Astrolabe: Modeling RTT Variability in LEO Networks

Vaibhav Bhosale, Ketan Bhardwaj, Ahmed Saeed
Georgia Institute of Technology, Atlanta, Georgia, USA
vbhosale6@gatech.edu,ketan.bhardwaj@cc.gatech.edu,asaeed@cc.gatech.edu

Abstract

Networking practitioners heavily rely on intuitive models of the behavior of networks when designing and analyzing protocols and algorithms. However, there is still a lack of such intuitive models of the behavior of LEO satellite networks, hindering innovation. In this paper, we provide a first step towards improving the intuitive understanding of the behavior of LEO satellite networks. In particular, we focus on developing a model that captures the RTT variability exhibited by such networks. We rely on simple and intuitive calculations instead of expensive simulations. To capture the high RTT variability exhibited by satellite networks, we estimate lower and upper bounds for the RTT between a pair of ground stations. We introduce Astrolabe, a novel approach that achieves accurate bounds, with a median lower bound within 1.15× the actual lowest RTT and a median upper bound within 2× the highest RTT in a few seconds instead of hours required by simulations or measurements.

CCS Concepts

• **Networks** → **Network performance modeling**; **Network dynamics**.

Keywords

Low Earth Orbit Satellite Networks, Round Trip Time Estimation

ACM Reference Format:
Vaibhav Bhosale, Ketan Bhardwaj, Ahmed Saeed. 2023. Astrolabe: Modeling RTT Variability in LEO Networks. In *Workshop on Low Earth Orbit Networking and Communication (LEO-NET '23), October 6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3614204.3616104

LEO-NET '23, October 6, 2023, Madrid, Spain

ACM ISBN 979-8-4007-0332-4/23/10…$15.00
https://doi.org/10.1145/3614204.3616104

1 Introduction

Networking practitioners rely on intuitive models of networks and workload characteristics when debugging application performance or customizing networking algorithms (e.g., congestion control). For example, such models have been very useful in the contexts of datacenter networks (e.g., incast, short buffer sizes, and tight latency requirements) [1, 17] and cellular networks (e.g., high variability due to physical layer artifacts and scheduling of the shared medium) [21–23]. Such an understanding, even in its abstract form, provides guidance when reasoning about problems unique to a specific type of network. Further, it helps identify the unique challenges and opportunities when designing new algorithms and protocols for that network. LEO satellite networks are no exception.

Our prior work shows that the high velocities of the LEO satellites introduce unavoidable high variability in the routes selected by a LEO satellite network [4]. The Round Trip Time (RTT) between two ground stations [1] can vary by up to 2× within a few seconds due to a change in the selected routes. Further, we found that, unlike in cellular networks, variability in LEO satellite networks has some structure. These observations highlight the need for an intuitive model of the behavior of satellite networks to enable the exploitation of that structure to design better-performing algorithms and protocols.

Our community currently lacks the intuitive understanding of the behavior of LEO satellite networks that is needed to predict and analyze their operational dynamics. This gap is especially pronounced when compared to well studied networks like data center networks, cellular networks, WiFi, and sensor networks. That lack of clarity is caused by several factors. First, it is prohibitively expensive to create a small-scale real satellite network for research, making it much harder to study compared to any other types of networks. Second, most large-scale LEO satellite network deployments are relatively new, limiting the amount of data that the community has on their behavior. Finally, operators of LEO satellite networks are not transparent with their data.

In this paper, we present Astrolabe, a first attempt at building an intuitive model of RTT variability in LEO satellite networks. We observe that generating a single estimate of the RTT between two ground stations is untenable due to the inherent high variability exhibited by the network. Thus, the

[1] In the context of this paper, ground stations and user terminals play the exact same role. For the rest of the paper, we will refer to both as ground stations.

Astrolabe model estimates a lower bound and an upper bound for the RTT between any pair of ground stations as a function of the constellation configuration. Existing approaches for quantifying RTT variability require computationally expensive simulations [2–4, 12, 13]. Running such simulations may not be always practical due to the substantial time and resource requirements. Concretely, we observed that replicating the latency observations for 100 cities for a single constellation can take 4-5 hours on a powerful machine. On the other hand, measurements would require expensive testbed deployments, capturing the complete extent of RTT variability over at least one full orbital time period, which is approximately 2 hours. Scaling these approaches for different routing policies or topologies would significantly increase the time and cost involved. Consequently, the expensive nature of current modeling methods impedes the pace of development of networking algorithms. Further, it is impractical to generate these estimates on the fly, especially while analyzing different ground-station pairs.

Our objective is to provide general and accurate, albeit slightly conservative, bounds that are easy to compute. These models can be utilized to fine-tune and potentially redesign networking algorithms for optimal performance in LEO satellite networks. Astrolabe offers a step-by-step process for determining both upper and lower bounds for any pair of ground stations using any constellation. For the lower bound estimation, Astrolabe starts with the geodesic distance and incorporates the characteristics of ground-satellite links (GSLs) and inter-satellite links (ISLs) to derive the lower bound estimates. These lower bounds, at the median, are within 1.15× of those obtained through simulation. As for the upper bounds, Astrolabe takes a conservative approach by always considering inter-orbit ISLs to travel from the source to the destination. This approach yields estimates within 2× of simulation-generated estimates in the median. Our estimates are closer for cases that require a pessimistic perspective. Computing Astrolabe can be done by hand or can be coded and evaluated in a matter of seconds, compared to the hours needed for simulations and measurements.

Our findings underscore the need for further exploration in this area to develop even more precise models that can accommodate diverse constellation topologies and routing strategies while continuing to be intuitive to compute. We hope that this line of work on producing intuitive models of LEO satellite networks will empower the community to customize networking algorithms and debugging tools for LEO satellite networks, making the most of the new and valuable resource.

2 Background

Our prior work observed the prevalence of route churn and RTT variability while using shortest path routing in LEO satellite networks [4]. We found that high route churn is quite common with a median path life of 30 seconds. While in most cases, route churn doesn't result in significant latency gains (more than 70% results in less than 25% gains), some cases may result in as much as a 2.5× increase in RTT. The magnitude of this RTT variability depends on the locations of the communicating ground stations, thus exhibiting a spatial structure. Notably, the highest RTT variability has been observed for shorter paths and paths that deviate from the orbital planes. This study, along with several other prior works, primarily focused on networks that rely on shortest-path routing, where routes are periodically selected by computing the shortest paths between all pairs of communicating ground stations [9–12, 16, 24]. In this paper, we focus on networks that leverage shortest-path routing, attempting to develop an intuitive model for predicting RTT variability in those networks.

Setup. We apply the Astrolabe model to the first orbital shells for Starlink, Kuiper, and Telesat constellations. We validate the bounds generated by Astrolabe using the Hypatia simulator [12]. For these simulations, we use the publicly available data for modeling the satellites with the ISLs connected in a +Grid topology, using a configuration similar to that used in prior work [3, 5, 6, 8, 9, 14, 15, 18–20]. For the workload, all pairs of the 100 most populous cities in the world are used as source-destination pairs of ground stations. This covers all different kinds of scenarios based on the length of the paths as well as the angle between the cities,[2] which have been known to impact the RTT variability [4] (Figure 6).

3 Understanding RTT Models

Astrolabe generates upper and lower bounds for RTTs along any given path, taking into account the positions of the communicating ground stations and various characteristics of the satellite constellation, including altitude, inclination angle, elevation angle, and number of orbital planes.

3.1 Determining Lower Bounds

In terrestrial networks, the theoretical lower bound for any path can be determined using the geodesic distance between the source and the destination and the speed of light propagation. We apply the same intuition, using the geodesic distance at the satellite altitude. We outline the steps used by Astrolabe to determine the lower bound in Algorithm 1.

Paths in satellite networks are made up of two GSLs and zero or more ISLs. The GSLs connect the ground stations to the satellites which will be connected by a sequence of ISLs. First, we estimate the lengths of the GSLs used to reach the first and last hop satellites. Then, we calculate the geodesic distance between the farthest satellites reachable through these links, all while minimizing the total path length. This concept is grounded in the triangle inequality which asserts

[2]The longest paths are all horizontal since the cities are within the Starlink coverage region and hence the latitude values are bounded

Algorithm 1 Calculating lower bound using Astrolabe

```
function EstimateLowerBound(src, dst)
    geodesicDistance ← calculateGeodesicDistance(src,dst,altitude)
    if geodesicDistance < 2*gslProjections then
        angle ← (geodesicDistance/2)/(earthRadius+altitude)
        gslLengths ← cosineRule(altitude,earthRadius+altitude,angle)
        minPathLength ← 2*gslLengths
    else
        remainingArc ← geodesicDistance − 2*gslProjections
        remainingChord ← getChordFromArc(remainingArc,altitude)
        minPathLength ← remainingChord+2*maxgslLength
    end if
    return minPathLength/speedOfLight
end function
```

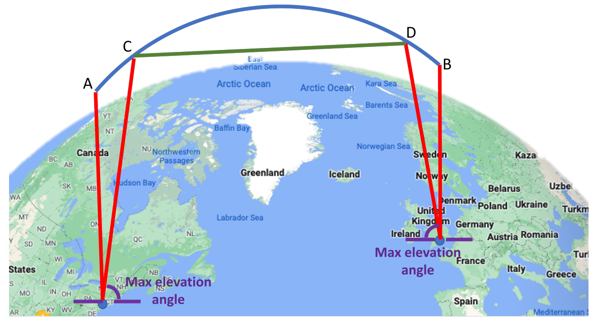


Figure 1: Depicting the calculation of lower bounds in Astrolabe

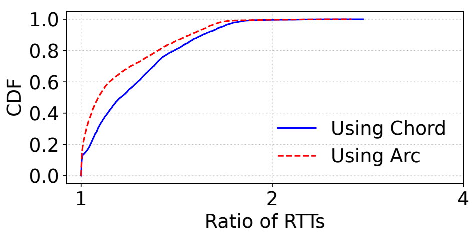


Figure 2: Variation of lower bounds calculated by Astrolabe using the different ISL projections.

Algorithm 2 Calculating upper bound using Astrolabe

```
function EstimateUpperBound(src, dst)
    horizontalPlanes ← Δlatitude/(360/numPlanes)
    verticalPlanes ← verticalOrbitalPlanes(src,dst)
    pessimisticPlanesCount ← max(horizontalPlanes,verticalPlanes)+2
    maxPathLength ← pessimisticPlanesCount*interOrbitISL+2*maxgslLength
    return maxPathLength/speedOfLight
end function
```

that the GSL lengths are shorter than the sum of the altitude and the geodesic distance covered by the GSL. This implies that longer GSL lengths inversely correspond to shorter overall path lengths. The longest GSL length is observed at the maximum elevation angle, which is what we use for the rest of our calculations.

After determining the lengths of the GSLs, we determine the lengths of the ISLs. We determine their lengths as shown in Figure 1. We first project both the ground stations at the satellite altitude (points A and B) and determine the geodesic distance between those two points. Next, we subtract the geodesic distance covered by the two GSLs (GSL geodesic projections) from the great-arc AB, resulting in the great-arc CD. To determine the GSL geodesic projections, we first find the angle covered by the great-arc AC and BD at the center by using the cosine rule with lengths as `earth radius, earth radius + altitude,` and `max GSL length`. Then, we use this angle to determine the lengths of the great arcs AC and BD.

For pairs of ground stations that are close to each other (i.e., when the sum of the two GSL projections is higher than the geodesic distance between the ground stations' projected points), the shortest path between these ground stations will only have a single satellite hop. Applying fundamental geometric principles, we deduced that the optimal route is achieved when a single satellite hop occurs precisely at the midpoint between the projected locations of the ground stations. These GSL lengths can be calculated employing the cosine law, which takes into account the angle formed by the geodesic distance and the two sides—represented as the `earth radius` and `earth radius + altitude`.

For the remaining majority of the cases, the geodesic component will involve the great-arc CD from Figure 1. However, since ISLs are straight lines, using the geodesic distance (which is a great arc) results in overestimating the path length in some cases, specifically, when the actual shortest path traverses the shortest possible distance between the communicating ground stations (the path has the maximum possible GSL lengths). In our 4950 source-destination pairs workload, we observed 25 such cases where the lower bound was 0.05% higher than the actual shortest path. To correct this, we instead used the chord length CD which resulted in accurate lower bound calculations in all the cases. However, this also resulted in the lower bound being slightly looser compared to the case of using the arc. Figure 2 compares the two methodologies for comparing the lower bound as the ratio between the estimate and the smallest RTT observed in our simulations.

Astrolabe's lower bound calculations are the same for all constellations operating at the same altitude because it only considers the altitude, elevation angle, and the maximum GSL length (which is determined by the altitude and elevation angle). The lower bounds can be made tighter by further incorporating the remaining characteristics of the constellation (e.g., ISL lengths, number of orbital planes, number of satellites), which we leave for future work.

Latitude Range	Latitudes per Orbital Plane
-53, -50	0.5
-50, -40	2
-40, 40	4
40, 50	2
50, 53	0.5

Table 1: Variations in density of orbital planes in the vertical direction with changing latitude ranges

3.2 Determining Upper Bounds

Determining upper bounds is considerably more challenging because the lengths of longest shortest paths[3] are influenced by many factors, including the distance between the ground stations and the alignment of the ground stations with the orbital planes [4]. Our goal with Astrolabe is to provide an intuitive and generic way of calculating rough estimates of the worst-case RTT while maintaining the accuracy of these bounds. Thus, we follow an overly pessimistic, yet simple, approach. Our approach leverages the observation that the worst case RTTs result from paths that traverse inter-orbit ISLs going through all the orbital planes from the source to the destination, maximizing the number of hops they go through. Algorithm 2 shows the upper bound calculations in Astrolabe.

To determine the number of inter-orbit ISLs, we compute the number of orbital planes between the pair of communicating ground stations. As a simplified measure, we consider the orbital planes crossed while traversing vertically (between the latitudes of the ground stations) and horizontally (between the longitudes of the ground stations) and choose the maximum of those two to determine the upper bound. This is clearly the pessimistic approach designed to effectively cover the scenarios where RTT variability is the highest, and hence overestimating the other cases (while being accurate).

Determining the number of orbital planes between ground stations horizontally is quite straightforward. The orbital planes are uniformly distributed, thus the angular separation between two orbital planes can be determined easily (360 / # orbits) and then this can be used to calculate the number of orbits between the ground station pair. However, doing so in the vertical direction is trickier. For inclined orbits used by most LEO satellite networks, satellite density is sparser at the equator and denser at higher latitudes. Therefore, we determined density buckets to determine the number of orbital planes vertically between ground stations. We use the latitudes per orbital plane as shown in Table 1 to determine the number of orbital planes separating a ground station pair vertically. These calculated values represent the horizontal and vertical separations between the ground stations. Therefore, to calculate the upper bound, we take the maximum of these separations and use that value as the number of inter-orbit ISLs. Since we are determining the upper bound, we assume the GSLs to be in the opposite direction (in contrast to the lower bound when they were towards the other ground station), and hence add one more orbital plane each. Finally, we add the length of the two GSLs to determine the upper bound.

For determining the upper bounds, Astrolabe only considers the positions of the ground stations along with the inter-orbit ISL lengths. These estimates can be improved by incorporating different scenarios such as the distance and the alignment of the ground stations along the orbital planes.

[3]The longest path between a pair of ground stations selected using shortest-path routing

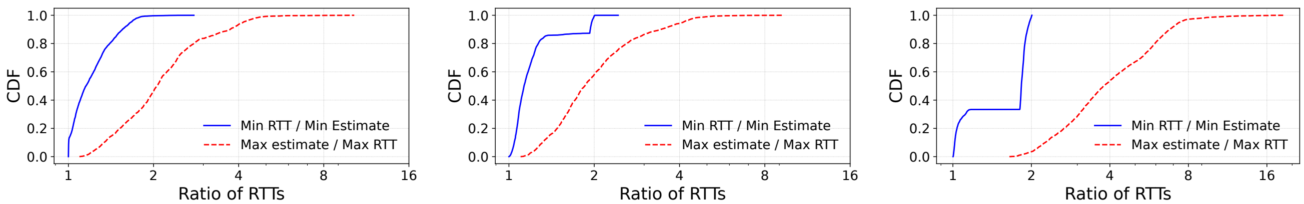
4 Evaluation

To validate the accuracy of these bounds, we compare them to the actual results observed in a simulated environment for three constellations: Starlink, Kuiper, and Telesat. We calculate the ratio between the actual lowest RTT and the lower bound predicted by Astrolabe, as well as the ratio between the upper bound predicted by Astrolabe and the actual largest RTT observed. A ratio greater than one indicates the accuracy of the bounds, with the magnitude of the ratio determining the looseness. Results for the three constellations are shown in Figure 3.

Effectiveness of Astrolabe lower bounds.

- For shorter paths where a single satellite hop will suffice all the times, Astrolabe estimates are always closer to the actual lowest RTT (Figure 4). The exception to this is the Moscow-St. Petersburg route where the estimate is 1.25× the lowest RTT due to the high latitude of St. Petersburg (59^o) resulting in very few satellites accessible from there.
- Estimates for longer paths show significant variations. Variations in estimating a lower bound on longer paths can be explained based on the angle between the ground station pair. In particular, ground stations along orbital planes (53^o or 127^o) have tight lower bound estimates. All other ground stations have looser estimates (Figure 5). This is because ground stations along orbital lines would use intra-orbit ISLs, traversing long distances in a straight line. On the other hand, deviating from the orbital lines would result in the use of inter-orbit ISLs which do not travel along the geodesic distance resulting in looser estimates.
- While Figure 4 might seem to indicate that the estimates become looser with increasing distance, it is rather an artifact of the nature of paths available. In particular, there are very few places on earth that are more than 17,000km apart (almost all the longer paths are for horizontal ground station pairs Figure 6).

Effectiveness of Astrolabe upper bounds. Due to the pessimistic nature of upper bound estimation, Astrolabe estimates for upper bounds are tighter for ground station pairs that actually suffer from high RTT variability (where the ratio of min RTT to max RTT is lower than 0.7 [4]).



(a) Starlink **(b) Kuiper** **(c) Telesat**

Figure 3: The CDF of the lower and upper bounds determined by Astrolabe for the three constellations

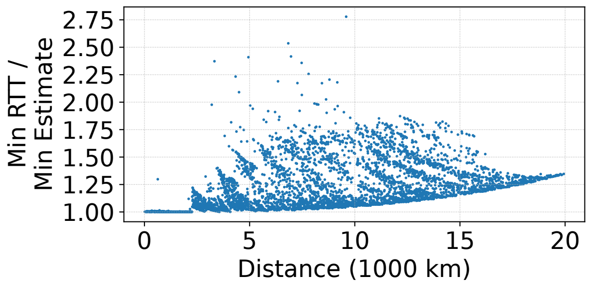


Figure 4: Variation of lower bounds calculated by Astrolabe based on the distance between the ground stations

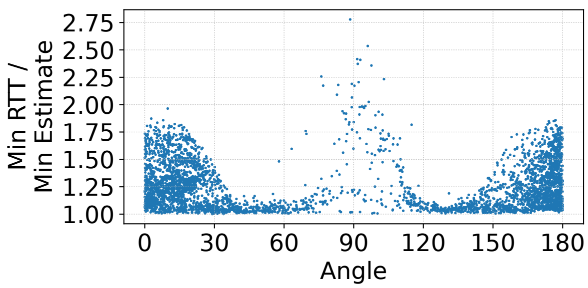


Figure 5: Variation of lower bounds calculated by Astrolabe based on the angle between the ground stations. This plot does not include the shorter routes which have a single satellite hop as the shortest path.

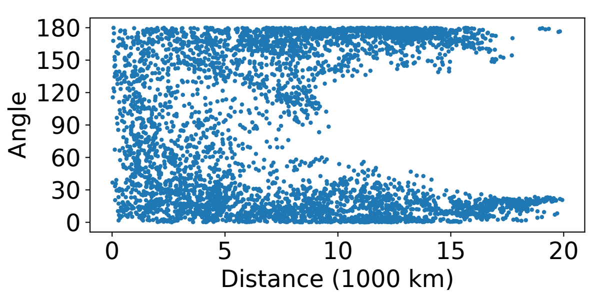


Figure 6: The distance and angular diversity of the source-destination pairs in our workload.

- As expected, Astrolabe has the poorest upper bound estimate for ground station pairs that are very close to each other (Figure 7), where the best case scenario of using a single satellite hop always happens. Astrolabe's upper bound assumes that a path has 2 inter-orbit ISLs whereas in reality there will be none.

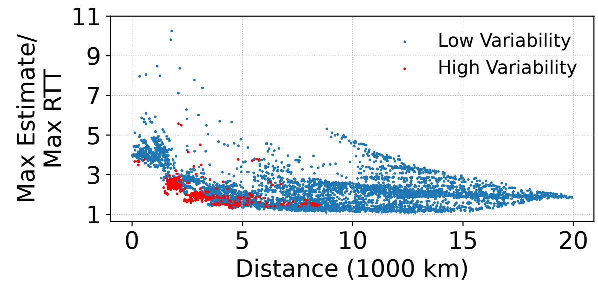


Figure 7: Variation of upper bounds calculated by Astrolabe based on the distance between the ground stations. Red points represent pairs of ground stations with high RTT variability.

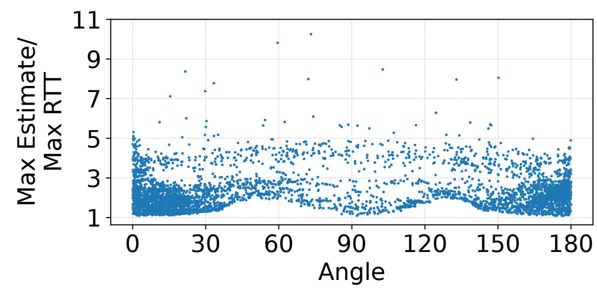


Figure 8: Variation of upper bounds calculated by Astrolabe based on the angle between the ground stations.

- For slightly longer routes where the shortest paths switch between zero or more ISLs, Astrolabe estimates are fairly accurate since the pessimistic estimates are in reality closer to the actual highest RTT observed (red dots in Figure 7).
- Longer paths show opposite trends compared to the lower bound estimates. Ground station pairs that are vertical or horizontal tend to be tighter since they are closer to the pessimistic estimates of Astrolabe, while pairs along the orbital planes are looser (Figure 8).

5 Discussion and Future Work

This paper focuses on the modeling of RTT variability in LEO satellite networks. While our current results are promising, these models need further work to become more accurate and easier to compute.

Astrolabe is currently designed for the shortest path routing strategy for a constellation that extensively uses ISLs. In practice, operators can employ a wide range of routing strategies like hot potato routing (satellites send all their packets to the nearest ground station) and sticky routing (ground stations stick to a satellite as long as it is accessible). Similarly, different operators tend to develop networks with different topologies such as networks with no ISLs (as was the case of Starlink initially) or a hybrid use of ISLs (Starlink today). We hope that our work highlights the need for more generic models of RTT that can accommodate the large design space of LEO satellite networks. In addition to assisting in the development of networking algorithms on currently deployed constellations, these models can also be used to create better constellations that minimize RTT variability.

We validated the accuracy of Astrolabe by running simulations for different constellations. However, to increase the confidence and efficacy of these models, we plan to validate the results on real constellations as part of future work. We posit that doing so for even a single constellation will greatly boost the significance of these models. Further, validating through real measurements will also help to identify and incorporate any other networking delays added due to satellite network artifacts not currently captured by Astrolabe.

Acknowledgments

This project was partially supported by NSF grant 2212098 as well as grants and gifts from Google and Cisco. We thank CloudLab [7] for providing a testbed to run our simulations.

References

[1] Mohammad Alizadeh, Albert Greenberg, David A. Maltz, Jitendra Padhye, Parveen Patel, Balaji Prabhakar, Sudipta Sengupta, and Murari Sridharan. 2010. Data Center TCP (DCTCP). In *Proceedings of the ACM SIGCOMM 2010 Conference* (New Delhi, India) *(SIGCOMM '10)*. Association for Computing Machinery, New York, NY, USA, 63–74. https://doi.org/10.1145/1851182.1851192

[2] Bhattacherjee, Debopam. 2021. *Towards Performant Networking from Low-Earth Orbit.* Ph. D. Dissertation. ETH Zurich.

[3] Bhattacherjee, Debopam and Singla, Ankit. 2019. Network topology design at 27,000 km/hour. In *Proc. of CoNext '19*. 341–354.

[4] Vaibhav Bhosale, Ahmed Saeed, Ketan Bhardwaj, and Ada Gavrilovska. 2023. A Characterization of Route Variability in LEO Satellite Networks. In *International Conference on Passive and Active Network Measurement.* Springer, 313–342.

[5] Olivier L. de Weck, Richard de Neufville, and Mathieu Chaize. 2004. Staged Deployment of Communications Satellite Constellations in Low Earth Orbit. *Journal of Aerospace Computing, Information, and Communication* 1, 3 (2004), 119–136.

[6] Inigo del Portillo, Bruce G. Cameron, and Edward F. Crawley. 2019. A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. *Acta Astronautica* 159 (2019), 123–135.

[7] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar, and Prabodh Mishra. 2019. The Design and Operation of CloudLab. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. 1–14. https://www.flux.utah.edu/paper/duplyakin-atc19

[8] Mark Handley. 2018. Starlink revisions, Nov 2018. Available at https://www.youtube.com/watch?v=QEIUdMiColU&ab_channel=MarkHandley.

[9] Handley, Mark. 2018. Delay is Not an Option: Low Latency Routing in Space. In *Proc. of HotNets '18*. 85–91.

[10] Handley, Mark. 2019. Using Ground Relays for Low-Latency Wide-Area Routing in Megaconstellations. In *Proc. of HotNets '19*. 125–132.

[11] Menglan Hu, Mai Xiao, Wenbo Xu, Tianping Deng, Yan Dong, and Kai Peng. 2022. Traffic Engineering for Software Defined LEO Constellations. *IEEE Transactions on Network and Service Management* (2022).

[12] Kassing, Simon and Bhattacherjee, Debopam and Águas, André Baptista and Saethre, Jens Eirik and Singla, Ankit. 2020. Exploring the "Internet from space" with Hypatia. In *Proc. of ACM IMC '20*. 214–229.

[13] Zeqi Lai, Hewu Li, Yangtao Deng, Qian Wu, Jun Liu, Yuanjie Li, Jihao Li, Lixin Liu, Weisen Liu, and Jianping Wu. 2023. StarryNet: Empowering Researchers to Evaluate Futuristic Integrated Space and Terrestrial Networks. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1309–1324.

[14] LeoSat. 2019. Technical Overview. Available at https://www.leosat.com/to/media/1114/leosat-technical-overview.pdf.

[15] Jiulong Ma, Xiaogang Qi, and Lifang Liu. 2017. An effective topology design based on leo/geo satellite networks. In *International Conference on Space Information Network*. Springer, 24–33.

[16] Mihael Mohorcic, Markus Werner, Ales Svigelj, and Gorazd Kandus. 2002. Adaptive routing for packet-oriented intersatellite link networks: performance in various traffic scenarios. *IEEE Transactions on Wireless Communications* 1, 4 (2002), 808–818.

[17] Arjun Roy, Hongyi Zeng, Jasmeet Bagga, George Porter, and Alex C. Snoeren. 2015. Inside the Social Network's (Datacenter) Network. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (London, United Kingdom) *(SIGCOMM '15)*. Association for Computing Machinery, New York, NY, USA, 123–137. https://doi.org/10.1145/2785956.2787472

[18] Afreen Siddiqi, Jason Mellein, and Olivier de Weck. 2005. Optimal reconfigurations for increasing capacity of communication satellite constellations. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. 2065.

[19] Kawsu Sidibeh. 2008. *Adaption of the IEEE 802.11 protocol for inter-satellite links in LEO satellite networks.* Ph. D. Dissertation. University of Surrey (United Kingdom).

[20] Tanya Vladimirova and Kawsu Sidibeh. 2007. Inter-Satellite Links in LEO Constellations of Small Satellites.

[21] Keith Winstein, Anirudh Sivaraman, and Hari Balakrishnan. 2013. Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. USENIX Association, Lombard, IL, 459–471. https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/winstein

[22] Yaxiong Xie, Fan Yi, and Kyle Jamieson. 2020. PBE-CC: Congestion Control via Endpoint-Centric, Physical-Layer Bandwidth Measurements. In *Proceedings of the Annual Conference of the ACM Special Interest Group on Data Communication on the Applications, Technologies, Architectures, and Protocols for Computer Communication* (Virtual Event, USA) *(SIGCOMM '20)*. Association for Computing Machinery, New York, NY, USA, 451–464. https://doi.org/10.1145/3387514.3405880

[23] Yasir Zaki, Thomas Pötsch, Jay Chen, Lakshminarayanan Subramanian, and Carmelita Görg. 2015. Adaptive Congestion Control for Unpredictable Cellular Networks. *SIGCOMM Comput. Commun. Rev.* 45, 4 (aug 2015), 509–522. https://doi.org/10.1145/2829988.2787498

[24] Shengyu Zhang, Xiaoqian Li, and Kwan Lawrence Yeung. 2022. Segment routing for traffic engineering and effective recovery in low-earth orbit satellite constellations. *Digital Communications and Networks* (2022).