A survey of LEO simulators and routing topologies How does the simulator/emulator benefit/limit the study of LEO

Shun Zhang, Yihe Bi, Chengming Li *Department of Electrical and Computer Engineering*
University of California San Diego
San Diego
Emails: shz094,y4bi, chl248@ucsd.edu

I. Motivation of project

With the realization of the Low Earth Orbit (LEO) Satellite Network, massive satellites are used to provide reliable low latency, high capacity, and high throughput. And several space companies, like SpaceX[4], are actively developing and deploying satellites into this orbit. However, the achievement of massive LEO satellites raises new challenges for research and scientists. How can the researchers study and improve the LEO Satellite Network's performance without touching them? How can experiments and tests be conducted without interacting with the LEO satellite network? And the answer is LEO Satellite Network, an experimentation framework that can simulate/emulate the behavior of satellites, ground stations, routing, network topology, etc.

Moreover, from the perspective of satellites, the consistent connection of satellites presents challenges due to their high mobility[3]. The dynamic movement of satellites can introduce instabilities across different network layers, making it difficult to maintain a seamless connection over time. There are more challenges that look like these. Those challenges make the development of the LEO satellite network experimentation framework far harder than we think.

II. Related work

Our research is based on the paper titled "STARRYNET: Empowering Researchers to Evaluate Futuristic Integrated Space and Terrestrial Networks"[9] from NSDI 2023 spring. This paper primarily delves into StarryNet, a simulation software designed for integrated space and terrestrial networks (ISTN). Within this paper, the authors conducted a comparative analysis of various tools, including live LSNs, simulators, and emulators. They identified several shortcomings in these tools, such as inconsistent constellations, unrealistic system and networking stack representations, limited flexibility, and relatively high costs. StarryNet, on the other hand, addresses many of these issues by leveraging a combination of real-data traces, model-based simulation, and large-scale emulation. The paper presents extensive performance testing of the StarryNet system, demonstrating its capability to meet multiple experimental requirements for ISTN. However, StarryNet is not without its limitations. It relies on parameters obtained from public networks, which may sometimes lack credibility, necessitating manual parameter adjustments. Additionally, it falls short in fully simulating the network's physical (PHY) layer.

Our case study will be divided into routing, topology, and existing simulators and emulators. Each case study will be analyzed based on the simulator/emulator they use. And how these simulators/emulators benefit/limit the study topic.

The routing case study will be divided into three routing topology-specific and two algorithm-specific case studies. They are listed as follows, "Delay is not an option: Low latency routing in space"[4], "Internet backbones in space"[3], "Using ground relays for low-latency wide-area routing in mega-constellations"[5], and "A novel DTN routing algorithm in the GEO-relaying satellite network"[14]. The pros and cons of their simulator/emulator will be discussed and analyzed.

The topology case study will look into two former publication and the emulation method and software they use. They are listed as follows, *"Network topology design at 27,000 km/hour"*[1] and *"'Internet from Space' without Inter-Satellite Links"*[15]. The pros and cons of their simulation and emulation method will be discussed and analyzed along with the possible impact from simulation method to the result of their research.

We will also discuss modern satellite tools including live platforms, simulators, and emulators. For live platforms, we will discuss Live Starlink, PlanetLab, and Emulab. For simulators, we will look into Hypatia

"Exploring the 'Internet from Space' with Hypatia"[7] and StarPerf *"StarPerf: Characterizing Network Performance for Emerging Mega-Constellations."*[9]. For emulator, we will discuss MiniNet *"Reproducible network experiments using container-based emulation."*[13]. We will compare different type of tools and review the pros and cons of each type and to summarize the conditions for an overall better tool.

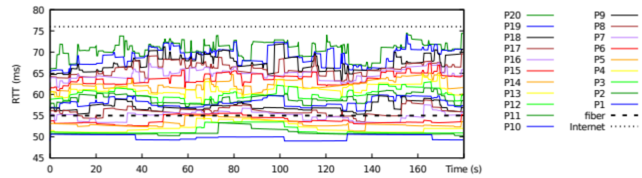

Fig. 1. MultiPath from "Delay is not an option: Low latency routing in space"

III. Project introduction

Our survey report aims to investigate the impact of several different simulators/emulators on the study of LEO Satellite Networks. First, We will perform several case studies about the performance of simulators/emulators that are used in the topology, routing and simulator-specific papers. The pros and cons of each simulator will be identified and evaluated. Some technical insights will also be provided along the way. i.e., whether having a functionality will benefit or limit the study of LEO satellite networks. This analysis help us to determine which functions/matrices are important for a general modern LEO emulator. Then, we will delve into a range of simulators/emulators that exist at the current time. The advantages and limitations of those simulators/emulators will be studied and analyzed. In addition, we will further evaluate one of the existing emulators we have reproduced. The evaluation will be based on scalability, realism, accessibility, and resource management perspectives. Last, the technical insight, challenges, and future research direction will be addressed.

IV. Routing Case Study

"Delay is not an option: Low latency routing in space"[4] has studied the latency properties of the LEO satellite network, which has phased array antennas for up and downlinks and laser communication between satellites. Their paper aims to provide early insight into the interactions between the dynamic topology of the constellation, how routing might work over such a novel network architecture, and emergent end-to-end latency properties[4]. Last they postulated a hybrid routing solution: high priority low-latency traffic always gets priority, and admission control limits its volume, preventing it from causing congestion and getting explicit routing ensuring minimum latency. For the remaining traffic, satellites monitor link load, and broadcast to all ground stations globally, so everyone is aware of hotspots[4]. Based on their simulation, they were able to prove the feasibility of the solution. Since simulation is mentioned multiple times in this paper, further study and analysis of their simulator will be conducted. However, the open-source code is not provided in the paper. Thus, the analysis will only be based on the plots and simulation scenarios provided and described in the paper. From the perspective of advantages, the simulator uses realistic constellations from Space X, which is the main trend in lots of other simulator settings. With such realistic constellations, the overall behavior of LEO satellites is tied closely to what happens in space. Second, it demonstrates its good scalability since all simulations in the paper are conducted based on the SpaceX phase 1 deployment, which consists of 1600 satellites. Third, it is very good at showing dynamic connectivity across a very long time frame. In Figure 1, this simulator shows the details of RTT variation of 20 paths up to 200s.

With a longer time frame, researchers can better study and understand the interactions between satellites and how routing topology really affects the performance of the LEO satellite network. From the perspective of disadvantages, as the authors mentioned in their paper, the simulator is built based on public details from the FCC filings[1]. Since the private details are hidden from the public, the hidden factors reduce the realism of the actual performance of the routing solution postulated by the authors. So, the final result didn't take into account the effects of the private details of the satellite. In addition, as the authors mentioned, they adopt satellite parameters from first principles[4]. The loss of realistic parameters will deviate the feasibility of the final solution from the correct path. The deviation makes the real routing design harder to achieve due to the adoption of ideal parameters and hidden factors.

"Internet backbones in space"[3] has studied the cost-performance tradeoffs in the design space for Internet routing that incorporates satellite connectivity, examining four solutions ranging from naively using BGP to an ideal, clean-state design[3]. Then afterward, the authors conclude that a path-aware networking architecture provides an optimal solution in which end-hosts obtain information and control over network paths[3]. There are four routing topologies have been simulated via their custom simulator. Thus, the analysis of their

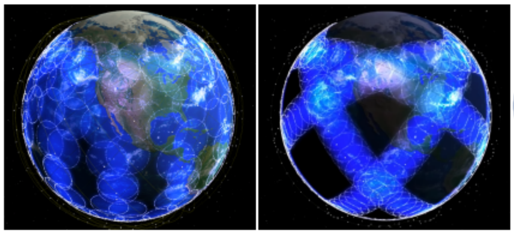

Fig. 2. Customization of Orbit planning ”Using ground relays for low-latency wide area”

simulator will also be conducted based on the plots and simulation scenarios provided and described in the paper. This custom simulator only involves a partial deployment of Starlink. The simulator simulates satellite orbits and resulting connectivity and latency. In addition, the historical rain data is imported into the simulator, which makes the simulation of the satellite network more realistic[3]. Taking into account the environmental effects can help researchers better understand the edge case when the non-ideal event happens. Also, it helps researchers identify and sort out the causes of bad things happening in the LEO satellite networks so that the researchers wouldn't identify the failure of normal communications between satellites as the failure of topology design. Second, as the authors mentioned, they have simulated four different topologies to show the advantages and disadvantages of each design. Such flexibility is what researchers really need in their research study. And having an integrated simulator can accelerate the development period, i.e., they don't have to switch platforms between different topologies. From the perspective of disadvantages, this simulator only models 10 percent of Space X phase two constellation[3]. The limited scale of simulation can't provide any detailed overview of the entire satellite network performance. Also, the portion of the simulated network is unknown in the paper. The latency performance can vary differently from high-population-density cities to low-population-density cities. i.e., the network traffic load can vary because of the population density. In addition, the latency performance also depends on the terrestrial part of the network. There are more ground stations in the urban area than in rural areas. The latency performance would be misunderstandd if the simulated satellites are in a rural area. Thus, the average performance of the proposed routing solution may deviate when different deployments are considered.

”Using ground relays for low-latency wide-area routing in mega-constellations”[5] has investigated the use of ground-based relays as a substitute for inter-satellite link(ISLs) due to the lack of ISLs in the early deployment stage of Starlinks. They have examined how such ground-relay routing improves the communication latency between satellites. Afterward, a hybrid solution, ground relay and ISLs, is proposed to show the LEO network performance can be improved further[5]. From the simulator side, the source of the simulator is not identified. Thus, the analysis of their simulator will be conducted based on the plots and simulation scenarios provided and described in the paper again. From the advantages side, this simulator has the freedom of customization of Orbit planning, Fig 2. The left figure shows 264 satellites comprising 11 satellites in each of the 24 orbital planes[5]. And the right figure shows 6 orbital planes with 66 satellites per plane[5]. This setup is exactly the same as the SpaceX FCC filing. In addition, the authors also simulated the SpaceX Starlink network in a range of possible configurations[5]. Since Starlink is the trending LEO satellite network in the LEO development, high freedom of customization with Starlink-specific can help the researchers refine the existing routing topology used in Starlink. So, the design trade-off can be better evaluated and improved when different configurations are applied. Second, this simulator maintains high dynamic connectivity for up to 4 hours, based on the plot from the paper. Compared to other existing simulators/emulators, some of them only extract one time instant of satellite network performance as a result, which doesn't fulfill the requirement of dynamic connectivity and high mobility of satellite. As seen in Fig 3, the simulator shows its dynamic connectivity for up to different time durations(from top to bottom, 4 hours, 4 hours, 5 minutes, and 5 minutes). A good dynamic

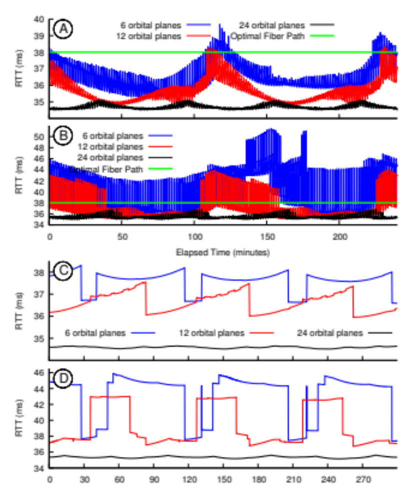

Fig. 3. Elapsed Time of Simulator ”Using ground relays for low-latency wide area”

connectivity gives the researcher a full overview of the long-lasting performance of the entire LEO satellite network. In the real communication system, bad things sometimes happen at the last mile/minute. So, the long-lasting result would provide an opportunity to find and study those edge cases happening in satellite communication. Beyond that, from the disadvantages perspective, this simulator has the highest computational intensity as we discussed so far. The high computational intensity is due to the large number of satellites, ISLs, and ground stations. A large number of nodes also brings challenges to the precision of the routing algorithm and path change. As stated in [5],"If we see that between the next two routing updates, the path to a destination has changed, we know that it should change sometime between 5 = 500ms and t = 1000ms, but Djkstra's algorithm does not tell us precisely when the change should occur" There are 500ms unknown during the communication. This number could be meaningful when the study goes down to the PHY layer. And it also blurs researchers for 500ms mystery if they try to improve the PHY layer performance. Lastly, a high computational intensity also requires better and more hardware machines in the simulations. Such costly requirements limit access to individual research or low-funding research groups. Moreover, higher computational intensity also means longer simulation duration. This makes the research inefficient if bad things happen in the middle way of simulation.

"A novel DTN routing algorithm in the GEO-relaying satellite network" has proposed a new routing algorithm that could find the best route by calculating each snapshot of time-varying topology[14]. The simulator they used is Linux-based, with more realistic data flow in a disruption-tolerant network-based satellite networ scene. The network model is built with Satellite To kits(STK)[14]. From the perspective of advantages, au thors import more realistic data into their models and ru their simulator for up to 1 day. The simulator maintai consistency and realistic data along this time fram So, the performance of the algorithm can be bett analyzed from multiple perspectives and a longer tim frame compared to a one-time snapshot of performanc With consistent connectivity in a long time frame, th interaction of the LEO satellite network can be full evaluated and analyzed. From the perspective of disa vantages, the STK is not well known for its system an networking stack realism. i.e., the network aspects are not considered. This could lead to underestimated results compared to realistic network communications.

V. Topology Case Study

A traditional network topology is the "+grid" topology. Which allows each satellite to connect to the closest 2 satellites in the same orbit plane and other 2 satellites from the neighboring plane. [B1] This topology ensures the inter-connectivity between all satellites, and, at the same time, allow each satellite to have a relatively small number of connections. Also, this method have small calculation overhead because the connection algorithm is simple. However, it is far from perfect because there are still unnecessary connection links and the performance (including throughput and latency) can be improved by large with better topology design. The following two research focus on the exploration of topology, and test if the new topology are applicable and beneficial to the real-world LEO satellite networks. To do so, they need to first determine the condition that they will consider, the criteria they will measure, how to decide if a new topology is better or not, and then to emulate their design and follow steps to generate their conclusion. So, following their research, we can summarize the commonly used parameters for constructing the topology, the parameters to measure after finishing the topology connection, the ways for providing interface for changing topology, which will help us understand the requirements of a good modern simulation tool.

"Network topology design at 27,000 km/hour."[1]: This paper proposed a new topology for low earth orbit satellite such that the link changes between satellites can be reduced and thus reduce latency. Their first step is to analyze satellite orbits and constellations. This means that they need to find a method of describing the satellites status. They tried to use less parameters because that will reduce data complexity. The following are the parameters

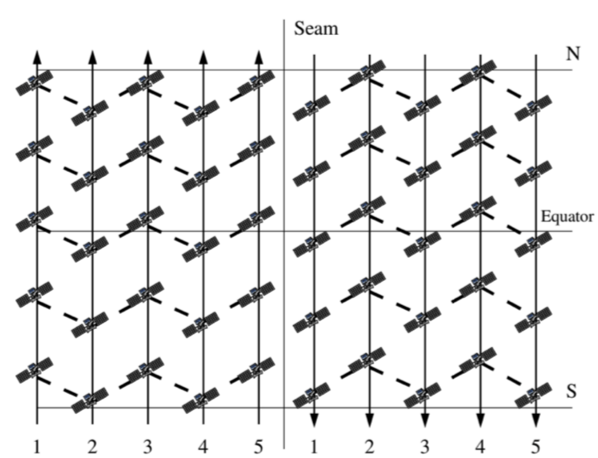

Fig. 4. +grid connection from "The Impact of Intersatellite Communication Links on LEO Performance"

they choose: Inclination, Altitude and Phase shifts. Since some other parameters such as orbit time and velocity can be derived from these three essential ones, they decided to discard others. Second, they need to take in account the system dynamics and consider the connectivity of each satellite. This step is also important because different dynamic and different satellite physical layer performance can affect their result drastically. Finally, they need to build a model that has all the previous parameters and come up with their proposed designs and test them. The tool they use for the simulation is not open sourced, but they mentioned it in their paper that "Even for just 25 cities, the [simulation] does not finish within 2 days on a machine with 64 cores and 500GB of memory. Further, extrapolating from smaller sizes, we estimate runtime for 1000 cities to require 1029 days, which would remain intractable even with perfect parallelism across a supercomputer."[1] They said that their simulator takes a very long time to finish even if with a relatively powerful modern workstation. However, they have not directly talks about the reason for this in their publication since their objective is not solving simulator issues. We propose that the reason for their long runtime is that they did not design the algorithm for simulation well. It is possible that they did not reuse some part of topology design and other layer (TCP/IP) simulation for their ground station(city). For many simulator, adding more cities does not increase the runtime by this much, but adding more control plane elements and adding more satellite increase the runtime by a large amount. Their closed source simulator is not following this pattern, so they must have different implementation when they are testing connection between cities. However, with this limited simulation capability, they successfully tested 3 different new topologies, and they showed that a "motifs" connection has the best overall performance.[1] To summarize, their simulation tool has advantage in customize parameters, enough measuring quantities, and customizable topology, and what they need to improve from this software is that they can fix the resource utilization and parallelism data reuse issue.

"'Internet from Space' without Inter-satellite Links?"[15]:

This research is about exploring the possibility of having the LEO satellite network working properly with less inter-satellite links. They propose that although previous works on reducing inter-satellite links failed, new technologies such as "satellite miniaturization, reusable rocket boosters, and the ability to set up and maintain high-bandwidth laser connectivity between satellites traveling at high velocity"[15] have been developed, so the result may have been changed since previous works.

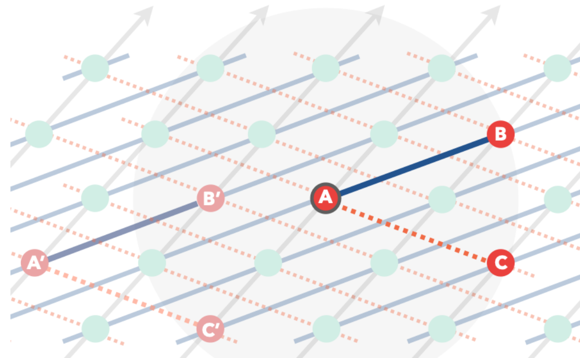

Fig. 5. motifs topology from *"Network topology design at 27,000 km/hour"*

The goal of their research is to find a topology with less inter-satellite links that can have similar performance as a regular topology. The matrices that they will measure is latency and its variability, the network-wide throughput, and the stability. The stability is important to this research because with less inter-satellite links, more data will be transferred through the terrestrial-satellite link, and this link is more vulnerable to variables such as weather condition and cloud movements. So, in this special case, they use a simulation library "ITU-Rpy"[6], which can simulate the attenuation of terrestrial-satellite link[fig6]. This is a proprietary simulator designed only for simulating attenuation So, it does not provide realistic simulation of other satellite properties such as link layer and transport layer protocols, and when they are comparing results, they compare latency, throughput, and stability separately. They cannot have a comparison of the overall performance. Given this, they still get valuable information that with less inter-satellite links, the network will have decrement in all three aspect. The network will have more latency, less overall throughput, and less reliability. So, inter-satellite links removal have a larger negative impact compare to its positive impact of reducing satellite cost and calculation overhead. From this research and the simulation software they used, we understand the procedure of implementing a proprietary, specific-functioned simulator. Although their implementation is not perfect because there is many other parameters can be taken into account, it does help them to derive a valuable conclusion. If such proprietary simulator can be integrated to a general purpose simulator, it will be much easier to use, and can help to generate more accurate data. This is another modification that current simulator/emulator can take.

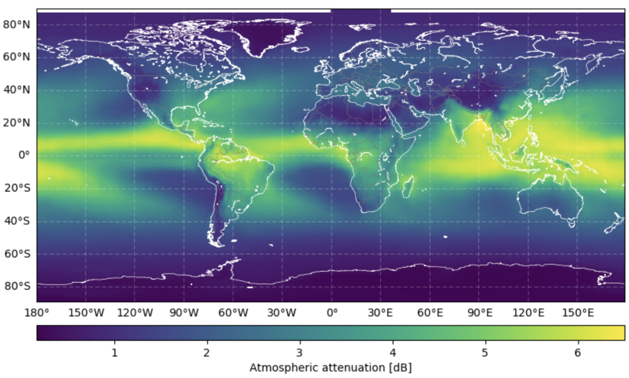

Fig. 6. Result from ITU-Rpy

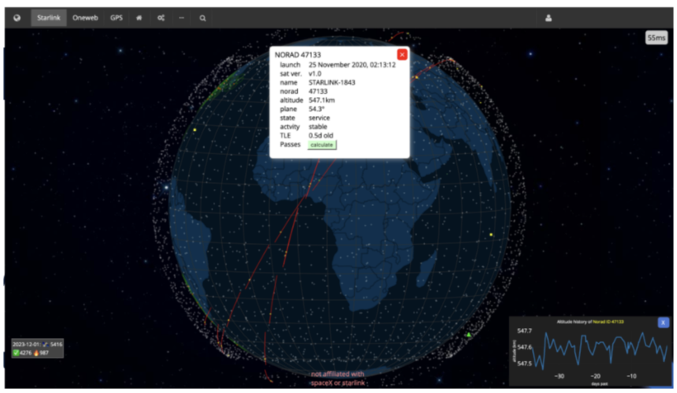

Fig. 7. Live Starlink User Interface

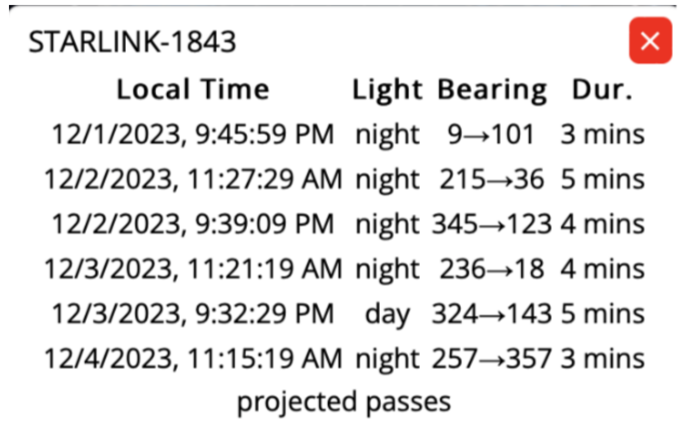

Fig. 8. Live Data from Live Starlink

Decision	Options and range of values
`Inclination`	inclination of orbit i (Inc_i)
`Altitude`	altitude of orbit i (Alt_i)
`Phase shift`	phase shift of orbit i (Pha_i)
`# of orbit`	total number of orbits (Num_{orb})
`# of satellite`	number of satellites in ith orbit ($SatN_i$)
`# of GS`	total number of ground stations
`Location of GS`	location distribution of GS
`Link band`	band range: S/X/Ku/Ka/optical
`Link type`	type range: bent-pipe, circuit- or packet-switched

Fig. 9. StarPerf Parameters

VI. Modern Satellite Software Tools

Modern Satellite Tools can be divided into three categories: Live platforms, Simulators, and Emulators. Each category have their own advantage and disadvantages. When comparing different categories of software tools, we are also going to correlate to the need from research former illustrated.

A. Live Platform

Live Starlink[11] is the most commonly used tool in this category. It provides live and real information from Starlink's already launched satellites. The advantage is that: First, the data are real. Second, have mechanism for user to specify certian ground location, orbit plane, or software/hardware versions. This method is often used to fetch data for other software tools. When it is used alone, the lack of flexibility lead to limited use case. It is expansive for research groups to explore their idea on real satellites. Also, its technical difficulty is also high.

B. Simulator

Starperf[10] is a typical simulator, it provides customization is many ways. User can customize topology and routing algorithm to develop and test possible alternatives. It also allows user defined satellite property such as inclination and phase shift. Also, this is a pure software implementation, so the cost of experiments is much lower than using real-world satellites. The downside for it is that it does not provide realistic simulation of real network traffic. This could lead to hidden problems caused by missing link layer or transport layer information.

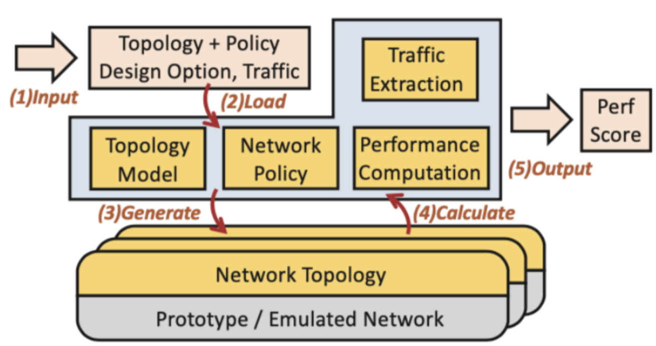

Fig. 10. Starperf Workflow Diagram

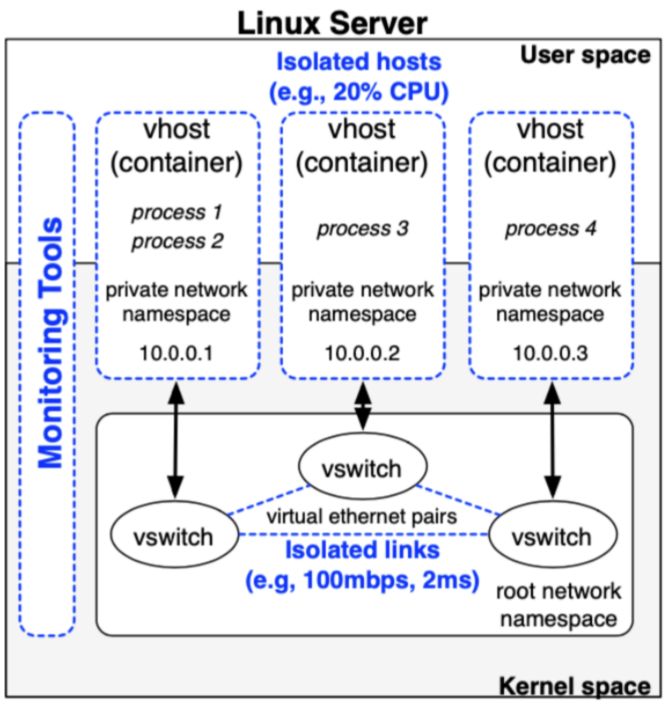

Fig. 11. Mininet Performance Isolation

C. Emulator

MiniNet Hi-Fi[13] is a broadly used emulator. It provides both real network environment simulation and the customizable factors. User can define their own topology and routing algorithm while having the real network application and protocols running in simulation. It also provides performance optimization with performance isolation. MiniNet uses 1. Control groups:allow a group of processes to be treated as a single entity for scheduling and resource management; 2. CPU Bandwidth Limits: enforce a maximum time quota for a cgroup within a given period of time. CPU time is fairly shared among all cgroups; and 3. Traffic Control: using tc configures link properties such as bandwidth, delay, and packet loss; to achieve high efficiency parallelism. However, even with performance optimization, MiniNet still takes significantly longer time to run the simulation due to much more complicated network realism. Another disadvantage of MiniNet is that it does not have good encapsulation for its source code, so implementing new design is possible here but more time consuming than using a simulator.

VII. Emulator Reproduction

StarryNet is one of the latest open-source emulator, and it does meet most of the requirements for a easy-to-use, accurate, and, versatile simulation software. So, we want to know how it actually performs. To reproduce the work, we follow the below steps: install the source-code, install required helpers such as docker and sgp, build the project, tune parameters, get result, and intemperate result. The environment we run the simulation on is a WSL Unbuntu 20.04 with 8 core CPU and 16GB of memory. The installation and set up process is smooth. Unlike other simulators such as STK, StarryNet is a fully open-source project. There is not license or payment needed. The parameters are also clearly listed in the user manual, which include most of our proposed requirements[fig13,fig14]. After setting up the environment and necessary parameter, we run a simulation of 100 satellite and 2 ground stations and got a reasonable delay number matrix[fig]. Then we tried to explore different parameters starting from number of satellite and ground stations. We found that increase number of ground stations does not have significant change in runtime. The runtime different is within margin of error. However, change in number of satellite does change the runtime behavior. As shown in [fig16], the runtime increase as number of satellite increase. Furthermore, at 70*70 satellite, the program show ”out of memory” error. This shows one disadvantage of this emulator. Another disadvantage is that although the simulator provides topology and routing parameter, the interface has not yet been implemented. To sum up, StarryNet does provide various functionality along with realistic data and ease of use, but it still has room to advance. First, it does not have sufficient performance optimization. We can only run a satellite network that has less than 2500 satellites. We believe that on a much more powerful computer and with more memory, the runtime and size of network can be improved, but performance optimization is still needed to prepare for even larger satellite network in the future. Second, there are functions that are yet to be implemented. Given StarryNet’s develop time and abundance of useful interfaces, we believe that functions such as supporting multiple topology design and change routing method can be implemented in the future.

VIII. Challenges and Gaps

In conclusion of the previous analysis, modern LEO satellite simulator need to satisfy four main properties. First, it need to have a realistic data set. Currently, most researches uses SpaceX’s published satellite data[11] because that data is real and reliable, but many simulation tools has not integrate this data fetching step into their workflow. If this can be implemented, the cost for researchers to implement an almost real simulation will be reduced. Second, topology and routing customization function or interfaces can be mounted. Since currently most use cases for simulation software

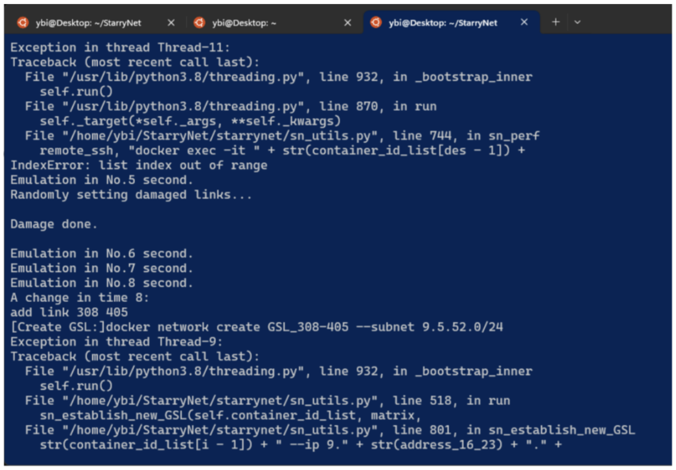

Fig. 12. Software environment

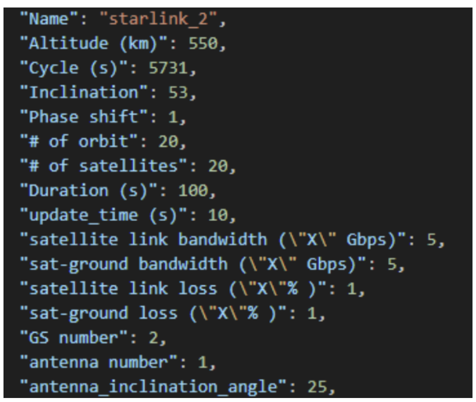

Fig. 13. Compile Time Changeable parameter

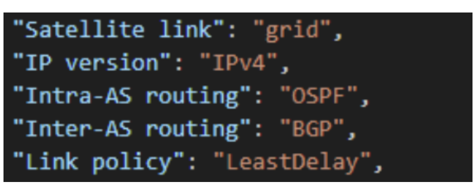

Fig. 14. Routing and Topology related parameter

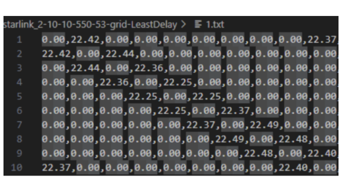

Fig. 15. Delay output from StarryNet

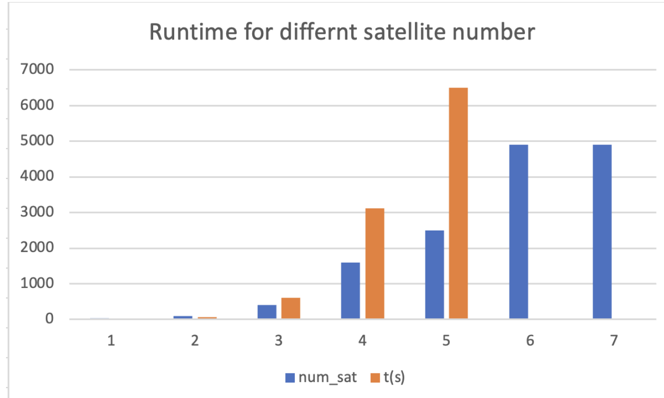

Fig. 16. Runtime

is to conduct related research, and such research often works with topology and routing. It is a necessary feature for an easy to use simulation software. The third is performance optimization. With the large data set and realism network protocol and environment simulation, the throughput of calculation and data will be very large. There needs to be more parallelism implemented optimized for CPU or memory usage. Forth, environmental factor simulation can be integrated. There are many research related to ground-satellite links, and now they can only use dedicated simulation software only to simulate the environmental variables. At the same time, those dedicated software are poor in simulation other factors that are already mature technique in other simulators. If general purpose simulator can integrate environmental simulation, more valuable research result will have a chance to appear.

IX. Individual Contribution

Individual contribution is shown as follows:

Shun Zhang: Conducted research simulation software, reproduction research and setups, midterm and final presentation preparation, and final paper write-up.

Yihe Bi: Did research on topology with their simulation software, reproduction procedure, midterm and final presentation preparation, and final paper write-up.

Chengming Li: Analyze the paper on routing with their simulator/emulator(4), reproduction initial steps, Midterm and Final Presentation preparation, and final paper write-up(Project introduction + motivation + routing case study).

X. Conclusion

With growing number of LEO satellites, more and more research are being done on this topic. Simulator, among all, also develops rapidly during recent years. We looked at various researches and the simulator they are using and generated future development suggestion for simulators based on current needs, which are realistic data, network customization, performance optimization, and environmental variables. Simulating software fulfilling these functionalities can largely help research groups. In the future, there might be other needs caused by development of LEO networks, but simulating software will still be important for related topic researches.

References

[1] Bhattacherjee, D., Singla, A. (2019, December). Network topology design at 27,000 km/hour. In Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies (pp. 341-354).

[2] Cao, X., Zhang, X. (2023, May). SaTCP: Link-Layer Informed TCP Adaptation for Highly Dynamic LEO Satellite Networks. In IEEE INFOCOM 2023-IEEE Conference on Computer Communications (pp. 1-10). IEEE.

[3] L2: Giuliari, Giacomo, et al. ”Internet backbones in space.” ACM SIGCOMM Computer Communication Review 50.1 (2020): 25-37.

[4] L1: Handley, Mark. ”Delay is not an option: Low latency routing in space.” Proceedings of the 17th ACM Workshop on Hot Topics in Networks. 2018.

[5] L3: Handley, Mark. ”Using ground relays for low-latency wide-area routing in megaconstellations.” Proceedings of the 18th ACM Workshop on Hot Topics in Networks. 2019.

[6] Inigo del Portillo. 2017. ITU-Rpy: A python implementation of the ITU-Rpy. Recommendations to compute atmospheric attenuation in slant and horizontal paths. https://github.com/iportillo/ITU-Rpy/.

[7] Kassing, S., Bhattacherjee, D., Águas, A. B., Saethre, J. E., & Singla, A. (2020, October). Exploring the” Internet from space” with Hypatia. In Proceedings of the ACM Internet Measurement conference (pp. 214-229).

[8] Kempton, B. S. (2020). A simulation tool to study routing in large broadband satellite networks. Christopher Newport University.

[9] Lai, Z., Li, H., Deng, Y., Wu, Q., Liu, J., Li, Y., ... Wu, J. (2023). StarryNet: Empowering Researchers to Evaluate Futuristic Integrated Space and Terrestrial Networks. In 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23) (pp. 1309-1324).

[10] Lai, Z., Li, H., Li, J. (2020, October). Starperf: Characterizing network performance for emerging mega-constellations. In 2020 IEEE 28th International Conference on Network Protocols (ICNP) (pp. 1-11). IEEE.

[11] Starlink analysis at the carinthia university of applied sciences (CUAS). https://forschung.fhkaernten.at/roadmap-5g/files/2021/07/Starlink-Analysis.pdf.

[12] Starrey Net, https://github.com/SpaceNetLab/StarryNet#what-are-the-components

[13] N. Handigol, B. Heller, V. Jeyakumar, B. Lantz, and N. McKeown. Reproducible network experiments using container-based emulation. In Conference on emerging Networking Experiments and Technologies (CoNEXT). ACM, 2012.

[14] L4: Wu, Yipeng, Zhihua Yang, and Qinyu Zhang. ”A novel DTN routing algorithm in the GEO-relaying satellite network.” 2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN). IEEE, 2015.

[15] Y. Hauri, D. Bhattacherjee, M. Grossmann, and A. Singla. ”Internet from Space” without Inter-Satellite Links. In Proceedings of the 19th ACM Workshop on Hot Topics in Networks (HotNets), page 205–211. ACM, 2020.