

LEOCN: Real-time and complete network simulation framework for LEO constellation networks¹

Cong Li^a, Haiyang Wang^a, Weipeng Liang^a, Yunhao Li^a, Hongyang Yan^{a,*}, Teng Huang^a, Bo Li^b and Jianwei He^b

^a *Institute of Artificial Intelligence, Guangzhou University, Guangzhou, Guangdong, People's Republic of China*
E-mails: 2462148389@qq.com, 2143845889@qq.com, 2909847341@qq.com, li.yunhao@foxmail.com, hyang_yan@gzhu.edu.cn, huangteng1220@buaa.edu.cn

^b *CASIC Space Engineering Development Co., Ltd, 430416 Hubei, People's Republic of China*
E-mails: kj0038@casic.com.cn, kj0030@casic.com.cn

Received 10 September 2022

Accepted 20 June 2023

Abstract. With the development of the space industry, LEO simulation platforms are increasingly important. Prior work mainly focused on the dynamics of constellation topology, routing algorithms against the topological motion, and the behavior of LEO constellation networks, but neglected real-time and completeness properties. In our work, we design a real-time and complete network simulation framework, named LEOCN, that is more meaningful and detailed for researchers than previous work on LEO satellite constellation simulations. LEOCN is supported by Linux System, which makes the framework more realistic and relieves the computational burden. Meanwhile, LEOCN deeply considers topology and routing algorithms, visualization, satellite mobility, communication protocols, and network traffic problems. On LEOCN, we can deploy routing algorithm experiments via SDN. We evaluate the performance of LEOCN in large-scale constellation scenarios, and the results demonstrate the superiority of the real-time and complete network simulation framework.

Keywords: LEO, Satellite, Constellation, Simulation, Mininet

1. Introduction

Low Earth Orbit (LEO) constellations such as Starlink [23] are groups of satellites that are below the altitude of 2,000 km, working together as a system. With the development of carrier rockets, thousands of LEO satellites pour into space, and the age of LEO constellation networks has been coming up.

As one of the most promising fields, the LEO satellites have the characteristics of high bandwidth and low latency, which are able to support more services than medium or geostationary Earth orbit (MEO/GEO) satellites. Delay-sensitive services, such as online games and real-time communication, are more fitted with the LEO network. The LEO satellite networks can be applied to sparsely populated areas that are challenging to cover by the Internet, such as deserts, forests, and oceans. Compared to conventional networks, the one based LEO constellations can bring lower costs than the fiber-optic connections, especially for ships and airplanes that are hard to use terrestrial networks.

¹Supported by Sec AI&Comp Lab.

*Corresponding author. E-mail: hyang_yan@gzhu.edu.cn.

Although the LEO constellation network has caught the eyes of the whole world, several challenges still stand in the way of it being used on a wide enough scale to replace the use of fiber networks as a primary linking method. The speed of LEO satellites reaches around 27,000 km/s [2], at which the LEO satellites circle the planet every 90 minutes. The high velocity severely affects the topology of the LEO constellation network, making the topology extremely unstable.

The instability of the topology nevertheless leads to more questions. Zhang *et al.* [42] proposed that there is a great need to study the handover strategies for the giant LEO satellite networks, like which satellites a user terminal needs to connect to during a handover. Therefore, the balance among handover delay, probability of erroneous handover, resource utilization need us a long way to go. On LEO constellation networks, conventional routing algorithms have to take much traffic for updating and synchronizing strategies because of frequent alteration of networks such as Open Short Path First (OSPF). The shortest path routing algorithm has large overhead [9] with the problem of instability, and it loses sight of trade-off against network preferences such as delay and bandwidth due to the instability problem, and it loses sight of the trade-off between network preferences such as latency and bandwidth [21]. The aforementioned challenges seek an appropriate LEO constellation network simulator to analyze and address them. In this paper, we propose a real-time and complete network simulation for the LEO constellation, LEOCN, to accelerate research.

LEOCN mainly focuses on the real-time and completeness of the network, which are missing from previous frameworks [10,15]. The real-time nature gives researchers the freedom to manipulate nodes and interact with the platform, and users can assign tasks or change system settings in real-time to observe system outcomes. There are nine task templates in LEOCN for providing examples to construct new tasks. Completeness in constellation networks digs up more unexpected phenomena that are possible in the real world. LEOCN has an analog module to simulate traffic that could be found on a traditional network based on the completeness of the framework. Mininet [22] plays a crucial role in ensuring these features, such as it provides the support of the data link layer and above, and possesses properties such as bandwidth, latency, jitter, and loss rate that make the network more realistic.

LEOCN consists of six modules, each of which focuses on different functional modularity and provides a diverse set of features designed for diverse objectives.

- **Node:** This module is responsible for the configuration and management of nodes in the LEO constellation. Each node has an independent environment, which is formed as a result of perfect isolation between any two nodes. There are 3 node scenarios in LEOCN: ground stations, satellites and mobility switches.
- **Link:** A link instance exists when two nodes are able to communicate with each other. It is in charge of link properties such as latency and loss rates, connection status, and whether two nodes can link to each other. The link module has 2 patterns, inter-satellite link (ISL) and ground to satellite link (GSL). They both require distance-based updates for all propagation delays.
- **Constellation:** It is designed to construct the initial topology of the entire LEO constellation by reading the configuration information from the configuration file. The constellation generator of LEOCN is based on Hypatia [15].
- **Topology:** This module dynamically updates the network topology over time, while the routing policy is refurbished. Two main modules are included here: the mobility module and the routing algorithm module. The topology of the LEO satellites is designed to be a staggered mesh [13,16].
- **Network:** The top layer in LEOCN, the network module, is able to control all the others. Therefore, it is mainly used to maintain the stable operation of the network.
- **Monitor:** The monitoring module keeps track of the state of each node, such as throughput and latency, for the convenience of the researcher, from which the experimental results are mainly obtained.

To summarize, we have the following contributions:

- We propose a real-time and complete network simulation framework that is more user-friendly and detailed for researchers than previous work against the LEO satellite constellation.
- LEOCN can deploy routing algorithms and conduct experiments via SDN or node-deployment.

Numerous researchers prefer the Systems Tool Kit (STK) [31] due to its substantial tool package. STK also supports the deployment of LEO satellite constellations. However, it is not very excellent at network experiments. LEOCN makes up for this shortcoming. We hope that more and more researchers will focus on both LEO constellations and networks.

The rest of this paper is organized as follows. In Section 2, we describe the background of LEO and the tools of our platform. In Section 3, we describe the related work on LEO constellation simulation. In Section 4, we proposed the architecture of LEOCN. Then we introduced concrete implementation of LEOCN, especially in terms of topology, mobility, routing algorithms and visualization in Section 5. The results and evaluation of simulation for the simulation utilization, memory usage and network latency are shown in Section 6. In Section 7, we discuss what has been done and what is to come, and finally, we conclude the paper in Section 8.

2. Background

The Low Earth Orbit (LEO) satellite, with altitude ranging from 250 km to 2,000 km above the Earth's surface and orbital periods ranging from about 84 to 127 minutes, has been developing in the 5G period [7,11] and is expected to play a key role in providing ubiquitous Internet and communications services of 6G period in the future. Compared to GEO and MEO satellites, LEO satellites have the advantages of lower emission cost [17], higher network capacity, higher resolution of Earth observation and better signal quality reception, and alleviate the problem of Antenna angle, polar coverage and time delay with regard to communication. For instance, recent work shows that it could offer lower latency than not only present fiber infrastructure, but also specialized free-space radio networking used in the high-frequency trading industry over long distances. Therefore, SpaceX Starlink, OneWeb and others are deploying hundreds to thousands of satellites to provide global broadband Internet service [10].

Nevertheless, orbital periods between 90 and 130 min render short visibility episodes from ground sites, where an LEO satellite appears on the horizon and disappears on the opposite side in about 6-12 min. Depending on the latitude of the location, an LEO satellite will appear over a spot two to four times a day contributing to the disadvantages of longer distances and faster satellite switching, and these networks also demand unique design and operational challenges especially in Physical topology design, routing [44] and congestion control [1]. As time goes by, the requirements toward data processing speed, network resources utility, energy saving, privacy protection, user quality of user service and survivability under link failures are higher and higher. Deng et al. [8] proposed an iterative optimization algorithm based on the Stackelberg game developed to maximize resource utility. By analyzing the shortcomings with regard to energy saving, Gregory et al. [29] present the crucial components of software infrastructure for safe, fully automated, energy-effective, dynamic, and reward-optimal operation of LEO satellites and satellite constellations. In order to cope with the problem of survivability, Shaoqing et al. [38] present k-means and β -means to protect against link failures, while Wang et al. [39] proposed to protect them after recognizing several key nodes in the network. Ting et al. [36] presented a blockchain-based trusted and privacy-preserved authentication scheme for inter-constellation cooperation in SGINs to not only protect the privacy, but improve the survivability of the LEO satellite network.

STK [44] is a prominent business analysis software developed by an American analytical graphics company in the space industry, supports the entire process of space missions, including design, testing, launch, running, and task applications and offers the communication simulation of satellite and ground stations in the graphical interface. It can not only calculate the location and position of the satellite at any time, and provide a satellite orbit generation guide to help users establish common orbits, but display all the information that is based on time, and have a multi-window real-time display of the task scene change. Cesium [4] is a JavaScript based map engine using WebGL which supports 3D, 2D, 2.5D map display and most browsers and mobile. It focuses on the visualization of dynamic geospatial data, with a high performance of system and precision.

The SGP4 (Simplified General perturbation modes) model was developed by Ken Cranford in 1970 for near-Earth satellites. It is an extensive analysis of Lane and Cranford (1969), takes the effects of perturbation forces

such as the non-spherical attraction of the earth, the attraction of the sun and moon, solar radiation pressure and atmospheric resistance into account and can be used for near-earth objects with orbit cycles less than 225 minutes. If the TLE track report is substituted into the SGP4 model, the space target with an orbital period of fewer than 225 minutes can be predicted successfully, and the position and velocity of the target object at any time can be solved [26]. Mininet [22] is a simulation tool for processing virtualization network developed by Stanford University based on Linux Container architecture and can create a virtual network containing hosts, switches, controllers, and links. Its switches support OpenFlow and have highly flexible customized software-defined networks. It provides a simple and inexpensive network testing platform for OpenFlow applications, topology-aware and OpenFlow-aware CLI for debugging or running network-wide tests, an extensible Python API for user network creation and experimentation, and enables complex topology tests without physical network connections.

3. Related work

3.1. Architecture

An excellent simulator consists of network, visualization, mobility, topology, routing bandwidth and latency, and most approaches focus on part of these features. Kassing *et al.* [15] developed a packet-level LEO simulator, Hypatia, with NS3 [25], it concentrated on the behavior of the LEO constellation networks. Hypatia builds an LEO constellation network with a few parameter configurations, and it splits the simulation into two modules, state generation and state execution. After execution of Hypatia, researchers acquire data on RTT, progress, ISL utilization and Congestion Window of TCP from the simulation. LEOCN can also obtain this information, but real-time and even more. Hypatia has no interaction with users while running, and its visualization is solitary from the constellation simulation. LEOCN is able to interact with users and shows the visualization of a constellation in real-time. Slowness is a significant defect of Hypatia. Its part of the simulation takes only one core of CPU and can not get the best of multicore CPU. 200 seconds of simulation time takes 20 minutes in optimized mode and it contains 100 ground stations and 1156 satellites. And the utilization of GSLs could not be tracked in Hypatia. Benefiting from techniques used by Mininet, LEOCN ensures that multicore CPU is used effectively and GSLs can also be traced.

Zhang *et al.* [44] use STK as aerospace physics engine, Mininet as the network engine and ONOS (Open Network Operating System) as SDN controller to build a three-layer simulation architecture (GEO/MEO/LEO) named Walker. It focuses on the dynamics of constellation topologies and routing algorithms with topology motion. Walker possesses the shortest routing algorithm and the minimum hop routing algorithm, but the latter is not suitable for its architecture due to existing MEO and GEO, it results in a long delay to use MEO/GEO as communication nodes. The structure of LEOCN only contains LEO and it does not exist this problem, so the minimum hop routing algorithm can get a better performance in LEOCN. In Walker, the operations of the adjacent orbit connection are so complex to take much of the time, the grid mode applied to LEOCN needs only to be configured at the beginning.

Kempton *et al.* [16] proposed a simulation framework referred to as SILLEO-SCNS for visualizing the LEO constellations through python3, and it mainly concentrates on the motion of LEO satellites in constellation and ground station nodes, and end-to-end connection dynamics. SILLEO-SCNS builds the LEO constellation networks by a graph that is generated by NetworkX [24]. It has no support for the link layer and above, and the computation of its latency uses only the sum of signal propagation times. Compared with SILLEO-SCNS, LEOCN possesses the support of the link layer and above, and the delay is more reasonable, which is benefited by Mininet.

Placement problems, physical layer, and congestion control in constellation architecture, part of researchers draw attention to them. SNS3 [28] designs a GEO simulation library focused on the physical layer. Chen *et al.* [5] use SDN to design an adaptive controller placement and assignment mode for LEO constellations. Dai *et al.* [6] research intelligent gateway placement to achieve the best network effectiveness in satellite-terrestrial integrated networks. Unlike the previous two papers, Vasisht *et al.* [37] research the distribution of ground stations to get a low latency LEO satellite network. Li *et al.* [20] use reinforcement learning to adjust congestion control to reach consistently high performances in satellite-ground integrated networks.

3.2. Routing algorithm

Conventional cooperative routing protocols show terrible performance because of the indifference to regular LEO constellation topologies and the long delay of ISLs. Based on traditional cooperative routing protocols, Tang *et al.* [32] propose a multi-path cooperative routing protocol that transfers a disparate portion of data through multiple link paths. It uses network coding technology which splits a data flow into multiple sub-flows in the middle nodes during the process of data transmissions, and then delivers them into disparate link paths. This protocol generates no redundant traffic and fits into the situation in that the LEO constellation network transfers enormous data.

Li *et al.* [18] improve security of routing algorithms with a distributed trust evaluation model in LEO satellite. Geng *et al.* [9] introduce unstable latencies into routing algorithms. Han *et al.* [12] proposed a routing algorithm with deep reinforcement learning for reliable anti-jamming routing. Zhang *et al.* [41] design a QoS preference routing algorithm to weigh bandwidth and latency.

3.3. Motion

As a vital issue in LEO satellite networks, mobility management is supposed to locate mobile nodes and to guarantee a stable data transmission upon the change in nodes position with IP update well known as an important procedure of it. Conventional location-independent network architecture for IPv6 (LIN6), requires mobile nodes to send binding update requests to the location directory once a handover occurs every time. Therefore, Tsunoda *et al.* [35] argues a handover-independent mobility management scheme for LEO satellite networks, purposes to reduce the number of IP update requests and eventually increases the system scalability in order to make the mobility management independent from handovers by exploiting geographical location information. Besides, the frequent IP handovers led to heavy mobility management load and large handover delays because of the high dynamic characteristics of the low-orbit satellite networks. To solve these problems, Zhang *et al.* [43] proposes a mobility management mechanism based on the virtual agent domain (VAD). Focused on improving call quality and balancing constellation network load, Wu *et al.* [40] propose a novel satellite handover strategy based on the potential game for mobile terminals in an LEO satellite communication network. To optimize related schemes in terms of response time and load balancing Chen *et al.* [5] propose the CRG-based controller placement and assignment (CCPA) algorithm with a bounded approximation ratio considering the mobility and load together.

4. Architecture of LEOCN

To make up for the question of real-time and interactivity, we built LEOCN, the structure of this LEO constellation network containing Node, Link, Constellation, Topology, Network and Monitor module and some submodules, as shown in Fig. 1.

4.1. Node

In LEOCN, each node has an isolated process and network interface environment, and they cannot communicate with each other, but through the network. Every node is considered a separate computer that is able to deploy services and experiment protocols. As nodes use the Linux kernel, they can utilize all the protocols supported by this kernel, and can also develop drivers to apply new network protocols to the LEO constellation simulator according to requirements. In addition, all nodes contain a property of CPU proportion, which is referred to as the different hash-rate of nodes specialized in various scenes, and the value is a float range from 0.0 to 1.0.

LEOCN possesses three node modules: ground station, satellite and mobility switch. Different scenes have their CPU proportion: ground stations are given a high hash rate, satellites are given a poor hash-rate, and mobility switches are without limitation of CPU proportion.

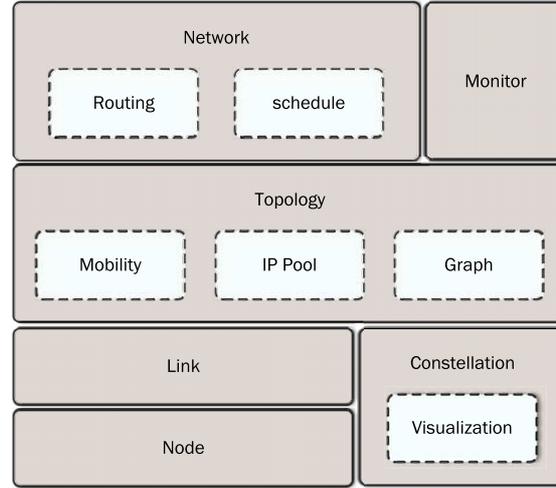


Fig. 1. Overall architecture of LEOCN.

Ground stations can be considered as both service nodes and user nodes, and the service nodes can provide SSH, Web and other services for the user nodes to use. Every ground station node has latitude, longitude, altitude, linked satellite and IP attributes. Latitude and longitude are primarily used for distance computation, and the latitude and longitude coordinates of different cities are readily retrieved from the Internet. These nodes will record service processes for releasing the host resources while ending the simulation. Unlike Hypatia [15], LEOCN abandons the viewpoint that ground stations can be regarded as routers to forward packets, which means all ground stations in LEOCN are the end node of the LEO constellation network and they only have no more than one connected satellite in the whole simulation.

In LEOCN, every satellite node has a mobility switch node, and they are one inseparable unit. Given the poor hash rate of satellite nodes, they deploy no service except for the routing function. An instance of satellite node consists of a two-line element set (TLE) and multiple IP in different network interfaces which are connecting to another satellite or ground station. Unlike ground station nodes, they do not contain any position properties, but functions that calculate position in real-time. Satellite nodes are directly connected, but ground station nodes connect to satellites through the mobility switch node of satellites. Both the mobility switch nodes and TLE, are used to calculate the real-time position and provide support for handover management.

Mobility switch nodes are responsible for connecting and disconnecting the satellite from ground stations. A mobility switch node can connect to any number of ground stations provided the connection conditions are satisfied.

4.2. Link

LEOCN has two link modes. Links between satellites are inter-satellite link (ISL), and directly they are connected to each other. Links between satellites and ground stations are ground-to-satellite links (GSL). Unlike ISL, all ground stations do not connect to satellite instead of their mobility switch node, as shown in Fig. 2. GSL simulates wireless communication with poor bandwidth, and ISL simulates laser communication with high bandwidth. However, the bandwidth of the link is not fixed, and it can be adjusted according to the requirements of researchers. In addition to bandwidth, there are delay, jitter and loss rate properties that can be controlled in the link module.

In the whole architecture, link modules are responsible for connecting the state between two nodes. It provides interfaces that modify its connecting state such as bandwidth, and record the instance of two link ends for the development of an add-on function. Each link instance has two independent network interfaces which control outgoing and ongoing connecting states, which supports future more complex architectural designs.

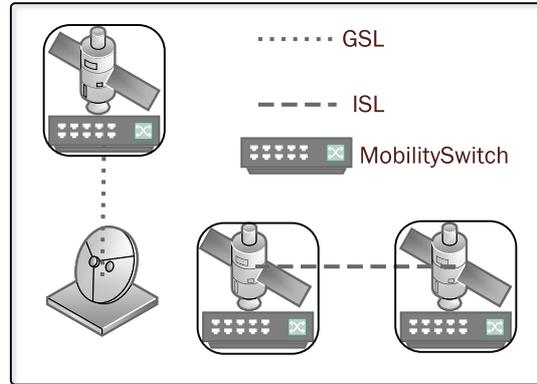


Fig. 2. Link modes in LEOCN.

4.3. Constellation

The constellation module is responsible for initialization, ISL and visualization, which makes it possible to generate constellations from existing TLE configuration files or specialized parameters such as the number of orbits, number of satellites in each orbit, inclination degree, eccentricity, perigee angle and altitude. The build process is not involved with the usage of any ground stations. This module can generate CZML data for Cesium Viewer [4] in real time.

The key points of this module are to support the TLE information of satellites and connect the state of ISL for the super architecture. The design philosophy of the LEOCN constellation module is that researchers can concentrate on building the ISL structure of LEO constellations without distraction from other factors such as the ground station.

4.4. Topology

As one of the most important modules, the topology module consists of a mobility module, IP pool module and graph module, as well as takes on the construction and updating of network topology. It is responsible for the entire life cycle of LEOCN.

The mobility module is responsible for the process of connecting ground station nodes to satellite nodes according to the distance between them. While the satellite nodes are connected by ground station nodes switching, the mobility module will inform the topology module of the altered state, and then the topology module updates IP pool and graph. In the mobility module, researchers can deploy their mobility algorithm to enhance the performance of the architecture of the constellation network.

4.5. Network

The network module is responsible for coordinating other modules to complete the overall scheduling. In this module, researchers can easily deploy their own algorithms such as routing algorithm and schedule algorithm in the form of SDN. This module has the privilege that access other module and control them, such as the graph of network topology from *Topology* module and the IP of ground station from *Node* module. It supports some APIs that manipulate the simulator for development of add-on function.

4.6. Monitor

The monitor module is responsible for the practice of overseeing the operation of the LEOCN network, which allows researchers to observe the state of ground stations, satellites and links. In LEOCN, the monitor module adopts sFlow-RT [27] that provides real-time network interface information such as protocol ratio and traffic size.

Table 1
Notations

Notations	Description
m	the number of satellite nodes
n	the number of ground station nodes
g	the ground station instance
f^{\max}	the maximum float
\mathcal{E}^{tle}	the satellite TLE set of constellation
\mathcal{E}^{sat}	the satellite set of constellation
\mathcal{E}^{gs}	the ground station set of constellation
\mathcal{E}	the all node set of constellation, where $\mathcal{E} = \{\mathcal{E}^{\text{sat}}, \mathcal{E}^{\text{gs}}\}$
\mathcal{V}^{gsl}	the GSL link set of constellation
$f(t, s^{\text{tle}})$	compute the coordinate of latitude and longitude of satellite at time t
$\text{dis}(t, e_1, e_2)$	compute the distance between e_1 and e_2 at time t

5. Implementation of LEOCN

To promote understanding of the following algorithm, the Table 1 and following definitions are provided.

Definition 1. s^{TLE} is expressed as a single satellite TLE, \mathcal{E}^{TLE} as the satellite TLE set of constellation, where $\mathcal{E}^{\text{TLE}} = \{s_1^{\text{TLE}}, s_2^{\text{TLE}}, s_3^{\text{TLE}}, \dots, s_m^{\text{TLE}}\}$, which m is defined as the total number of satellites in the constellation.

Definition 2. t is defined as timestamp, \mathcal{T} is defined as $\mathcal{T} = \{t_1, t_2, t_3, \dots\}$, where $\forall i \in 1, 2, 3, \dots, t_i < t_{i+1}$. To emphasize the characteristics of the real-time property in LEOCN, t_1 is defined as the timestamp of the starting simulation.

Definition 3. At each timestamp $t \in \mathcal{T}$, the constellation topology is defined as an undirected graph $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$, where \mathcal{V} is the set of ground station nodes and satellites nodes at the timestamp of t , and \mathcal{E}^t is the set of ISL link and GSL link at the timestamp of t .

5.1. Topology

At the process of building network topology, topology module obtains the information of satellites and their ISL from constellation module and ground station ones from configuration file, and then generates corresponding instances. These instance nodes are granted for the specialized IP ranges. Each ISL is allocated as a 30-bit subnet mask, which accommodates the two end nodes. The GSL of ground station nodes which connect to the same satellite node are considered as a same subnet, and its capacity depends on the range of subnet mask designed by the researchers.

Each construction and destruction of ISL and GSL require the IP pool module of the topology module to control. This module allocates IP to the instance node from the IP pool while construction, and releases IP from the instance node to the IP pool while destruction. Topology module utilities graph of networkx [24] to record all instance nodes.

As can be seen from Fig. 3, each satellite is able to connect any number of ground stations except for more than the capacity of the subnet, and has 4 ISL links, 2 links in the same orbit and 2 links connected to adjacent orbit. Throughout the whole process of constellation simulation, topology module need to record all IP of the network interfaces and all subnet for each network interfaces. These data support for the implementation of routing algorithm mentioned later in this section.

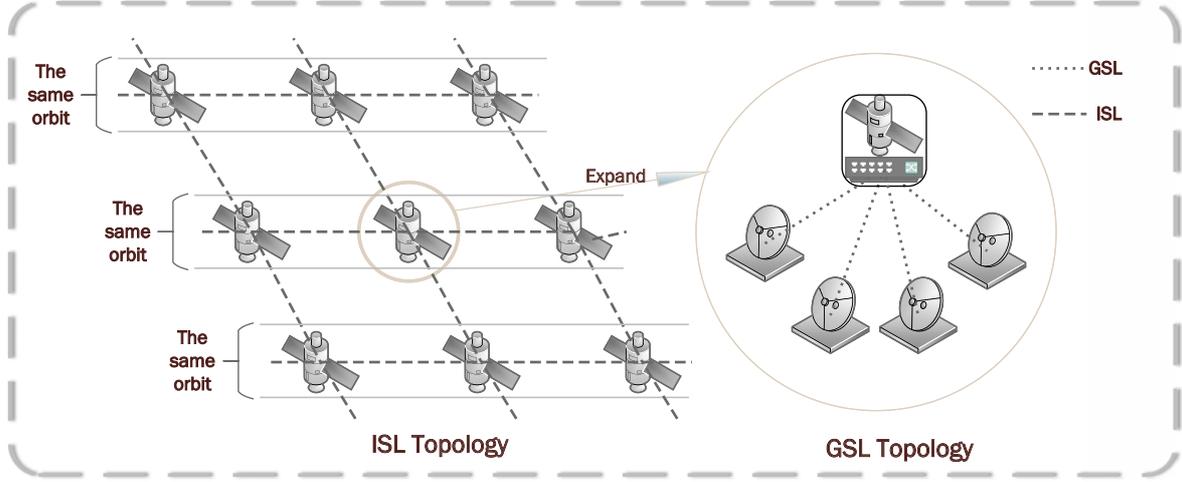


Fig. 3. Topology of LEOCN, including ISL topology and GSL topology. ISL topology describes the connection between satellites, GSL topology describes the connection between satellite and ground station.

Algorithm 1 Mobility algorithm

Input: $\mathcal{E}^{\text{tle}}, t$

```

1:  $t \leftarrow \text{now}()$  ▷ Obtain the current time
2: for all  $g \in \mathcal{E}^{\text{gs}}$  do
3:    $m \leftarrow f^{\text{max}}$ 
4:    $s^m \leftarrow 0$  ▷ The shortest distance satellite
5:   for all  $s \in \mathcal{E}^{\text{sat}}$  do
6:      $d \leftarrow \text{dis}(t, g, s)$ 
7:     if ( $d < m$ ) then
8:        $m \leftarrow d$  ▷ Record shorter distance
9:        $s^m \leftarrow s$ 
10:    end if
11:  end for
12:  if  $\text{check}(g, s^m)$  then ▷ Check connectivity
13:     $\text{link}(g, s^m)$  ▷ Build GSL link
14:  end if
15: end for

```

Output: $\mathcal{G} = (\mathcal{E}, \mathcal{V}^{\text{gsl}})$

5.2. Mobility

The LEOCN utilizes the time slice algorithm for the implementation of mobility, which updates topology states with unceasing regularity. The update algorithm seeks the shortest satellite for each ground station, so the distance between every ground station and all satellites must be computed, and the time complexity is $O(n^2)$, as shown in Algorithm 1. After this process, the new graph generated by the mobility process will be stored in the topology module.

The link process of mobility algorithm, as shown in Fig. 4, consists of multiple verification and link switch processes. First, check whether the link needs to be replaced. Connectivity validation is then required, regardless of whether changes are required. Lastly, disconnecting the current link if both connection links fail to verify

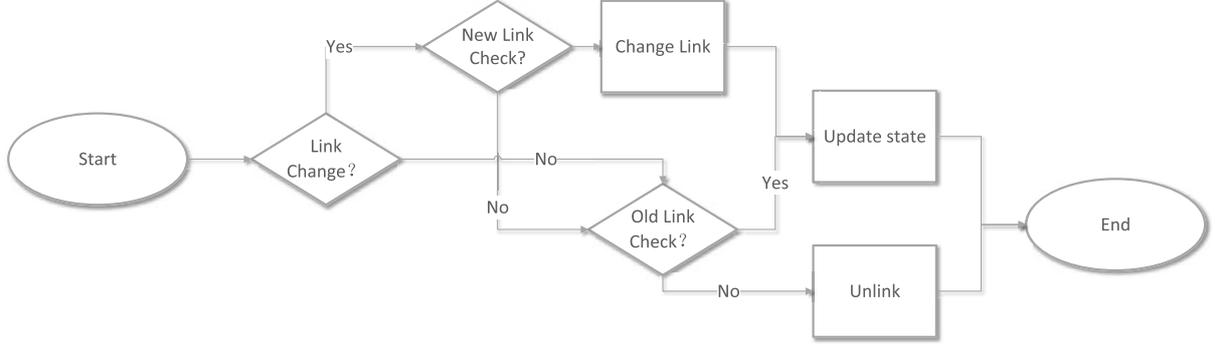


Fig. 4. Workflow of link updating.

Algorithm 2 The shortest hop routing algorithm**Input:** $\mathcal{G} = (\mathcal{E}, \mathcal{V})$

- 1: $\text{path} \leftarrow \text{Dijkstra}(\mathcal{G})$ ▷ Compute the shortest path of graph \mathcal{G} by Dijkstra method
- 2: $\mathcal{E}^{\text{per}} \leftarrow \text{permutations}(\mathcal{E}, 2)$ ▷ Compute all possible 2-length orderings with no repeated elements
- 3: **for all** $(e^{\text{src}}, e^{\text{dst}}) \in \mathcal{E}^{\text{per}}$ **do**
- 4: $\text{forwarding} \leftarrow \text{path}[e^{\text{src}}][e^{\text{dst}}]$ ▷ Obtain complete routing policy from e^{src} to e^{dst}
- 5: $\text{next_hop} \leftarrow \text{forwarding}[1]$ ▷ Obtain the next-hop of node e^{src}
- 6: $\text{addRoute}(e^{\text{src}}, e^{\text{dst}}, \text{next_hop})$ ▷ Add a static route to e^{src}
- 7: **end for**

in connectivity validation, otherwise update the current connection state that contains bandwidth, IP and delay computed according to distance.

5.3. Routing algorithm

LEOCN deploys the minimum hop routing algorithm with Dijkstra method, which can be well applied to the topology mentioned earlier in this section, as shown in Algorithm 2. Since initialization, the ISL topology has been fixed, and so has the subject of every satellite. Consequently, fixed are paths to any subnet of satellite from other satellites. In this case, the minimum hop routing algorithm only requires to update routing strategies once at the initialization. Compared to the perpetual updating strategies of the shortest path routing algorithm, it has the lower overhead.

The implementation of routing algorithm requires the information of next-hop IP, which comes from topology implementation. Instead of computing the next-hop of all subnet, the routing algorithm of the LEOCN only calculates that part of the subnet of the ground station, as shown in Fig. 5, ① and ② are the reachable subnets. ③, ④, ⑤, ⑥, ⑦, ⑧ and ⑨ are the unreachable ones that the simulation does not compute the routing strategies to. The number of subnet of constellation simulation is $3m$, of which $2m$ is ISL subnets and the other m is the subnet of satellites for ground stations. In LEOCN, all ground station nodes employ the connected satellite as the gateway, so the implementation of routing algorithm only depends on the routing strategy of satellite. And one routing updating process takes $m \times (m - 1)$ routing strategies, where each of the m satellites has $m - 1$ routing strategies. It is considerably smaller than the full subnet routing strategies.

5.4. Visualization

The visualization of LEOCN implements with Cesium [3], which can show the satellite trajectories, the topology information, the ISL link state and link traffic size. The generation of satellite trajectories depends on TLE, which

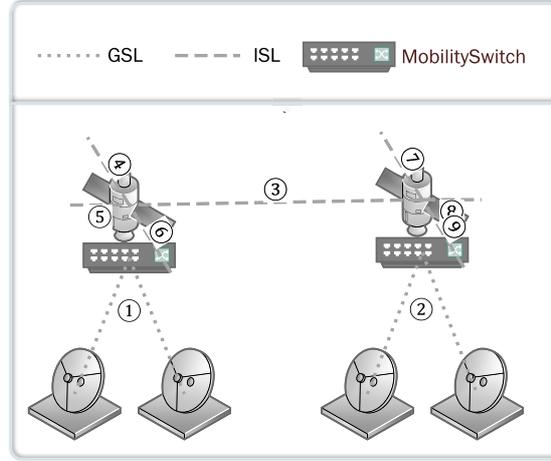


Fig. 5. Subnet mode of LEOCN, thin dotted lines indicate GSL link, thick dashed lines indicate ISL link.

LEOCN utilizes TLE to generate CZML data for Cesium in real-time. The topology information and ISL link state are retrieved from the topology module and visualized in the Cesium. The link traffic size is acquired from the monitor module and visualized on a separate page.

6. Evaluation

In this section, we present four experiments to show the status of constellations in different environments. Since LEOCN is the first real-time and complete framework for network simulation, we can not conduct comparative experiments with the frameworks of other papers. The environment of the four experiments:

- Ubuntu 20.04
- Python 3.8.10
- Kernel 5.13.0-51-generic
- Mininet 2.3.1b1
- Google Chrome 103.0.5060.114
- Memory 16G
- Intel Xeon Processor 64 cores
- 256 satellites
- 10 ground stations
- 1000 Mb/s ISL bandwidth
- 100 Mb/s GSL bandwidth
- 10 seconds time-slice
- iputils-ping 3:20190709-3
- procs 2:3.3.16-1ubuntu2.2

Fig. 6 shows the visualization of the environment of experiment. It can see the different orbits, ISL state, satellite position and ground station position.

6.1. Resource consumption

The consumption of system resources is one of the important indicators to evaluate the performance of LEOCN. Fig. 7 shows the state of system resources from the beginning of the simulation. The top [34] from *procs 2:3.3.16-1ubuntu2.2* is used to collect the system resource state every 100 ms in these experiments. There are three labels in

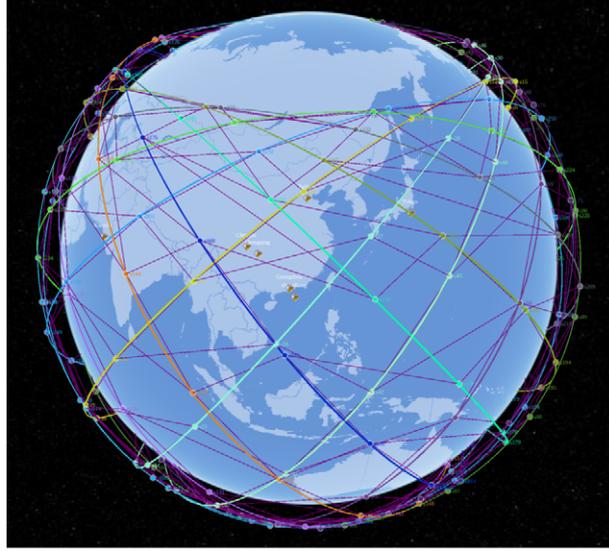


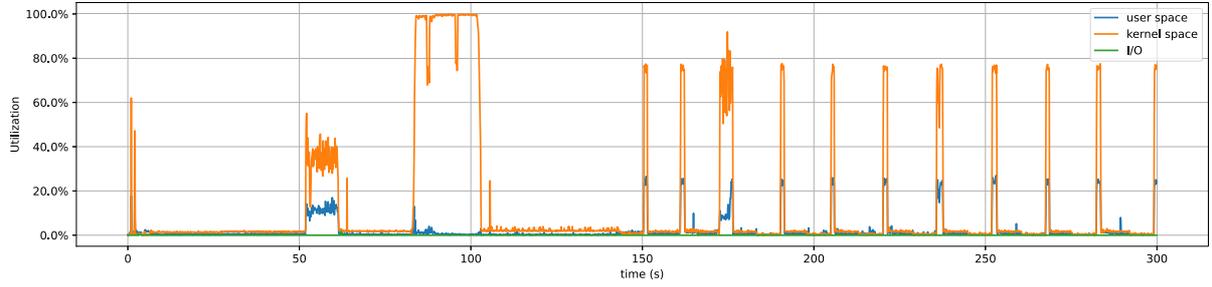
Fig. 6. Visualization of experiment environment, each thick line in a different color indicates a satellite orbit, and the same colored dots above indicate satellites in that orbit, and the thin purple lines indicate ISL link.

the first two figures Fig. 7a and Fig. 7b, *user space* as the time running user processes of the CPU state percentages, *kernel space* as the time running kernel processes of the CPU state percentages, and *I/O* as the time waiting for I/O completion of the CPU state percentages. Two kinds of experiments were tested here, *No-Load Simulation* and *General-Load Simulation*. There is no network communication in *No-Load Simulation*. *General-Load Simulation* will randomly pick some nodes and make them communicate with each other. The third figure Fig. 7c shows the state of physical memory usage for two kinds of experiments, and as can be seen in the figure, the Ubuntu system itself consumes about 1,500 MB of physical memory.

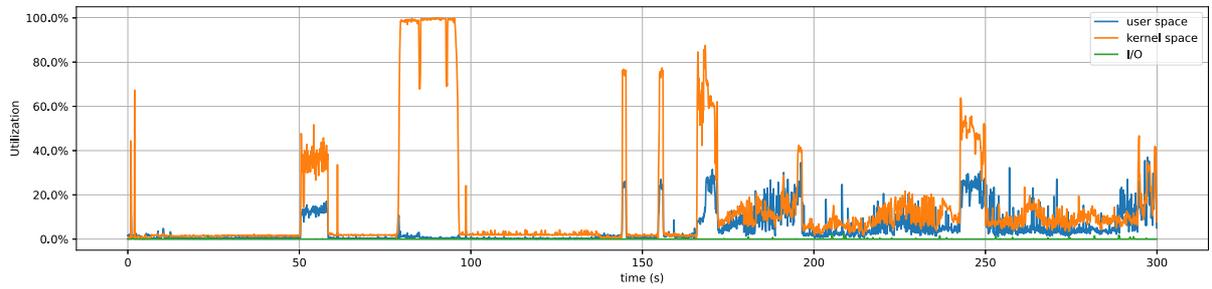
The time from 0 to 150 seconds is the initial process of LEOCN, and after this process, the simulation is up and running. As shown in Fig. 7, the first *kernel space* peak around 50 seconds is due to the initialization of the node network interface, and the second one around 100 seconds is due to the initialization of routing strategies. In *No-Load Simulation* experiment, there is a *kernel space* peak every 10 seconds after 150 seconds where the simulation started, and they are due to the updating of the mobility module. In the running period of *General-Load Simulation* experiment, *user space* and *kernel space* keep dynamically changing due to the random network behavior. The utilization of *I/O* had been low in the whole process. The LEOCN is not a memory-consuming system, and even with so many nodes, the entire system utilizes less than 5,000 MB of physical memory.

6.2. RTT and path-hop

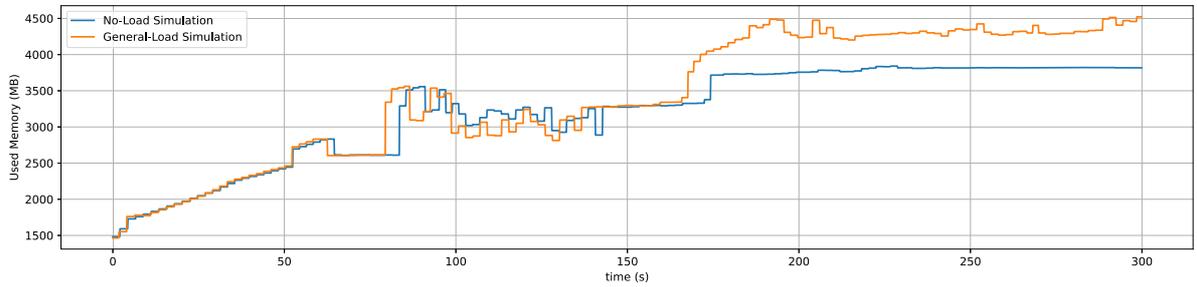
We examine how the end-end RTT and path hop differ over time, its result is shown at Fig. 8. The mobility of satellites has a great influence on the stability of the satellite network while the handover of satellites happening. The ping measurements from `iputils-ping` and the simulation utilities are the minimum hop routing algorithm. From Fig. 8, the RRT could have shot up and bounced back while the handover of satellites happening. For path-hop, the distance as far apart as Guangzhou and Chicago maintains between 8 and 11 hops, but even the shorter distance from Guangzhou to Beijing has the probability that path-hop jumps to a high value of around 11 and stays there for a while.



(a) No-Load Simulation Utilization



(b) General-Load Simulation Utilization



(c) Memory Usage

Fig. 7. Resource consumption statistical figure, Fig. 7a was experimented in the environment of no network communication, Fig. 7b was experimented in the environment of randomly network communication, Fig. 7c shows the memory usage of the preceding two environments.

6.3. Loss

Most previous works do not pay attention to the loss information, here is the experiment for loss, as shown in Fig. 9. This experiment utilizes iputils-ping to collect the information of the loss packet, of which timeout has been set to 1 second and it measures the information every 100 ms for one hour. There are five paths that have been experimented, *Beijing-Chongqing*, *Chicago-Paris*, *Lagos-Karachi*, *Chengdu-Tokyo* and *HongKong-Guangzhou*. As can be seen from the figure, *HongKong-Guangzhou* and *Beijing-Chongqing* have a better overall amount of loss packets than others, and *Chicago-Paris* have the worst overall amount of loss packets in the whole experiment.

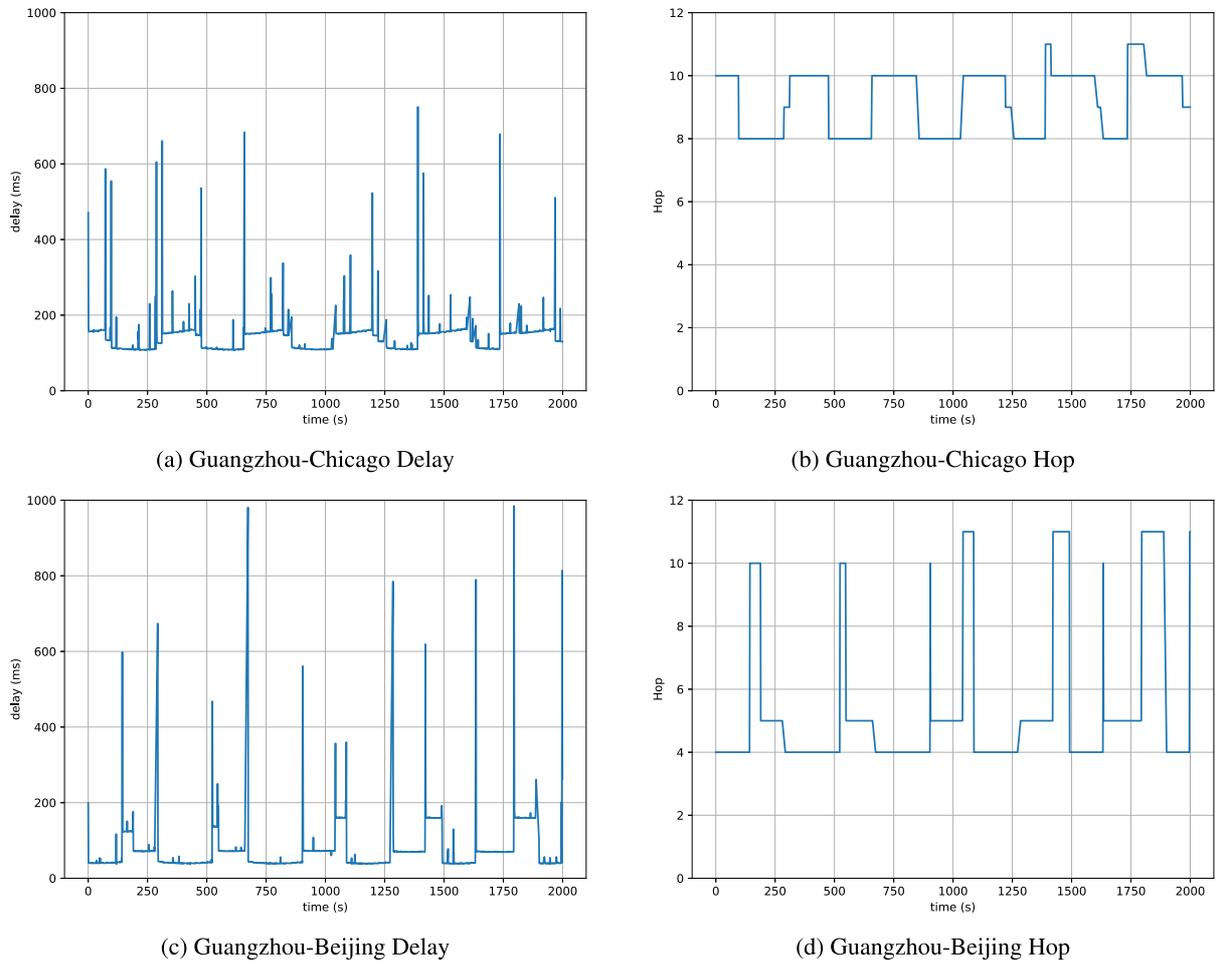


Fig. 8. RTT and path-hop evolution, the experimental data were collected every 100 ms.

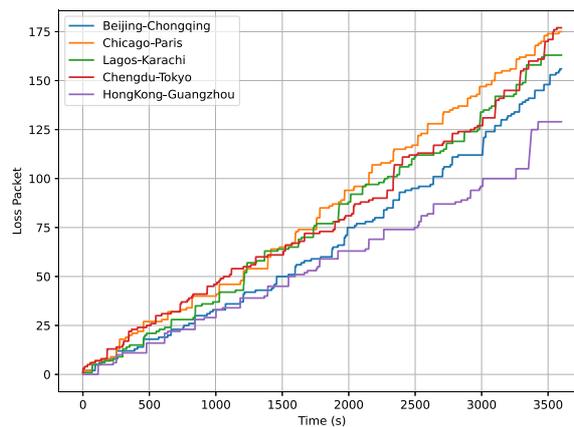


Fig. 9. Loss packet overall, this experiment utilizes iputils-ping to collect the information of the loss packet, its timeout has been set to 1 second and it measures the information every 100 ms for one hour.

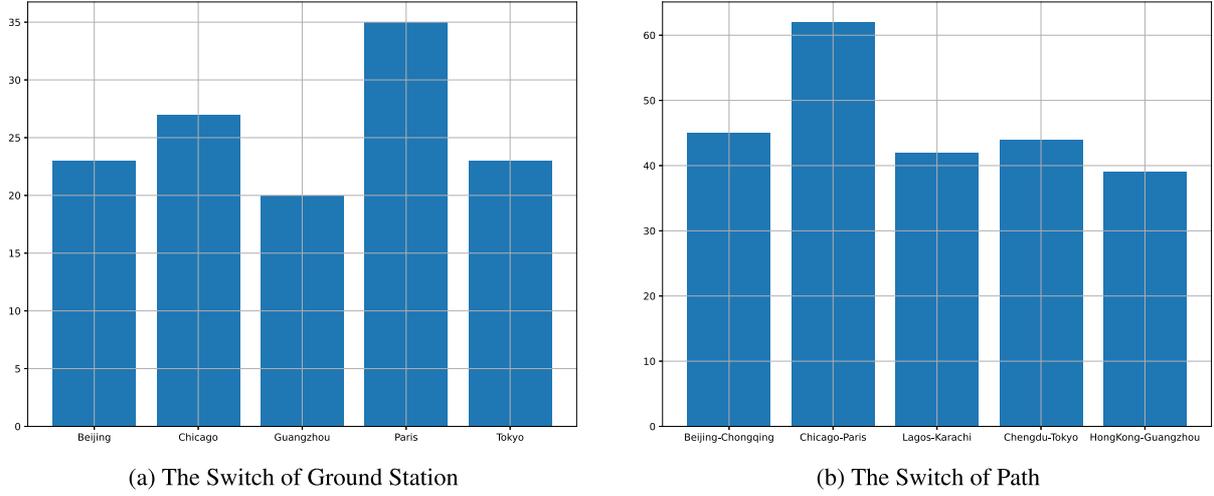


Fig. 10. Mobility experiment: in this experiment, the simulation stays running continuously for one hour and records the switch times Fig. 10a of satellite connected by ground station. The values of Fig. 10b are the sum of the switch times of the two target ground stations.

Table 2
Part of ground station

	Beijing	Chicago	Guangzhou	Paris	Tokyo
Latitude	39.9075	41.85003	23.8	48.85341	35.6895
Longitude	116.39723	-87.65005	113.17	2.3488	139.69171

6.4. Mobility

We also explore the effects of the position of a ground station on mobility. In this experiment, the simulation stays running continuously for one hour and records the switch times of the satellite connected by the ground station. The result is shown at Fig. 10. In the *The Switch of Ground Station* of the figure, it displays the switch time of five ground stations, and *Paris* has the largest switch time while *Guangzhou* has the minimum one. These just correspond to the latitude of the ground station, as shown in Table 2: the higher latitude is, the more times the ground station needs to switch. At the same time, it has a huge influence on the end-to-end path, as shown in *The Switch of Path* of Fig. 10.

7. Discussion

On this platform, we achieve the following two experiments. The first experiment is about a network attack on our platform that is involved with five attacks including a DDoS attack, FTP burst, PortScan, SSH burst, and web scan attack. We use a Machine Learning (ML) algorithm to recognize the sort of network attack in LEO link flow. To prevent our model from identifying potential attacks by mistakes in the future, we conduct another experiment, and we create the adversarial example, meaning that this sort of flow will happen in a few days to enhance the robustness of our model. For generating better images, we transfer our flow bytes into images and train our model by those images. These two experiments are detailedly written in other papers. And we evaluate our platform with the two experiments above conducted, and it shows good link utilization and low network delay.

Applying NWDAF(Network Data Analytics Function) to leo installation will have been becoming a trend taking the special structure of 5G into account. By using edge computing, we can process network edge cache and unload calculation when facing insufficient resources, and keep user data security and privacy. Shaoqing Wang et al.

[30] proposed an AI-driven F-RAN (fog radio access networks) paradigm combining the edge computing with ML. Their thoughts include Artificial Intelligence (AI) capsules alleviating the transmission burden incurred by network data collection through utilizing computation and cache resources at various fog nodes, and AI brain that considers completely minimum energy consumption for offline AI model training and evaluation. Our platform can conduct some AI analysis toward both resources computation offloading and emergent data processing combining their ideas. It will be combined with the method proposed by Tianqing Zhu et al. [33] in the future. Over and above, we can avoid the overhead incurred by massive training data collection by federated learning and solve cost-inefficiency of re-designing AI models by transferring learning in the future.

Considering the lack of trust among different constellations, communication security and privacy protection are important issues that need to be solved during inter-constellation cooperation. Blockchain has the advantages of privacy security, non-tamperability of data and decentralization and it will properly satisfy the demand for LEO satellite constellation. Therefore, Shaoqing Wang et al. [36] presents a trusted and privacy-preserved authentication scheme for inter-constellation collaboration. They design the permanent identity used for inter-constellation communication, and the temporary identity used for inter-constellation cooperation and privacy leakage for each satellite respectively. When satellites go commercial, these methods else proposed by Jiang Nan et al. [14] and Li Jin et al. [19] will also be used. Besides, a consortium blockchain for sharing information is introduced, while a replica storage node method for efficient authentication under limited resources is proposed. Based on their research, our platform will conduct further experiments to design intelligent contracts customized for LEO installation.

We have achieved the function of judging whether our network suffers from an attack and recognizing classification towards network attack and which terminal launches this attack, and we make defense based on that. To prevent some attacks that can happen in the future and enhance the robustness of our model, we make adversarial samples by network flow on the LEOCN platform. The following is edge computing [30] based Deep Learning to enhance the function of data analysis and reduce traffic congestion which contributes to our following experiments. Finally, we need to achieve the blockchain [36] to keep the encryption and security of communication among LEO satellites and ground terminals.

8. Conclusion

We presented LEOCN, a new platform for network simulation that includes topology and routing algorithm, visualization, satellite mobility, communication protocol and network traffic. LEOCN exhibits dynamics of the constellation with special celestial movement in our implementation. It offers a real-time and complete network simulation platform on which we have conducted some experiments about network defense and adversarial examples offering topology and routing algorithm in advance for LEO satellite constellations. With this platform, you can conduct some LEO experiments about blockchain, edge computing, deep learning and so on, without the trouble of special celestial movement, routing algorithms, etc., especially for low orbit satellites.

Acknowledgements

Support by the Key Research and Development Program of Guangzhou (No.202103050003).

Conflict of interest

None to report.

References

- [1] D. Bhattacharjee, W. Aqeel, I.N. Bozkurt, A. Aguirre and A. Singla, Gearing up for the 21st century space race, in: *The 17th ACM Workshop*, 2018.
- [2] D. Bhattacharjee and A. Singla, Network topology design at 27,000 km/hour, in: *Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies, CoNEXT'19*, Association for Computing Machinery, New York, NY, USA, 2019, pp. 341–354. ISBN 9781450369985. doi:10.1145/3359989.3365407.
- [3] Cesium, <https://cesium.com/>.
- [4] Cesium Viewer, <https://cesium.com/cesiumjs/cesium-viewer/>.
- [5] L. Chen, F. Tang and X. Li, Mobility- and load-adaptive controller placement and assignment in LEO satellite networks, in: *IEEE INFOCOM 2021 – IEEE Conference on Computer Communications*, 2021, pp. 1–10. doi:10.1109/INFOCOM42981.2021.9488806.
- [6] C.-Q. Dai, Q. Yang, J. Wu and Q. Chen, Intelligent gateway placement in satellite-terrestrial integrated network, in: *IEEE INFOCOM 2021 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6. doi:10.1109/INFOCOMWKSHPS51825.2021.9484491.
- [7] T. Darwish, G.K. Kurt, H. Yanikomeroğlu, M. Bellemare and G. Lamontagne, LEO satellites in 5G and beyond networks: A review from a standardization perspective, 2021, [arXiv:e-prints](https://arxiv.org/abs/2108.00000).
- [8] R. Deng, B. Di and L. Song, Dialogue between satellite and cellular networks: Pricing game for data offloading assisted by ultra-dense LEO, *Constellations* (2019).
- [9] S. Geng, S. Liu, Z. Fang and S. Gao, An optimal delay routing algorithm considering delay variation in the LEO satellite communication network, *Computer Networks* **173** (2020), 107166, <https://www.sciencedirect.com/science/article/pii/S1389128619313040>. doi:10.1016/j.comnet.2020.107166.
- [10] G. Giuliari, T. Ciussani, A. Perrig and A. Singla, ICARUS: Attacking low Earth orbit satellite networks, in: *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, USENIX Association, 2021, pp. 317–331, <https://www.usenix.org/conference/atc21/presentation/giuliari>. ISBN 978-1-939133-23-6.
- [11] A. Guidotti, B. Cioni, C. Colavolpe, A. Conti, C. Foggi, B. Mengali, D. Montorsi, C. Piemontese and A. Vanelli-Coralli, Architectures, standardisation, and procedures for 5G Satellite Communications: A survey, *Computer Networks* **183** (2020).
- [12] C. Han, A. Liu, L. Huo, H. Wang and X. Liang, Anti-jamming routing for Internet of satellites: A reinforcement learning approach, in: *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2877–2881. doi:10.1109/ICASSP40776.2020.9052911.
- [13] Iridium Communications Inc, <https://www.iridium.com/network/>.
- [14] N. Jiang, W. Jie, J. Li, X. Liu and D. Jin, GATrust: A multi-aspect graph attention network model for trust assessment in OSNs, *IEEE Transactions on Knowledge and Data Engineering* (2022).
- [15] S. Kassing, D. Bhattacharjee, A.B. Águas, J.E. Saethre and A. Singla, Exploring the “Internet from space” with hypatia, in: *Proceedings of the ACM Internet Measurement Conference, IMC'20*, Association for Computing Machinery, New York, NY, USA, 2020, pp. 214–229. ISBN 9781450381383. doi:10.1145/3419394.3423635.
- [16] B. Kempton and A. Riedl, Network simulator for large low Earth orbit satellite networks, in: *ICC 2021 – IEEE International Conference on Communications*, 2021, pp. 1–6. doi:10.1109/ICC42927.2021.9500439.
- [17] Z. Lai, H. Li, Q. Zhang, Q. Wu and J. Wu, Cooperatively constructing cost-effective content distribution networks upon emerging low Earth orbit satellites and clouds, in: *2021 IEEE 29th International Conference on Network Protocols (ICNP)*, 2021, pp. 1–12. doi:10.1109/ICNP52444.2021.9651950.
- [18] H. Li, D. Shi, W. Wang, D. Liao, T.R. Gadekallu and K. Yu, Secure routing for LEO satellite network survivability, *Computer Networks* **211** (2022), 109011, <https://www.sciencedirect.com/science/article/pii/S1389128622001712>. doi:10.1016/j.comnet.2022.109011.
- [19] J. Li, X. Hu, P. Xiong, W. Zhou et al., The dynamic privacy-preserving mechanisms for online dynamic social networks, *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [20] X. Li, F. Tang, J. Liu, L.T. Yang, L. Fu and L. Chen, AUTO: Adaptive congestion control based on multi-objective reinforcement learning for the satellite-ground integrated network, in: *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, USENIX Association, 2021, pp. 611–624, <https://www.usenix.org/conference/atc21/presentation/li-xu>. ISBN 978-1-939133-23-6.
- [21] Z. Luo, T. Pan, E. Song, H. Wang, W. Xue, T. Huang and Y. Liu, A refined Dijkstra’s algorithm with stable route generation for topology-varying satellite networks, in: *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 2021, pp. 1146–1147. doi:10.1109/ICDCS51616.2021.00129.
- [22] Mininet, <http://mininet.org/>.
- [23] MStarlink, <https://www.starlink.com/>.
- [24] NetworkX, <https://networkx.org/>.
- [25] NS3, <https://www.nsnam.org/>.
- [26] J.F. San-Juan, I. Pérez, M. San-Martín and E.P. Vergara, Hybrid SGP4 orbit propagator, *Acta Astronautica* **137** (2017), 254–260, <https://www.sciencedirect.com/science/article/pii/S0094576516311535>. doi:10.1016/j.actaastro.2017.04.015.
- [27] sFlow-RT, <https://sflow-rt.com/>.
- [28] SNS3, <http://sns3.org/>.

- [29] G. Stock, J.A. Fraire, T. Momke, H. Hermanns and E. Cruz, Managing fleets of LEO satellites: Non-linear, optimal, efficient, scalable, usable, robust, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **39**(11) (2020), 1–1. doi:[10.1109/TCAD.2020.3028016](https://doi.org/10.1109/TCAD.2020.3028016).
- [30] Y. Sun, M. Peng, Y. Ren, L. Chen, L. Yu and S. Suo, Harmonizing artificial intelligence with radio access networks: Advances, case study, and open issues, *IEEE Network* **35**(4) (2021), 144–151. doi:[10.1109/MNET.011.2000656](https://doi.org/10.1109/MNET.011.2000656).
- [31] Systems Tool Kit, <https://www.agi.com/products/stk/>.
- [32] F. Tang, H. Zhang and L.T. Yang, Multipath cooperative routing with efficient acknowledgement for LEO satellite networks, *IEEE Transactions on Mobile Computing* **18**(1) (2019), 179–192. doi:[10.1109/TMC.2018.2831679](https://doi.org/10.1109/TMC.2018.2831679).
- [33] Z. Tianqing, W. Zhou, D. Ye, Z. Cheng and J. Li, Resource allocation in IoT edge computing via concurrent federated reinforcement learning, *IEEE Internet of Things Journal* **9**(2) (2021), 1414–1426. doi:[10.1109/JIOT.2021.3086910](https://doi.org/10.1109/JIOT.2021.3086910).
- [34] Top, <https://man7.org/linux/man-pages/man1/top.1.html>.
- [35] H. Tsunoda, K. Ohta, N. Kato and Y. Nemoto, Supporting IP/LEO satellite networks by handover-independent IP mobility management, *IEEE Journal on Selected Areas in Communications* **22**(2) (2004), 300–307. doi:[10.1109/JSAC.2003.819977](https://doi.org/10.1109/JSAC.2003.819977).
- [36] A. Tx, Z. Ran, L. Jiang, H. Tao, A. Yi and C. Fry, A blockchain-based and privacy-preserved authentication scheme for inter-constellation collaboration in Space-Ground Integrated Networks.
- [37] D. Vasisht, J. Shenoy and R. Chandra, L2D2: Low latency distributed downlink for LEO satellites, in: *Proceedings of the 2021 ACM SIGCOMM 2021 Conference, SIGCOMM'21*, Association for Computing Machinery, New York, NY, USA, 2021, pp. 151–164. ISBN 9781450383837. doi:[10.1145/3452296.3472932](https://doi.org/10.1145/3452296.3472932).
- [38] S. Wang, Y. Zhao and H. Xie, Improving survivability of LEO satellite network with guaranteed based approach, in: *Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies, CoNEXT'19 Companion*, Association for Computing Machinery, New York, NY, USA, 2019, pp. 47–48. ISBN 9781450370066. doi:[10.1145/3360468.3368173](https://doi.org/10.1145/3360468.3368173).
- [39] S. Wang, Y. Zhao and H. Xie, Improving survivability of LEO satellite network with guaranteed based approaches, in: *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020.
- [40] Y. Wu, G. Hu, F. Jin and J. Zu, A satellite handover strategy based on the potential game in LEO satellite networks, *IEEE Access* **7** (2019), 133641–133652. doi:[10.1109/ACCESS.2019.2941217](https://doi.org/10.1109/ACCESS.2019.2941217).
- [41] T. Zhang, H. Li, S. Zhang, J. Li and H. Shen, STAG-based QoS support routing strategy for multiple missions over the satellite networks, *IEEE Transactions on Communications* **67**(10) (2019), 6912–6924. doi:[10.1109/TCOMM.2019.2929757](https://doi.org/10.1109/TCOMM.2019.2929757).
- [42] T. Zhang, L. Yang, T. Dong, J. Yin, Z. Liu and Z. Wang, A multi-attribute decision handover strategy for giant LEO mobile satellite networks, in: *Smart Computing and Communication*, M. Qiu, K. Gai and H. Qiu, eds, Springer International Publishing, Cham, 2022, pp. 64–73. ISBN 978-3-030-97774-0. doi:[10.1007/978-3-030-97774-0_6](https://doi.org/10.1007/978-3-030-97774-0_6).
- [43] X. Zhang, K. Shi, S. Zhang, D. Li and R. Xia, Virtual agent clustering based mobility management over the satellite networks, *IEEE Access* **7** (2019), 89544–89555. doi:[10.1109/ACCESS.2019.2926432](https://doi.org/10.1109/ACCESS.2019.2926432).
- [44] Y. Zhang, B. Wang, B. Guo, Y. Yuan, T. Dong, J. Yin, K. Li, X. Guo and S. Huang, A research on integrated space-ground information network simulation platform based on SDN, *Computer Networks* **188** (2021), 107821, <https://www.sciencedirect.com/science/article/pii/S1389128621000104>. doi:[10.1016/j.comnet.2021.107821](https://doi.org/10.1016/j.comnet.2021.107821).