

With the preference lists set up, the SPA-(S,P) algorithm can be adopted to find a stable matching between FNs and ADSSs [12]. In the algorithm, ADSSs first propose to their currently most preferred resource pair in their preference lists. For each resource pair, if the requested proposal from the ADSSs exceeds its capacity, the DSO will find the worst combination of FNs and ADSSs in its preference list and reject this ADSS. Receiving the rejected notification, the ADSS will continue to engage with the next favorite resource pair in its preference list. The procedure terminates when all ADSSs are either matched with a resource pair or have proposed to every resource pair in their preference lists. By the sequential proposing and rejecting actions of ADSSs and DSOs, the convergence of the algorithm is guaranteed, and a stable matching result exists.

In Fig. 3, we evaluate the performance of transmission delay with the proposed SPA-(S,P) algorithm. When the number of ADSSs increases, due to the limited amount of computing resource blocks (CRBs), the ratio of ADSSs satisfying delay requirement generally decreases. However, compared to the random matching result, the algorithm that matches SPA-(S,P) is able to maintain the high ratio, where most of the ADSSs are able to achieve low transmission delay and high utility.

Moral Hazard Game Analysis for the Interactions between DSOs and FNs

According to the possible matching results between the physical resources provided by FNs and the virtualized resource requested by the ADSSs, given the total amount of virtualized resource, each DSO is required to consider the motivation strategy for each FN to provide the optimal amount of physical resource and achieve a high utility.

The motivation problem from DSOs to FNs can be modeled as a moral hazard in contract theory [13]. The problem arises when both parties have incomplete information about each other. For example, the employees' actions are hidden from the employers [14]. As the DSOs do not know the resource usage information within different FNs, if one DSO offloads its data services to FNs with limited available resource, the ADSSs will suffer poor QoS and will switch to other DSOs. Therefore, such an asymmetric information situation between DSOs and FNs will severely reduce the utility of both DSOs and ADSSs.

In order to avoid such a situation, considering the amount of physical resource provided from one FN to one DSO and the payment from the DSO to the FN, we propose a resource-payment bundle in the contract between DSOs and FNs. In order to motivate the FNs to provide larger amounts of physical resource, the DSO is required to pay more to the FNs correspondingly. Furthermore, according to different requirements of its serving ADSSs and different transmission delay between the FN and serving ADSSs, the relations between the amount of provided physical resource and the payment for different DSO and FN pairs may be different.

In [14], we evaluate the motivation strategy from each DSO to FN with contract theory. In order to motivate FNs to serve ADSSs, when one FN agrees to provide resources for ADSSs, the DSO will pay a fixed amount of money to the FN. Furthermore, if the ADSS is able to provide more resource to improve the QoS of the ADSSs during the service, the DSO will offer an additional bonus. Accordingly, the rent from each DSO to FN can be defined as a combination of the fixed payment plus bonus for providing ADSSs with higher QoS. The utility of each FN is the total rent paid by the DSOs minus the operation and measurement costs. The utility of each DSO is denoted as the revenues from ADSSs minus the rent to FNs. Aiming to maximize the utility of each DSO based on the selfish behaviors of FNs, we obtain the optimal value of rent for each FN in a contract.

In the simulation, we compare our proposed payment plan with four other plans. In the single bonus plan, we assume each FN can offer at most one CRB to the DSO. In the stochastic independent plan, we assume the measurement error from the DSOs to all FNs is zero. For the technologically independent plan, we consider the cost for adopting each CRB within each FN is independent of each other. The independent plan combines both stochastic independent and technologically independent plans. As shown in Fig. 4, when the cost coefficients of CRB within FNs increase, as the DSO is required to pay more to motivate FNs, the utility of the DSO generally decreases for all plans. Moreover, with the amount of asymmetric information between DSOs and FNs increasing, the utility of DSO decreases. Thus, the utility of the DSO in the independent payment plan is the highest, followed by the utilities in the stochastic independent plan, technologically independent plan, and our proposed plan. The single bonus plan has the lowest utility due to the limited amount of offered CRBs in each FN.

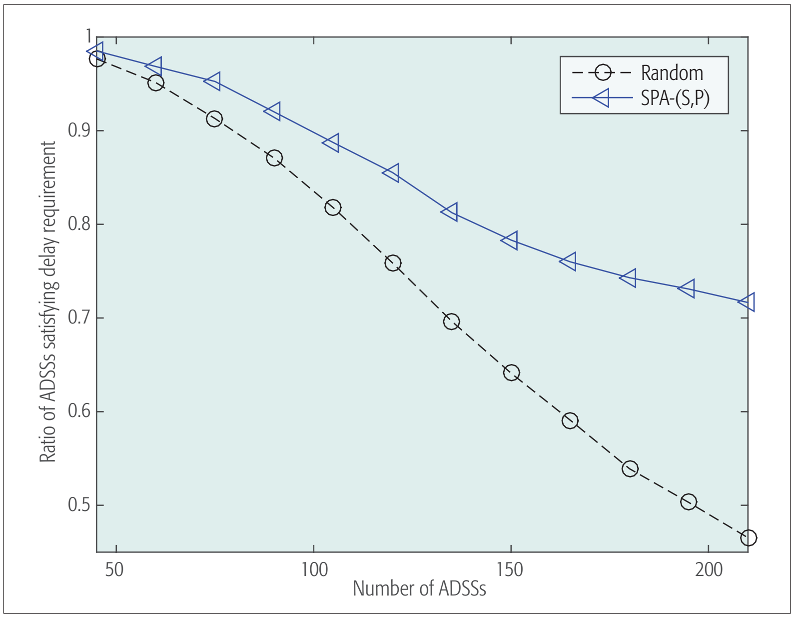

Figure 3. The performance evaluation of SPA-(S,P).

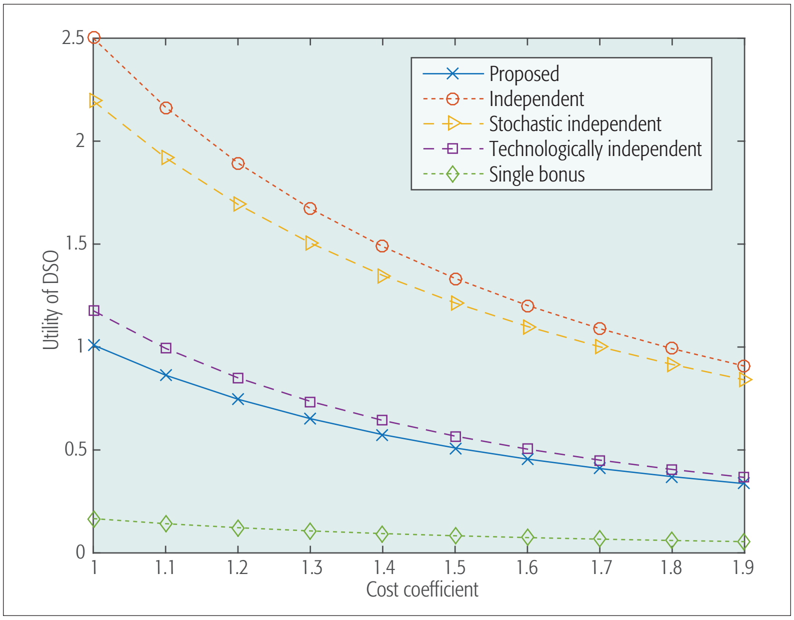

Figure 4. The performance in moral hazard between DSOs and FNs.

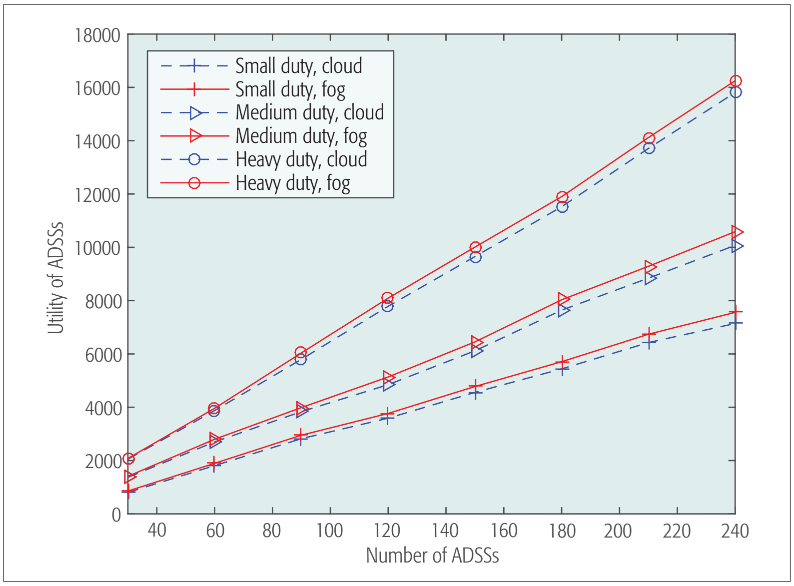

Figure 5. The utility of all ADSSs vs. the number of ADSSs.

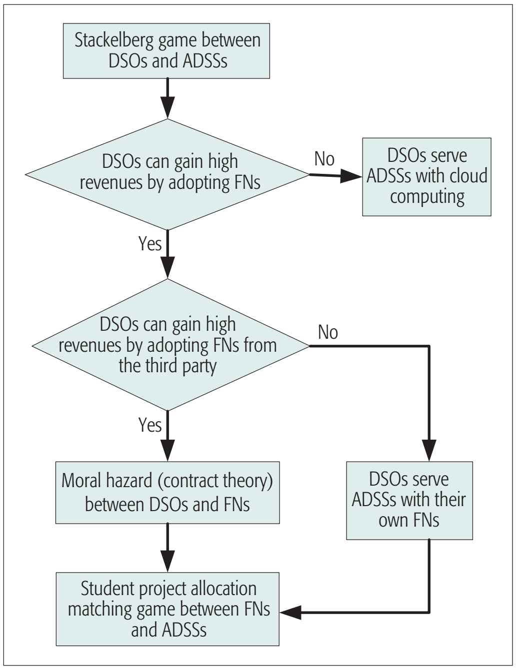

Figure 6. The flowchart for the three-layer hierarchical game.

Stackelberg Game Analysis for the Interactions between DSOs and ADSSs

Predicting the payment for motivating the FNs and the matching results between the provided physical resources and the requested virtualized resource, each DSO determines its optimal service price charging to its serving ADSSs.

As in the virtualized data market, all DSOs first announce their prices for their virtualized services. Based on the announced prices, each ADSS chooses its preferred DSO and determines the optimal amount of resources to purchase. The interaction between DSOs and ADSSs can be modeled as a Stackelberg game, where the DSOs act as the leaders, and the ADSSs act as the followers. In the game, because of the first-mover advantage, each DSO is able to predict the reactions of its serving ADSSs and determine its optimal price for the highest utility. Thus, the Stackelberg equilibrium exists between DSOs and ADSSs [5, 15].

Moreover, the competition also exists among DSOs. If one DSO sets its price too high, the ADSSs may switch to other DSOs for high utility. Therefore, there is also a non-cooperative game among all DSOs. In order to attract more ADSSs and maintain high utility for each DSO, following the proposals in [5], a sub-gradient algorithm can be adopted. In the algorithm, each DSO does not know the existence of other DSOs at the beginning and sets its initial service price at a high value. Then each DSO predicts the utility that it can achieve by adjusting its price with a small value $\pm\Delta$. If adjusting the price is able to improve the utility, the DSO will follow the adjustment in the next round. Otherwise, in the next round, the DSO keeps the current price unchanged. The game continues with reduced value of Δ until no DSO is able to adjust prices to achieve a higher utility.

According to the proposed hierarchical game framework, we evaluate the performances of ADSSs in fog computing, given the cost of motivating the FNs and the matching results between FNs and ADSSs. As shown in Fig. 5 [15], with the number of ADSSs increasing, regardless of the computing data size for each ADSS, the total utility of ADSSs generally increases. Furthermore, considering the fixed service price of the DSO and the same computing data size, because of the low transmission delay, the utility of the ADSS in fog computing can be better than the utility in cloud computing. However, in fog computing, the DSOs are able to set high prices in the virtualized network to gain high revenues. With our proposed algorithm, we guarantee that the utility of the ADSS in fog computing is always higher than that in cloud computing, and the improvement gap of the ADSSs' utility from fog computing to cloud computing is small.

The Three-Layer Hierarchical Game for Fog Computing

From the interaction analyses between DSOs and FNs, between DSOs and ADSSs, and between FNs and ADSSs, a general data service with the proposed three-layer hierarchical game can be summarized as in Fig. 6. In the flowchart, the DSOs first play the Stackelberg game with all ADSSs, and the DSOs are required to make decisions based on the investigation of all the FNs' and ADSSs' information. If the DSOs cannot gain higher revenues by adopting the FNs, the cloud computing will be adopted where FNs will not be included in the data services. If the fog computing is able to bring high profits (i.e., revenue minus costs) for the DSOs, the DSOs then consider whether or not renting the FNs from the third party is beneficial. If the DSO is able to achieve higher utility by renting the FNs from the third party, moral hazard can be applied to motivate the FNs. Otherwise, the DSOs will apply their own FNs for the data services. Finally, with the provided resources from the FNs and the requirements from the ADSSs, the student project allocation matching game is employed to achieve stable results.

Conclusions

In this article, we have proposed a three-layer hierarchical game framework for resource management in the multi-DSO, multi-FN, and multi-ADSS scenario. In the proposed framework, we have first introduced a Stackelberg game between DSOs and ADSSs, where DSOs act as the leaders, providing virtualized services to ADSSs, the followers. Second, based on the total amount of requested virtualized resources, a moral hazard model in contract theory is adopted between the DSOs and FNs to motivate the FNs to offer efficient physical resources. Third, based on the offered physical resources and provided virtualized resources, a student project matching game has been proposed for resource allocation. Finally, based on the hierarchical game framework, we have summarized our work and shown it in the flowchart in Fig. 6.

Acknowledgment

This work was supported in part by the U.S. National Science Foundation Nos. CPS-1646607, ECCS-1547201, CCF-1456921, CNS-1443917, ECCS-1405121, NSFC61428101, and ECCS-1554576.

References

[1] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing: Concepts, Applications and Issues," *Proc. 2015 Wksp. Mobile Big Data*, Hangzhou, China, pp. 37–42.

[2] Y. Mao *et al.*, "Mobile Edge Computing: Survey and Research Outlook," arXiv preprint arXiv:1701.01090, 2017.

[3] CISCO, "Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are," white paper; https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf, accessed Apr. 2015.

[4] I. Goiri et al., "Intelligent Placement of Datacenters for Internet Services," *Proc. IEEE ICDCS*, Minneapolis, MI, June 2011, pp. 131–42.

[5] H. Zhang *et al.*, "Fog Computing in Multi-Tier Data Center Networks: A Hierarchical Game Approach," *Proc. IEEE ICC*, Kuala Lumpur, Malaysia, May 2016.

[6] G. S. S. Sardellitti and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Trans. Signal Info. Processing over Networks*, vol. 1, no. 2, June 2015, pp. 89–103.

[7] Z. Jiang *et al.*, "Improving Web Sites Performance Using Edge Servers in Fog Computing Architecture," *Proc. 2013 IEEE 7th Int'l. Symp. Service Oriented System Engineering* Redwood City, CA, Mar. 2013, pp. 320–23.

[8] Y. Feng, B. Li, and B. Li, "Price Competition in an Oligopoly Market with Multiple IaaS Cloud Providers," *IEEE Trans. Computers*, vol. 63, no. 1, Jan. 2014, pp. 59–73.

[9] J. C. Harsanyi and R. Selten, *A General Theory of Equilibrium Selection in Games*, MIT Press, 1988.

[10] Z. Han *et al.*, *Game Theory in Wireless and Communication Networks: Theory, Models and Applications*, Cambridge Univ. Press, 2011.

[11] A. H. A. El-Atta, and M. I. Moussa, "Student Project Allocation with Preference Lists over (student, project) Pairs," *Proc. Second Int'l. Conf. Computer and Electrical Engineering*, Dubai, UAE, Dec. 2009.

[12] D. J. Abraham, R. W. Irving, and D. F. Manlove, "The Student-Project Allocation Problem," *Proc. 14th Int'l. Symp. ISAAC*, Kyoto, Japan, Dec. 2003, pp. 474–84.

[13] P. Bolton, and M. Dewatripont, *Contract Theory*, MIT Press, 2004.

[14] Y. Zhang *et al.*, "Multi-Dimensional Payment Plan in Fog Computing with Moral Hazard," arXiv preprint arXiv:1701.07877, 2017.

[15] H. Zhang *et al.*, "Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching," arXiv preprint arXiv:1701.03922, 2017.

Biographies

Huaqing Zhang [S'14] received his B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in June 2013. He is now pursuing a Ph.D. degree in the Department of Electronic and Computer Egineering at University of Houston, Texas. He has been a reviewer for TWC, TCCN, TBC, and JWCN. His research interests include wireless communications and networking, zero-determinant strategy, and hierarchical game.

Yanru Zhang [S'13, M'16] received her B.S. degree in electronic engineering from the University of Electronic Science and Technology of China in 2012, and her Ph.D. degree from the Department of Electrical and Computer Engineering, University of Houston in 2016. She is now working as a postdoctral fellow at the Network Communications and Economics Lab, Chinese University of Hong Kong. Her current research involves the contract theory and matching theory in network economics, Internet and applications, and wireless communications and networking. She received the best paper award at IEEE ICCS 2016.

Yunan Gu received her M.S. in computer science from Texas Southern University in 2013 and her Ph.D. degree in electrical and computer engineering from the University of Houston in 2016. Now, she is in an R&D engineer in the IP Technology Research Division of Huawei. Her research interests include matching theory, LTE-Unlicensed, D2D, V2X, and so on.

Dusit Niyato [M'09, SM'15, F'17] is currently an associate professor in the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He received his B.Eng. from King Mongkuts Institute of Technology Ladkrabang, Thailand, in 1999 and his Ph.D. in electrical and computer engineering from the University of Manitoba, Canada, in 2008. His research interests are in the area of energy harvesting for wireless communication, the Internet of Things (IoT), and sensor networks.

Zhu Han [S'01, M'04, SM'09,F'14] received his B.S. degree in electronic engineering from Tsinghua University in 1997, and his M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was an R&D engineer of JDSU, Germantown, Maryland. From 2003 to 2006, he was a research associate at the University of Maryland. From 2006 to 2008, he was an assistant professor at Boise State University, Idaho. Currently, he is a professor in the Electrical and Computer Engineering Department as well as in the Computer Science Department at the University of Houston His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing* in 2015, the IEEE Leonard G. Abraham Prize in the Field of Communications Systems (best paper award in *IEEE JSAC*) in 2016, and several best paper awards at IEEE conferences. Currently, he is an IEEE Communications Society Distinguished Lecturer.

With our proposed algorithm, we guarantee that the utility of the ADSS in fog computing is always higher than that in cloud computing, and the improvement gap of the ADSSs' utility from fog computing to the cloud computing is small.