AI-SPACE: A Cloud-Edge Aggregated Artificial Intelligent Architecture for Tiansuan Constellation-Assisted Space-Terrestrial Integrated Networks

Xiaobin Xu, Qi Wang, Yanzhao Hou, and Shangguang Wang

Abstract

Space-Terrestrial Integrated Networks (STINs) and ubiquitous artificial intelligence are considered as important infrastructures of 6G networks. Due to the long transmission distance and high moving speed of on-orbit satellites, STINs face the challenges of unreliable data transmission, untimely data processing, and low utilization of satellite resources. Tiansuan constellation implements the edge computing platform, which builds the foundation of edge intelligence for STINs. This article proposes a cloud-edge aggregated artificial intelligent architecture for STINs. With the assistance of Tiansuan constellation, an implementation strategy of the intelligent network architecture is proposed. Furthermore, three use cases are performed and discussed to achieve intelligent data transmission, intelligent remote sensing, and intelligent network slicing. Simulation results demonstrate that the proposed architecture can fully utilize the resource of STINs and improve the overall network throughput effectively.

Introduction

In the recent years, Space-Terrestrial Integrated Networks (STINs) play an important role in terms of military reconnaissance and communication support. Countries around the world pay attention to the development of STINs. The International Telecommunication Union (ITU) stipulates that the right to use satellite frequencies and orbital resources is obtained on a first come first served basis [1]. Therefore, to preempt the satellite frequency and orbit resources, more and more satellites are launched into the space in a short time, which incurs an explosive growth of on-orbit data. Due to the lack of future application scenarios of STINs, the on-orbit satellites and under-construction constellations are usually with poor hardware capacity and insufficient computing power [2]. Both the huge amount of on-orbit data and the unstable transmission link require on-orbit intelligence in STINs. In this case, it is necessary to empower STINs with computing capabilities and intelligent services.

As the first computation constellation, Tiansuan constellation provides STINs with an on-orbit lightweight 5G core and edge computing platform [3]. Satellite edge computing supported by Tiansuan constellation makes STINs possible to perform intelligent data processing and forwarding. The simplified architecture of Tiansuan edge computing platform is shown in Fig. 1.

The future networks are mostly cloud-edge fusion architectures [4, 5], and there have been many kinds of research on the combination of cloud computing and the Internet of Things [6, 7]. To make STINs more intelligent, more and more solutions are proposed. In 2018, Stergiou et al. proposed a cache decision system that combines the functions of the IoT with cloud computing, edge computing, and big data to achieve a Smart and Secure environment [8]. In 2020, Stergiou et al. proposed a big data management system framework based on a cloud federated network to significantly improve performance under large workload scenarios, thereby saving costs and better managing data [9]. In 2021, Plageras et al. studied the system of collecting and managing sensor data in smart buildings operating in the Internet of Things (IoT) environment to build a Green Smart Building [10].

Inspired by the current Service-based architecture (SBA) of 5G networks and the Software Defined Network (SDN) technology, this article proposes an intelligent networking architecture of STINs combining distributed preprocessing and centralized management. The architecture is based on Tiansuan constellation and provides several types of intelligent network services. The contributions are summarized as follows:

- We propose a cloud-edge aggregated architecture, which aggregates centralized intelligent controllers on cloud servers and distributed intelligent agents on satellite edge servers. The architecture adopts a space-terrestrial collaboration strategy, in which the ground station can assist the controller in model training.
- We propose an implementation strategy of intelligent network architecture and a resource management scheme based on the capabilities of Tiansuan constellation. The controller is placed on the main satellite of Tiansuan constellation, and the edge server is placed on the edge satellite of Tiansuan constellation. The controller and edge server are combined to implement tasks intelligently.

Digital Object Identifier: 10.1109/MNET.003.2200401

Xiaobin Xu and Qi Wang are with Beijing University of Technology, China; Yanzhao Hou (corresponding author) and Shangguang Wang are with Beijing University of Posts and Telecommunications, China; Yanzhao Hou is also with Peng Cheng Laboratory, China.

- We performed three use cases to achieve intelligent data transmission, intelligent remote sensing, and intelligent network slicing. We use data coding to improve the reliability of data transmission, we also propose to integrate image recognition algorithms into the satellite to ensure real-time processing, and use network slicing to flexibly allocate network resources.

The remainder of this article is organized as follows: First we introduce the cloud-edge aggregated architecture and its implementation. Following that we propose three types of use cases. We then evaluate the performance of the proposed architecture. The final section summarizes this article.

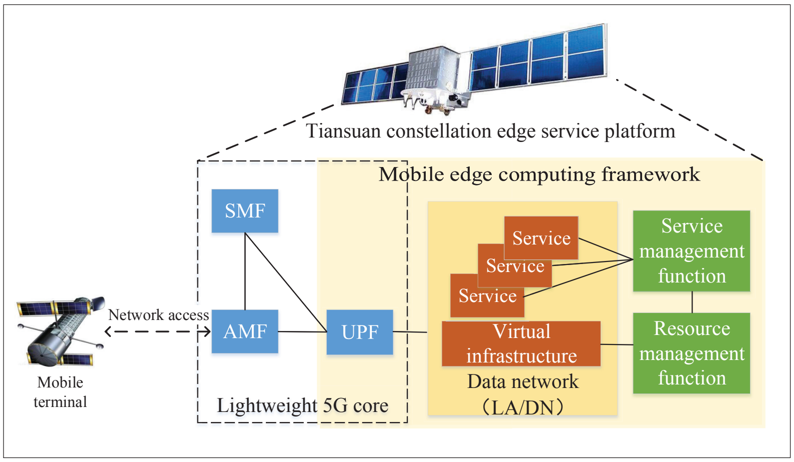

FIGURE 1. The simplified architecture of Tiansuan edge computing platform.

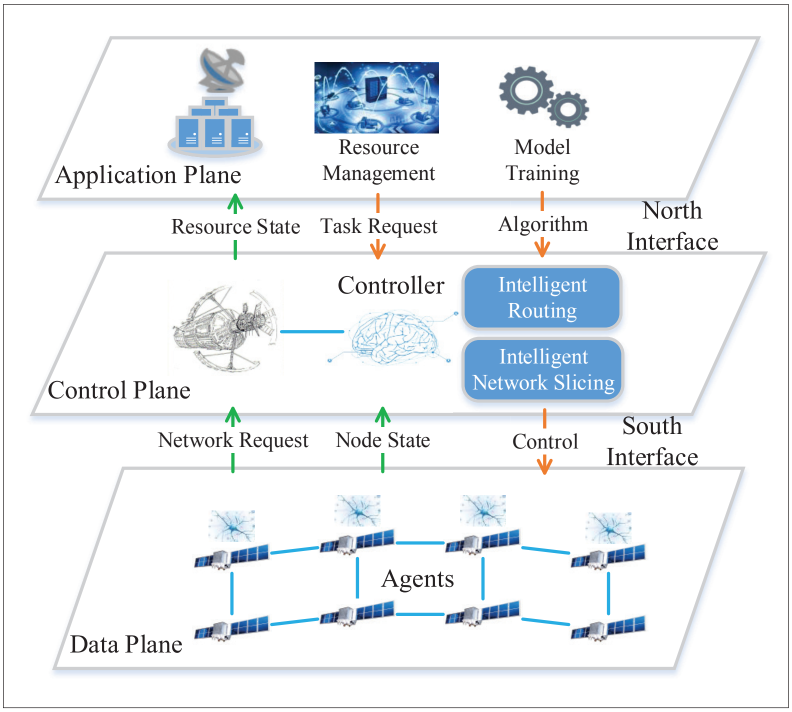

FIGURE 2. Cloud-edge aggregated architecture for intelligent STIN.

Cloud-Edge Aggregated Architecture

SDN-Based Networking Architecture

Due to the high cost, typically each owner only has a small number of space-based infrastructure nodes. Therefore, it is difficult to unify the operation standards of satellite networks as globally as terrestrial Internet. The globalization of satellite networks will not be achieved at least soon. For this reason, this article envisages that future STINs will be application-driven and form multiple independent constellations. Each constellation is established based on SDN, and its network protocol can be defined on its own. STIN applications with the same network protocol can communicate with each other. The typical network architecture is shown in Fig. 2.

As shown in Fig. 2, the overall structure of the network is divided into two parts: the space end and the ground end. The space end consists of one controller and multiple edge satellites in several orbital planes. The controller is deployed in the main satellite of Tiansuan constellation. The controller can remotely deploy services into edge satellites to allocate bandwidth and computing resources. For intelligent transmission tasks, the main task of edge satellites is to transmit data, and the transmitted data also requires encoding and processing in edge satellites. Therefore, for the intelligent transmission scenario, the resources allocated are mainly bandwidth resources, as well as computing resources. For intelligent remote sensing tasks, edge satellites mainly complete image recognition tasks, but also need to forward data. Therefore, for the intelligent remote sensing scenario, the resources allocated are mainly computing resources, as well as bandwidth resources. The edge server is deployed in the edge satellite. The edge server places the services deployed remotely by the controller, such as data transmission services, data encoding, and decoding services, intelligent processing services, and so on. The ground end consists of a ground control center and many ground receiving stations. The ground control center mainly interacts directly with high-orbit satellites for the management of the whole network.

The software implementation strategies of each part of the architecture are described as follows: edge satellites belong to the data plane of SDN architecture. They carry the hardware equipment needed for computing and are installed with the operating system. Each satellite adopts an integrated network architecture based on SDN and Network Function Virtualization (NFV), and the computing, storage, and transmission resources are modeled and managed in a unified manner. When performing specific tasks, resources are pre-allocated in the container, and the specific tasks are carried out in the container. As the control plane of SDN, the controller is responsible for the control of the network and the distribution of tasks. The controller stores information on all the ongoing missions in the constellation and the real-time resource utilization of all edge satellites. The controller receives new task requests in a real-time manner. When a request arrives, the task is assigned according to the intelligent algorithm. As the application plane of SDN, the ground control center is responsible for the development of application functions, the training of artificial intelligence (AI) algorithms [11], and the operation and maintenance of the network.

Space-Terrestrial Collaboration Strategy

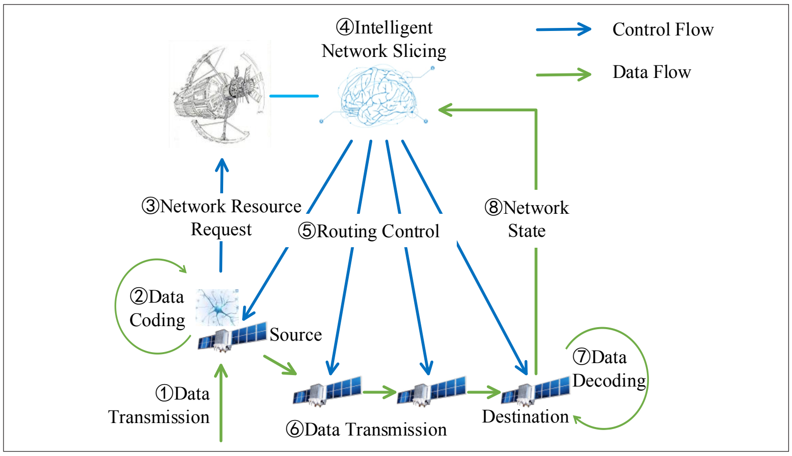

FIGURE 3. Scenario of intelligent data transmission.

Machine learning (ML), as a representative AI algorithm, can be divided into two steps: training and execution. The training of the model usually requires big computational capability and occupies a significant amount of computing resources. Therefore, most existing ML algorithms are not feasible to be implemented in satellites. For this reason, we propose a space-terrestrial collaboration strategy for the following scenarios.

Training of Massive Static Historical Data: For many applications, although real-time data may change, the overall characteristics of the data remain the same. In this scenario, the ground control center performs model training based on the massive static historical data and sends the trained model to the controller.

Training of a Small Amount of Real-Time Data: For real-time applications, the controller adopts a lightweight algorithm and carries out simple model training based on a small amount of real-time data. The training process does not pursue obtaining the optimal solution, but can quickly obtain an acceptable and optimal strategy and adjust it in real-time. Short response time and fast convergence are the core requirements of this type of algorithm.

Static and Dynamic Hybrid Training: For most real-time applications, a combination of static and dynamic training methods can be adopted. First of all, a base model is trained by the ground control center and sent to the controller. The controller updates the base model according to the real-time data.

Implementation Scheme of Cloud-Edge Aggregated Architecture

With the existing capabilities assistance of Tiansuan constellation, the cloud-edge aggregated architecture proposed in this article can place the controller on the main satellite of Tiansuan constellation and the edge server on the edge satellite of Tiansuan constellation. Each intelligent processing function can be deployed to the satellite in the form of a container and published as a service in the satellite, which will be published as "services" in the edge platform, as shown in Fig. 1. In a typical task, the controller and edge server are combined to implement tasks intelligently. The process is presented as follows.

Stage 1 (Edge End) — The Initiation of Network Requests and the Intelligent Pre-Allocation of Network Resources: When an edge satellite of Tiansuan constellation receives transmission requests, it analyzes recent requests and predicts the possible data volume in the tasks in the future. An agent estimates the requirements of resources and duration of the set of nodes required to complete the entire task and makes a resource request to the controller.

Stage 2 (Cloud End) — Intelligent Planning and Task Distribution of Network Resources: After receiving a network request, the controller intelligently decomposes the task according to the current occupation of network resources and divides a complete task into task chains of several agent nodes. Each node in the linked list records the tasks required for each agent. After the task is assigned, the controller broadcasts the task to each agent and updates the remaining resources.

Stage 3 (Edge End) — Task Execution and Status Collection: When an edge satellite of Tiansuan constellation receives a forwarding task, it establishes a container and allocates computing and transmission resources for this task. The first edge satellite of Tiansuan constellation encodes the data and sends the data to the next hop. For each forwarding, an agent records the current resource consumption of this node and stores it in the header of the packet. After receiving the data, the last satellite takes out the packet header information and sends it to the controller to realize a real-time collection of network status. This can be implemented based on in-band network telemetry (INT) and programming protocol-independent packet processors (P4).

Stage 4 (Cloud end): real-time recording of the network status. After receiving network status information, the controller updates the remaining resource information according to that provided by each agent node.

Intelligent Application of Architecture

Based on the cloud-edge aggregated architecture proposed in this article, we perform and discuss three use cases to achieve intelligent data transmission, intelligent remote sensing, and intelligent network slicing.

Intelligent Data Transmission

To improve the reliability, stability, and throughput of STINs, we integrate data coding technology into the proposed architecture. The intelligent data transmission scenario in STINs is illustrated in Fig. 3. The edge satellite encodes data and sends network resource requests to the controller. The controller performs routing control for multiple edge satellites. The edge satellite sends the data to the next hop, and the edge satellite in the last hop decodes the data and sends the network status information to the controller.

The implementation of intelligent data transmission mainly includes three parts: data coding, resource requirement mapping, and resource allocation optimization.

Data Coding: In STINs, each data transmission request includes three main parts: source node, destination node, and data transmission volume. To achieve intelligent data transmission, the source node should first adopt a specific coding algorithm to encode the data and send the coding packet.

After receiving enough coding packets, the destination node decodes them to obtain the original data. After coding, each data block is no longer unique, and the data transmission becomes a simple superposition of the number of packets.

Resource Requirement Mapping: In our considered scenario, the source node and destination node not only need data transmission but also need data coding/decoding operations. The tasks of the source node and destination node need to utilize both transmission resources and computing resources. The intermediate nodes of the transmission only need to forward data without processing, which only utilizes its transmission resources. Therefore, data transmission can be considered as the computing and transmission resource overhead of the source node and the destination node, as well as the transmission resource overhead of the intermediate node. The costs on the source node and the destination node are fixed expenses while those on the intermediate nodes are variable expenses since there can be various ways of routing. In Fig. 3, the intermediate nodes can be adjusted, as long as there is a sufficient amount of coding packets transmitted to the destination.

Resource Allocation: Resource allocation is one of the most important techniques of intelligent data transmission. The optimal allocation of resources mainly includes two aspects:

- Determination of the data transmission rate of the source node. For a data transmission request, the agent at the source node determines the data processing and transmission rate based on the remaining amount of its resources. In other words, the processing speed and transmission rate are generally faster with more remaining resources.
- Selection and resource allocation of intermediate nodes. As shown in Fig. 3, the structure of STINs is usually a Manhattan network. When the source node and destination node are determined, after data coding, the data packet can be transmitted through multiple paths at the same time, and the sum of the data transmission rate in each path is equal to the data transmission rate of the source node. The data transmission rate can be considered as the utilization of transmission resources. The transmission resource utilization of each node cannot exceed its remaining resources. Therefore, the problem of optimal allocation of resources can be converted to an optimization problem of average link utilization with constraints. The more balanced the link utilization is, the less likely it is to cause network congestion and the stronger its adaptability is to new tasks. This problem can be solved by a variety of intelligent algorithms such as reinforcement learning.

The coding complexity depends on the specific coding algorithm. To ensure the real-time forwarding of continuous tasks when they arrive, if the amount of data to be transmitted for each task is n, the complexity of the coding algorithm should not exceed O(n). The network addressing strategy is usually simple, so it mainly depends on the task priority scheduling strategy when multiple tasks arrive. The satellite network link is unstable, with long end-to-end delays and low reliability of data transmission. Data coding can effectively improve the reliability of transmission and the success rate of transmission. In the intelligent network architecture proposed in this article, multiple types of data encoding algorithms can be selected and flexibly adjusted for data transmission to further improve the reliability of data transmission and the controllability of the network.

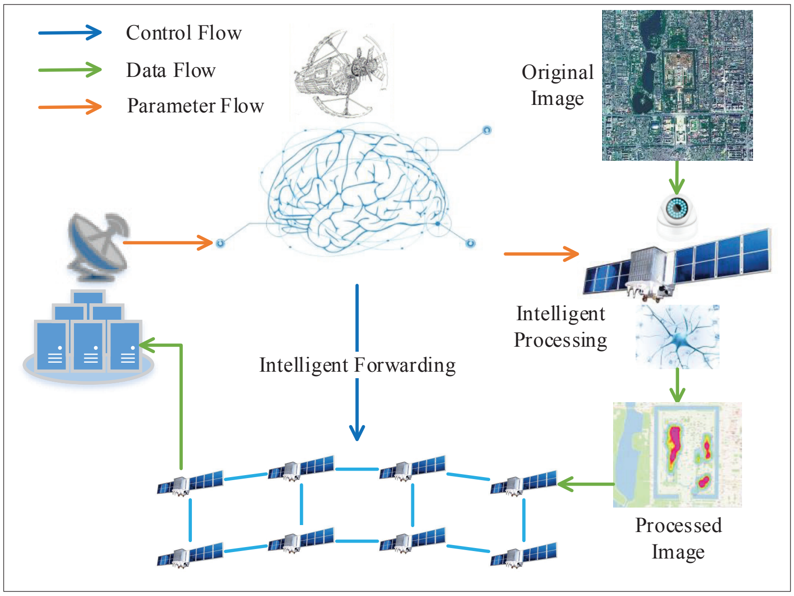

FIGURE 4. Scenario of intelligent remote sensing.

Intelligent Remote Sensing

In remote sensing applications, we propose the use case of intelligent remote sensing. This use case can effectively reduce the amount of data transmission and improve the real-time performance of remote sensing applications. As shown in Fig. 4, the implementation of intelligent remote sensing includes intelligent processing and intelligent forwarding. The controller sends the model to the edge satellite, which processes the data. Each hop processes and forwards the data, and sends the results to the next hop.

Intelligent Processing: The amount of remote sensing image data connected is typically very large and therefore needs to be processed before being used. After processing, the sizes of the processed image files are usually much reduced and can be directly used for display and analysis [12]. Therefore, we propose to integrate the image recognition algorithm into the operation of satellites. The ground stations perform model training based on the specific needs of remote sensing images and send the trained models to the controller. The controller then sends these models to the corresponding edge satellites. The edge satellites collect data and process it based on the received models, and then encode the processed data and send them to the ground.

Intelligent Forwarding: In the transmission process, each hop can be transmitted by processing and forwarding to enhance reliability. The intelligent forwarding process of remote sensing images is essentially Intelligent data transmission with preprocessing of the data. Due to the need to image capturing and processing, the demand for computing resources is higher. On the other

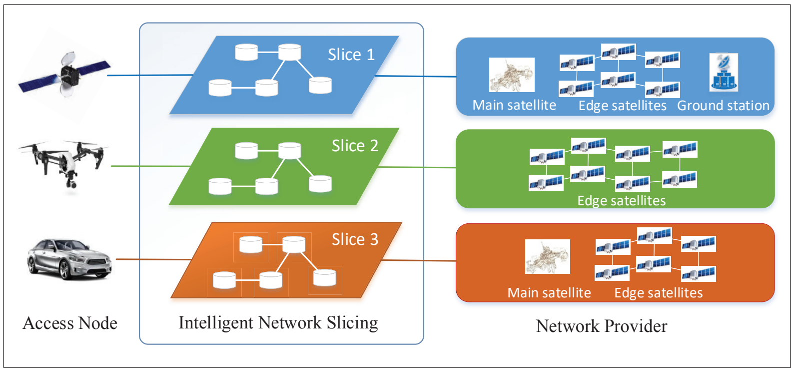

FIGURE 5. Scenario of intelligent network slicing.

hand, the data processing and transmission rate of the source node not only needs to be optimized according to its own needs but also needs to meet the user requirements, which can be implemented as constraints for designed algorithms.

For the controller, the additional cost mainly depends on the storage and distribution of intelligent algorithms. Assuming there are m remote sensing image processing models, the controller needs to maintain additional storage space for m models. Since the controller needs to distribute the model before the arrival of the intelligent remote sensing task, the controller is not required to participate in the real-time operation process, so the cost of the controller can be ignored. The integration of an image recognition algorithm into satellites can effectively improve the efficiency and quality of returned data. The proposed intelligent architecture makes on-orbit data processing algorithms adaptive and controllable, which can further improve data processing efficiency.

Intelligent Network Slicing

The network slicing method can adjust the network resource allocation scheme according to the user's requirements for the network, which is an important way to improve the efficiency of network resource utilization. In the architecture proposed in this article, we use the application-oriented resource management method to realize the intelligent slicing of the network. Each type of network service needs the support of different types of network resources. Operators can provide users with different communication and computing resources and charge fees according to their requirements. In this process, STINs operators aim to maximize profits while STINs users aim to get the highest degree of satisfaction through leasing resources. Resource allocation is a classic problem in current network applications, which is often modeled as a resource management [13]. The goal of intelligent network slicing is to optimize resource allocation and make effective use of network resources to maximize the benefits for both STIN operators and STIN users. The scenario of intelligent network slicing is illustrated in Fig. 5.

As shown in Fig. 5, the Internet of Vehicles applications requires massive access points and high bandwidth. Therefore, the network element should be deployed on the main satellite and the edge satellites; For Unmanned Aerial Vehicle applications, it has high requirements for response speed. Network elements should be deployed on the edge satellites; For satellite applications, huge remote sensing data requires ultra-wide bandwidth. Network elements should be deployed on the main satellite, edge satellites, and ground stations. Intelligent network slicing implementation consists of two steps as presented below.

Problem Modeling: For different types of services, we first need to model the utilization of computing and communication resources used according to specific application scenarios (i.e., intelligent data transmission and intelligent remote sensing, as previously mentioned). According to the resource consumption, the application charge needs to be modeled. The price is usually set according to the two dimensions of computing resources and communication resources, and the charge increases with the increase in resource consumption. Based on resource cost and resource charge, the satisfaction of users and the profit of operators are modeled.

Optimal Strategy Analysis: After the establishment of the model, it is necessary to analyze the user satisfaction function and operator profit function. Under various constraints, whether the maximum values of the user satisfaction function and the operator profit function exist need to be determined, respectively. If maximum values exist, the optimal resource allocation policy exists.

Currently, an algorithm based on Q-learning has been proposed for the intelligent allocation of computing resources [14]. However, this method is designed for scenarios when the user resource demand is static. When the user resource demand is generated dynamically, we need to adapt the algorithm to accelerate its convergence speed to meet the needs of dynamic scenarios.

Both main satellites and edge satellites can quickly calculate the optimal prices and resource requirements. In the worst case, the amount of computation on a main satellite is the same as the number of messages from the edge satellite. The computational complexity of each round is O(m), where m is the number of edge satellites. Similarly, in the worst case, the computational complexity of each round of each edge satellite is O(n), where n is the number of main satellites. The proposed intelligent architecture can allocate network resources more flexibly and utilize network resources more effectively.

As for the extra cost of different tasks arriving, when there are many arriving tasks, the controller may have a bottleneck because it needs to coordinate each type of task. The multi-controller model can effectively solve this problem. Therefore, Tiansuan constellation has designed a mode that two main satellites jointly provide services to share the cost of the controller [3].

Simulation Results

To evaluate the performance of the proposed architecture, we design a basic application scenario and show the results of its average throughput.

Simulation Setup

In this section, the considered scenario includes a ground control center, a main satellite, and 64 edge satellites. Among them, edge satellites are uniformly distributed in 8 orbital planes with 8 satellites in each orbit plane, forming a Manhattan network. The average transmission speed of the inter-satellite link, transmission delay, transmission packet loss rate, and computing power of each satellite is set to be 100 MBps, 50 ms, 0.1, and 5GB/s, respectively. Many satellite task models adopt Poisson distribution to simulate the arrival of tasks [15]. In our simulation experiments, the number of arrived edge satellite data transmission requests per second follows the Poisson distribution and is recorded as

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, ..$$

where λ is set to be 1. The data transmission amount follows the normal distribution $\mathcal{N}(10, 1^2)$ (MB). The destination nodes of data transmission are uniformly distributed within the range and are selected randomly as any edge satellite node except the source node. The ground control center can accept users' remote sensing image acquisition requests. The arrival of data transmission requests per second follows the Poisson distribution

$$P(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, k = 0, 1, ...$$

with λ set to be 5. The amount of collected data follows the normal distribution $\mathcal{N}(2, 0.1^2)$ (GB). The position of the receiver of remote sensing images is as assumed to be static. The location of remote sensing image acquisition follows the uniform distribution, which is randomly assigned to one other remote sensing satellite. After remote sensing image processing, the data is compressed to 1 percent of the original size.

Comparison of Throughput

In this simulation, the main satellite first uses the Stackelberg game method to determine the price of its resources. The optimal amount of resources to rent is then obtained by users according to the resource pricing. An edge satellite uses Luby Transform (LT) code to encode the data, with redundancy set to 0.3. The main satellite determines transmission paths to control data transmission. The delay of determination and distribution of the transmission strategy is set to be 300ms. To better demonstrate the performance of the proposed method, we design two greedy algorithms in this scenario to select the shortest path for data transmission:

- Greedy algorithm without data coding. Packets are retransmitted when it is lost, referred to as greedy-retransmission (Greedy-RT) algorithm
- Greedy algorithm with coding before data transmission, referred to as greedy-network coding (Greedy-NC) algorithm.

The computer simulation runs for one minute, during which we assume that the satellite's movement relative to the ground does not affect the existing data links. We compare the throughput of the three aforementioned methods, as shown in Fig. 6.

Figure 6 shows that in the initial stage of simulation, the average throughput of the greedy network coding algorithm is close to that of the proposed algorithm. This is because, with less network congestion, most of the data can be successfully transmitted. However, the throughput of the greedy algorithm without network coding is relatively low. This is because each packet needs to be confirmed and retransmitted when packet loss occurs, which increases the transmission time and reduces the average throughput. With time, the throughputs of the three algorithms gradually tend to be stable, and the throughput difference between the greedy algorithm with network coding and the algorithm proposed in this article increases. This is because as time goes on, the congestion in the greedy algorithm increases gradually, leading to a decrease in throughput. The experimental results show that the proposed framework can effectively improve the overall throughput of the network.

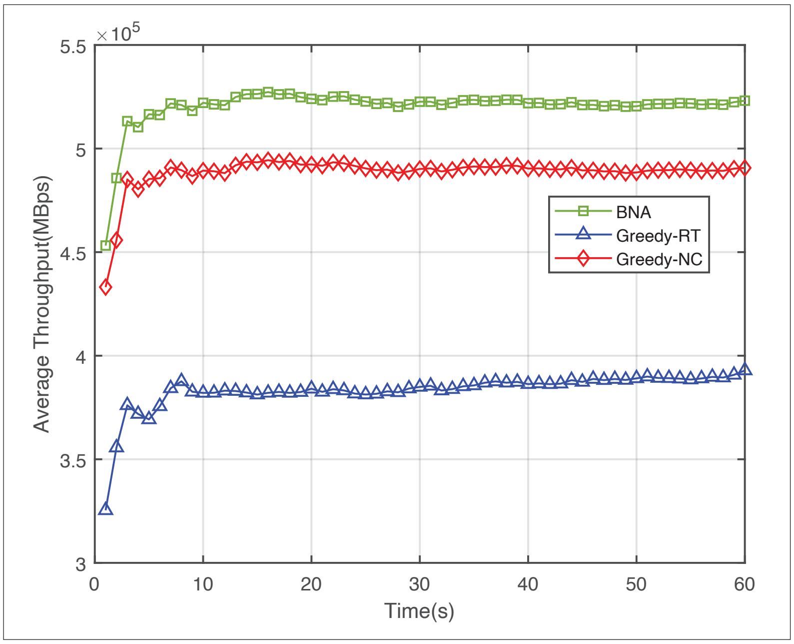

FIGURE 6. Comparison on average throughput.

Conclusion

With the on-orbit computation capability provided by Tiansuan constellation, this article proposes a hybrid architecture for Intelligent Space-Terrestrial Integrated Networks. The proposed architecture aggregates a centralized intelligent controller and distributed intelligent agents. The proposed architecture can provide intelligent data transmissions, intelligent remote sensing, and intelligent network slicing. The simulation results demonstrate the effectiveness of the improvement of the proposed architecture on the effectiveness of the intelligent architecture. As future work, we will consider the method of multi-controller cooperation, focusing on solving the load balancing problem facing massive tasks, to further improve the intelligent network architecture.

Acknowledgment

This work was supported by Beijing Natural Science Foundation (Grant No.L212032), the National Key R&D Program of China (Grant 2019YFE0114000) and the National Natural Science Foundation of China (Grant No.62032003).

References

[1] K. Lin *et al.*, "Virtualized QoS-Driven Spectrum Allocation in Space-Terrestrial Integrated Networks," *IEEE Network*, vol. 33, no. 1, 2019, pp. 58–63.

[2] S. Chen *et al.*, "Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed," *IEEE Wireless Commun.*, vol. 27, no. 2, Apr. 2020, pp. 218–28.

[3] S. Wang *et al.*, "Tiansuan Constellation: An Open Research Platform," *Proc. IEEE Int'l Conf. Edge Computing*, Chicago,

IL, USA, Sept. 5-10, 2021, pp. 94–101, DOI: 10.1109/EDGE53862.2021.00022.
[4] J. Fang and A. Ma, "IoT Application Modules Placement and Dynamic Task Processing in Edge-Cloud Computing," *IEEE Internet of Things J.*, vol. 8, no. 16, 2021, pp. 12,771–81.
[5] S. Tu *et al.*, "Social Phenomena and Fog Computing Networks: A Novel Perspective for Future Networks," *IEEE Trans. Computational Social Systems*, vol. 9, no. 1, Feb. 2022, pp. 32–44.
[6] C. L. Stergiou *et al.*, "Secure Machine Learning scenario from Big Data in Cloud Computing via Internet of Things Network," *Handbook of Computer Networks and Cyber Security, Principles and Paradigms*, Sept. 2020, pp. 525–54.
[7] C. L. Stergiou and K. E. Psannis, "Digital Twin Intelligent System for Industrial Internet of Things-based Big Data Management and Analysis in Cloud Environments," *Virtual Reality & Intelligent Hardware*, vol. 4, no. 4, 2022, pp. 279–91.
[8] A. P. Plageras *et al.*, "Efficient IoT-Based Sensor BIG Data Collectionprocessing and Analysis in Smart Buildings," *Future Generations Computer Systems: FGCS*, vol. 82, May 2018, pp. 349–57.
[9] C. L. Stergiou, K. E. Psannis, and Y. Ishibashi, "Green Cloud Communication System for Big Data Management," *Proc. 2020 3rd World Symposium on Communication Engineering*, Thessaloniki, Greece, Oct. 9–11, 2020, pp. 69–73, DOI: 10.1109/WSCE51339.2020.9275579.
[10] C. L. Stergiou, K. E. Psannis, and B. B. Gupta, "IoT-Based Big Data Secure Management in the Fog Over a 6G Wireless Network," *IEEE Internet of Things J.*, vol. 8, no. 7, 2021, pp. 5164–71.
[11] S. Tu *et al.*, "Reinforcement Learning Assisted Impersonation Attack Detection in Device-to-Device Communications," *IEEE Trans. Veh. Tech.*, vol. 70, no. 2, Feb. 2021, pp. 1474–79.
[12] H. Huang *et al.*, "Online Green Data Gathering from Geo-Distributed IoT Networks via LEO Satellites," *Proc. IEEE Int'l. Conf. Commun.*, Kansas City, MO, USA, May 20-24, 2018, pp. 1–6, DOI: 10.1109/ICC.2018.8422522.
[13] X. Xu *et al.*, "On the Aggregated Resource Management for Satellite Edge Computing," *Proc. IEEE Int'l. Conf. Commun.*, Montreal, QC, Canada, June 14-23, 2021, pp. 1–6, DOI: 10.1109/ICC42927.2021.9500539.
[14] C. Qiu *et al.*, "Deep Q-Learning Aided Networking, Caching, and Computing Resources Allocation in SoftwareDefined Satellite-Terrestrial Networks," *IEEE Trans. Veh. Tech.*, vol. 68, no. 6, June 2019, pp. 5871–83.
[15] P. Zou *et al.*, "Optimizing Information Freshness Through Computation–Transmission Tradeoff and Queue Management in Edge Computing," *IEEE/ACM Trans. Networking*, vol. 29, no. 2, Apr. 2021, pp. 949–63.

Biographies

Xiaobin Xu [M] (xuxiaobin@bjut.edu.cn) is currently an Associate Professor at the Beijing University of Technology. He obtained his Ph.D. at the Beijing University of Posts and Telecommunications in 2014. His current research areas include space-air-ground integrated information network, internet of things, mobile edge computing.

Qi Wang (wangqi74081@emails.bjut.edu.cn) received the B.S. degree from Beijing Information Science and Technology University in 2020. She is currently pursuing the master's degree with the Faculty of Information Technology, Beijing University of Technology. Her current research areas include space-air-ground integrated information network, internet of things, mobile edge computing.

Yanzhao Hou [M] (houyanzhao@bupt.edu.cn) received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2014. He is currently with the National Engineering Research Center for Mobile Network Technologiest, BUPT. His research interests include 6G networks, software defined radio, vehicular networks, terahertz communications and trial systems.

Shangguang Wang [SM] (sgwang@bupt.edu.cn) is a Professor at the Beijing University of Posts and Telecommunications. He is the founder of Tiansuan Constellation. His research interests include service computing, cloud computing, and satellite computing.