STCC-Sim: A Satellite-Terrestrial Collaborative Computing Modeling and Simulating Toolkit for Resource Provisioning

Tianjiao Ma*, Wei Xu*, Zhe Ding‡, Tianyu Wo†, and Xudong Liu*
*School of Computer Science and Engineering, Beihang University†School of Software, Beihang University
‡School of Cyberspace Security, Beijing University of Posts and Telecommunications
{matianjiao, xuwei, woty, liuxd}@buaa.edu.cn*†, zhedingqd@126.com‡

***Abstract*—The Satellite-Terrestrial Integrated Network (STIN) technology based on low-earth orbit (LEO) satellites provides a seamless network with low latency and high reliability to global users. The Space-Terrestrial Collaborative Computing (STCC) paradigm based on STIN has become a promising solution for ubiquitous computing. Due to the mobility of satellites and the vulnerability of inter-satellite and satellite-terrestrial network connections, the research on resource scheduling strategy, configuration, and deployment of STCC services is more complex than that of ground cloud services. The lack of simulators further restricts the development of STCC research. In order to solve this problem, we propose an STCC simulation tool for the satellite-terrestrial hybrid cloud in this paper. The tool can simulate the hybrid network models and support the simulation of the flow computing model. By evaluating the performance of the task offloading policy, we demonstrate the effectiveness of the simulation tool.**

***Index Terms*—LEO satellite constellation, satellite-terrestrial collaborative computing, modeling and simulation, task offloading, resource provisioning**

I. INTRODUCTION

LEO constellations have become an important infrastructure for business giants to provide users with low-latency and high-bandwidth Internet services. The Satellite-Terrestrial Integrated Network (STIN) is a promising architecture combining LEO constellations with ground cellular communication systems. Through 6G technology, satellite, ground, and air network facilities can be interconnected to achieve a highly reliable, low-latency, and seamlessly covered network, which has attracted widespread attention from industry and academia[1] and has broad application prospects in many fields such as telemedicine, autonomous driving, smart cities, and industrial Internet[2].

The Space-Terrestrial Collaborative Computing (STCC) paradigm represents an innovative edge computing framework based on the Space-Terrestrial Integrated Network (STIN). The Satellite Edge Cloud (SEC) is constructed from an array of satellites that are interconnected via inter-satellite links (ISL) and are equipped with advanced computing, storage, and networking devices[3]. The SEC establishes connections with terrestrial cloud infrastructures through satellite-ground links (SGL) to collaboratively execute computing tasks, thereby enhancing task performance and Quality of Service (QoS)[4]. SEC nodes, which are endowed with spatial location awareness and network topology awareness capabilities, possess the ability to dynamically adjust their workloads, allocate resources, and migrate computing tasks following the specific requirements of the tasks[5]. The STCC paradigm can provide resource services to users globally and significantly reduce application latency while improving QoS. The optimal organization and scheduling of SEC resources, as well as the effective realization of satellite-ground collaboration, hinge upon the accurate prediction of the positions of highly mobile LEO satellites and the robust modeling of inter-satellite links and satellite-ground links.

There is currently a substantial body of research on resource scheduling and task offloading for space-ground collaborative computing [4][5][6][7]. However, these studies employ diverse network architectures and lack a unified modeling approach for satellite network links. This heterogeneity has led to a lack of consistent quantitative evaluation within the field, highlighting the urgent need for an evaluation tool. Therefore, the optimal solution is a comprehensively modeled, highly flexible, and scalable simulation software. However, simulation software for Space-Terrestrial Collaborative Computing (STCC) systems faces numerous challenges:

- LEO satellites move with high speed and follow a periodic pattern. The Inter-Satellite Links (ISLs), including intra-orbit and inter-orbit links, and Satellite-to-Ground Links (SGLs) are highly fragile. Accurate prediction of satellite positions is crucial for simulating network links, as link bandwidth and connectivity are subject to fluctuations.
- The modeling and simulation of inter-satellite and satellite-to-ground network routing depend on the satellite constellation. For example, routing within intra-orbit ISLs may follow different rules from inter-orbit ISLs. The complexity of routing algorithms

further complicates the simulation process.

- The STCC system comprises diverse entities with complex relationships. Resources vary in type, size, and requirements. Applications differ in performance, load, and scale. Moreover, different users have heterogeneous, dynamic, and potentially conflicting Quality of Service (QoS) requirements. These factors collectively impose stringent demands on the flexibility of the simulation system.

These challenges make the software simulation of resource provisioning and task offloading for Space-Terrestrial Integrated Network (STIN)-based systems extremely difficult.

According to our research, no existing simulation software can comprehensively address the aforementioned challenges. Therefore, this paper proposes STCC-Sim: a scalable Space-Terrestrial Collaborative streaming Computing Simulator based on STIN. This simulator extends the CloudSim[8], which is an open-source simulation tool for modeling and simulating cloud computing environments. With STCC-Sim, researchers can rapidly evaluate the effectiveness and performance of resource scheduling, task orchestration, load-balancing strategies, and applications in space-terrestrial collaborative computing scenarios. The advantages of this simulation system are as follows:

- Network Transparency: Leveraging ephemeris data, STCC-Sim achieves sub-second simulation of mega constellation networks. The network modeling process is significantly simplified with minimal configuration required from the user. Furthermore, the simulator incorporates built-in routing and forwarding models for inter-satellite links (ISLs) and satellite-to-ground links (SGLs).
- Flexibility: With few configurations, the system can support the modeling and simulation of various computing scenarios. Moreover, it enables the performance evaluation of multiple applications and strategies within each scenario.

The main contributions are:

- We propose a simulation framework for STCC that integrates satellite constellation simulation, satellite network link simulation, and event-driven streaming computing simulation.
- The framework enables sub-second modeling of end-to-end satellite links and network routing.
- Task modeling for space-terrestrial collaborative streaming computing.

Additionally, the effectiveness of STCC-Sim is demonstrated through the evaluation and comparison of the performance of four task offloading strategies.

II. RELATED WORK

This section introduces related research work in STIN and STCC, including the network topology of LEO satellite constellations, STCC architectures, resource scheduling policies, satellite network simulation tools, and other related simulators.

LEO satellite constellation networks have become an important part of the communication system, providing low-latency, high-bandwidth, and highly reliable network services to areas far from densely populated areas. StarLink is the largest low-orbit satellite constellation, with nearly 12,000 satellites planned to be deployed to provide network services to more than 1 million users[9]. In terms of LEO satellite network topology, Bhattacherjee et al.[10] calculated the maximum and minimum distances of inter-satellite links, proposed a grid-based LEO network topology planning scheme, and explored the possibility of using different network topology planning schemes in different dimensions. Handley et al.[11] discussed how to achieve low-latency routing in space and concluded through StarLink simulation that naive routing algorithms will produce large network fluctuations, and proposed three improvement methods: multi-channel communication, separation of uplink and downlink routing, and multiple satellite communication methods. Giuliari et al.[12] discussed the inter-domain routing problem of how LEO giant constellations can be integrated with the terrestrial Internet, and analyzed the advantages and disadvantages of four solutions.

Using computing resources carried by LEO satellites to achieve on-orbit computing has been a research hotspot in recent years. The primary issue is the architecture that should be used to achieve satellite edge computing. Pfandzelter et al.[13] explored the application organization form of edge computing platforms based on low-orbit satellite networks and proposed that considering the high mobility of satellites, applications based on stateful virtual machines and containers face significant difficulties in migration. Therefore, applications based on stateless serviceless computing are the best choice for on-orbit computing. Bhosale et al.[14] proposed a service orchestration framework Krios based on Kubernetes for satellite scenarios, which implemented a service placement and migration strategy aware of satellite-ground collaborative orbits. Qiu et al.[15] proposed a software-defined STN framework to jointly manage and orchestrate network, cache, and computing resources, described the joint resource allocation problem as a joint optimization problem, and used a DQN-based method to solve it to improve performance. Xu et al.[16] proposed a cloud-edge collaborative computing architecture based on STIN, and verified in multiple scenarios that this computing architecture effectively improved the data transmission efficiency of satellite applications. In addition, the study also proved that a global integrated satellite network is challenging to achieve shortly, and a more realistic goal is the application-oriented coordination of multiple LEO constellations.

The above studies show that hierarchical architecture is the mainstream architecture for realizing satellite edge computing, so how formulating the optimal strategy in hierarchical architecture has become the main challenge of the research. Zhang et al.[17] proposed an intelligent computing offloading scheme based on a deep deterministic policy gradient algorithm. The scheme consists of multiple different deep neural networks, which can output discrete and continuous variables at the same time. It can adapt to dynamic environments and achieve simultaneous decision-making on offloading locations and resource allocation under multi-task concurrency. Song et al.[18] proposed a novel MEC framework to explore the task processing capabilities of satellites for IoT mobile devices and decomposed the optimization problem into two hierarchical sub-problems corresponding to the space and ground parts. Tang et al.[19] proposed a hybrid cloud and edge computing LEO satellite network to provide users with heterogeneous computing resources.

Traditional satellite network simulation tools, such as OPNET[20] and OMNet++[21], are capable of modeling satellite network resources in detail but lack the simulation of computing models and algorithms. CloudSim[8] is a software framework for simulating cloud computing environments, which can help researchers and developers test and evaluate different cloud computing strategies and applications. However, it cannot fully reflect some factors in real cloud environments, such as network latency, security risks, and reliability. EdgeCloudSim[22] is a simulator for evaluating the performance of edge computing systems. It provides a modular architecture and supports multiple core functions, such as network modeling for WLAN and WAN, mobility models for devices, and load generation models. It uses configuration files to process parameters in the computing system and supports the generation of tasks in a probabilistic distribution manner. SatEdgeSim[23] focuses on tsimulating three-layer satellite network models: cloud, edge, and fog. However, SatEdgeSim has insufficient support for details on the satellite network implementation.

III. System Architecture

This section presents the architecture of STCC-Sim. Ohter than the overview about the newly created and extended module, it also presents link model, network topology model, route forward model, and task model. The simulator aims to ease-up the process of building satellite-terrestrail computing scenario and to speed up the provisioning policy research.

A. Architecture

Fig. 1 shows the three-layer system architecture of STCC-Sim, where the red box content is the module newly created or extended from CloudSim. Regarding the network, STCC-Sim has added the Location Service module to manage the positioning of satellites and ground terminals. Users can manage the positions of various

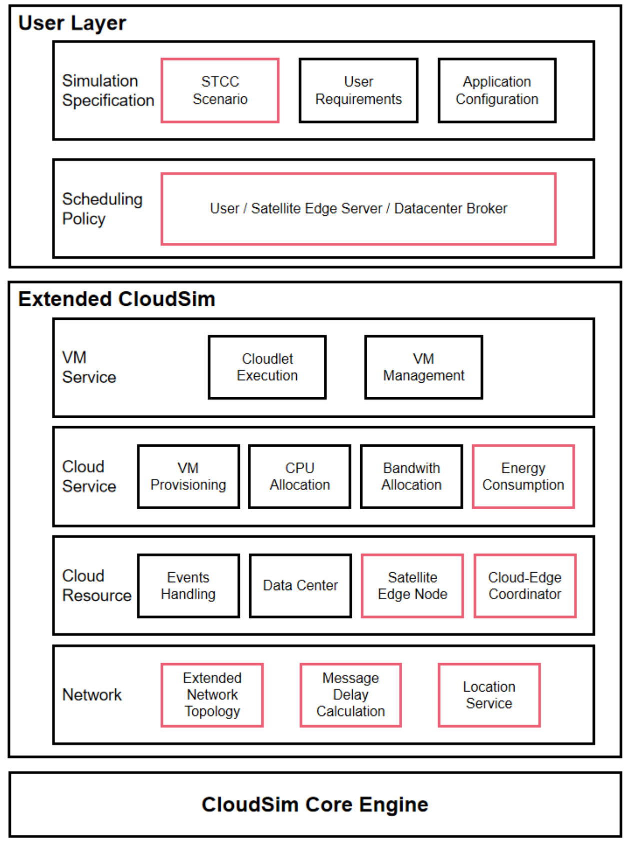

Fig. 1. STCC-Sim Architecture

entities in the simulator at different times by configuring the ephemeris of satellite entities or the coordinate transformation table of ground entities, thereby realizing the simulation of high dynamic characteristics in satellite-terrestrial collaborative computing. At the same time, STCC-Sim has expanded the network topology and message delay calculation module, added a routing model for giant constellations, and realized the simulation of network transmission delay and bandwidth of inter-satellite links and satellite-ground links. In terms of cloud resources, the Satellite Edge Nodes and Cloud-Edge Collaboration modules have been added to simulate satellite edge computing resources and realize satellite-ground collaborative resource scheduling. Due to the limited power provided by satellites, in order to simulate energy constraints, the cloud service layer has added an energy consumption module to evaluate the performance of scheduling strategies in terms of energy consumption. For users' convenience, STCC-Sim has added scenario configuration and expanded the cloud-edge agent. Users can simulate scenarios through simple configuration.

STCC-Sim is easy to use. The only extra job that user should do is to provide the configuration to STCC Scenario module in order to build a scenario. The Location Service module generates satellite ephemeris, ISL length, and other necessary information based on user's configuration. These information is then stored in time-varying topology information database. Then like the CloudSim, user could run simulations of satellite-terrestrial coordinative computing.

B. Link Model

STCC-Sim uses the Walker constellation [24] as the configuration of the LEO constellation by default. The Walker constellation has the characteristics of uniform distribution and high symmetry. It can be represented by $(N \times P, P, F, h, i)$, where N represents the number of satellites in each orbit, P represents the number of orbits, $F \in \{0, 1, ..., N-1\}$ is the phase factor, h is the orbit height, and i represents the orbit inclination. Users can set constellation parameters through the STCC scenario configuration module to simulate different constellations.

The relative position between two adjacent satellites in the same orbit is relatively fixed in a constellation, so a relatively stable inter-satellite link can be formed. For inter-satellite links across orbits, since the relative position changes between satellites in adjacent orbits are relatively large, STCC-Sim stipulates that inter-satellite links can only be established between satellites in adjacent orbits when the conditions are met. As shown in Fig.2, a cross-orbit inter-satellite link can only be established when the angle $\Delta\omega$ between two satellites in adjacent orbits is less than the value set by STCC-Sim. In addition, if the number of orbits is small and the orbital altitude is insufficient, it may be impossible to establish a cross-orbit inter-satellite link due to the influence of the earth's curvature.

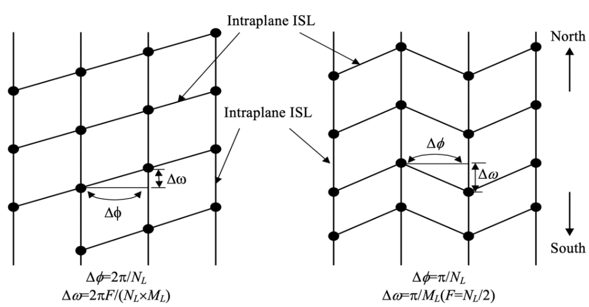

Fig. 2. ISL Connectivity

The connectivity of the satellite-ground link depends on the ground coverage model of the satellite signal. As shown in Fig.3, O is the center of the earth, R is the radius of the earth, G is the ground station, S is the sub-satellite point, h is the orbital height of the satellite, λ is the elevation angle of the ground station relative to the satellite, e is the geocentric angle between the user and the satellite, α is the semi-depression angle of the satellite, and d is the distance between the ground station and the satellite. The satellite-ground link can be established only when the ground station G is within the satellite coverage area.

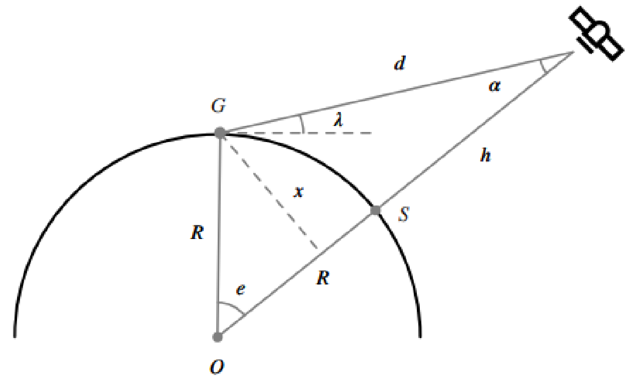

Fig. 3. Satellite Coverage

After the user configures the constellation parameters and simulation time through the STCC Scenario module, the Location Service module of the network layer can generate ephemeris, calculate the position of each satellite within the simulation time, and assist in the simulation of the network topology.

C. Network Model

Fig. 4 illustrates the network architecture of the scenario simulated by STCC-Sim. The ground data center constitutes the ground cloud, which has sufficient computing, storage, and network resources to provide cloud computing services for ground users and satellite terminals. The low-orbit satellite constellation composed of LEO edge computing nodes constitutes the satellite edge cloud, which provides edge computing services for users and satellite terminals that cannot access the ground network. The ground-satellite terminal[18] (Terrestrial-satellite terminal, TST) serves as an access point for mobile devices, enabling ground devices to communicate with satellites but does not have computing capabilities. Based on virtualized network technology, TST serves as a gateway to connect the ground cloud and the satellite edge cloud.

The most important part of the network model is the calculation of network latency, of which the calculation of inter-satellite links and satellite-ground links is the core. Since network latency is proportional to the length of the link, thus the calculation of network latency depends on the calculation of link length. As shown in Fig.2, if the number of satellites in orbit is N, the satellite orbit height is h, and the radius of the earth is R, the length of ISL in orbit can be calculated as

$$l_{intra} = (R+h)\sqrt{2 - 2\cos(\frac{2\pi}{N})}$$

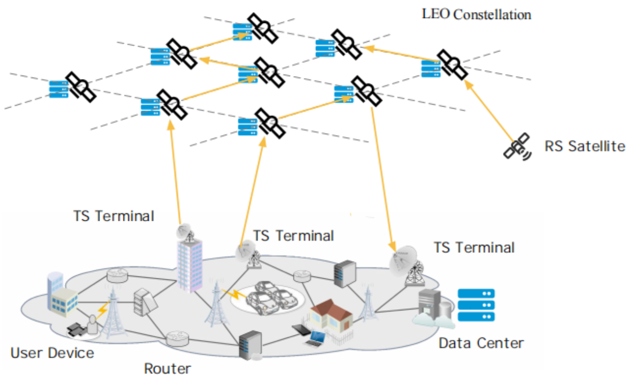

Fig. 4. STCC-Sim Network Architecture

Similarly, the interplane ISL length is

$$l_{inter} = \frac{\cos(lat)(R+h)\sqrt{2-2\cos(\frac{2\pi}{N})}}{\cos\Delta\omega}$$

where lat is the latitude of the satellite. The corresponding link propagation latency are $\frac{l_{intra}}{c}$ and $\frac{l_{inter}}{c}$, where c is the speed of light.

The satellite-ground link calculation model is shown in Fig.3. It can be seen that the satellite-ground link length is

$$d = \sqrt{R^2\sin^2\lambda + 2Rh + h^2} - R\sin\lambda$$

The satellite-to-ground link delay is $\frac{d}{c}$.

Satellites are always moving at high speed, but the satellite network topology is periodic and predictable, so the network topology module of STCC-Sim adopts a topology snapshot strategy. This strategy divides the network topology into multiple time slices according to the time dimension, assuming that the network topology within the time slice remains unchanged. Therefore, the network topology can be calculated at any time based on the satellite position predicted by the ephemeris.

D. Routing Model

The packet forwarding process in the node is shown in Fig.5. When the node has data to transmit through the network, it first enters the storage buffer to queue and waits for transmission. The storage buffer transmits packets to the network at a specific transmission rate. The packets in the storage buffer and the packets sent from other links are forwarded to the transmission queue of the corresponding link after the node queries the routing table. The transmission delay from node i to node j includes queuing, routing, and link latencies. Since the processing latency of the router is generally within milliseconds, it is ignored here.

Let $q(t)$ represent the length of the sending queue at time t, then the queuing latency is

$$T^{ij} = \frac{P_{avg}}{\Delta t B_{ij}} \int_t^{t+\Delta t} q(x)dx$$

Where B_{ij} represents the bandwidth of link (i, j) and P_{avg} is the average packet size.

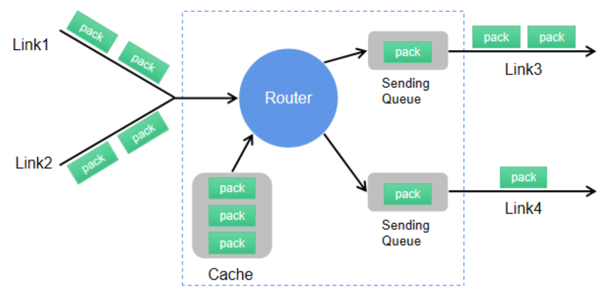

Fig. 5. Network Route Forward Model Sketch

E. Task Model

STCC-Sim supports streaming computing models, and the computing task is defined as a Service Function Chain (SFC). The task source can be a ground user terminal or a functional satellite (such as a remote sensing satellite), and the computing service can be deployed in a data center or on a satellite edge cloud node. The yellow arrow in Fig.4 shows an example of the task data flow.

In terms of implementation, SFC is defined by a DAG consisting of several Cloudlets. Each Cloudlet is sent to the target node through a routing strategy. The node's resource allocation strategy implements resource allocation to complete the calculation of the Cloudlet and triggers a completion event to generate the next Cloudlet until the entire computing task is completed. Basic data such as resource allocation and node resource utilization during the task completion process are collected to evaluate routing selection, resource allocation, and task offloading strategies.

IV. Experiments and Evaluation

This section evaluates the performance of several task offloading strategies in ground computing and remote sensing satellite image processing tasks through a series of experiments to verify the effectiveness of the simulator.

A. Offloading Policy

There are four task offloading strategies evaluated, namely random strategy, greedy strategy, local offloading strategy, and e-greedy strategy. The random strategy means the task randomly selects a satellite edge cloud node or data center as the offloading destination. The greedy strategy is to offload the task to the globally idlest node. The local offloading strategy is a strategy that selects the idlest node in the node group where the task-receiving node is located. The e-greedy strategy is an adaptive local offloading strategy, and the offloading decision is based on the load of all nodes whose offloading path length is less than e at the current node.

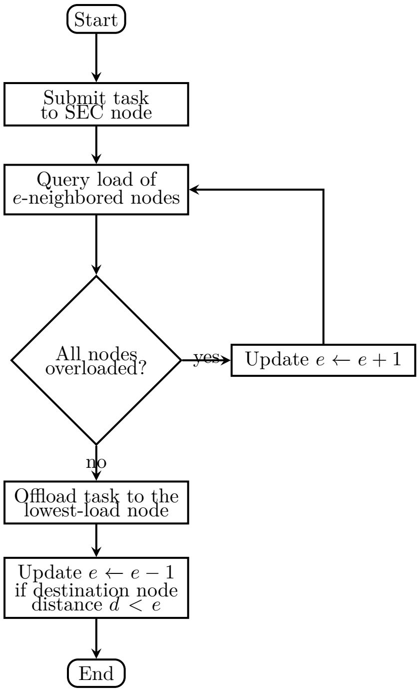

Fig. 6. Flow Chart of e-greedy Policy

As shown in Fig.6, the e-greedy policy maintains a load table of nodes with an adjacent distance of e. Each offloading decision selects the node with the lowest load from the table as the offloading destination. If the load of all nodes is higher than the threshold, the search range is increased. Finally, the parameter e is updated according to the distance between the target and decision nodes.

B. Experiment Setup

In order to verify the effectiveness of STCC-Sim, the experiment uses a complete satellite-ground collaborative computing scenario. The task types include ground computing tasks and Remote Sensing (RS) tasks. The SFCs of the two types of tasks are linear, with lengths of 4 and 6, respectively. The experiment is divided into two groups, and the above four strategies are used to make offloading decisions for the two types of tasks to observe whether STCC-Sim can evaluate the performance of these strategies.

The two groups of experiments have the same configuration except for the number of tasks. The total number of tasks for ground computing tasks is 50 to 300, divided into six groups of simulation experiments, and the task submission time follows a Poisson distribution. The total number of remote sensing tasks is 50 to 250, with five groups of experiments, and the task submission time follows a uniform distribution. The evaluation indicators include resource utilization and task delay. Other parameters are shown in Table I.

Name of Parameter	Value
Number of TST	20
Number of DC	10
Number of orbital planes	10
Number of satellite per Plane	12
Orbital inclination	86°
Bandwidth of ISL	25Mbps
Bandwidth of SGL	10Mbps
Routing table update interval	1s
Satellite edge host CPU capacity	1,000MIPS
DC host CPU capacity	10,000MIPS
Input data of ground task	100-500KB
Input data of RS task	1,000KB
Input-output ratio of ground task	10-50%
Input-output ratio of RS task	90%
Simulation time	10min

TABLE I
Configuration of Simulation Experiments

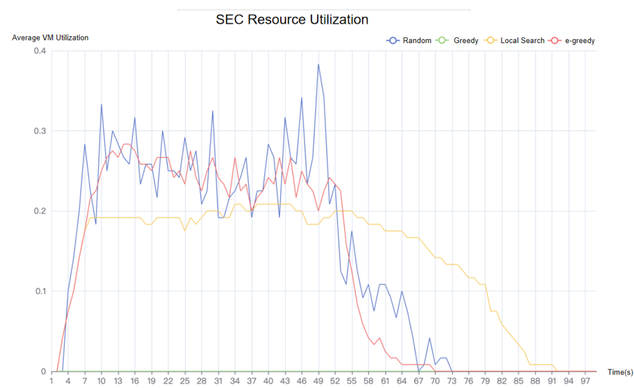

Fig. 7. SEC VM Utilization

C. Result Analysis

1) Ground Task: The ground user terminal submits the ground task to the SEC. After receiving the task, the SEC node selects the appropriate computing node for calculation according to the offloading policy. The task may be calculated at the satellite edge node or offloaded to the node in the data center, and the final result is returned to the ground user terminal.

As shown in Fig.7 and Fig.8, these are the average resource utilization of SEC and ground cloud nodes when the total number of tasks is 300. It can be seen that when the random strategy is adopted, the resource utilization of SEC is the highest, followed by the e-greedy strategy. The e-greedy makes the most effective use of ground cloud node resources, followed by the greedy policy.

In addition to resource utilization, the experiment uses latency indicators to evaluate the strategy. As shown in Fig.9 and Fig.10, they are the task's transmission delay and end-to-end delay, respectively. The greedy policy always selects the globally idlest node as the offloading target, so the transmission and end-to-end delays are the highest. The e-greedy strategy performs best in latency control.

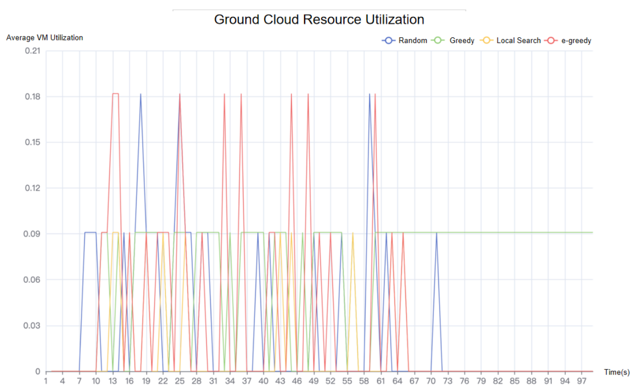

Fig. 8. Ground Cloud VM Utilization

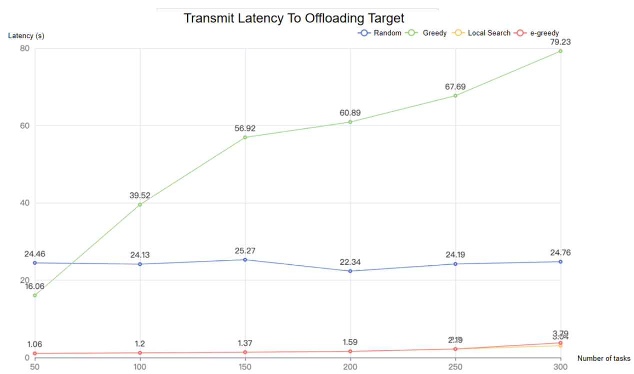

Fig. 9. Total Transmit Latency

2) RS Task: The remote sensing task is an image processing task initiated by remote sensing satellites. It is a latency-sensitive task. This type of task needs to be transmitted (or calculated) by the SEC node and finally transmitted to the ground data center. As shown in Fig.11, the end-to-end latency increases with the number of tasks. Among all strategies, the *e*-greedy strategy has the best effect and can effectively reduce the latency.

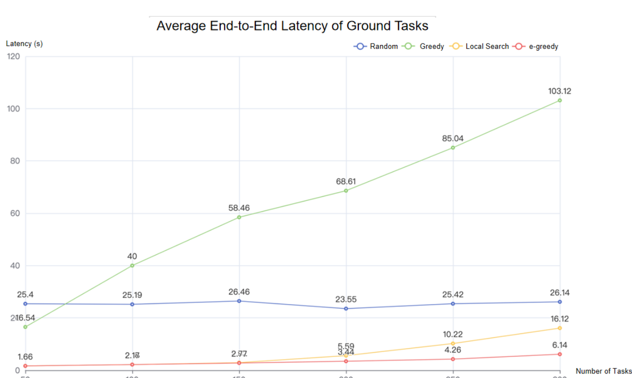

Fig. 10. Average End-to-end Latency of Ground Tasks

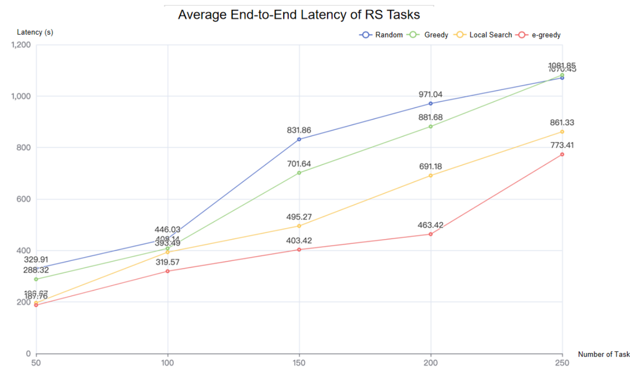

Fig. 11. Average End-to-end Latency of RS Tasks

V. Conclusion and Future Work

In satellite-ground collaborative computing, researchers need to deal with the highly dynamic and time-varying topology of satellite-terrestrial collaborative networks while considering computing paradigms such as cloud computing and edge computing. However, the existing simulators do not support this scenario enough, which restrains the research process. In response to the above problems, this paper proposes a simulator STCC-Sim, focusing on solving the simulation problem of the dynamic topology of the satellite-terrestrial collaborative network. It helps researchers evaluate multiple strategies, such as task offloading and resource allocation, in multiple application scenarios based on a few configurations. Using this simulation system, researchers can conduct research such as scheme comparison, parameter evaluation, and task scheduling and no longer need to repeatedly and tediously build satellite-ground collaborative computing scenarios.

At present, STCC-Sim is still far from being fully completed. First, STCC-Sim only supports two types of task models for streaming computing. The parallel tasks are not yet supported, which is a gap from the actual usage scenarios. Secondly, many parameters in STCC-Sim's scenario modeling are not configurable, and the lack of flexibility restrains users from researching more scenarios. Third, STCC-Sim has not yet considered the disturbance and fault tolerance of wireless communications in the link model, which is quite different from the actual communication scenario. Fourth, the simulation evaluation indicators of the STCC-Sim system are not rich enough, and more indicators need to be added to evaluate the strategy performance in more detail.

References

[1] S. M *et al.*, "Terrestrial-satellite communication techniques and challenges for 6g," in *ICICNIS 2024*. DOI: 10.1109/ICICNIS64247.2024.10823263.

[2] R. Wang *et al.*, "Acbot: An iiot platform for industrial robots," *Frontiers of Computer Science*, vol. 19, no. 4, p. 194 203, 2025.

[3] B. Denby and B. Lucia, “Orbital edge computing: Nanosatellite constellations as a new class of computer system,” in *ASPLOS 2020*, Association for Computing Machinery, 2020, ISBN: 9781450371025. DOI: 10.1145/3373376.3378473.

[4] X. Wen *et al.*, “A hierarchical game framework for win-win resource trading in cognitive satellite terrestrial networks,” *IEEE Transactions on Wireless Communications*, vol. 23, no. 10, pp. 13 530–13 544, 2024. DOI: 10.1109/TWC.2024.3403097.

[5] W. Lan *et al.*, “Security-sensitive task offloading in integrated satellite-terrestrial networks,” *IEEE Transactions on Mobile Computing*, vol. 24, no. 3, pp. 2220–2233, 2025. DOI: 10.1109/TMC.2024.3489619.

[6] X. Zhang *et al.*, “Cost-effective hybrid computation offloading in satellite-terrestrial integrated networks,” *IEEE Internet of Things Journal*, vol. 11, no. 22, 2024. DOI: 10.1109/JIOT.2024.3424782.

[7] C. Zhou *et al.*, “Deep reinforcement learning for delay-oriented iot task scheduling in sagin,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 911–925, 2021. DOI: 10.1109/TWC.2020.3029143.

[8] R. Andreoli, J. Zhao, T. Cucinotta, and R. Buyya, “Cloudsim 7g: An integrated toolkit for modeling and simulation of future generation cloud computing environments,” *Software: Practice and Experience*, 2025.

[9] J. C. McDowell, “The low earth orbit satellite population and impacts of the spacex starlink constellation,” *The Astrophysical Journal Letters*, vol. 892, no. 2, p. L36, 2020.

[10] D. Bhattacherjee and A. Singla, “Network topology design at 27,000 km/hour,” in *CoNEXT 2019*, New York, NY, USA: Association for Computing Machinery, 2019, ISBN: 9781450369985. DOI: 10.1145/3359989.3365407.

[11] M. Handley, “Delay is not an option: Low latency routing in space,” in *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, New York, NY, USA: Association for Computing Machinery, 2018, ISBN: 9781450361200. DOI: 10.1145/3286062.3286075.

[12] G. Giuliari *et al.*, “Internet backbones in space,” *SIGCOMM Comput. Commun. Rev.*, vol. 50, no. 1, pp. 25–37, Mar. 2020, ISSN: 0146-4833. DOI: 10.1145/3390251.3390256. [Online]. Available: https://doi.org/10.1145/3390251.3390256.

[13] T. Pfandzelter, J. Hasenburg, and D. Bermbach, “Towards a computing platform for the leo edge,” in *EdgeSys 2021*, New York, NY, USA: Association for Computing Machinery, 2021, ISBN: 9781450382915. DOI: 10.1145/3434770.3459736.

[14] V. Bhosale, K. Bhardwaj, and A. Gavrilovska, “Toward loosely coupled orchestration for the LEO satellite edge,” in *HotEdge 2020*, USENIX Association, Jun. 2020.

[15] C. Qiu *et al.*, “Deep q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, 2019. DOI: 10.1109/TVT.2019.2907682.

[16] X. Xu *et al.*, “Ai-space: A cloud-edge aggregated artificial intelligent architecture for tiansuan constellation-assisted space-terrestrial integrated networks,” *IEEE Network*, vol. 37, no. 2, 2023. DOI: 10.1109/MNET.003.2200401.

[17] H. Zhang *et al.*, “Satellite edge computing with collaborative computation offloading: An intelligent deep deterministic policy gradient approach,” *IEEE Internet of Things Journal*, vol. 10, no. 10, 2023. DOI: 10.1109/JIOT.2022.3233383.

[18] Z. Song, Y. Hao, Y. Liu, and X. Sun, “Energy-efficient multiaccess edge computing for terrestrial-satellite internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 202–14 218, 2021. DOI: 10.1109/JIOT.2021.3068141.

[19] Q. Tang, Z. Fei, B. Li, and Z. Han, “Computation offloading in leo satellite networks with hybrid cloud and edge computing,” *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9164–9176, 2021. DOI: 10.1109/JIOT.2021.3056569.

[20] X. Chang, “Network simulations with opnet,” in *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1*, 1999, pp. 307–314.

[21] A. Varga, “Omnet++,” in *Modeling and tools for network simulation*, Springer, 2010, pp. 35–59.

[22] C. Sonmez, A. Ozgovde, and C. Ersoy, “Edgecloudsim: An environment for performance evaluation of edge computing systems,” in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, 2017, pp. 39–44. DOI: 10.1109/FMEC.2017.7946405.

[23] J. Wei *et al.*, “Satedgesim: A toolkit for modeling and simulation of performance evaluation in satellite edge computing environments,” in *2020 12th International Conference on Communication Software and Networks (ICCSN)*, 2020, pp. 307–313. DOI: 10.1109/ICCSN49894.2020.9139057.

[24] J. Walker, “Some circular orbit patterns providing continuous whole earth coverage,” *Journal of the British Interplanetary Society*, vol. 24, pp. 369–384, 1971.