Dynamic Computing Resource Adjustment in Edge Computing Satellite Networks

Feng Wang[1], Dingde Jiang[1](✉), Sheng Qi[1], Chen Qiao[1], and Jiping Xiong[2]

[1] School of Astronautics and Aeronautic, UESTC, Chengdu 611731, China
jiangdd@uestc.edu.cn
[2] College of Physics and Electronic Information Engineering, ZJNU, Jinhua, China

Abstract. The LEO constellation has been a valuable network framework due to its characteristics of wide coverage and low transmission delay. Utilizing LEO satellites as edge computing nodes to provide reliable computing services for accessing terminals will be the indispensable paradigm of integrated space-air-ground network. However, the design of resource division strategy in edge computing satellite (ECS) is not easy, considering different accessing planes and resource requirements of terminals. To address these problems, we establish the resource requirements model of various terminals. Meanwhile, the advanced K-means algorithm (AKG) is provided to realize ECS resource allocation. Then, a fleet-based adjustment (FBA) scheme is proposed to realize dynamic adjustment of resource for ECSs. Simulation results show that the proposed dynamic resource adjustment scheme is feasible and effective.

Keywords: Edge computing · LEO satellite network · Resource adjustment · Space-air-ground network

1 Introduction

The Low earth orbit (LEO) constellation network is making an important role in the space-air-ground integrated communication network. Compared with about 500 ms communication delay utilizing geosynchronous earth orbit (GEO) satellites, LEO constellation has the advantages of low delay, high capacity, full coverage and manageability [1, 2]. Moreover, it can guarantee the efficient communication in the areas such as polar region, desert, oceans and air that are difficult to reach by GEO satellite or terrestrial base stations, so as to achieve global network coverage [3]. With the development of intelligent terminals, the demand for real-time data processing is impending. LEO satellite has been difficult to meet the real time data computing for terminals [4]. Therefore, the fusion of LEO constellation and the edge computing paradigm to enhance the real-time management of intelligent terminals has gradually attracted attention. As Fig. 1 shows, the edge server can be deployed in the LEO satellite, making LEO satellites become edge computing satellites (ECSs), where the data processing module for terminals are installed. Computing resources of ECS can be split into virtual machines (VMs) with different specifications to implement data

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 135–145, 2019.
https://doi.org/10.1007/978-3-030-32216-8_13

processing for various terminals [5]. However, the design of resource division strategy in edge computing satellite (ECS) is not easy, considering different accessing planes and resource requirements of terminals. Since LEO satellites are moving at a relatively high speed, its topology and coverage area in the next time slot may change, leading to the reconfiguration of the resources in ECSs. The resources and time spent on reconfiguration may affect the ability of ECS for real-time data processing. Therefore, a dynamic resource adjustment scheme is needed to help the ECS realize real-time data processing for intelligent terminals.

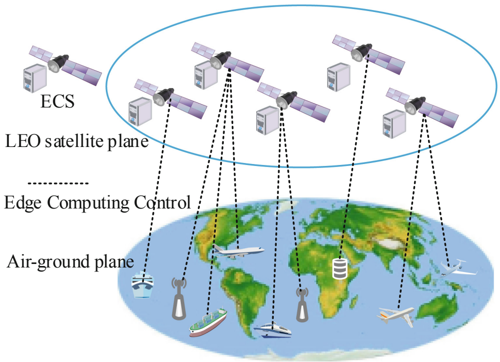

Fig. 1. The edge computing satellite network model.

The scheduling of resources in satellite networks is an ongoing research area and there have been some important studies. Authors in [6] considered the cooperative mechanism of relay satellites deployed in GEO and LEO and proposed a multiple access and resource allocation strategy for GEO relay in LEO satellite network. But they didn't consider the high data transmission latency of GEO satellites, about 500 ms round trip time. Sinha et al. [7] presented a multi-agent based modeling of LEO satellite network. Satellites were modeled as autonomous agents and could collaborate with other satellite agents. The allocation of tasks by the agents was modeled as a distributed constraint optimization problem. But small-scale self-organizing networks may affect other satellites, causing unnecessary link congestion. Sheng et al. [8] constructed a novel graph model to describe the evolution of multi-dimensional resources in satellite network. They proposed a resource allocation strategy to facilitate efficient cooperation among various resources. Although the performances of the LEO satellite network were improved, these studies could not provide dynamic management schemes for resources in each LEO satellite. In order to solve the above shortcomings, this paper considers the resources management in ECSs. In particular, the dynamic management strategies in ECS are provided. Traffic modeling [9], traffic estimation [10], network selection [11], energy efficiency [12] and network behaviors [13] are studied in previous work.

The rest is arranged as follows. Section 2 constructs the mathematical model of resource division in ECS and propose the dynamic resource adjustment scheme. The simulation results and analysis are shown in Sect. 3. Finally, we conclude our work in Sect. 4.

2 Mathematical Model

In edge computing satellite network, each ECS can connect multiple terminals by the user data links (UDLs). The ECS generates the corresponding VM to provide computing services for the terminal. Therefore, a problem need to be considered that how to allocate the computing resources of ECSs for the terminals, so as to not only meet the computing demand, but also realize reasonable resource configuration.

2.1 Resource Allocation in ECS

Different terminals have different data computing demands. For example, due to the relatively high-speed motion between UAVs and satellites, low delay data transmission is required to reduce signal distortion. Considering the data center has fixed location and need to receive various types of terminal data, there is a high bandwidth demand for it when accessing ECS. Frequent link switching may result in partial data loss, so for sensor nodes and ships, it is necessary to select ECS for data processing with long connection time. Therefore, intelligent terminals can be divided into three types: delay sensitivity, bandwidth sensitivity and connection time sensitivity.

Delay Sensitivity: Without considering the time taken by the signal transmission and reception, the time taken by transmitting the information from the terminal to the ECS is assumed as the transmission delay $d_{iE}(t)$:

$$d_{iE}(t) = \frac{\sqrt{(x_E(t) - x_i(t))^2 + (y_E(t) - y_i(t))^2 + (z_E(t) - z_i(t))^2}}{c} \tag{1}$$

where $(x_i(t), y_i(t), z_i(t))$ and $(x_E(t), y_E(t), z_E(t))$ respectively represent the three-dimensional coordinates of ECS E and terminal i, and they are all functions of time. c is the speed of light. If terminal i has a fast speed it will has a large value of $d'_{iE}(t)$

$$d'_{iE}(t) = \frac{\Delta d_{iE}(t)}{\Delta t} \tag{2}$$

The value of $d'_{iE}(t)$ is denoted by D_{iE} to express the delay sensitivity of terminal i.

Bandwidth Sensitivity: The data transmission rate always depends on the bandwidth provided by the ECSs' transceiver and the unit is Mbps. If the communication link is established between ECS E and terminal i, the maximum available bandwidth B_{Ei} is the smaller available bandwidth value B for both nodes:

$$B_{Ei} = \min(B_E, B_i) \tag{3}$$

The value of B_{Ei} can be used to describe the bandwidth sensitivity of terminal i.

Connection Time Sensitivity: Frequent link switching may result in partial data loss of terminals like sensors and ships, so these terminals have high connection time sensitivity. The link connection time starts with the establishment of transmission links and ends with the satellites lost their physical visibility. We define the critical time when link establishment as T_0 and when two nodes lose physical visibility as $T_{\max}$, the link connection time T_{Ei} can be expressed as follows:

$$T_{Ei} = T_{\max} - T_0 \tag{4}$$

The three types of sensitivities of n terminals can be expressed as $\{D_{1E}, D_{2E}, \ldots D_{nE}\}, \{B_{1E}, B_{2E}, \ldots B_{nE}\}$ and $\{T_{1E}, T_{2E}, \ldots T_{nE}\}$. For convenient comparison with the other indicators, we normalize the D_{iE} as D^*_{iE}, B_{iE} as B^*_{iE} and T_{iE} as T^*_{iE}:

$$D^*_{iE} = \frac{D_{iE} - D_{\min}}{D_{\max} - D_{\min}} \times 100 \tag{5}$$

$$B^*_{iE} = \frac{B_{iE} - B_{\min}}{B_{\max} - B_{\min}} \times 100 \tag{6}$$

$$T^*_{iE} = \frac{T_{iE} - T_{\min}}{T_{\max} - T_{\min}} \times 100 \tag{7}$$

where $D_{\max}$ $B_{\max}$ $T_{\max}$ and $D_{\min}$ $B_{\min}$ $T_{\min}$ are the maximum and minimum values in the dataset. Finally, the terminal indicator set can be expressed as X:

$$X = \{x_1, x_2, \ldots x_n\} \tag{8}$$

where $x_i = (D^*_{iE}, B^*_{iE}, T^*_{iE})$ is a three-dimension vector to represent the sensitivity characteristics of terminal i. In order to achieve the terminal clustering, Dis_{ij} is defined as the Euclid distance between terminals of the same cluster:

$$Dis_{ij} = \sqrt{(D^*_{iE} - D^*_{jE})^2 + (B^*_{iE} - B^*_{jE})^2 + (T^*_{iE} - T^*_{jE})^2} \tag{9}$$

Suppose the classified cluster is $\{C_1, C_2, \ldots C_n\}$, the average distance $\overline{Dis}$ in the same cluster can be expressed as

$$\overline{Dis_{ij}} = \frac{\sum_{i=1}^{n} \sum_{j=1, j \neq 1}^{n} Dis_{ij}}{C_n^i} \tag{10}$$

According to above equations, the standard deviation SD_i of the terminals in the same cluster is:

$$SD_i = \sqrt{\frac{\sum_{i=1}^{n} \sum_{j=1, j \neq 1}^{n} (Dis_{ij} - \overline{Dis_{ij}})^2}{C_n^i}} \tag{11}$$

In the cluster i, the smaller the SD_i value, the higher the similarity of the terminals. In the results obtained by the clustering algorithm, the minimum of the maximum standard deviation of all clusters is pursued:

$$\min\{\max SD_i\},\ i = 1, 2, 3, \ldots \tag{12}$$

The clustering algorithm will choose k-means algorithm, which has simple and fast clustering ability. At the same time, the number of initial clusters and clustering centers can be preset to reduce the computation complexity and clustering error. The quadratic mean of each dimensional data is used to generate the initial cluster center:

$$D_{sq} = \sqrt{\frac{D_{1E}^{*2} + D_{2E}^{*2} + \cdots + D_{nE}^{*2}}{n}} \tag{13}$$

$$B_{sq} = \sqrt{\frac{B_{1E}^{*2} + B_{2E}^{*2} + \cdots + B_{nE}^{*2}}{n}} \tag{14}$$

$$T_{sq} = \sqrt{\frac{T_{1E}^{*2} + T_{2E}^{*2} + \cdots + T_{nE}^{*2}}{n}} \tag{15}$$

The initial coordinates are $(D_{sq}, 0, 0)$, $(0, B_{sq}, 0)$ and $(0, 0, T_{sq})$. In this paper, the advanced K-means algorithm (AKG) is used to cluster the terminals. The detailed steps are as follows:

Step 1: Select $(D_{sq}, 0, 0), (0, B_{sq}, 0)$ and $(0, 0, T_{sq})$ as the centers of three clusters;
Step 2: Calculate the distance from the remaining terminals $\{x_1, x_2, \ldots x_n\}$ to each cluster center and classify these terminals into the nearest cluster center;
Step 3: Calculate the average values $\overline{Dis_{ij}}$ of terminals in each cluster respectively and designate them as the new cluster centers.
Step 4: Iterate Step 2 and Step 3 until reaching the threshold or cluster centers are not changed.

Through the AKG algorithm, terminals are divided into three categories. The ECS calculates resource requirements of each type of terminals for VM resource allocation, so that each terminal can obtain appropriate computing resources.

2.2 Dynamic Adjustment

The above scheme help the ECS to realize the reasonable division of resources. However, the resource configuration of one ECS is different when flying over different regions. Therefore, a fleet-based adjustment (FBA) scheme is proposed to reduce the unnecessary consumption of ECSs. Actually, the ECS network can be modeled by the IM matrix like:

$$IM = (\omega_{ij})_{m\times n} = \begin{bmatrix} \omega_{11} & \omega_{12} & \cdots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ \omega_{m1} & \omega_{m2} & \cdots & \omega_{mn} \end{bmatrix} \tag{16}$$

where ω_{ij} is denoted as the weight of satellite j in orbit i. Meanwhile, ω_{ij} is the function of parameter $D_{ij}, B_{ij}, T_{ij}, \deg_{ij}$:

$$\omega_{ij} = f(D_{ij}, B_{ij}, T_{ij}, \deg_{ij}) \tag{17}$$

where $\deg_{ij}$ is the accessing number of terminals in each ECS. The ω_{ij} can be regarded as a multi-dimensional vector. Among the parameters, D_{ij} and T_{ij} may not be independent for each other. So the principal component analysis (PCA) method is utilized to reduce the data dimension and improve the accurate of evaluation for each ECS. The covariance matrix of the two variables is calculated as:

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} \tag{18}$$

where

$$c_{11} = \mathrm{cov}(D_{ij}, D_{ij}) = E\left[D_{ij} - E\left(D_{ij}\right)\right]^2 \tag{19}$$

$$c_{12} = \mathrm{cov}(D_{ij}, T_{ij}) = E\left[(D_{ij} - E[D_{ij}])(T_{ij} - E[T_{ij}])\right] \tag{20}$$

$$c_{21} = \mathrm{cov}(T_{ij}, D_{ij}) = E\left[(T_{ij} - E[T_{ij}])(D_{ij} - E[D_{ij}])\right] \tag{21}$$

$$c_{22} = \mathrm{cov}(T_{ij}, T_{ij}) = E\left[T_{ij} - E\left(T_{ij}\right)\right]^2 \tag{22}$$

After that, the eigenvalues λ_1 and λ_2.of the matrix C is calculated. The corresponding eigenvectors are α_1 and α_2. Then the eigenvector corresponding to the largest eigenvalue is selected, such as α_2 to λ_2. Finally, the random variable Y is obtained as:

$$Y = \alpha_2^T * (D_{ij}, T_{ij}) \tag{23}$$

where Y is the final random variable with only one dimension:

$$Y = (y_1, y_2, \ldots, y_n) \tag{24}$$

So the function of ω_{ij} is further transferred as:

$$\omega_{ij} = f(Y_{ij}, B_{ij}, \deg_{ij}) \tag{25}$$

To compare the similarity of each ECS, firstly the values of three parameters should be converted from different dimensions into one-dimensional weight value. The weighted average method is used to calculate the final weight value of each ECS. The calculation process is as follows:

$$P_{ij} = \left(\sum_{i=1}^{l} \omega_i d_{ij} + \sum_{i=1}^{k-1} \omega_i d'_{ij} \right) / \sum_{i=1}^{k} \omega_i \tag{26}$$

where

$$\begin{aligned} d_{ij} &= \frac{x_{ij} - x_i^{(s)}}{x_i^{(h)} - x_i^{(s)}}, \\ i &= 1, 2, \cdots, k, \ \ j = 1, 2, \cdots, n \end{aligned} \tag{27}$$

$$\begin{aligned} d'_{ij} &= \frac{x_i^{(h)} - x_{ij}}{x_i^{(h)} - x_i^{(s)}}, \\ i &= 1, 2, \cdots, k, \ \ j = 1, 2, \cdots, n \end{aligned} \tag{28}$$

P_{ij} is the performance evaluation of ECS E_{ij}. k is the total number of the parameters selected. l is the number of positive parameters selected. n is the number of ECSs in the evaluation. w_i is the weight of the parameters i. x_{ij} is the value of the parameters i in E_{ij}. $x_i^{(h)}$ is the optimal value of the parameters i. $x_i^{(s)}$ is the average value of the parameters i. d_{ij} is the evaluation score of the parameters i in E_{ij}. d'_{ij} is the evaluation score of the inverse parameters i in E_{ij}. The inadmissible value of a parameter is the worst value that should not appear in the evaluation. The optimal value is the best value that the parameter can achieve. The ECS network evaluation can be calculated as:

$$P_N = \left(\sum_{i=1}^{n} P_{L_i} \omega_i \right) / \sum_{i=1}^{k} \omega_i \tag{29}$$

where P_N is the performance evaluation of the network. n is denoted as the number of ECSs. ω_i is the performance weight of E_{ij}. Meanwhile, supposing that the ECS fleet is $X = (x_1, x_2, \ldots x_m)$ and the control time of x_i is t_i:

$$x_1 = t_1, x_2 = t_2, x_3 = t_3, \ldots, x_m = t_m \tag{30}$$

Then the whole control period Pe is

$$Pe = \sum\nolimits_{i=1}^{m} t_i \tag{31}$$

It means satellite x_i will be the ECS of the region once again after Pe time. The evolution of the configuration for one certain region will be carried out after Pe time. During Pe, ECSs with similar ω_{ij} have similar computing requirements. Configurations of ECSs can be inherited from the above ECSs in the fleet, as shown in Fig. 2. The resource for ECSs achieve dynamic management in spatio-temporal dimension.

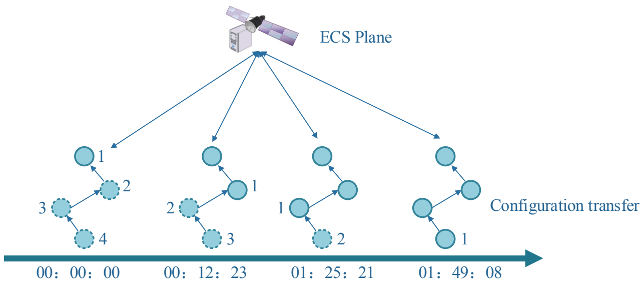

Fig. 2. The dynamic configuration transfer model.

3 Simulation Results and Analysis

The simulation is carried out in the integrated simulators where the space-air-ground network has established. A predictable LEO satellite network is constructed with the use of the Satellite Tool Kit (STK) simulator. The satellite model is established with reference to iridium constellation to guarantee the practical application. Meanwhile, we randomly created 100 UAV terminal nodes, 200 ship nodes and 200 sensor nodes in the Pacific Ocean initially using STK. Then we connect STK with MATLAB to obtain the satellites and terminal data, where the calculation and comparison of the proposed scheme and other methods are also carried out. Meanwhile, the node data is imported to the Qualnet simulator to establish the heterogeneous network for network performance evaluation. Each terminal node is assigned different values of three sensitivities according to their characteristics of resource demand.

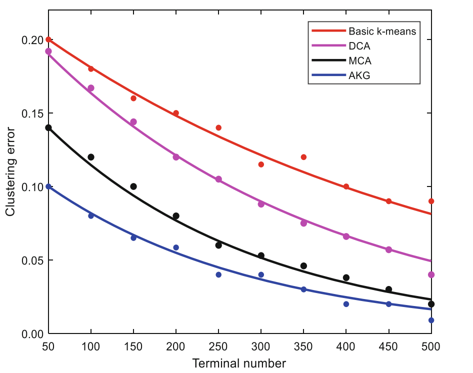

Fig. 3. Error comparison of two clustering algorithm.

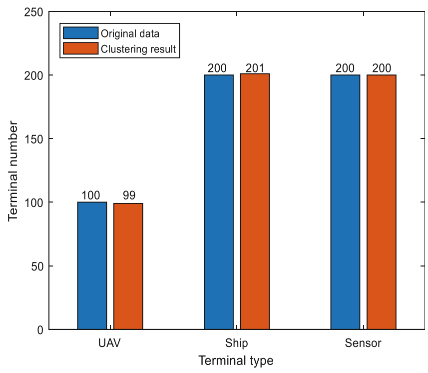

Fig. 4. The clustering result of AKG.

In the simulation, we first verified the classification precision of the proposed advanced k-means algorithm (AKG). As shown in Fig. 3, we compare the clustering errors of the basic k-means algorithm, meanshift clustering algorithm (MCA), density-based clustering algorithm (DCA) and the proposed AKG algorithm with the increase of terminal numbers. The trends of error change is plotted by curves. It can be seen that the clustering error of AKG is significantly lower than the other three algorithms. This is because the clustering center at the initialization of the AKG has been calculated regularly, while the basic k-means algorithm and the other two methods just randomly select the clustering center. The selection of clustering center will greatly affect the accuracy and computational complexity of the algorithm. The clustering result is expressed as Fig. 4, which shows a good terminal classification result.

The additional calculation of the ECS network for dynamic resource adjustment will be analyzed. As Fig. 5 shows, the methods of basic K-means, AKG and fleet-based adjustment (FBA) are compared. The additional calculation of method basic K-means rises rapidly with the increase of the number of involved ECSs, followed by AKG. The FBA method occupies the least amount of additional calculation. This is because the configuration of FBA is obtained from the previous ECS in the fleet, so there is no need to reuse the clustering algorithm for terminal classification. It only requires the adjustments based on the obtained configuration. Therefore, the FBA method can save more computing resources for ECSs.

The resource utilization of ECSs has a great influence on the performance of the satellite network. Next we will analyze the resource usage of ECSs under different methods to reflect the change of load balancing, total resource usage and robustness of the ECS network. In addition to the methods of AKG, DCA, MCA, Basic k-means, the TEG method in [14] is added to the comparison, which separated the computing resources and cache resources to realize efficient resource usage. Actually, the variance of resource usage of ECS nodes can reflect the load balancing of the ECS network to some extent. Figure 6 compare the variance change of different methods. It can be seen that the variance values of all five methods is increasing with ECS number increase. This is because different ECS has undertaken different amounts computing services of terminals. But the variance of the AKG method grows the slowest. Referring back to the above, the AKG method can help ECS to partition computing resources accurately, so that the uneven utilization of resources caused by allocation errors is reduced. Similarly, Fig. 7 shows the difference of total resource usage of the five methods. It can be seen that the total resource usage of AKG method is always the lowest. Therefore, accurate resource allocation can reduce resource waste and improve load balancing performance of the whole network. Meanwhile, we use the number of high capacity nodes to reflect the robustness of the ECS network in Fig. 8. The threshold of utilization of each ECS node is set to 0.7. We can see that the AKG method always has the most available ECSs with terminal number increase, which maintains the robustness of the ECS network. All the above show that the proposed scheme provides higher clustering precision and resource utilization, compared with other methods.

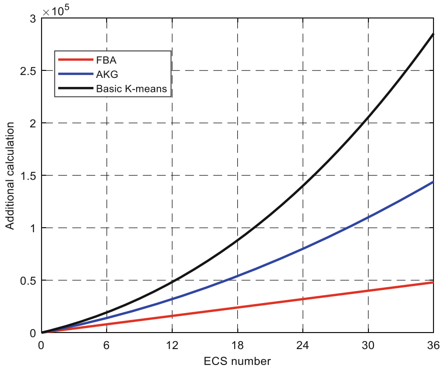

Fig. 5. The comparison of additional calculation.

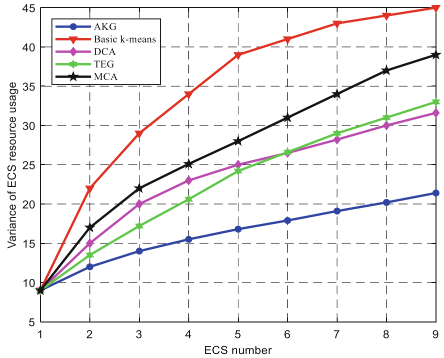

Fig. 6. The comparison of load balancing.

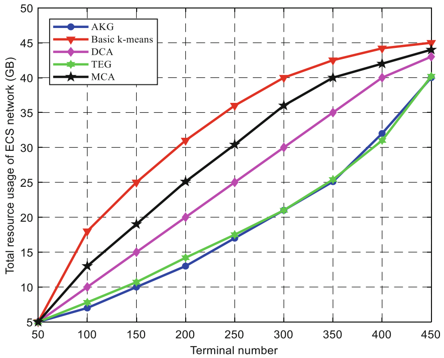

Fig. 7. The comparison of total resource usage.

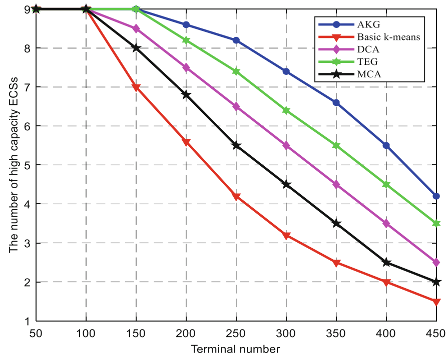

Fig. 8. The comparison of robustness.

4 Conclusions

This paper studies the resource management in the edge computing satellite network. We jointly consider the dynamic resource division and adjustment strategies in edge computing satellite (ECS). An advanced K-means algorithm (AKG) is proposed to guide the resource division in ECS. Meanwhile, a fleet-based adjustment (FBA) scheme is designed to realize dynamic adjustment of resource for ECS. The simulation results show that the proposed dynamic resource adjustment scheme is promising.

Acknowledgment. This work was supported by National Natural Science Foundation of China (No. 61571104), Sichuan Science and Technology Program (No. 2018JY0539), Key projects of the Sichuan Provincial Education Department (No. 18ZA0219), Fundamental Research Funds for the Central Universities (No. ZYGX2017KYQD170), and Innovation Funding (No. 2018510007 000134). The authors wish to thank the reviewers for their helpful comments.

References

1. Qu, Z., Zhang, G., Cao, H., et al.: LEO satellite constellation for Internet of Things. IEEE Access **5**, 18391–18401 (2017)
2. Liu, J., Shi, Y., Zhao, L., et al.: Joint placement of controllers and gateways in SDN-enabled 5G-satellite integrated network. IEEE J. Sel. Areas Commun. **36**(2), 221–232 (2018)
3. Zhang, Z., Jiang, C., et al.: Temporal centrality-balanced traffic management for space satellite networks. IEEE Trans. Veh. Technol. **67**(5), 4427–4439 (2018)
4. Tang, F., Zhang, H., Fu, L., et al.: Multipath cooperative routing with efficient acknowledgement for LEO satellite networks. IEEE Trans. Mob. Comput. **8**(1), 179–192 (2019)
5. Li, H., Ota, K., Dong, M.: Learning IoT in edge: deep learning for the internet of things with edge computing. IEEE Netw. **32**(1), 96–101 (2018)
6. Du, J., Jiang, C., Wang, J., et al.: Resource allocation in space multiaccess systems. IEEE Trans. Aerosp. Electron. Syst. **53**(2), 598–618 (2017)
7. Sinha, P.K., Dutta, A.: Multi-satellite task allocation algorithm for earth observation. In: Proceedings of TENCON 2016, pp. 403–408 (2016)
8. Sheng, M., Wang, Y., Li, J., et al.: Toward a flexible and reconfigurable broadband satellite network: resource management architecture and strategies. IEEE Wirel. Commun. **24**(4), 127–133 (2017)
9. Jiang, D., Wang, W., Shi, L., et al.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Netw. Sci. Eng. **5**(3), 1–12 (2018)
10. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)
11. Jiang, D., Huo, L., Lv, Z., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **PP**(99), 1–15 (2018)
12. Jiang, D., Zhang, Y., Song, H., et al.: Intelligent optimization-based energy-efficient networking in cloud services for multimedia big data. In: Proceedings of IPCCC 2018, pp. 1–6 (2018)
13. Jiang, D., Huo, L., Song, H.: Understanding base stations' behaviors and activities with big data analysis. In: Proceedings of Globecom 2018, pp. 1–7 (2018)
14. Wang, P., Zhang, X., Zhang, S., et al.: Time-expanded graph based resource allocation over the satellite networks. IEEE Wirel. Commun. Lett. **8**(2), 360–363 (2019)