DISS. ETH NO. 27845

Towards Performant Networking from Low-Earth Orbit

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZÜRICH

(Dr. sc. ETH Zürich)

presented by

DEBOPAM BHATTACHERJEE

Master of Science (Technology) in Security and Mobile Computing
KTH Royal Institute of Technology (Sweden), Aalto University (Finland)

born on 02.08.1987

citizen of India

accepted on the recommendation of

Prof. Dr. Ankit Singla (ETH Zürich), examiner
Prof. Dr. Adrian Perrig (ETH Zürich), co-examiner
Dr. Ranveer Chandra (Microsoft Research), co-examiner

2021

Abstract

Internet infrastructure is potentially at the cusp of a radical change. While Starlink, SpaceX's proposed low-Earth orbit constellation of 40,000+ satellites, already has 1,600+ satellites in orbit and has started offering beta service to users across the globe, it is only one competitor in a new "space race" to build satellite-based global Internet services. A large number of competing companies like Amazon, Telesat, OneWeb, and others, with their projects at various stages of maturity, make it likely that a significant fraction of the global population will be able to connect to the Internet via satellites in the near future.

While consumer-facing satellite networks have already existed for decades, the new developments differ from those of the past in their goals, their scale, and the technology involved. The largest "NewSpace" constellations target cheap, global low-latency Internet coverage using thousands of satellites in low-Earth orbit. In contrast, currently deployed constellations like HughesNet serve niches like rural coverage using a few satellites in geostationary orbit, resulting in large latencies of hundreds of milliseconds. While Iridium and Iridium NEXT operate in low-Earth orbits, they target the even narrower niche of satellite telephony.

The scale of the new constellations being deployed enables global coverage and low-latency connectivity at the cost of high dynamicity unforeseen in today's terrestrial Internet. In order to maintain orbit, satellites fly at speeds of tens of thousands of kilometers per hour, thus continually changing link and path properties. To tackle the high dynamicity of these systems, we need to revisit all of the core networking questions like topology design, routing, and congestion control.

This work first identifies various networking opportunities and challenges that arise in low-Earth orbit mega-constellations. Motivated by the findings, it makes progress on two of these fronts: (a) topology design for low-Earth orbit networks and (b) a framework for

simulating and visualizing them, revealing the fundamental challenges facing congestion control, routing, and traffic engineering in such networks.

We propose one topology design method explicitly aimed at tackling the high temporal dynamism inherent to low-Earth orbit satellites. We exploit repetitive patterns in the network topology to avoid expensive link changes over time, while still providing near-minimal latencies at nearly 2× the throughput of standard past methods.

Despite the excitement around the upcoming planned low-Earth orbit constellations, there has been a serious dearth of analysis tools for such unforeseen networks. We built Hypatia, a packet-level simulation and visualization framework, which models the inherent dynamicity of these networks, and allows network enthusiasts to simulate and analyze these networks under varied settings, traffic demands, and protocol assumptions. This work quantifies in-depth how both routing and end-to-end transport are affected due to low-Earth orbit network dynamics. Hypatia can act as a vehicle for enabling broader research in this area.

Zusammenfassung

Der Internetinfrastruktur stehen möglicherweise radikale Änderungen bevor. Starlink, SpaceX's low-earth-orbit Konstellation besteht aus 40'000+ Satelliten, von denen bereits 1'600+ Satelliten im Orbit sind. Starlink hat begonnen erste Dienste im Beta Stadium anzubieten. Dabei handelt es sich nur um ein Wettbewerber in einem neuen "space race" um einen globalen, satellitenbasierten Internetzugang anzubieten. Eine grosse Anzahl Konkurrenten wie Amazon, Telesat, OneWeb und andere haben Projekte in unterschiedlichen Entwicklungsstadien. Diese ermöglichen wahrscheinlich in naher Zukunft einem Grossteil der Weltbevölkerung sich über Satelliten ins Internet zu verbinden.

Von den bestehenden Sattelitennetzwerken, die direkt auf Endkunden ausgerichtet sind und seit Dekaden existieren, unterscheiden sich die neuen Entwicklungen im Ziel, Ausmass und der involvierten Technologie. Die grössten "NewSpace" Konstellationen zielen auf günstige globale niedriglatenz Internetabdeckung mit tausenden Satelliten im low-Earth orbit. Im Gegensatz dazu haben aktuelle Konstellationen wie "HughesNet" das Ziel Nischen abzudecken, wie zum Beispiel abgelegene ländliche Gebiete. Sie erreichen dies durch wenige, geostationäre Satelliten. Das bringt jedoch Latenzzeiten im Bereich von Hunderten von Millisekunden. Iridium und Iridium NEXT befinden sich zwar im low-Earth orbit, aber sie zielen auf eine noch kleinere Nische: Satellitentelephonie.

Das Ausmass von diesen neuen Konstellationen ermöglicht globale Abdeckung und Konnektivität mit kleiner Latenz, bringt aber eine Dynamik die im bisherigen stationären Internet nicht auftritt. Um im Orbit zu bleiben, fliegen die Satelliten mit Zehntausenden von Stundenkilometer und ändern ständig ihre Link- und Pfadeigenschaften. Um diese hohe Dynamik zu handhaben müssen wir alle Netzwerkkernfragen wie Topologieentwurf, Routing, und Congestion control überdenken.

Diese Arbeit identifiziert zuerst verschieden netzwerktechnische Chancen und Heraus-

forderungn welche von diesen low-Earth Megakonstellationen ausgehen. Ausgehend von diesen Herausforderungen schlägt diese Arbeit neue Techniken vor für (a) Topologieentwurf für low-earth-orbit Netzwerke und (b) ein Framework um diese zu simulieren und zu visualisieren. Dabei identifizieren wir fundamentale Herausforderungen zu Congestion control, Routing und Trafficengineering in solchen Netzwerken.

Wir schlagen explizit eine Design Methode vor welche die inhärent hohe temporale Dynamik solcher low-Earth orbit Satellitensysteme beachtet. Wir nutzen repetitive Muster in der Netzwerktopologie um teure Linkwechsel zu verhindern, während wir immer noch minimale Latenzzeiten bieten und beinahe 2× den Durchsatz von aktuellen Standardmethoden erreichen.

Trotz der Begeisterung über geplante low-Earth orbit Konstellationen mangelt es an Analysetools für solche neue Netzwerke. Wir entwickelten Hypatia, ein Paketlevel Simulator und Visualisierungsframework, welches die inhärente Dynamik dieser Netzwerke modelliert. Es ermöglicht NetzwerkexpertInnen solche Netzwerke unter unterschiedlichen Einstellungen, Traffic Anforderungen und Protokollen zu simulieren und zu analysieren. Diese Arbeit quantifiziert detailliert wie beides, Routing und End-to-end transport, von den low-Earth Dynamiken beeinflusst werden. Hypatia kann als Tool für weitere Forschung auf diesem Gebiet verwendet werden.

Acknowledgments

I thank Ankit Singla for being a great PhD advisor, for trusting in my abilities, for allowing me to explore, and for being the guiding star. I am also indebted to my PhD examination committee members, Adrian Perrig and Ranveer Chandra, for their valuable feedback.

I express my gratitude to Systems Group colleagues at ETH Zürich, especially my Network Design lab-mates (read friends) – Vojislav Dukic, Simon Kassing, Melissa Licciardello, Satadal Sengupta, Max Grüner, Catalina Alvarez, and Elham Ehsani Moghadam, for the countless brainstorming sessions and for offering me a friendly & productive environment. Outside work, I am grateful to Vojislav, Milena, Dunja, Melissa, Lukas, and Maya for knitting so many adorable memories to cherish forever.

I have been extremely fortunate in being able to work with some amazing collaborators and without them doing research would not be as fun – Muhammad Tirmazi, Sangeetha Abdu Jyothi, Ilker Nadi Bozkurt, Waqar Aqeel, William Sentosa, Balakrishnan Chandrasekaran (Bala), Brighten Godfrey, Bruce Maggs, Gregory Laughlin, Anthony Aguirre, and Anja Feldmann. I thank Bala also for exhaustively mentoring me during my job search and for organizing (along with Roze) numerous weekend trips during my summer fellowship at MPI für Informatik, Saarbrücken.

I thank my parents for their love & moral support. Finally, I thank Taniya, my wife, for being beside me through thick and thin and for the everyday magic that keeps me chasing my dreams.

Contents

1

Introduction

The Internet is potentially taking "one giant leap" into space, with plans afoot for large satellite constellations (hundreds to tens of thousands of satellites; mega-constellations) to blanket the globe with low-latency broadband Internet. Several companies have disclosed efforts along these lines, including SpaceX [1], Amazon [2], Telesat [3], and OneWeb [4]. With 1,600+ [5] (in May, 2021) satellites already in orbit, and an increasing launch cadence, SpaceX's Starlink constellation is already offering limited availability of its Internet service since 2020 [6]. It is thus unsurprising that these ambitious plans for an "Internet from space" have captured the public imagination [7–11].

This chapter starts with discussions on the previous generation satellite networks in order to emphasize their differences with the recently proposed broadband mega-constellations, and the recent technology advances which enable their deployments. Then it discusses the new goals of these proposed deployments, the scale which supports these goals, and the extreme dynamicity of such networks which needs to be tackled to unleash their full

potential. Finally, it defines the scope of this thesis and the organization, lists the corresponding publications, and discusses the significance of this work in a nutshell.

Previous generation deployments. Satellite networks like HughesNet [12] and ViaSat [13] have been operational for many years. HughesNet, for example, serves niches like rural coverage for at most a few million subscribers using only tens of satellites in geosynchronous (GSO) and/or geostationary[1] (GEO) orbits. These GSO constellations have a fundamental limitation—a height of 35,786 km that results in high latency, with reported round-trip times (RTTs) often exceeding 600 ms [14]. They also provide very limited bandwidth. Non-geosynchronous orbit (NGSO) satellites are also in operation, but presently cater to niche communication needs. For instance, the medium Earth orbit (MEO) zone, with heights ranging from 2,000 km to below that of GSO, is occupied by navigation systems including GPS [15], GLONASS [16], and Galileo [17]. Also operating in this band is O3b [18], a 16-satellite constellation providing communication for ships, offshore platforms, and regions with poor terrestrial connectivity. O3b claims 140 ms RTTs and a maximum throughput of 2.1 Mbps per connection [19]. The Iridium [20] and Iridium NEXT [21] constellations have even lower altitude, operating in the low-Earth orbits (LEO; at most 2,000 km above Earth's surface), but focus on satellite telephony. Thus, no operational constellation from the previous generation addresses global broadband Internet connectivity at low latency. This is the space newer players seek to occupy by leveraging recent technology breakthroughs.

Recent developments in space technology. Various recent space-technology advances have made deploying mega-constellations consisting

[1] equatorial GSO deployments making them static relative to the Earth

of tens of thousands of LEO satellites practical, the most significant being the reduction in launch costs. Reusable boosters [22] restrict the launch costs to building second stage, fairing, and ground handling, thus reducing satellite launch costs by more than 10× [23] compared to conventional satellite launches. Satellites could also be replenished in a few years thus allowing companies to build and deploy cheaper ones with reduced lifetimes. Also, individual launches can now accommodate hundreds of satellites, thus allowing large-scale batch production and deployment. While SpaceX Falcon-9 [24] currently deploys 60 satellites per launch, this number could soon increase to ∼400 with Falcon-heavy [25]. SpaceX Starlink is currently being deployed at a pace of 120+ (2+ Falcon-9 launches) satellites per month. Other significant developments, which enable mega-constellation deployments, include miniaturization of satellites (driven by the CubeSat [26] advancements) enabling the packing of more satellites per launch and advances in inter-satellite laser technology that promises high throughput at the speed-of-light at long distances. Even more than a decade ago, laser inter-satellite links (ISLs) at 5.6 Gbps over distances up to 4,900 km were achieved between LEO satellites [27]. Multi-Gbps rates have been achieved even across 45,000 km between GEO and LEO satellites at high relative velocity [28]. Two firms advertise multi-Gbps LEO ISL offerings [29, 30], with plans to soon offer 100+ Gbps [31, 32]. Recent work [33] claims that their production-ready ISL equipment meets the needs of LEO networks in terms of size, weight, and power. SpaceX recently deployed [34] a few polar satellites with ISL capabilities.

New goal. The under-construction "NewSpace" constellations differ fundamentally from past efforts in terms of capabilities, as discussed above, and objectives. Unlike existing satellite networks [12, 20, 21], the new ones are targeting not only traditional niches such as shipping, satellite telephony,

and connectivity for rural areas, but also mass market broadband that addresses global coverage issues and also competes with current terrestrial networks in many markets. SpaceX's stated goal, for instance, is "to have the majority of long distance Internet traffic go over this network" [35]. Furthermore, with altitudes of a few hundred kilometers in LEO orbits, these constellations promise latencies comparable to terrestrial ISPs.

New scale. One design manifestation of these goals is scale: to provision enough access bandwidth for their larger target user population, the new systems deployed at much lower heights (lower latency) than GEO/MEO need many more satellites than past ones. Starlink, with its 1,600+ satellites in orbit, is already the largest ever satellite fleet in space history, but eventually, the largest planned constellations will each comprise thousands of satellites [1, 2]. The planned 40,000+ Starlink constellation [36] could provide capacity comparable to the entire Internet's long-haul fiber [37]. This has only become possible due to favorable trends in space technology as discussed above.

High dynamicity. Moving from GEO to LEO satellites lowers latency, but it makes the constellations highly dynamic. At a 550 km altitude, a satellite travels at ∼27,000 kmph, covering in a minute distances comparable to those between Munich–Berlin, Delhi–Lahore, or San Jose–Los Angeles. Satellites in adjacent or nearby orbits of the same altitude and inclination have low relative velocities near the Equator, but move faster relative to each other at higher latitudes. Thus, which satellites connect to each other, and to which ground stations (GSes) evolves over time. The changing distances between satellites, their relative velocities, and the technology used, determine which connections are feasible, and how fast they can be setup.

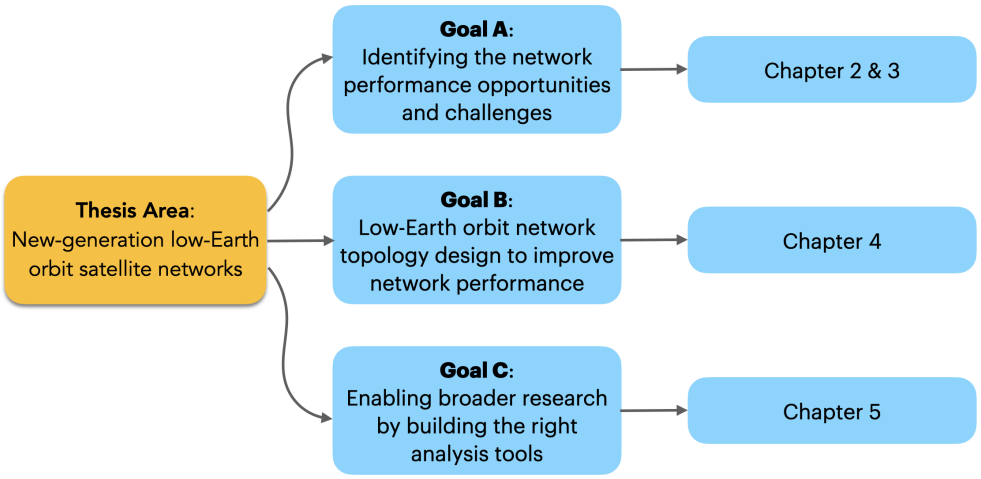


Figure 1.1: ***Thesis structure:*** *A bird's eye view of the goals and the corresponding chapters.*

Broad opportunities & challenges. Large LEO constellations promise **global Internet coverage** at **low-latency** and **high-bandwidth**. However, realizing the full potential of these networks requires addressing new research challenges posed by their **unique dynamics**. In such constellations, each satellite orbits the Earth every ∼100 minutes. This high-velocity movement of satellites creates not only high churn in the ground to satellite links, but also fluctuations in the structure of end-end paths as the satellites comprising the paths move. Our observations highlight several research challenges these networks would pose across all layers of the network stack, including how the physical topology for these networks could be designed; how routing may need to account for greater diversity and variability in route performance; and how the new latency-focused congestion control proposals may need to be reevaluated.

Scope of this thesis. As Fig. 1.1 shows, this thesis explores the LEO network opportunities and challenges, proposes and evaluates intuitive topology design for globally spanning LEO networks, quantifies routing and con-

gestion control challenges that arise due to the high dynamicity of such systems, and builds a framework for enabling broader community research in this area. More specifically some of the questions this dissertation addresses are:

- **Goal A**: What are the network performance opportunities and challenges that arise due to the scale and dynamics of planned LEO mega-constellations?
- **Goal B**: How would one inter-connect satellites with laser ISLs in a globally spanning LEO network in order to optimize performance?
- **Goal C**: How could one simulate and visualize these new-age LEO networks at a granularity that allows to effectively quantify the network-centric challenges and enable broader research in this area?

Thesis organization. The thesis is structured as follows:

Goal A: Identifying the network performance opportunities and challenges.

In Chapter.2 we introduce the unique features and settings of LEO satellite networks and the low-latency opportunity that these new constellations potentially offer. In Chapter.3 we discuss the network-centric design challenges that arise due to LEO network dynamicity.

Goal B: LEO network topology design to improve network performance.

Chapter.4 explores the topology design challenges in further depth and proposes a motif-based solution that uses repetitive patterns to inter-connect satellites thus avoiding link-churn while offering high performance.

Goal C: Enabling broader research by building the right analysis tools.

Chapter.5 introduces Hypatia, a framework for simulating and visualiz-

ing these "NewSpace" satellite networks, that allows us to quantify the LEO networking challenges and address the urgent need for tools enabling broader research in this area.

Finally, we touch upon an ambitious goal of building a globally spanning low-latency hybrid network in Chapter.6 and future LEO networking research & open problems in Chapter.7.

Related publications. Part of this work has already been published and is listed in Table. 1.1 for reference. Necessary permissions have been taken from co-authors as needed.

Significance of this work. As discussed above, this work quantifies LEO networking opportunities and challenges, proposes a topology design approach that could improve the network performance of LEO constellations by as much as 54%, and lays the foundation of broader LEO networking research by offering a simulation and visualization framework which allows fine-grained packet-level insights on these networks. Note that broadband constellation topology design is currently in a state of flux, as is evident from amendments to FCC filings over the last few years. Hence our work is set to offer network performance-centric insights to stakeholders while they finalize their designs. The simulation framework, Hypatia, is not only useful to the constellation providers, but also to the research community who could bring in their experience and intuitions to address performance issues that arise due to LEO dynamicity and drive these networks at high efficiency. In recognition of its significance, this work received the following awards:

- The LEO topology design work published at CoNEXT 2019 (discussed in Chapter.4) received IRTF's Applied Networking Research Prize [38] in 2020.

- HYPATIA received the Best Paper Award at IMC 2020 with the following public statement [39] issued by the award committee:
  "As the next generation of satellite networks is being deployed, being able to accurately simulate satellite communications in all its richness is increasingly important. This paper makes a notable contribution to this challenge and is rich in opportunities to further extend the work."

While working on this dissertation, I co-organized the following events in order to bring together people from both industry and academia interested in LEO broadband and applications.

- LEOCONN 2021 – A webinar [40] on satellite-based networking. This event hosted talks from both industry leaders and academic researchers. Attendees spread across 38 countries included 150+ industry participants, 30+ participants in top leadership positions, 20+ professors from top-50 universities, 100+ academic participants from top-50 universities, and 10 Government space agencies.
- IETF-111 side-meeting [41] on 'SATCOM activities' – The work in this thesis has also seeded ongoing efforts at IRTF to potentially setup a research group focused on "new space" satellite networks.

Table 1.1: *List of related publications in reverse chronological order. Co-authors marked with * are BS/MS thesis students supervised by me.* † *represents joint-first authorship.*

Venue & **Award**	*Paper Title* & Authors
NSDI'22 arXiv'18 Preprint	*cISP: A Speed-of-Light Internet Service Provider* D. Bhattacherjee†, W. Aqeel†, S. A. Jyothi, I. N. Bozkurt, W. Sentosa, M. Tirmazi, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla
IMC'20, **Best paper**	*Exploring the 'Internet from space' with* Hypatia S. Kassing†, D. Bhattacherjee†, A. B. Águas*, J. E. Saethre*, and A. Singla
HotNets'20	*'Internet from Space' without Inter-satellite Links* Y. Hauri*, D. Bhattacherjee, M. Grossmann*, and A. Singla
HotNets'20	*In-orbit Computing: An Outlandish thought Experiment?* D. Bhattacherjee†, S. Kassing†, M. Licciardello, and A. Singla
CoNEXT'19 **IRTF ANRP'**20	*Network topology design at* 27,000 *km/hour* D. Bhattacherjee and A. Singla
arXiv'19 Preprint HotCloud'17	*Measuring and exploiting the cloud consolidation of the Web* D. Bhattacherjee, M. Tirmazi, and A. Singla
HotNets'18	*Gearing up for the 21st century space race* D. Bhattacherjee, W. Aqeel, I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla

2

Background

In this chapter, we focus on explaining the design parameters defining LEO satellite constellations followed by specifications of a few key constellations currently under design and/or deployment. We then highlight the low latency opportunity leveraging inter-satellite connectivity. Finally, we discuss the impact of low latency on Internet applications.

2.1 What describes an LEO satellite?

We describe the parameters that specify a satellite's orbit and connectivity. We limit ourselves to circular (non-eccentric) orbits, which every proposed constellation plans to use. As the "NewSpace" constellations plan to offer global broadband, it is intuitive that they use circular orbits which offer uniformity in service.

Inclination, i: The angle made by a satellite's orbital plane with the Equator (during northward travel across the Equator) is called the angle of inclination. For polar orbits, the inclination is 90°. Orbits with lower

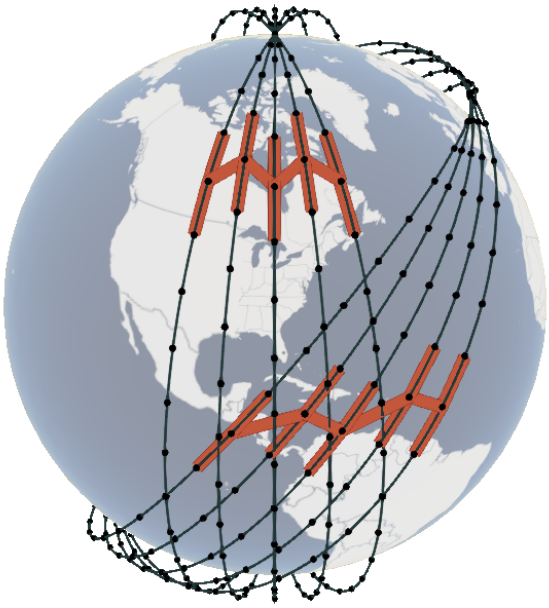


Figure 2.1: *Polar and inclined orbits with +Grid connectivity. An interactive 3D visualization is also available online [42].*

inclinations (prograde orbits) do not travel over the poles, spending more time at lower latitudes. Fig. 2.1 shows 5 polar and 5 non-polar orbits. As shown, for both types of orbits, the density of satellites increases away from the Equator, simply due to the involved geometry.

In principle, the inclination, i, can range from 0° to even higher than 90° for retrograde orbits. Sun-synchronous [43] orbits are retrograde close-to-polar orbits such that the satellite is synchronous with respect to the sun and come back to the same Earth location at the same time every day, thus finding interesting satellite imagery applications. Note, that some of the LEO constellations [44, 45] have limited sun-synchronous deployments planned, thus hinting at their possible usefulness also in broadband offerings. On the other end of the range, it is desirable to avoid too-low inclinations for two reasons: (a) needing to avoid interference with GEO satellites which operate over the Equator thus limiting the operations of

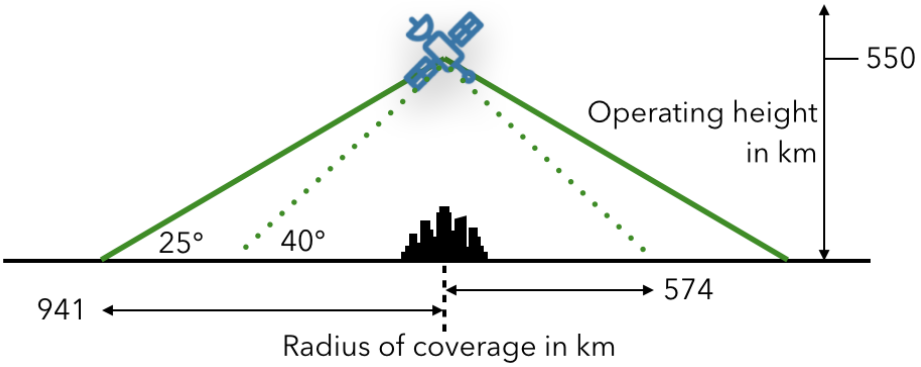


Figure 2.2: *The radius of coverage depends on operating height and minimum angle of elevation. While the schematic does not show this, the angles are correctly computed considering Earth's curvature.*

such LEO satellites [36]; and (b) inclinations lower than the latitude of the launch locations require additional maneuvers for placing in orbits, which is expensive [46]. For SpaceX, the current southern-most launch site is at Boca Chica, Texas [47] at a latitude of 25.99°N.

Height, *h*: The height of a satellite's orbit is measured from Earth's surface. For LEO satellites, $h \leq 2{,}000$ km. A satellite's height also determines its velocity and orbital period. Satellites around the heights specified by upcoming LEO constellations complete an orbit in ∼100 minutes, and travel at ∼27,000 km/h, more than 20× the speed of sound in air.

A satellite's height is an outcome of a complex set of operational and design choices, and technological barriers: lower height facilitates fast de-orbiting and orbit maneuvering after initial testing and reduces radiation hazards while increasing the effects of atmospheric drag. To counteract higher drag, orbit-raising technology has to be deployed. The operating height, h, is subject to regulation by the FCC and ITU.

Minimum angle of elevation, *e*: This parameter is not specific to an orbit, but rather to connectivity. A satellite can only communicate with

ground stations that can see it sufficiently above the horizon. Specifically, the elevation of the satellite in the sky must be higher than the minimum angle of elevation, e, from a ground station, for connectivity to be possible. For instance, as shown in Fig. 2.2, for Starlink, initially $e = 25°$, to provide higher coverage with a smaller number of satellites, while $e = 40°$ later, once enough satellites are deployed [48].

Together, h and e determine the coverage cone of a satellite, as illustrated in Fig. 2.2. Larger h and smaller e increase coverage. The trade-off is that larger h results in proportionally higher latency, while lower e leads to greater path loss for connections over longer radio links. Another downside of lower e is potentially increased interference, but we assume that this is not a substantial issue. This is reasonable given the steps [48, 49] taken by large LEO constellations to avoid interference: large number of satellites visible from each ground station, lower operating heights to reduce beam contours, steerable narrow spot-beams which can be re-assigned bands or allocated wider bands if needed, prioritization of higher angle of elevation, in-orbit traffic re-routing, etc.

Orbit and satellite placement: Globally spanning constellations use orbits that are equidistant from each other, and also uniformly space satellites within an orbit. For specifying the trajectory of a satellite, 7 orbital elements [50] (including an epoch and the 6 Keplerian elements) need to be specified. One can uniformly vary the right ascension of ascending node (RAAN) to create different orbital planes, and the mean anomaly (MA) to position satellites within the same plane. Eccentricities are all set to 0, such that orbits are circular. For zero-eccentricity orbits, perigee formally occurs at the ascending node (the point where the satellite crosses the equator while traveling from the Southern to the Northern hemisphere); the argu-

ments of perigee (AP) are thus set to 0. The mean motion ($2\pi/P_{\text{orbit}}$) varies according to the height of the satellite, h (which defines the orbital period P_{orbit}), which we discussed earlier. We also discussed inclination, i, separately above.

phase offset, *p*: The relative position of satellites in adjacent orbits is specified in terms of the p. Prior work [51] explores this parameter in the context of intra-shell collision probability. We do not address this in this work and instead assume that deployed LEO satellites will be equipped to autonomously dodge collisions [52, 53]. For this reason, we choose a fixed value, $p = 0.5$, across all experiments to spread satellites most uniformly across a shell. Uniformity maximizes the coverage over time for each constellation, which is a desirable characteristic.

2.2 What describes an LEO constellation?

The most generic model[1] of a satellite has 3 parameters: height h, inclination, i, and minimum angle of elevation e. In theory, these *could* be configured independently across the thousands of satellites planned for the largest constellations. However, this has a large downside for connectivity: when ISLs are deployed, connectivity between satellites traveling in arbitrary trajectories with respect to each other will be necessarily short-lived, requiring frequent hand-offs that can take tens of seconds [54]. Thus, for ISLs to be useful in providing long-distance end-to-end low latency connectivity, sets of satellites must travel in cohorts or shells. Of course, satellites may still be organized into multiple such shells. A fortunate side-effect of this is a reduction in the design problem's complexity, with design decisions needed only at the scale of shells instead of satellites.

[1] All broadband LEO networks currently under deployment use circular orbits.

Table 2.1: *The number of orbits, o; the number of satellites per orbit, n; the inclination, i; and operating height, h; for different shells of Starlink, Kuiper, and Telesat. We shall frequently refer to the first shell for each constellation, $S1$, $K1$, and $T1$, in the text.*

	shell	o	n	i	h (km)
Starlink	$S1$	72	22	53°	550
	$S2$	72	22	53.2°	540
	$S3$	36	20	70°	570
	$S4$	6	58	97.6°	560
	$S5$	4	43	97.6°	560
Kuiper	$K1$	34	34	51.9°	630
	$K2$	36	36	42°	610
	$K3$	28	28	33°	590
Telesat	$T1$	27	13	98.98°	1,015
	$T2$	40	33	50.88°	1,325

We assume that a constellation C consists of s **shells**. Each shell is a set of o **orbits** with the same inclination, i. Each orbit in a shell has the same **number of satellites**, n. Satellites sharing an orbit travel at uniform separation from each other; and orbits sharing a shell cross the Equator at equal separation from each other. These constraints ensure that satellites in adjacent orbits travel in the kinds of long-lived ISL-capable cohorts described below in §2.2.1.

To add concrete numbers to the above abstract description of LEO constellations, we describe the design parameters for the largest three proposed constellations, summarized in Table. 2.1.

SpaceX Starlink: Table. 2.1 details the first phase of Starlink, with 4,409 satellites planned across 5 orbital shells [36, 48, 55, 56]. SpaceX is currently deploying S1, with 1,584 satellites (72 orbits, each with 22 satellites), $h =$ 550 km, and $i = 53°$. The minimum elevation, $e = 25°$. S1 will cover most of the world's population, but will not extend service to less populated

regions at high latitudes. This coverage issue will be addressed by the higher inclination shells, S3-S5. SpaceX's stated plan is to deploy more than 42,000 satellites, but it is unclear how much of this is posturing to secure spectrum [57].

Note that, at the time of writing this dissertation, SpaceX is actively changing Starlink design parameters. While a past FCC filing [48] specified S1 with 24 orbits and 66 satellites per orbit, this has recently changed [44, 56] to the one given in Table. 2.1.

Amazon Kuiper: Kuiper plans three shells, with a total of 3,236 satellites at slightly different operating heights [58–60]. Kuiper entirely eschews connectivity near the poles, with all its shells having inclinations under 52°. The FCC filings mention a few possible values of e: "20(min)/30/35/45" [58].

Telesat: Telesat plans two shells with a total of 1,671 satellites [45], roughly a fifth of which will cover the higher latitudes, using an inclination of 98.98°, with the rest focused on improving capacity at lower latitudes. Telesat plans $e = 10°$, but the feasibility of this is unclear — unlike Starlink and Kuiper, whose filings detail how to address beam contour and antenna gain changes for e, Telesat's filings thus far omit such information.

While constellations would eventually consist of multiple shells, the current deployments target the first shells. Also, we believe that the problem setting is so new that it calls for separate analyses of individual shells. Shells could operate independently of each other still providing end-to-end connectivity by leveraging a combination of radio up/down links and inter-satellite laser connectivity, as we discuss next. The dynamicity inherent in these LEO networks is manifested in single shells without loss of generality, and hence this dissertation primarily focuses on single shells.

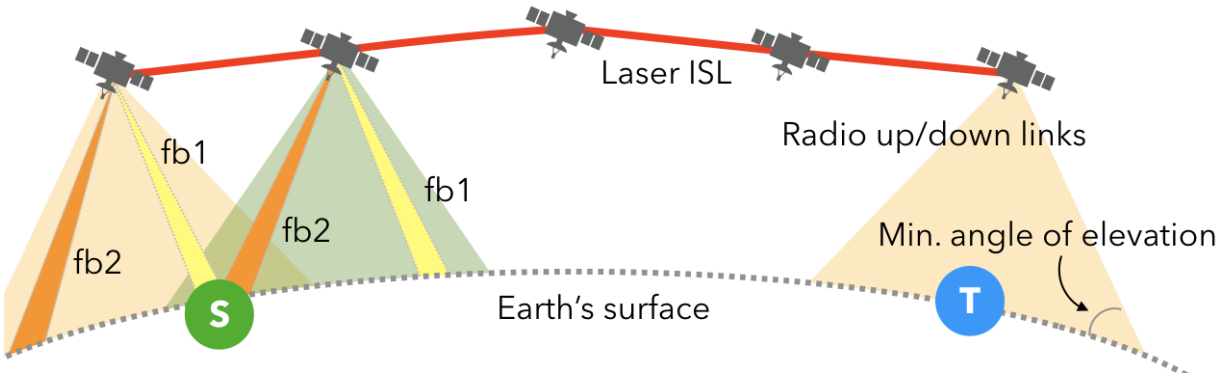


Figure 2.3: *Each satellite covers a cone defined by the minimum angle of elevation, e. A satellite uses steerable beams of different frequency bands (e.g., $fb1$, $fb2$) to connect to different GSes.*

2.2.1 Constellation connectivity

Now we describe the ground-satellite and satellite-satellite connectivity in LEO networks.

Ground-satellite links (GSL): Satellites can connect to ground stations (GS) using bidirectional radio links, as shown in Fig. 2.3, which are more resilient to cloud cover than lasers. We model a satellite's GSL as having a fixed bandwidth. According to Kuiper's FCC filings [49], each satellite will have multiple antennas, with each antenna supporting multiple steerable beams ($fb1$ and $fb2$ in Fig. 2.3); the beam steering and frequency band allocation will be software-defined, with the goal of maximizing throughput. Whether each GS can also simultaneously connect to multiple satellites depends on the type of GS: a user terminal uses a single phased-array antenna, while an enterprise user or gateway terminal uses multiple parabolic antennas with more flexibility [49].

A satellite's GSL bandwidth is shared among ground stations within the satellite's field of coverage. In order to accommodate free space path-loss,

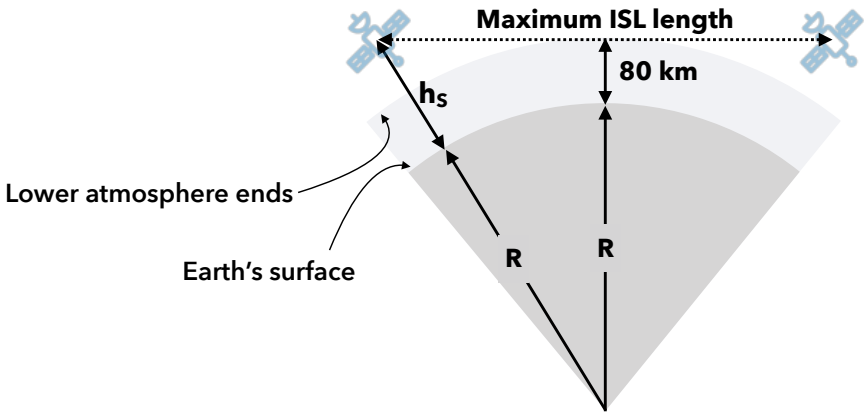


Figure 2.4: *An ISL must not enter the Mesosphere. Thus, given the altitude of satellites, one can easily calculate the maximum ISL length.*

the maximum bandwidth that can be assigned to a ground station at a line-of-sight distance of d_s from a satellite operating at a height h is modeled as h^2/d_s^2. In order for a ground station to receive an entire 1 unit bandwidth from a satellite, it must be located at the nadir of the satellite, with no other terminals in the field of view of that satellite. For Starlink, each satellite is claimed to have 20 Gbps downlink [61] GSL capacity.

Inter-satellite links (ISL): Inter-satellite links or ISLs, as shown in Fig. 2.3, are expected to be laser due to the extremely low beam divergence properties and narrow beam widths, minimizing interference issues. A satellite may connect to other satellites in its range using ISLs. Existing non-GEO constellations like Iridium [20] use and Starlink's filings [36, 48] mention a grid-like approach for their ISLs, considered standard in the industry and related academic literature [62]: each satellite has 4 bi-directional ISLs with its nearby neighbors, 2 in the same orbit, and 2 with immediate neighbors in the 2 adjacent orbits. Such connections can be maintained for long periods, as the connected satellites have low relative velocities, except at higher latitudes. Minor variants of such connectivity

are possible, *e.g.,* to a satellite with a positive or negative phase shift in the adjacent orbit; one variant is shown in Fig. 2.1. For convenience, we refer to this connectivity pattern as **+Grid**, reflecting its shape. There is a common, implicit assumption that all new constellations that support inter-satellite connectivity will use the same approach. This is reflected in visualizations and analyses of connectivity in these constellations [51, 63–67]. However, a simple (standard) calculation shows that this view is unnecessarily restrictive. The key visibility constraint is to avoid the ISL entering the atmosphere any lower than the Thermosphere [68]. This is the lowest atmospheric layer devoid of water vapour, and starts at ∼80 km above Earth's surface. Thus, the minimum clearance of an ISL above the Earth's surface should be 80 km, as illustrated in Fig. 2.4. Given these constraints, it is trivial to calculate the maximum ISL range, *e.g.,* for satellites operating at an altitude of $h_S = 550$ km (Starlink's $S1$ altitude), the maximum ISL length can be calculated as $d_{ISL} = 5{,}014$ km.

For small constellations with tens of satellites, this range constraint limits ISLs to satellites in the same or adjacent orbits. However, for the denser constellations being proposed, there are many more possibilities: for a uniform 40^2 constellation at 53° inclination and 550 km altitude, a satellite at the Equator could potentially set up ISLs with 190 others.

There is, of course, another constraint: power, and relatedly, equipment size and mass. For fixed desired link characteristics, longer distances require more transmission power, and larger and heavier equipment, which can be expensive in satellite systems.

These constraints will later be important in the context of LEO constellation topology design. While visibility is the likely limiting constraint, we also analyze in Chapter. 4 settings where range is limited by these other

constraints.

Bent-pipe versus end-to-end connectivity: A constellation can also operate without ISLs, using "bent pipe" connectivity, with data either exiting at the nearest base station (connected to the Internet) or bouncing up and down through satellites and ground stations. Both result in higher latency [69, 70] than those possible with long-distance line-of-sight laser [51, 71]. In Chapter. 3 we quantify the challenges of long-distance bent-pipe connectivity. While Starlink has announced successful launching of 10 satellites with ISL capabilities [72, 73], we still await more information on this at the time of writing this dissertation.

2.2.2 System dynamics

Moving from GEO to LEO satellites lowers latency, but it makes the constellations highly dynamic. At a 550 km altitude, a satellite travels at 27,306 kmph, covering in a minute distances comparable to those between Munich–Berlin, Delhi–Lahore, or San Jose–Los Angeles and completing an orbit around the Earth in ∼100 minutes [54]. As satellites travel fast across GSes, GS-satellite links can only be maintained for a few minutes, after which they require a handoff. ISLs also continuously change in length. The Earth's shape and orbital geometry results in satellites in different orbits coming closer at higher latitudes. This results in a continuous change in their relative positions and velocities and hence the ISL lengths and latencies. Thus, which satellites connect to each other, and to which ground stations (GSes) evolves over time. The changing distances between satellites, their relative velocities, and the technology used, determine which connections are feasible, and how fast they can be setup.

The end-end path between two GSes thus changes both in terms of which

satellites are involved, and in terms of the lengths of both the GS-satellite links and the ISLs.

Mobility is, of course, well-studied in a variety of contexts, including cellular networks, high-speed trains, drones and airplanes, and swarms of mobile nodes. For many of these settings, there are also models of mobility, together with simulation and analysis infrastructure. However, LEO satellite mobility is unique for several reasons:

- LEO mobility features much larger distances and velocities than terrestrial mobile networks.
- Unlike most other settings, LEO networks' core infrastructure itself is mobile, rather than just the end-points.
- LEO mobility is predictable; this is not the case for the most well-studied setting, cellular networks.
- LEO networks feature thousands of network switches (satellites) capable of providing Tbps of connectivity. This scale is far beyond other networked swarms.

Each previously well-studied setting features one or two of the above characteristics, but not all of them. For instance, trains, and to a lesser extent, airplanes, also feature predictable motion, but none of the other characteristics.

2.2.3 Impact of interference

Before we end our background discussions on LEO constellation connectivity, let us briefly mention various interference mitigation strategies followed by the proposed LEO constellations.

Satellite constellations deploy multitude of interference mitigation techniques in order to maximize throughput offered per satellite. Well known schemes include dividing the terrestrial area of coverage into tessalating shapes or spots, and then reusing frequencies by assigning adjacent spots different non-overlapping frequency bands. Starlink and Kuiper have extensive plans [36, 49] beyond such simple mitigation, which include (for either or both):

- Operating at low heights to reduce beam contours.
- Many steerable, shapeable beams with varying bandwidth. Starlink plans to deploy spot beams as narrow as 1.0° to 1.5° in order to minimize interference.
- Multiple antennas per satellite, each with multiple beams.
- Each spot can be served by multiple beams or satellites at non-overlapping frequencies. Starlink takes a step further and allows spots being served by the same frequency if the angle of separation is 10° or higher.
- Beam splitting and merging in order to address interference as well as spatially varying demands.
- Prioritizing higher angle of elevation, if need be, in order to reduce chances of interference.
- In-orbit traffic re-routing in order to use fewer beams.

Given this broad set of online, software-defined mitigation techniques, throughout this dissertation we assume negligible impact from interference.

2.3 Low-latency opportunity

Large latencies in today's Internet infrastructure are partly explained by poor use of existing fiber infrastructure: two communicating sites often use a longer, indirect route because their service providers do not peer over the shortest fiber connectivity between their locations. We find, nevertheless, that even latency-optimal use of *all* known fiber conduits, computed via shortest paths in the recent InterTubes dataset [74], would leave us 1.93× away from c-latency. This gap stems from the speed of light in fiber being $\sim\frac{2}{3}c$, and the unavoidable circuitousness of fiber routes due to topographic and economic constraints of buried conduits.

LEO satellite infrastructure is aimed at operating at the speed of light in air (∼vacuum) leveraging a combination of radio and laser. Although there is a latency cost of sending bits from the terrestrial plane to the satellite plane, over long distances, not only this cost is compensated but also much lower communication latencies could be achieved by operating at c-speed.

The remaining part of this chapter explores this LEO low latency communication opportunity, while Chapter. 3 focuses on the LEO networking challenges, thus addressing our goal:

Goal A: Identifying the network performance opportunities and challenges.

Although we focus more on the low latency aspects of network performance, the topology design work in Chapter. 4 takes into account end-to-end path hop-count and hence congestion, and Chapter. 5 quantifies further performance aspects like RTT variations, routing and traffic engineering challenges, and transport performance. In that sense, Goal A of this thesis is intertwined with the remaining goals.

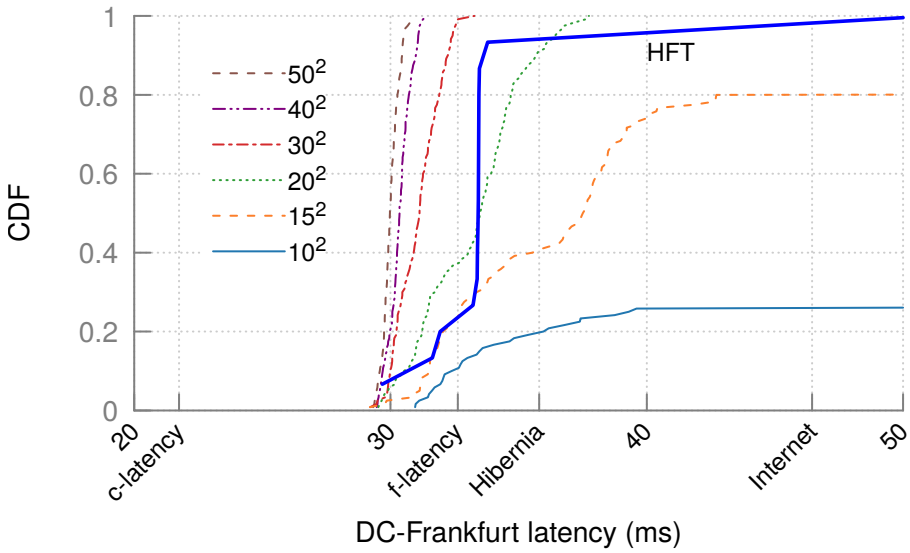


Figure 2.5: *LEO constellations of suitable density can achieve sub-fiber latencies over long distances. They can even beat trans-Atlantic latencies seen in the latency-obsessed HFT industry.*

2.3.1 Estimating end-to-end latencies in LEO

With orbits for all satellites specified, together with inter-satellite links, we can estimate at any instant, the latency between two different ground locations using this network. We compute these estimates at a granularity of 1 minute over a period of 2 hours. For each minute, we consider the topology to be static. We identify the satellites visible from the 2 target ground locations and compute the shortest path between them through the satellites using Dijkstra's algorithm. We translate the computed distance to latency assuming data transmission at the speed of light in vacuum (and ignoring error correction and other overheads).

Fig. 2.5 shows the latency between Washington, D.C. and Frankfurt for different constellation sizes. We vary constellation sizes in $\{10^2, 15^2, \ldots, 50^2\}$, with a constellation of size o^2 using o orbits, each with $n = o$ satellites. For clarity, Fig. 2.5 shows results for a subset of these constellations. To give the appropriate context, Fig. 2.5 also includes the latency between the same

locations over today's Internet, 46.4 ms, as reported in WonderNetwork's global ping statistics [75]; the latency when using the GTT (Hibernia) Express trans-Atlantic cable [76], 35.8 ms; f-latency, *i.e.,* the best latency achievable were a fiber cable laid along the geodesic between the same locations, 32.6 ms; and c-latency [77], *i.e.,* the fundamental latency limit, achievable if the geodesic were traversed at the speed of light in vacuum, 21.7 ms.

As the results show, even the relatively small 30^2 constellation can (almost always) achieve latencies better than the best possible with fiber. The median path uses 12 satellite hops, but this could potentially be reduced with a different ISL configuration than the simple one we tested. Denser constellations, as expected, can not only achieve lower latencies, but also reduce the variation. Sparse constellations experience periods where the two locations are disconnected.

2.3.2 Beating today's bleeding edge

For the Frankfurt-DC segment, our estimates suggest that dense LEO satellite networks could achieve latencies 35% lower than today's Internet, and 16% lower than the best available (and costly, using the Hibernia cable) fiber connectivity. Even the faster Hibernia cable, however, is not at the bleeding edge of minimizing latency. While high-frequency traders are already well known [78] to have achieved sub-fiber latencies on certain *intra*-continental routes, how low are trans-Atlantic latencies in this latency-obsessed industry? How would satellite networks compare to their latencies?

We can estimate trans-Atlantic Frankfurt-DC latency in the HFT industry by examining trading data. The key premise is that certain economic news

triggers trading activity, and is transmitted from its source to financial centers over the fastest available connectivity. Thus, the timing of the news release and the trading at financial centers reveal the lowest available latency between these locations.

We use US Bureau of Labor Statistics (BLS) non-farm payrolls estimates, released in Washington **DC** at 8:30 AM ET on the first Friday of each month. The trade timings we use are for (a) the E-mini S&P 500 Futures (ES) which trade at the CME (Chicago Mercantile Exchange data center located in **Aurora**, Illinois); and (b) the Euro-Bund Futures (FGBL) which trade at Eurex (in **Frankfurt**, Germany). We assume that the BLS news is neither known nor traded on in advance, and that the trade timestamps are accurate at the ∼10-100 μs level (for regulatory compliance).

The time differential between Aurora and Frankfurt trading activity, Δ_{AF}, can be inferred with high confidence from uniquely identifiable trading bursts after the BLS news. Given that DC-Aurora and DC-Frankfurt news transmissions begin simultaneously, if we can estimate DC-Aurora latency, L_A, we can estimate the DC-Frankurt latency as $L_A + \Delta_{AF}$.

The DC-Aurora locations are 1,004.52 km apart (*i.e.,* minimally, 3.35 ms). We estimate $L_A = 4$ ms, based on the reasonable assumption that HFTs use similar networks here as in other previously analyzed intra-continental segments [78].

We estimated DC-Frankfurt latency for 15 events, each corresponding to a BLS news announcement during Q1-2 2016 [79]. Fig. 2.5 includes the resulting CDF of these 15 estimates. Some of the observed latencies beat the best achievable with fiber; speculation is that opportunistic short wave radio communications are used [80], which would explain these measurements. But regardless of the method, the measurements establish that networks

with latency lower than even the hypothetical ideal fiber are already being used in niche deployments even across the Atlantic divide. Even more interestingly, satellite constellations smaller than those planned can match or improve on this tighter baseline, thus beating today's bleeding edge in terms of latency.

2.4 Benefits of low-latency

By now, we see that LEO constellations could offer long-distance low-latency connectivity, also bridging the continental divides which otherwise rely on fiber. But, how crucial is the role of low latency in improving the quality of experience (QoE) of Internet applications?

User experience in many interactive network applications depends crucially on achieving low latency. Applications focused on user interactivity, such as augmented and virtual reality, tele-presence and tele-surgery, music collaboration over long-distances, etc., can all benefit from low-latency network connectivity. Likewise, less visible and user-centric applications, such as real-time bidding for Web page advertisements and block propagation in block-chains, would also benefit from a low-latency network. Even seemingly small increases in latency can negatively impact user experience, and, subsequently, revenue for the service providers: Google, for example, quantified the impact of an additional 400 ms of latency in search results as 0.7% fewer searches per user [81]. Further, wide-area latency is often the bottleneck, as Facebook's analysis of over a million requests found [82]. Indeed, content delivery networks present latency reduction and its associated increase in conversion rates as one of the key value propositions of their services, citing, *e.g.,* a 1% loss in sales per 100 ms of latency for Amazon [83].

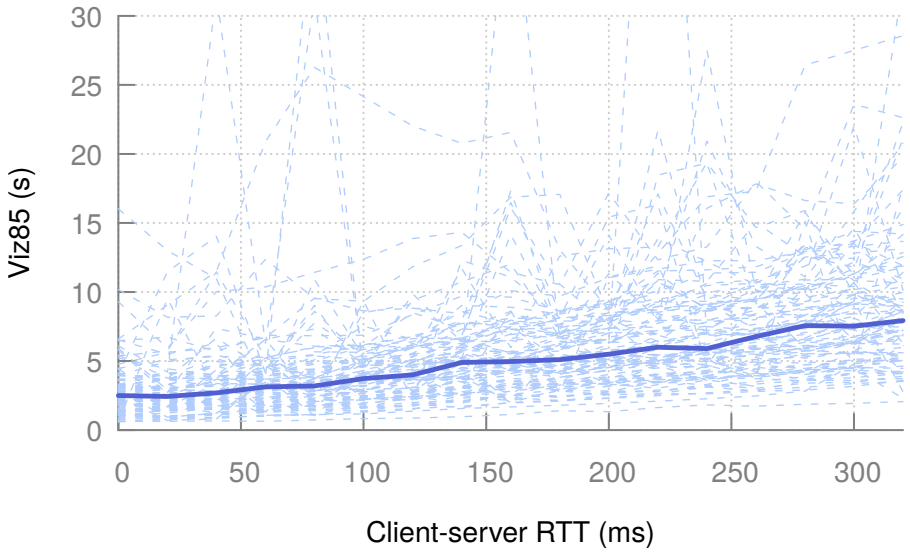


Figure 2.6: *As client-server RTT increases, Web page visual completion time (*85%*) increases linearly, with every* 10 *ms of RTT increase, adding* 186 *ms in the median. The individual dashed, light lines are for individual pages, and also show the linearity, albeit with variations for some pages, and with different slopes. The solid line represents the medians.*

While it is beyond (and orthogonal to) the scope of this work to analyze this in significant detail, we now assess, in a simplified environment, the user experience improvements lower latency could achieve for Web browsing.

2.4.1 Impact of latency on Web browsing

In order to demonstrate the impact of latency on Web page visual completion times, we deploy a Web client in Azure's US WEST 2 data center and fetch Web content that could be reached with negligible latency.

In order to identify Web servers deployed in or near the Azure's US WEST 2 data center, we quantify round-trip latency to Web services from this data center. We measure round trip times to Web servers hosting the most popular Web sites (per Alexa's list [84]) in May 2018 and identify 100

servers within negligible latencies (less than 10ms) from the client. We use `hping` [85] to conduct our RTT measurements, allowing us to send TCP SYN packets to the Web servers and record when the TCP SYN-ACKs were received at our Amazon nodes.

For the shortlisted 100 servers, we investigate the dependence of 85% visual completion times (viz85, indicative of when most of the visual content is populated) on client-server RTTs. The most frequently cited work on the impact of increasing RTT is Mike Belshe's 2010 measurement of 25 popular Web pages [86]. Others have also quantified the relationship between measured *last-mile* latency and page load times [87, 88] over small numbers of pages (less than 10). We provide fresh measurements of this using our setup, which allows tight control of latency starting from nearly zero (under 10 ms). Note that this kind of measurement is only made possible by our experimental setup allowing us to place the client in close proximity of the servers, with other setups starting from the already significant latencies they observe. While record-and-replay tools could also be used to produce such results, they often add significant inaccuracy (*e.g.,* 8% in the median with MahiMahi [89]). In our setup, bandwidth and latency can be controlled at the client which runs the Chrome browser on Ubuntu virtual machines with 2 cores and 8 GB memory. For automating Web page loads and recording performance metrics, we use `sitespeed.io` [90].

Page load time or PLT depends heavily on ad content (*e.g.,* when the network RTT is 160 ms, enabling and disabling ad blocking changes median PLT by 8%, while there is no change in viz85). Thus, we only present results on viz85. Fig. 2.6 shows for each of the 100 pages tested, how viz85 increases nearly in linear fashion with RTT (with bandwidth fixed to 10 Mbps). The regression-based best-fit (over the medians at each RTT value, with RTTs

being in seconds) is:

$$t = 18.6 * RTT + 1.9 \tag{2.1}$$

Thus, for every 10 ms of increase in RTT, (median) viz85 time increases by 186 ms. (Faster compute affects the constant in that equation but not the linear factor.) Of course, there is substantial variation across pages, as shown by the individual lines in the plot, with some pages incurring substantially more RTTs than others. The large linear factor indicates how important the role of lower latency is in improving Web browsing experience. Better page load times translate to larger traffic and higher revenue as discussed above.

2.5 Related publications

The plots and the corresponding discussions in this chapter have been taken from the following publications:

- D. Bhattacherjee, M. Tirmazi, and A. Singla, "Measuring and exploiting the cloud consolidation of the Web," *arXiv:1906.04753*, 2019, Preprint
- D. Bhattacherjee, W. Aqeel, I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Gearing up for the 21st century space race," in *ACM HotNets*, 2018

3

Low-Earth orbit networking challenges

Having gone through the LEO communication opportunities, in this chapter we discuss the various LEO networking challenges that need to be addressed. We first quantify the utility of having ISLs versus using long-distance bent-pipe (BP) connectivity (bouncing network traffic between the terrestrial and satellite planes) for low-latency LEO communications. Then we discuss topology design, routing and traffic engineering, and end-to-end congestion control challenges in the light of LEO dynamics.

3.1 Is BP connectivity enough?

Starlink, Kuiper, and Telesat all claim in their regulatory filings that their constellations will feature high-bandwidth laser ISLs [45, 49, 55]. This would enable moving data in space across a series of satellites, along nearly the shortest path, and at the speed of light in vacuum, c. Unfortunately,

despite the public claims thus far, there is still uncertainty about if and when all of the new constellations will be able to use ISLs.

One hurdle is the "burn on reentry" requirement that regulators are asking operators to satisfy. Given the large number of proposed satellites, operators are being asked to ensure that *every* component burns up during reentry to the atmosphere, thus not risking injury and damage from de-orbiting satellites. However, the silicon-carbide components that are often used in the mirrors for ISL equipment [51] do not satisfy this requirement.In order to comply, SpaceX's Starlink announced [92] elimination of those components during their first few deployments to ensure complete "burn on reentry" of satellites. Only recently, in January 2021, Starlink deployed 10 LEO satellites with ISL capabilities, although no information on their capabilities is publicly available yet. There is still uncertainty about the incorporation of ISLs in other constellations, while OneWeb plans to deploy their constellation without ISLs.

We thus attempt to quantify the utility of ISLs for such constellations: how important are ISLs to the capabilities of such networks, and how do the properties of LEO constellations with and without ISLs differ? Without ISLs, transoceanic distances must be bridged in some other manner than on-land ground stations; following the lead of recent work [93, 94], we use in-flight aircraft to serve as relays in such settings. We then analyze the latencies for end-end network paths and their variations over time, attenuation due to weather, and network throughput under intuitive traffic scenarios.

While prior work argued [69] that absent ISLs, such networks still provide low latency, our findings are more mixed. Indeed, LEO networks without ISLs still provide low latency between many ground locations. However,

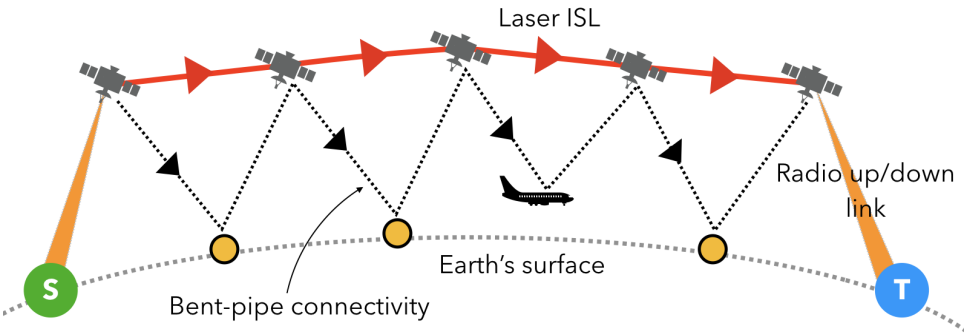
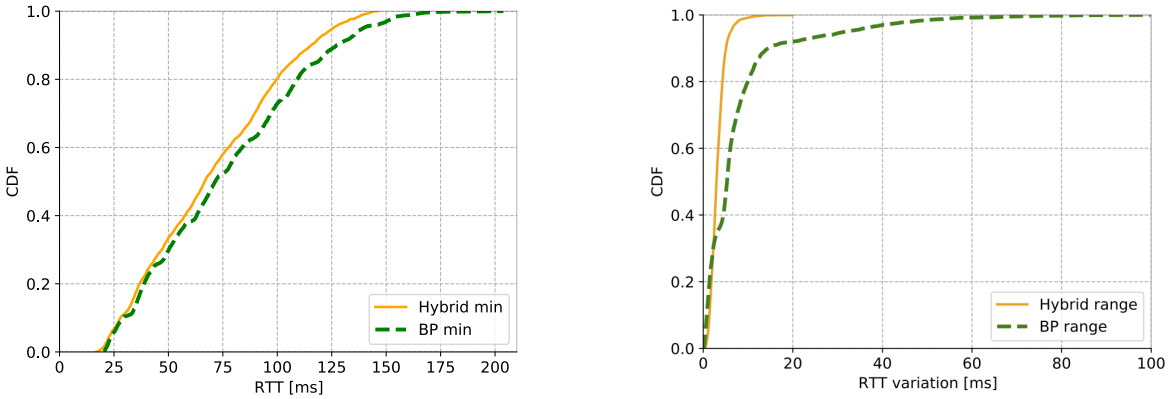


Figure 3.1: *ISL-path (solid) versus zig-zag BP path (dashed). Smaller circles and an aircraft represent ground stations (GSes).*



(a) (b)

Figure 3.2: *Incorporating ISLs result in (a) lower and (b) more stable RTTs over time than with BP-only connectivity. (a) plots the minimum RTTs seen across city pairs, while (b) plots the RTT variations (i.e., max-minus-min RTT) seen across city pairs.*

we quantitatively show that even with dense ground station deployments, there are significant downsides to foregoing ISLs.

3.1.1 BP versus ISL connectivity

LEO networks do not need ISLs to provide service. Without ISLs, connections between far-separated GSes bounce up-and-down between satellites and on-path GSes, yielding BP connectivity. Fig. 3.1 shows BP and ISL connectivity. Besides the natural sources and destinations of traffic as GSes,

BP requires additional GSes to *transit* traffic. In particular, across large water bodies, we allow this approach to use in-flight airplanes as transit GSes.

With exclusively ISL connectivity, the first radio hop between the source GS and a satellite is followed by a series of laser ISLs, and a radio last hop to reach the target GS. Satellites are the only intermediate hops between the two communicating GSes. A *hybrid* approach uses both ISL and BP connectivity in end-end paths.

Our goal is to assess the utility of ISLs by comparing networks that are restricted to only BP connectivity to hybrid ones that additionally feature ISLs. A recent effort [69] suggested that BP with a dense-enough deployment of GSes could achieve latencies comparable to constellations with ISLs. Another effort [63] coarsely compared the throughput with and without ISLs, using an extremely lax model, where traffic entering the constellation could exit *anywhere*, treating the entire network as one maximum flow instance with many sources and *one* large sink, instead of imposing any constraints on the destinations of traffic flows. The latter effort also did not account for the possibility of a dense relay deployment, using only tens of terrestrial gateways.

Thus, prior work does not yield a full, clear picture on the utility of ISLs. To address this issue, using models more in line with networking realities, we quantify three network properties with and without ISLs: latency and its variability, network-wide throughput, and resilience to weather. We also discuss other differences that can impact BP versus ISL connectivity.

Traffic matrix: The source/sink GSes are located in the 1,000 most populous cities in the world [95]. We only allow traffic between city-pairs separated by more than 2,000 km apart along the geodesic. This minimum

distance constraint is used to model the fact that for most nearby city pairs, terrestrial connectivity will provide lower latency, while also being cheaper. Further, to keep the traffic matrix to a tractable size for simulations, instead of running traffic between all pairs of cities, we uniform randomly pick 5,000 city pairs.

Relays for BP: For a conservative view of the utility of ISLs, we use a dense deployment of GSes. The city-GSes serve as both traffic sources and sinks, as well as transit relays. GSes which only transit traffic are placed uniformly every 0.5° on the latitude-longitude grid within a radius of 2,000 km of the cities. This is the highest density of GSes tested in prior work arguing that BP could achieve low latency [69]. In addition to these GSes, to help BP achieve transoceanic connectivity, we use all in-air commercial aircraft as GSes. Note that aircraft-to-satellite radio links are already used in aircraft that offer satellite Internet to passengers. For the positions of such aircraft, we use FlightAware's data [96] for a period of 1 day from 2018. We include only those aircraft as possible intermediate hops which are flying over water bodies [97] to supplement the on-land GSes.

3.1.2 Latency and its variability

LEO satellite networks are highly dynamic. Due to the high velocity of satellites, end-to-end paths and their latencies change continually. We compare the impact of such variations on BP and hybrid connectivity on Starlink $S1$.

We simulate the networks for 1 day. At every 15-min snapshot, we find shortest paths between the 5,000 city pairs. Fig. 3.2 shows the minimum (across snapshots) RTTs and range of RTTs seen over time for both BP and hybrid networks. The min. RTT, in Fig. 3.2(a), is strictly lower for

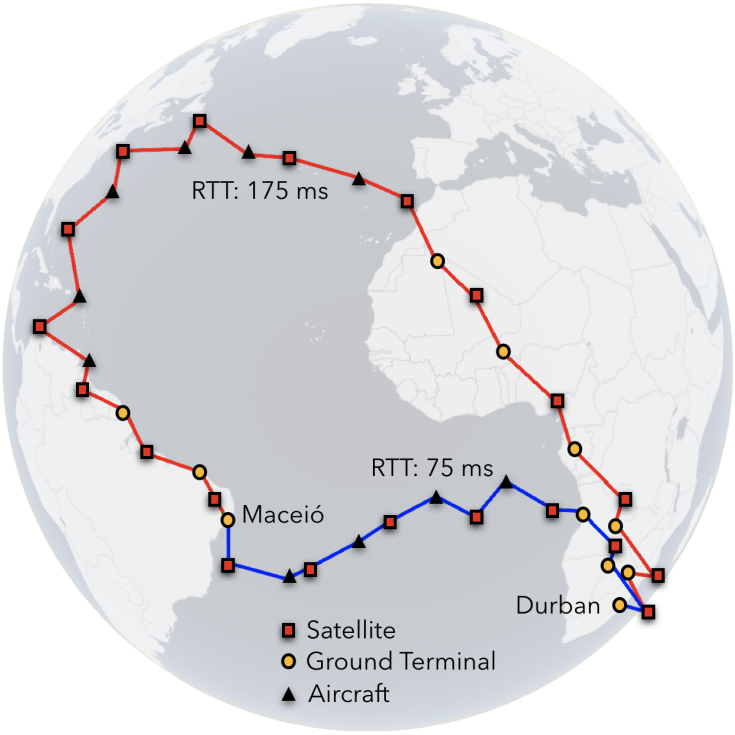


Figure 3.3: *The path between Maceió, Brazil and Durban, SA changes a lot depending on aircraft availability.*

the hybrid approach, as expected. Along the lines of prior work [69], the differences are small for most city-pairs. There are, however, substantial differences in the tail, the maximum difference being 57 ms. With exclusively BP connectivity, some paths see high latencies due to sub-optimal intermediate hops.

RTT variations reveal larger differences. Across time snapshots, we compute the max-minus-min RTT difference for each city-pair, and show the distribution across city-pairs in Fig. 3.2(b). The results show that latencies vary much more with BP. While with hybrid connectivity, the maximum range of RTTs across city-pairs over time is under 20 ms, with BP this range is as high as 100 ms. Thus, with hybrid connectivity, RTT stability improves by as much as 80%.

As Fig. 3.3 shows, with BP, the path between Maceió, Brazil and Durban, South Africa sees an inflation of 100 ms. This is because the density of air traffic is much sparser over the south Atlantic than the north. Hence, the path often ends up using aircraft flying over the north Atlantic as intermediate hops. Note that this behavior, as discussed above, not only inflates the RTT of this path significantly, but also makes the heavily used paths over the north Atlantic (due to busy routes between north America and Europe) even more congested.

RTT variations result in varying quality of experience for latency-critical applications. For instance, past work [98, 99] has shown how QoE in gaming deteriorates not only with higher latency, but also with higher latency variations.

3.1.3 Network-wide throughput

The network's throughput will determine how much revenue the operator can obtain while offering low-congestion connectivity. We thus compare the aggregate throughput offered by LEO networks with BP-only connectivity and hybrid connectivity. With BP, all traffic needs to be routed via up/down radio links, using up more constrained capacity at these links, instead of using higher-capacity ISLs. For experiments in this section, we use floodns [100] which simulates routed flows in a network.

We evaluate the throughput of both approaches on Starlink $S1$ and Kuiper $K1$. We first show results with each GS-satellite link having up- and downlink capacities of 20 Gbps, and ISLs of 100 Gbps, and later, with different ratios of these capacities. We use the same 5,000 city-pairs, and for each pair, we route over the k edge-disjoint shortest paths, with $k = 1$ and 4.

We find a max-min fair allocation of link capacities to flows in order to

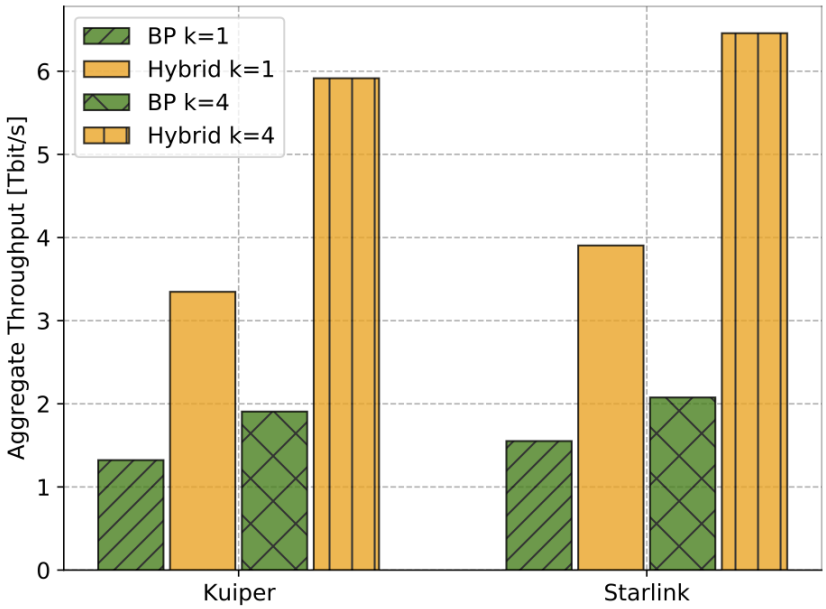


Figure 3.4: *Aggregate throughput over Starlink $S1$ and Kuiper $K1$ for* 5,000 *city pairs, with traffic sent along k edge-disjoint shortest paths per pair.*

find out the aggregate throughput of the system. Traffic between each city-pair uses k sub-flows, each along one of the k edge-disjoint shortest paths. These sub-flows are treated independently by the simple max-min fair-share algorithm [101], which iteratively and greedily finds the most congested link in the network, and shares the bottleneck link capacity fairly among the competing flows. Note that because of edge-disjoint paths used for sub-flows, sub-flows of one flow do not compete with each other. The exploration of superior routing schemes is left to future work.

Fig. 3.4 shows the achieved aggregate throughput. For only shortest-path routing ($k = 1$), the hybrid approach achieves more than 2.5× higher throughput than BP for both Starlink $S1$ and Kuiper $K1$, while with $k = 4$, this improvement is even larger, at least 3.1×. Also noteworthy, is that the improvement from using multiple paths, instead of just the shortest, is larger for the hybrid approach: 1.65× and 1.76× for $S1$ and $K1$, compared

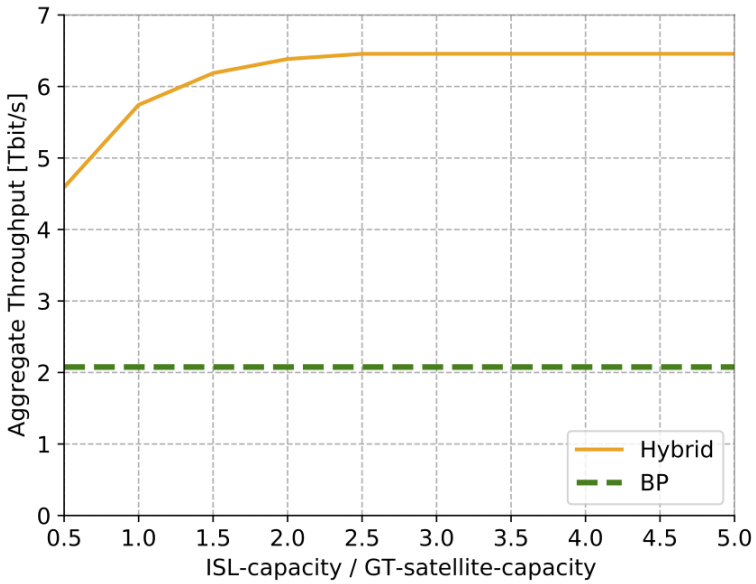


Figure 3.5: *Starlink S1's throughput with varying ISL capacities.*

to 1.34× and 1.44× for BP.

We also examine the impact of the relative capacities of GS-satellite links and ISLs. We fix the GS-satellite link capacity at 20 Gbps, and vary ISL capacity from 0.5×–5× of this. Fig. 3.5 shows $S1$'s throughput with $k = 4$. Even with an ISL capacity of 0.5× GS-satellite link capacity, the hybrid approach, with its greater path diversity, increases throughput by 2.2× compared to BP. With $k = 4$, the aggregate throughput does not improve for ISL capacities beyond 3×, this is an artefact of the routing scheme we use; with more efficient routing and traffic engineering, we can expect larger improvements overall, as well as a continued increase with higher capacity ISLs. A routing scheme that minimizes the maximum utilization, for example, can offer higher throughput, albeit at the cost of increased latency.

Besides the use of scarce GS-satellite capacity for transit in addition to only sourcing and sinking traffic, another reason BP fairs poorly on throughput

is that at any time, it is unable to utilize a large fraction of the satellites for networking at all. For Starlink $S1$, we find that across a day, the number of satellites that are entirely disconnected from the rest of the network varies between 25.1% and 31.5% of all satellites.

3.1.4 Resilience to weather

We use the *ITU-Rpy* [102] library to measure the attenuation for GS-satellite paths. The library implements ITU recommendations to model atmospheric attenuation due to rain, cloud, gaseous cover, and tropospheric scintillation in slant paths. Attenuation due to path loss is not considered, reflecting the assumption that the link design accounts for that.

For each of the 5,000 city pairs, we find the shortest paths using Dijkstra's algorithm. Next, we find the worst attenuation seen across all links in the path. Note that for BP paths, this is the worst attenuation seen across all links of the zig-zag path bouncing between the satellites and GSes. For paths consisting of ISLs, this value is either the first or last hop attenuation, whichever is worse. For calculating the atmospheric attenuation along BP and ISL paths, we use different up-link and down-link frequencies for Starlink $S1$ (14.25 GHz and 11.7 GHz respectively; Ku-band) which are within the ranges specified in their FCC filing [55]. For the ISL paths, we exclude GSes as intermediate hops, in order to quantify the maximum improvements in attenuation possible with ISLs. If such intermediate hops through GSes are used, the attenuation of the path would be the worst link attenuation across all the up/down radio links used along the path. Note that this model and the experiments in this section (exclusively) assume that the signal for BP-paths uses error correction and regeneration at each GS; otherwise, the multiplicative impact of attenuation would be

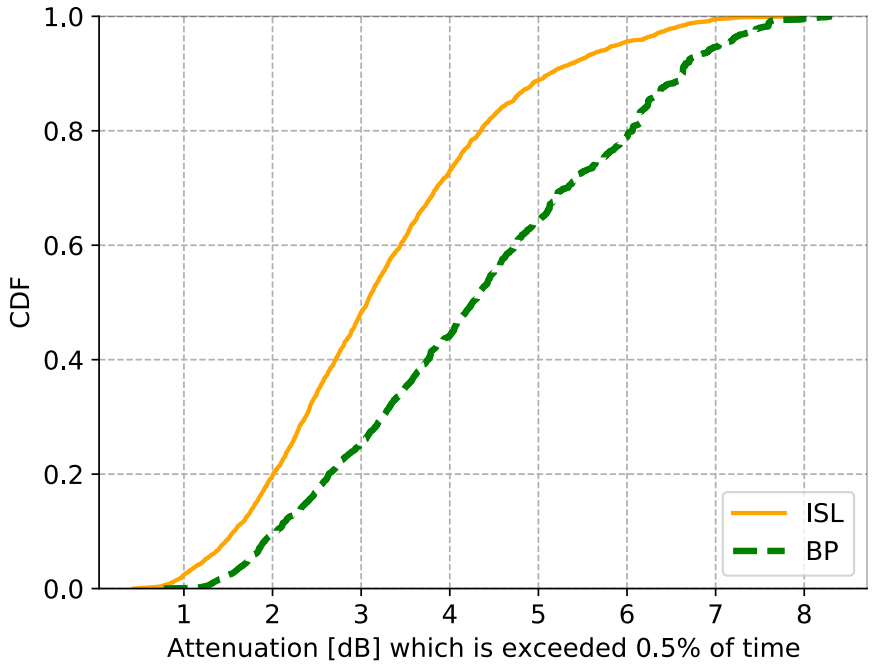


Figure 3.6: *Attenuation is much higher for BP connectivity.*

prohibitively high.

Attenuation across city-pairs: For each city-pair, we compute the 99.5$^{\text{th}}$ percentile attenuation across time. This percentile corresponds to more than 7 minutes a day, and almost 2 days every year. We compute a distribution (across city-pairs) of this 99.5$^{\text{th}}$ percentile attenuation. Fig. 3.6 shows that the attenuation is much higher for BP; the median with ISLs is more than 1 dB lower. This translates to an 11% reduction in received power. This number would be even higher for Ka-band communication (intended for use for larger terrestrial gateways), which is affected more by weather conditions [103]. Higher attenuation has to be dealt with by appropriate design for modulation and error correction schemes (MODCOD) [104], and trades off bandwidth for reliability.

Attenuation along one example path: For more insight into why BP suffers more from attenuation, we use an example end-end path between

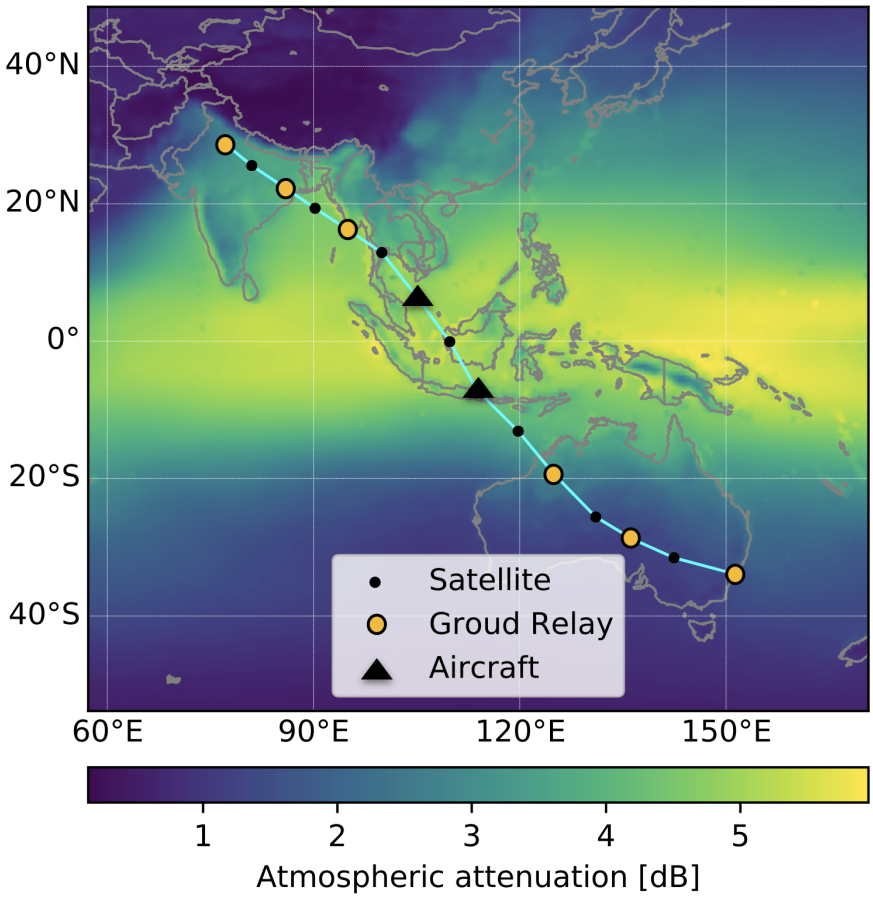


Figure 3.7: *Delhi-Sydney path with 2 aircraft and 4 GSes.*

Delhi and Sydney. Note that the Delhi-Sydney pair is not among the 5,000 randomly picked city-pairs. We pick this city-pair because the path between them covers the tropical region, which experiences high annual precipitation [105]. Fig. 3.7 shows a random time, at which the BP-path uses 2 aircraft and 4 on-land GSes as intermediate hops. The heat-map depicts 99.5^{th} percentile attenuation across south-east Asia. Although both end-points, Delhi and Sydney, are in low attenuation areas, BP ends up using intermediate hops in regions with higher attenuation. In contrast, the ISL path avoids this entire high-attenuation region. This is evident in Fig. 3.8, which plots the attenuation along the path. At least 1% of the time, the BP attenuation is 5 dB (44% reduction in received power on the affected link)

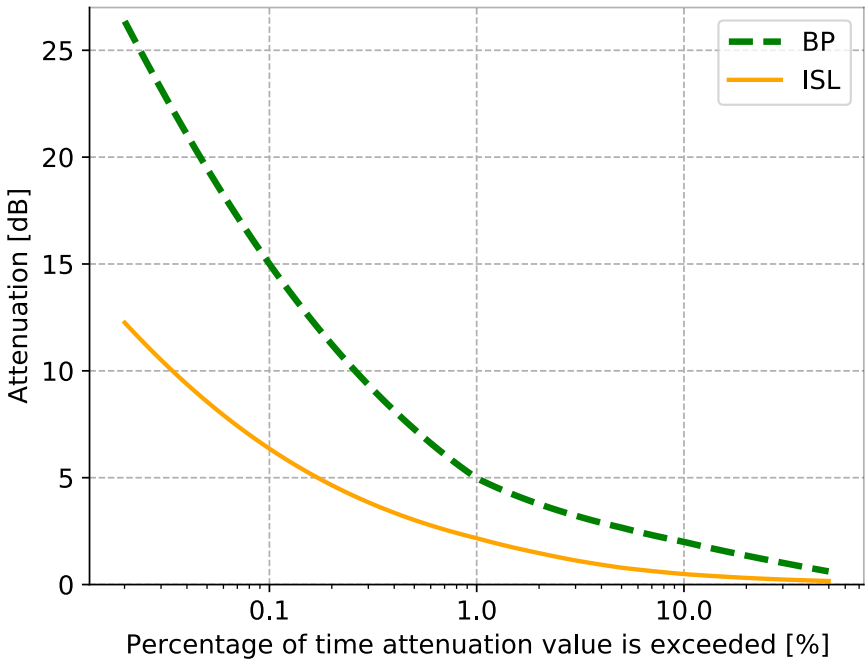


Figure 3.8: *Impact of Attenuation on Delhi-Sydney path.*

while ISL attenuation is 2.2 dB (32% reduction in received power on the affected link). Thus, ISL connectivity can reduce the attenuation due to weather by 39% (56% received power with BP versus 78% received power with ISL) at least 1% of the time.

3.1.5 Other benefits of ISLs

We quantitatively compared BP and ISL connectivity across three network properties, showing that constellations with ISLs would have a substantial edge on latency, throughput, and reliability. However, there are several other aspects where ISLs offer improvements, which we have not yet quantified.

Crossing unfriendly territory: BP connectivity between certain sources and destinations is bound to require GSes in countries and regions that an operator would like to avoid either because the topography is challenging

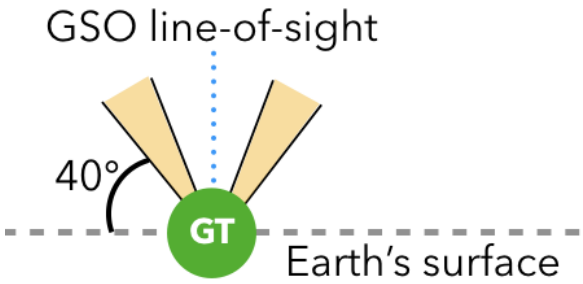


Figure 3.9: *GSO arc-avoidance: at the Equator, only satellites in the small shaded regions of elevation are reachable.*

for construction and maintenance, or for political reasons. ISLs side-step this issue, crossing such unfriendly territory entirely in space.

Spectrum efficiency: For Ka- or Ku-band radio spectrum operation, companies need licenses from bodies like FCC and ITU. The spectrum is shared among multiple interested parties. In contrast, thanks to the narrow beams and negligible interference issues, inter-satellite laser connectivity is unlicensed [106]. Thus, with ISLs, interference and spectrum contention only arise at the sources/sinks of data.

GSO arc-avoidance: GSO satellites fly above the Equator, and operate in the same frequency bands sought for LEO communication. Thus, LEO Satellites, when crossing the lower latitudes near the Equator, must avoid interference with GSO satellites. Both Starlink and Kuiper explicitly note in FCC filings [49, 55] that they would address this by only allowing up/down-links with at least a minimum angular separation from the bore-sight of a GSO base station. For Starlink this angle of separation is 22°, while Kuiper mentions that this angle would gradually increase from 12° to 18° over deployment. The consequent reduction in the field-of-view from a GS is illustrated in Fig. 3.9 for Starlink with the 40° minimum angle of

elevation planned for its full deployment.

With BP, any traffic between the northern and southern hemispheres would use GSes near the Equator. Thus, the impact of the reduced GS field-of-view will be much higher on BP than on ISL connectivity, as for the latter, only sources and destinations in the Equatorial region will be affected.

ISLs are important: Given the multitude of benefits of ISLs in offering long-distance low-latency connectivity at high throughput, reducing the latency variability, avoiding bad weather conditions in the lower layers of the atmosphere, and others, and also as the majority of LEO constellation providers planning to deploy ISLs, for the most of this dissertation we consider constellations with ISLs. As mentioned before, SpaceX has already launched a few Starlink satellites [73] with laser ISL capabilities thus demonstrating the practicability of such deployments. Nevertheless, for the packet-level simulator, Hypatia, discussed in Chapter. 5, we touch upon its ability to simulate constellations without ISLs in order to accommodate simulations of OneWeb-like constellations which do not plan to incorporate ISLs.

3.2 Topology design in LEO networks

LEO constellation topology design is a high-dimensional network optimization problem. Recent work shows that increasing the capacity of the inter-satellite network would substantially improve throughput between terrestrial ground stations [63], even after appropriate accounting for other system bottlenecks like ground-satellite connectivity in an end-to-end manner. However, this prior work focuses on higher-cost inter-satellite links as the vector for increasing the capacity of the satellite-satellite interconnect, ig-

noring the potential for superior *network design* instead. This approach is reflective of a more widespread, usually implicit, assumption that inter-satellite links *must* be local, and +Grid-like, *e.g.,* with a satellite connecting to its nearest neighbor in each direction, similar to that in (smaller) past constellations. Available patents [107], visualizations [108] and analyses [51, 63–67, 109, 110] feature this assumption. However, this assumption is needlessly limiting (§2.2.1): in many cases, it will be possible to connect to more distant satellites. These longer links can improve the network's throughput efficiency, as each end-end connection then uses capacity on fewer, longer inter-satellite hops. Allowing longer links greatly expands the design space, and makes satellite topology design a highly non-trivial problem: *given a small number of inter-satellite links (ISLs) per satellite, how should these be connected into a topology for maximizing network bandwidth and minimizing latency?* To the best of our knowledge, this question itself, let alone answers to it, has not been put forth in prior work, at least in the context of the mega-constellations under development.

At first glance, this may appear to be a traditional network optimization problem, very much like that arising in, for instance, data centers, where there is a long line of research on switch interconnects rooted in graph theory [111–114]. However, two key aspects differentiate satellite network design from other well-studied settings: (a) ISLs are limited by range, so that only relatively nearby satellites can connect to each other; and (b) satellites are moving with respect to the Earth and each other. The former issue creates challenges for graph theoretic abstractions that do not account for link locality, while the latter requires additional consideration of the changing set of links that are feasible, as well as their evolving distances from terrestrial endpoints. Network design is NP-hard even in simple, static cases [115], so how do we tackle the added complexity of the temporal

dynamics? This problem and potential solutions are explored in Chapter. 4

LEO trajectory design: While Chapter. 4 primarily focuses on LEO topology design, we also present a first-cut systematic analyses of LEO trajectory design parameters at the end of that chapter. Each satellite's orbit has many degrees of freedom that must be configured, and worse, thousands of satellites must be configured jointly in order to offer optimized network performance.

3.3 Intra-domain & inter-domain routing challenges

Large LEO constellations promise global coverage at low-latency and high-bandwidth. However, realizing the full potential of these networks requires addressing new research challenges posed by their unique *dynamics*. The high-velocity movement of satellites creates not only high churn in the ground to satellite links, but also fluctuations in the structure of end-end paths as the satellites comprising the paths move. This has serious implications on both intra-domain and inter-domain routing in LEO networks.

Intra-domain routing: As satellites move continually at high speeds, inter-satellite links need to carry different end-to-end flows at different times. Imagine a pair of satellites over Atlantic moving from lower latitudes close to the Equator to mid-latitudes seeing large volume of traffic between the US and Europe. As flows traversing a bottleneck ISL try to converge to a fair-share, new flows might arrive thus continually shifting the point of convergence. We discuss this problem in greater details in Chapter. 5. Satellite trajectories are highly predictable and routing and traffic engineering might need to take into consideration LEO dynamics in proactively adapting to foreseeable spatial and temporal traffic demand

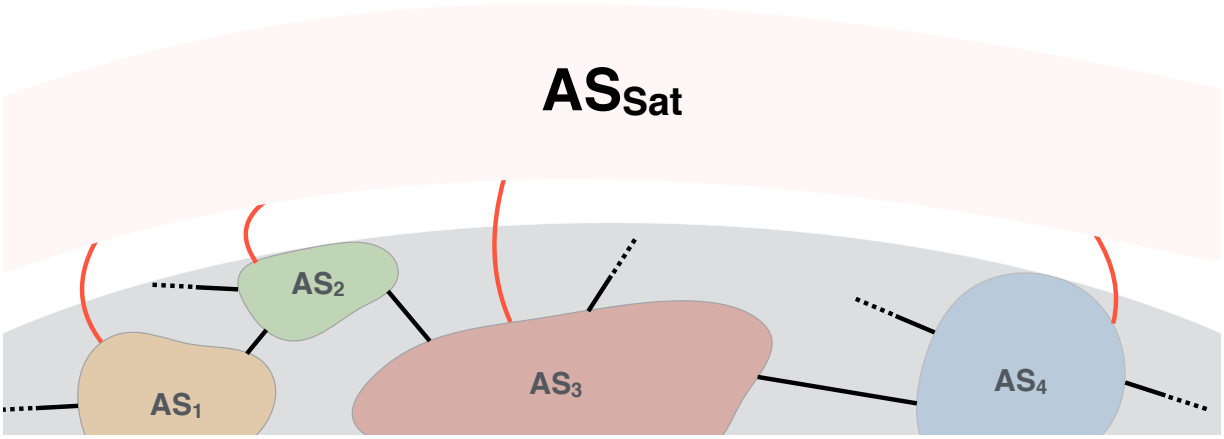


Figure 3.10: *Satellite ASes may create challenges for BGP, but also several opportunities for improving Internet routing.*

changes. Constellations can pre-compute routes for the future [116, 117]. Of course, more sophisticated schemes can also be built that are aware of the link and congestion state [118–122].

Inter-domain routing: Another interesting routing implication of high density LEO satellites lies in their interactions with today's Internet ecosystem. Consider the example in Fig. 3.10, where each of the 4 terrestrial ISPs is peering with a satellite AS, AS_{Sat}. AS_1 has two equal-AS-length paths to AS_3, through AS_2 and AS_{Sat}. Likewise, AS_2 has two similar paths to AS_4. The geographic distances could mean that were ASes choosing routes based on latency, AS_1 should prefer the terrestrial route to AS_3, while AS_2 should prefer the satellite route to AS_4. While it is already the case that AS path lengths in today's Internet are poor proxies for performance, LEO satellite networks may make this discrepancy larger in magnitude and more commonplace. The performance and availability of paths through the satellite network(s) is also likely to be more variable. These observations create obvious challenges in Internet route selection – while there is a long

history of research on performance-aware Internet routing [123–126], satellite networks could dramatically increase the pressure to find deployable solutions.

Another implication already hinted at in Fig. 3.10 is the possibility that all or a large fraction of terrestrial networks may peer with a single large satellite network, especially due to the large performance advantage over long-distance routes. This would be an extreme point in the "flattening" of the Internet [127], which may have several implications on Internet reliability and security [128]. If multiple satellite networks are deployed and compete for peering with terrestrial networks, this presents another unique setting: unlike terrestrial ISPs, the topology and network size for a satellite ISP are known, creating greater transparency for peering. Note that the ISLs may not be precisely known for dense constellations, but could likely be inferred from end-to-end latency measurements.

It is also unclear how a satellite ISP would offer its services. Should it deploy ground stations at locations good for peering, such as IXPs, or compute a distribution of ground stations for more uniform coverage? Should it expose more flexibility to customers and peers in picking routes through it (given the aforementioned natural transparency of this setting), perhaps even enabling on-demand long-haul connectivity, or expose a more traditional interface, by handling these complexities internally? What would the service-level agreements look like, particularly with higher variability in latency, and to a lesser extent, in the availability of links? Thus, a raft of routing issues are worth investigating.

While recent parallel work [129] performs cost–performance trade-offs in the design space for incorporating LEO inter-domain routing in today's Internet, in the context of this dissertation, we do not focus on inter-domain

routing beyond what has been already discussed in this section.

3.4 Congestion control faces LEO dynamics

Congestion control for traditional satellite networks is a well studied problem, with specialized TCP variants [130, 131] modeling satellite paths with high bandwidth-delay product and high loss rates. However, for LEO satellites, with latencies being lower by more than an order of magnitude, these design assumptions may need adjustment. Another unique characteristic of the new breed of satellite networks is the latency variation over time – unlike GSO-based networks, LEO-based networks see path length changes over time as the satellites move. Fig. 3.11 shows an example of this variation for a Frankfurt-DC link over a 25^2 LEO constellation. The latency varies in a $\sim$5 ms range around the 32 ms median. The magnitude of these variations depends on satellite density, with smaller constellations seeing larger variation. Note that even the large planned constellations will be put in place incrementally, making this a significant concern.

It is unclear how even the recent crop of congestion control proposals like PCC [132, 133], BBR [134], and Copa [135] fare in this setting. PCC Vivace [133] filters out small random RTT changes and jitter, but the magnitude of variation in our setting exceeds its thresholds. BBR [134] and Copa [135] try to estimate queueing-free RTTs as the minimum over end-to-end RTT measurements, but here, the minimum RTT itself is time-changing. Overall, end-to-end protocols may easily confuse the network's change in propagation delay for queueing dynamics. Thus, even our best congestion control ideas may need to be reworked, or at least, reevaluated.

A potential way forward is to expose knowledge of the changing (but predictable) physical layer latencies to the congestion control mechanisms,

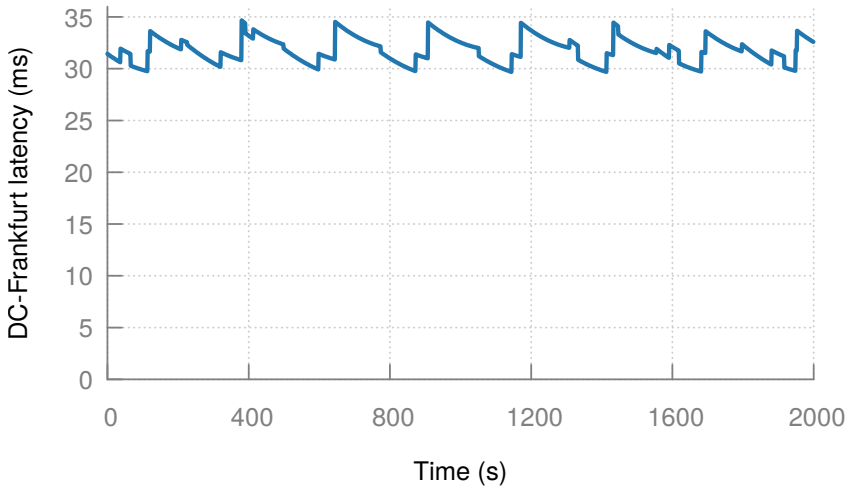


Figure 3.11: *The Frankfurt-DC latency over a* 25^2 *LEO constellation for a period of 2,000 seconds at a granularity of* 1 *second.*

such that they can correct for it. Such cross-layer machinery could be implemented by splitting the end-to-end transport connection into three segments, where the middle is a custom system operated by the satellite provider; or it could be implemented end-to-end with more significant deployment hurdles.

While in the context of this dissertation we do not propose targeted congestion control solutions for LEO networks, we do quantify the impact of LEO dynamicity on both loss-based and delay-based congestion control schemes using Hypatia in Chapter. 5. Our evaluations and Hypatia itself are aimed to drive research in this area in the years to come.

3.5 Related publications

The plots and the corresponding discussions in this chapter have been taken from the following publications:

- Y. Hauri, D. Bhattacherjee, M. Grossmann, and A. Singla, "'Internet

from Space' without Inter-satellite Links," in *ACM HotNets*, 2020

- D. Bhattacherjee, W. Aqeel, I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Gearing up for the 21st century space race," in *ACM HotNets*, 2018

I supervised the Bachelor and Master theses of Yannick Hauri and Manuel Grossmann respectively, culminating in the HotNets'20 paper.

4

Topology design at 27,000 km/hour

In this chapter, we posit that the high density of these new LEO constellations and the high-velocity nature of such systems render traditional approaches for network topology design ineffective, motivating new methods specialized for this problem setting. We propose one such method, explicitly aimed at tackling the high temporal dynamism inherent to low-Earth orbit satellites. We exploit repetitive patterns in the network topology to avoid expensive link changes over time, while still providing near-minimal latencies at nearly 2× the throughput of standard past methods. This chapter addresses the following goal:

Goal B: LEO network topology design to improve network performance.

At the end of this chapter, we touch upon the LEO trajectory design problem and quantify the impact of various trajectory design parameters on LEO network performance. While mega-constellation trajectory design is itself an interesting problem and calls for further in-depth quantitative analyses, it is driven by launch locations and rocket launch trajectories, satellite

de-orbiting, and other non-networking challenges which is difficult to zero in on at these early stages of the LEO broadband industry.

4.1 A new topology design challenge

Along the lines of discussion in §3.2, we make a concrete case in this chapter that intuitive network topology design approaches drawn from experience in traditional settings are unsuitable for this new, non-traditional problem context. As LEO satellite positions change very fast relative to the Earth (ground stations) and with respect to each other, were are faced with the challenge of finding design strategies that take into account the extreme dynamicity of the network elements unforeseen in terrestrial networks or geostationary satellite constellations. Our exploration shows that even with substantial problem-specific customization, methods like Integer linear programming, random graphs, and ant-colony optimization, fall short.

As a first attempt at addressing this problem, we propose a novel approach exploiting repetitive patterns: if the topology is restricted such that each satellite's local view is the same as that of any other, then one can limit topology design to the space of all possible *local* views at just one satellite. We refer to each such local view as a ***motif***. This motif is then repeated across all satellites, with each connected to its neighbors in the same way. Even for the densest proposed constellations, the space of possible motifs, while non-trivial, is small enough to search exhaustively and identify the optimal motif for a target traffic matrix.

While this simple approach already provides a $\sim 2\times$ efficiency improvement over the neighbor-grid baseline, we further observe that satellites are closer to each other at higher latitudes than at the Equator, implying a larger set of possibilities for ISLs. Thus, the motifs can be customized at different

latitudes, providing further improvements in network capacity. This insight enables efficiency improvements even when we make worst-case assumptions about the range of power-limited ISLs – for StarLink, with our approach, network performance would improve over the neighbor-grid by 37% even in the most pessimistic scenario.

We also show that our approach effectively tackles the temporal variations inherent to LEO satellite systems: ISLs based on motifs are maintained for long time periods, thus avoiding frequent, expensive link changes. Note that link changes can require tens of seconds, thus being a substantial overhead if links are operational only for a few minutes before each change.

We make the following contributions:

- We frame the problem of inter-satellite topology design for large LEO constellations, showing why intuitive approaches like Integer programming, random graphs, and ant-colony optimization are unsuitable.
- We propose a new approach for designing such networks using regular repetitive patterns, *i.e.,* motifs.
- We further show how the spatial geometry of LEO constellations admits customization of motifs applied to different parts of the topology.
- We study the impact of allowed ISL range and setup time on the effectiveness of our approach.
- We evaluate our ideas on the two largest proposed constellations, Starlink and Kuiper, showing network performance improvements of as much as 54% and 45% over neighbor-grid connectivity respectively for a natural, population-weighted traffic model.
- For Starlink, we find that even with pessimistic assumptions about ISL range, network performance can be improved by as much as 40%.

- The code and data used to generate the key results in this chapter are publicly available [137].

4.2 Typical network design?

Given (a) a constellation's satellite trajectories, (b) a small number of inter-satellite connection units at each satellite, and (c) a target traffic matrix between terrestrial endpoints; our goal is to decide which satellite-satellite connections to build to minimize latency and hop-count in end-end paths.

Static variants (ignoring satellite motion and Earth's rotation) of similar network design problems are known to be NP-hard [115], but one could imagine the methods typically used in that setting — Integer programming, linear program rounding, ant-colony optimization, random graphs, etc. — to be effective here. One might even argue that the ISL design problem is a costly but one-time effort that can perhaps be tackled using supercomputing resources.

We thus explored customizing three intuitive approaches to address satellite topology design. We discuss why such techniques are doomed to fail in the face of the problem's complexity and the involved temporal dynamics. We also use this exploration to help draw out the constraints and objectives more concretely.

4.2.1 ILP for a static snapshot

To handle the temporal variations in a somewhat brute-force manner, one could potentially optimize a series of short-term static snapshots using Integer linear programming.

Inputs:

- L: Maximum number of ISLs allowed per satellite.
- v_{ab}: Is sat_a visible to sat_b and within its range?
- d_{ab}: Linear distance between satellite sat_a and sat_b.
- A traffic matrix, H, between terrestrial sites.

While this approach can obviously use arbitrary inputs, for a reasonable *concrete* instantiation, we assume that: (a) $L = 4$, in line with information that can be gleaned from Starlink's regulatory filings[1]; (b) only the visibility constraints determine v_{ab}, an assumption we relax later (§4.3.6); and (c) H specifies traffic distribution between the 1,000 most populous cities (2025 population estimates [95]) as ground sites, with city-city traffic volume scaled $\in [0, 1]$ in proportion to the population products of the city pairs. (We also examine a different workload based on economic activity instead of population in §4.3.7.)

Note that we take a simplified view of connectivity between ground sites and satellites, assuming that as long as a GS is within range of a satellite, it can connect to it. In our concrete instantiation with populous cities, we reduce the likely multitude of GSes in such cities to just one per city, with arbitrary bandwidth and connectivity towards satellites, limited only by range. Considering GS placement and GS-satellite connections jointly in the optimization is left to future work for several reasons: (a) it is unclear how much of these decisions will be controlled by the constellation operators if customers can buy and deploy GSes, as is being suggested by Starlink [138]; (b) bottlenecks are likely to arise in both ground-satellite and satellite-satellite connectivity, even if there is substantially higher capacity in ISLs than for satellite up/down links [63]; and (c) we believe a

[1]While recent work [51] concluded that there would be 5 ISLs, SpaceX revised their filings later, with the new filings indicative of 4 ISLs.

decomposed treatment of these complex problems is fully justified, especially at this nascent stage, as even in mature areas like data centers, there is precedent for such decomposition, *e.g.,* with routing and topology design.

Outputs: We must set (a) binary variables y_{ab} that capture whether an ISL between sat_a and sat_b is active; and (b) binary variables x_{ab}^{st} that capture whether traffic from endpoint c_s to c_t is carried over an ISL between sat_a and sat_b.

Objective function: The output decisions must minimize a combination of latency and ISL hop-count between the end points. For each endpoint pair, we quantify latency in terms of *stretch*, *i.e.,* the ratio of the shortest-path distance across the designed network and the geodesic distance. To aggregate our stretch (S) and hop-count (B) measures, arbitrary linear combinations may be used. We define $M_\alpha = \alpha S + B$, where α controls how much more we value stretch. Most of our analysis weighs both factors equally ($\alpha = 1$), but in §4.4.5, we examine the impact of varying α.

Finally, we define our objective function, Φ_α, as the sum of M_α across endpoint pairs, weighted by the traffic matrix H. Minimizing Φ_α minimizes system-wide stretch and hop count *per unit of traffic.*

Constraints: Below we provide intuitive textual description of the constraints:

- ISLs are duplex.
- Each satellite should have ≤ 4 ISLs active.
- An ISL can be connected iff visibility allows.
- An ISL can carry traffic iff it is active.
- Flow should be conserved at satellites.

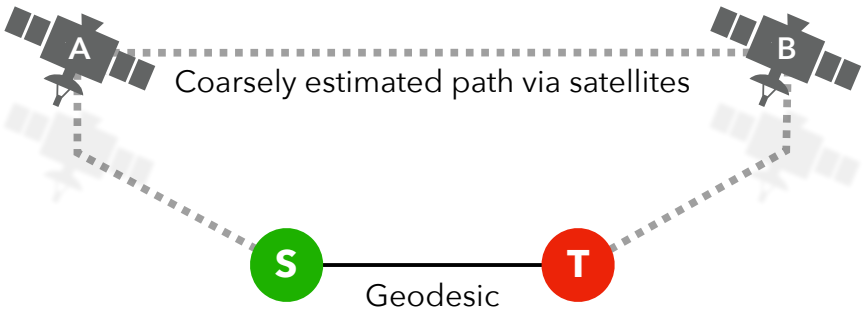


Figure 4.1: *ISLs far from S-T should not carry S-T traffic. The estimated S-A-B-T path traverses the geodesics between S and A's terrestrial nadir, up to A, the ISL, down to B's nadir, and the geodesic to T.*

- End points should source or sink flow correctly per H.

Customized ILP: We invested substantial effort in studying and improving the scalability of an ILP approach. To reduce problem size, we added a heuristic, the intuition for which is shown in Fig. 4.1: satellites "too far away" from the geodesic between an endpoint pair shouldn't carry its traffic. We obtain an estimate of the path length between two end points, S and T, through an ISL between two particular satellites, A and B by adding the geodesic distance between S and A's terrestrial nadir, the A-B ISL length, and the geodesic distance between B's nadir and T. If this estimate exceeds the geodesic S-T distance by $\geq 1.5\times$, we restrict the ILP from considering an A-B ISL to carry S-T traffic. This eliminates many variables of x_{ab}^{st} type. For the largest scales we could test, this heuristic causes no compromise in optimality.

Results: We use a 40^2 LEO constellation (53°, 550 km) with a maximum ISL length of 5,014 km. We generated ILPs for increasing numbers of cities, selected in order from the 1,000 most populous cities. For smaller numbers of cities, the ILPs execute within reasonable time, a few minutes

to a few days, and provide optimal ISL setup schemes. For 20 cities, the ILP-generated network's Φ_1 is lower than +Grid's by 54%, showing the substantial improvements possible.

Issue #1: Limited scalability. Even for just 25 cities, the ILP does not finish within 2 days on a machine with 64 cores and ∼500 GB of memory. Further, extrapolating from smaller sizes, we estimate runtime for 1000 cities to require 10^{29} days, which would remain intractable even with perfect parallelism across a supercomputer.

Issue #2: Temporal dynamics. LEO satellites travel hundreds of kilometers in minutes. Thus, ILP solutions generated minutes apart would use very different ISLs, requiring a large number of ISL changes. For 20 cities (the largest scale we could run the ILP for), ILP topologies generated just one minute apart share only 9% of links. Even with (optimistic) ISL setup times of a few seconds, this would be an unacceptable amount of link churn. With such a large link churn, it's unclear if/how any incremental link change strategy would work either.

4.2.2 Random graphs

Inspired by network design in the data center context [111], we also explored a generic solution that does not specialize to the traffic, but works well across a variety of workloads: random regular graphs (RRGs). These are graphs uniform randomly sampled from the space of all regular graphs (*i.e.,* with all nodes having the same number of connections) of a target number of nodes (satellites) and degree (4 links per satellite).

Customized RRG: Unfortunately, unlike the well-studied standard setting [139], it is unknown how to *uniformly* sample RRGs in our setting, *i.e.,* from the more restricted space of regular graphs with link locality. However,

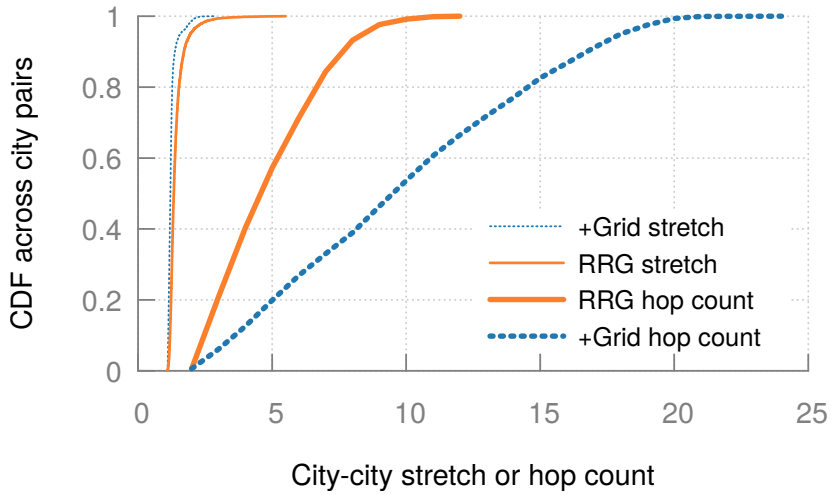


Figure 4.2: *Compared to +Grid, random graphs (**RRG**) reduce hop counts substantially, at the cost of a marginal increase in stretch.*

to gain intuition for the shortcomings of this approach, such rigor is not necessary, and we design a heuristic process that samples the permissible edges (but gives no guarantees about uniform graph sampling).

Our heuristic first lists all permissible ISLs based on visibility constraints (at a time instant), and picks ISLs to add from this list uniformly at random. An ISL can only be added if both satellites involved still have fewer than 4 connections. We repeat this procedure until no more new ISLs can be added. At this point, if only a small fraction of ISLs are missing (*i.e.,* some connectivity is unused), we stop. Otherwise, for each satellite with fewer than 4 ISLs, we remove all its ISLs and those of its neighbors, and add them back to our sampling list, and continue sampling.

Results and challenges: To compare random graphs against +Grid, we randomly select 5,000 city pairs and calculate the stretch, hop count, and M_1 with both topologies for each city pair. As Fig. 4.2 shows, by sacrificing median stretch by 11%, random graphs decrease the median hop count by

53%, and thus, the median M_1 by 43%, and Φ_1 by 42%.

Issue #1: Temporal dynamics. While random graphs overcome the ILP's scalability issue, unfortunately, they also suffer from the system's temporal dynamics: within 2 (5) minutes, more than 8% (19%) of the ISLs become infeasible. Naively refreshing the entire topology every few minutes would change nearly all ISLs, causing massive disruption. However, incremental changes are non-trivial because a satellite for which an ISL becomes infeasible may not be within reach of other satellites also in the same state, to which it could connect. Thus, reconnecting lost connectivity in an incremental fashion would require breaking and remaking even more connections near such disconnected links.

Issue #2: Inflexibility. Unlike the ILP, random graphs do not admit explicit optimization towards a target traffic matrix, or for an arbitrary α, *i.e.,* they do not allow configuration of the tradeoff an operator may want to make between stretch and hop-count. Instead, they yield only one fixed choice in the trade-off space, as we discuss later in comparison with our more flexible approach (§4.3).

We also spent substantial effort in customizing **ant-colony optimization** [140] to optimize the ISLs starting from a random graph. This approach performs well for small problem sizes (few tens of city pairs) but does not converge for larger cases, while also causing high link churn. As the drawbacks are similar to the above two methods, we omit the details.

4.2.3 Summary of the challenges

For large constellations like those proposed, we must configure several thousand ISLs to meet desired latency and throughput goals in a highly dynamic setting. We find that the problem's complexity even for a single snapshot,

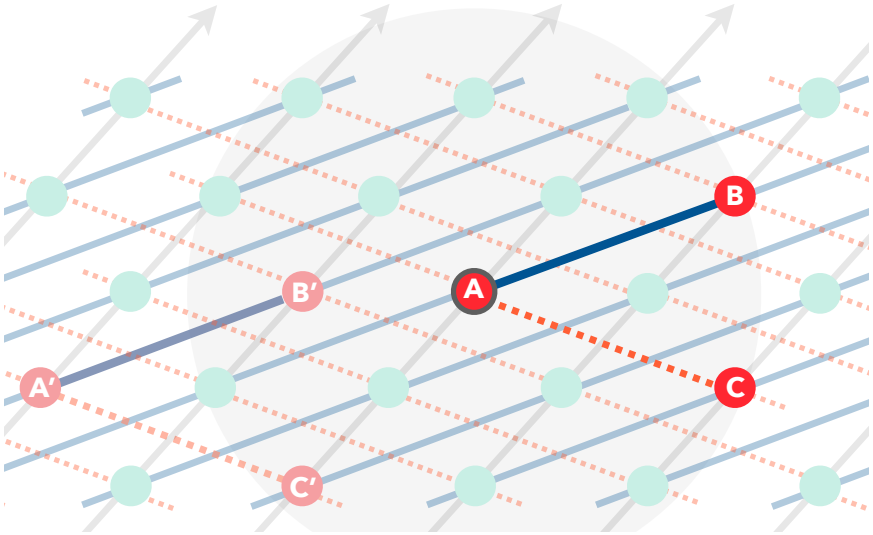


Figure 4.3: *A can connect to any satellites within its view (circle). If A chooses B and C, repeating this pattern across the constellation fully specifies its ISLs. Then any satellite A′ has the same local view of its connectivity, as shown. We call such local views "motifs".*

coupled with its temporal dynamics, defeats traditional methods. An ideal solution would not only yield substantial improvements over +Grid for a static snapshot of the system, but also minimize churn in links to avoid the few seconds to tens of seconds of overheads incurred from link changes.

4.3 Motifs: simple yet effective

Our observations about the deficiencies of standard methods also reveal the features of a practical solution: avoiding overly complex optimization and link churn. We need solutions that marry the simplicity of the +Grid with performance gains clearly shown possible by other methods. Our proposed solution is thus a generalization of the +Grid.

4.3.1 Generalizing +Grid with motifs

As discussed in §3.2, for dense constellations, satellites can reach (*i.e.,* communicate with, using a single point-to-point link) other satellites beyond only their nearest neighbors. Fig. 4.3 illustrates the reach of satellite A in such a setting. If A picks two satellites, B and C, to connect to from this reachable set, we can repeat this connectivity pattern across the entire network (as shown in Fig. 4.3) to obtain an ISL topology. Note that A's other two ISLs are decided by other satellites connecting to it in the same manner. Thus, every satellite's local view is identical. The 2 ISL choices for A are what we refer to as a **motif**. More generally, a motif is a 3-satellite, 2-ISL connectivity pattern, repeating which throughout the constellation fully describes its ISL topology. Henceforth, we use motif to refer to both the connectivity pattern, and its resulting topology.

Different motifs describe a family of topologies, with the +Grid being a member of this family. The simplest version of our approach, which we improve upon later, involves exhaustively evaluating all possible motifs to pick the one with the best performance. For any satellite, enumerating all possible connectivity patterns involves finding nearby satellites to which direct point-to-point links are feasible based on any specified ISL range constraints. We only consider satellites in orbits traveling in the same direction to define motifs. As one such set of satellites travels northward, another set crosses them traveling southward; connections across these sets would be short-lived. Note however, that satellites are farther apart at the Equator and closer to each other at higher latitudes, implying that a near-Equator satellites have the fewest in-range satellites. Thus, for enumerating motifs, we use a satellite at the Equator, ensuring that for any other satellite, the considered motifs will only contain feasible links.

Having computed the set of satellites reachable from a satellite at the Equator, it is easy to enumerate all feasible motifs: every possible choice of 2 satellites from this reachable set is a motif. Due to the symmetries involved, some of these motifs are equivalent, *e.g.,* in Fig. 4.3, for A, picking any two of the 4 satellites it is connected to would lead to the same topology. Thus, using the spacial symmetry around a satellite, we can cut the number of motifs in an enumeration to only motifs that result in different topologies.

With the unique motifs enumerated, we simply evaluate any metrics of interest, *e.g.,* Φ_α, across them all. The motif with the best metric value, *i.e.,* lowest Φ_α, is chosen.

To summarize this procedure:

- Consider a satellite at the Equator, e.
- Let S_e be the set of all satellites within e's range.
- The set of all motifs, $M = [S_e]^2 = \{\{a, b\} : a, b \in S_e, a \neq b\}$
- As an (optional) optimization, cull equivalent motifs.
- Output the best motif $m = \arg\min_{x \in M} \Phi_\alpha(x)$.

4.3.2 No link churn

Motifs obviously cover a limited subset of the space of possible ISL topologies, but they describe a structured set of topologies with *stable* connectivity, *i.e.,* without ISL churn. In the example in Fig. 4.3, A-B and A-C travel in sync, such that their relative velocities close to the Equator are small and grow larger only at higher latitudes. These changes are of precisely the same nature as in the +Grid. Thus, each satellite is continuously connected to the same satellites — even at higher latitudes as satellites change

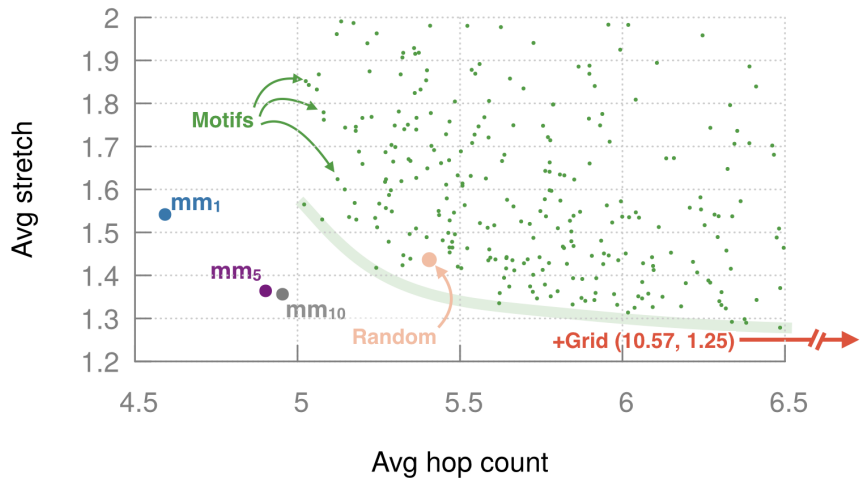


Figure 4.4: *Different motifs present a large number of design points, with different stretch and hop counts. The curve shows the Pareto frontier across motifs trading off one metric for the other. +Grid provides low stretch, but with very high hop count. The mm_1, mm_5, mm_{10} points are for the more sophisticated approach we discuss later in §4.4.*

directions, we still maintain the same connections. Thus, motifs provide long-term, stable connections throughout the topology.

4.3.3 Performance at an arbitrary snapshot

For the same 40^2 constellation as in §4.2, we find that 1029 unique motifs exist. Fig. 4.4 shows the topology's average stretch and hop-count for each such motif at one arbitrarily chosen system snapshot in time. The metrics are computed in the same manner as in §4.2, using the population-product traffic matrix across popular cities. For clarity, we trim the plot to leave out motifs with overly large stretch or hop count. The plot includes the random graph approach; we used 100 trials and show the mean across them. +Grid achieves an average weighted stretch and hop-count of 1.25 and 10.57 respectively. We can draw several insights from these results:

- Different motifs differ widely in their performance.

- Motifs expose a trade-off between stretch and hop count, with several motifs at the Pareto frontier. Random graphs provide only *one* point in the design space.
- Motifs at the Pareto frontier can exceed the random graph's performance, while allowing greater flexibility for optimizing whichever metric we value more.
- +Grid also has nearly the lowest stretch, but very high hop count: a motif that compromises 2% (10%) on stretch, can improve hop count by 32% (47%).

Besides system-wide average metrics, we also examine their distribution across city-pairs. The best motifs improve median (95^{th} percentile) M_1, M_5, and M_{10} by 44.5% (54%), 26.2% (37%), and 16.8% (22.3%) respectively over +Grid.

Unfortunately, we could not run the ILP at the same scale for inclusion in Fig. 4.4. At the largest scale we could run the ILP (20 cities), it achieves 54% lower (better) Φ_1 than +Grid. The motif's Φ_1 is 45% better than +Grid. We remind the reader that the ILP is not a practical approach for the reasons discussed earlier (§4.2.1).

4.3.4 Performance over time

While motifs provide link stability, how does a motif's performance for a given traffic matrix evolve with time, as the constellation moves in sync across the Earth's surface? To address this question, we use the same (fixed) motif as in §4.3.3, which achieves the best Φ_1. We evaluate Φ_1 at a minute-by-minute granularity across a 2 hour period, which is more than one orbital period for the constellation.

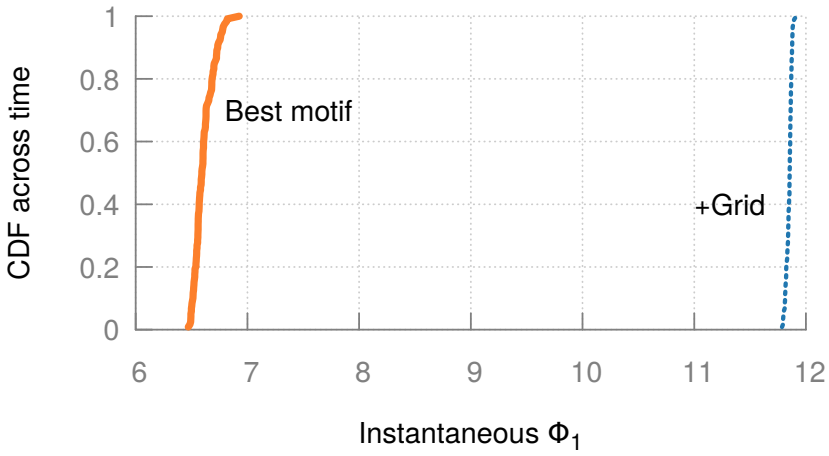


Figure 4.5: *Motifs, like +Grid, show little variation in* Φ_1 *across time.*

Fig. 4.5 shows the distribution of Φ_1 over these per-minute snapshots. The best motif achieves Φ_1 44% (43%) lower than +Grid in the median (95^{th} percentile). Variation in Φ_1 over time is within 10% of the median for both. Thus, motifs provide consistent improvements over the naive +Grid approach, without depending on *any* dynamic ISL reconfiguration.

4.3.5 Effect of constellation configuration

We also assess the utility of motifs for constellations of different configurations in terms of size and orbital inclination.

Size: We evaluate motifs and +Grid on uniform constellations (53°, 550 km) of various sizes: 16^2, 24^2, 32^2, 40^2. For small constellations, both +Grid and the best motif leave many city pairs without connectivity through the constellation. Both topologies have the same numbers of disconnected city pairs, because these stem from lack of GS-satellite visibility rather than ISLs. Thus, we evaluate Φ_1 across *connected* city-pairs.

Fig. 4.6 shows that for denser constellations, motifs yield larger benefits.

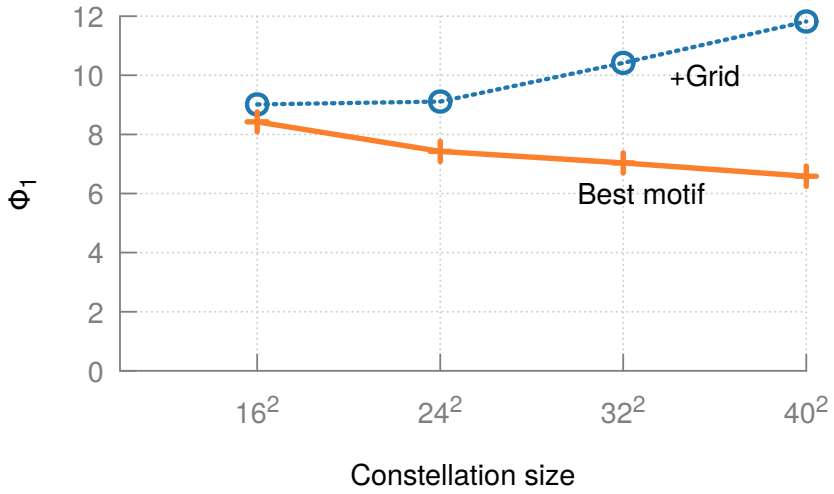


Figure 4.6: *As constellation size increases, the utility of motifs increases. For +Grid, stretch improves only marginally, while hop counts increase substantially with size due to neighbor-only connectivity.*

This is expected, as the number of candidate motifs increases with density. The increase in Φ_1 for +Grid with larger constellations may appear odd, but is easily explained: while improvements in stretch are small, particularly beyond 24^2, the hop count increases rapidly with size as satellites only connect to their nearby neighbors.

These results also capture how the nature of the problem changes going from the smaller past constellations (much smaller than even 16^2) to the planned mega constellations. For smaller constellations, the problem is rather finding the lowest altitude such that the +Grid links clear the Mesosphere; see [141] for discussions along these lines.

Inclination: We also examined motifs across a polar constellation (90° inclination) of the same size (40^2) and altitude (550 km). These results are shown in Fig. 4.7. The median improvement in M_1 going from +Grid to motifs is 46% for the polar constellation, compared to 44.5% for the 53° one. As expected, the 53° constellation has a 14% lower M_1 (with motifs,

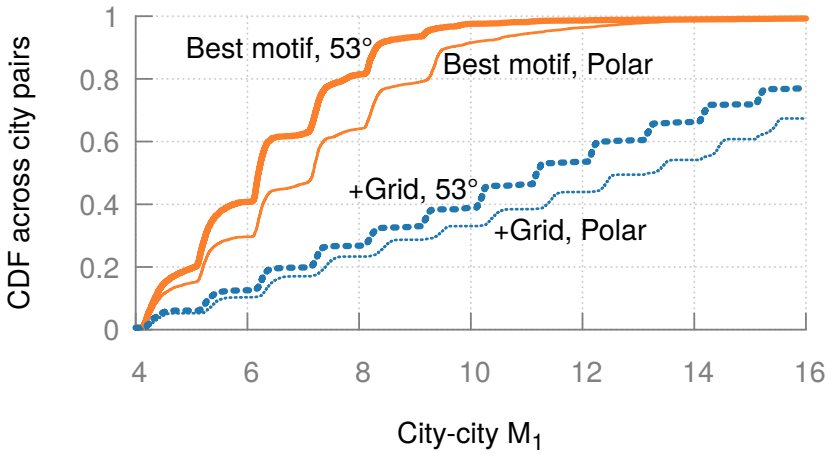


Figure 4.7: *For the same constellation size, performance improves with inclined orbits as satellites spend less time over sparsely populated polar regions. Improvement over +Grid is similar for both constellations.*

in the median) than the polar one, because satellites spend more time over the densely populated regions and thus fit our population-product traffic matrix better.

4.3.6 Accounting for power-limited range

As discussed in §2.2.1, for a constellation at an altitude of 550 km, visibility limits ISL range to 5,014 km. While our understanding is that range is primarily limited by visibility, we also evaluate motifs in settings where range is additionally stunted, *e.g.,* by a severely constrained power budget. For the 40^2 constellation, the longest links in +Grid are 1,467 km, giving us a lower bound on range. We thus evaluated ranges between this bound and the maximum of 5,014 km. Fig. 4.8 shows that with 3,000 km range, the performance improvements are already similar to the setting with the longer only-visibility-limited range. Given that the *primary* purpose of these satellites is network connectivity, it is unlikely that design decisions

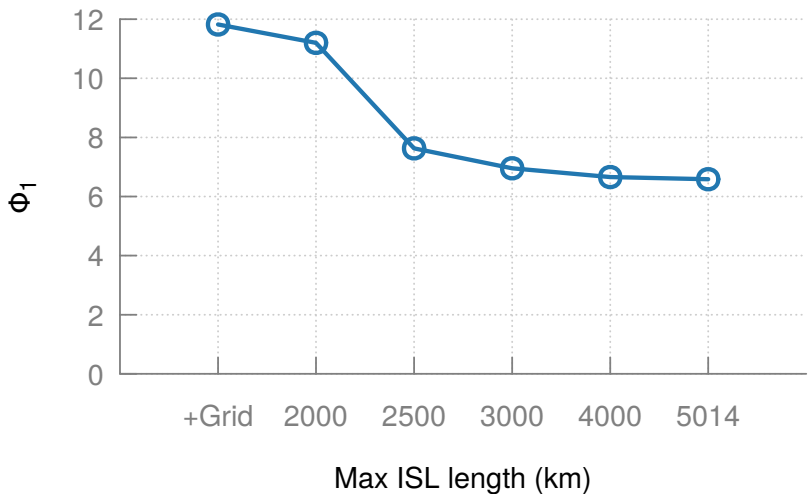


Figure 4.8: Φ_1 *for the best motifs across different ISL ranges: a modest increase in range beyond the minimum required for +Grid connectivity can substantially improve performance.*

that stunt range to be so severely limited will be made. As discussed earlier (under "Recent developments in space technology" in Chapter. 1), much longer ISL ranges are practical today. In fact, some constellations like Telesat will require higher range links (6,000 km [142]) due to their lower density. Also, operators would likely want partial deployments to already use ISLs, and given that satellites will be farther apart in such sparser partial deployments, longer ISL ranges than in the final deployment would be needed.

4.3.7 A different traffic matrix

While +Grid connectivity is completely traffic matrix agnostic, motifs permit *some* degree of customization towards the traffic – from the sizable space of motifs, one can pick the motif that performs best for a target traffic matrix. Of course, this is more limited customization compared to

solutions like ILPs, which allow complete traffic-directed optimization (but are unsuitable for other reasons, as discussed in §4.2.1).

The traffic that the topology is designed for will ultimately be driven by market forces, regulators, and the geographic variation in competing terrestrial connectivity. As such, it is difficult for us to evaluate for *the* right traffic matrix. Throughout, we have picked one reasonable, intuitive traffic matrix, in the form of the population-product model. Briefly, we discuss another similarly intuitive traffic matrix.

Instead of population, we consider economic activity, in terms of Gross Domestic Product (GDP), as a proxy for Internet traffic, based on the known correlation between Internet penetration and GDP [143]. Thus, we use the top 100 cities ordered by their GDP [144] as ground sites, with city-city traffic volume scaled $\in [0, 1]$ in proportion to the GDP products of the city pairs. For the 40^2 inclined constellation, the best motif achieves Φ_1 34% lower (better) than +Grid for this traffic matrix. While smaller than for the population-weighted traffic model (48%), this is still a large improvement.

Considering time-varying traffic is left to future work, we note that one could evaluate the potential motifs against snapshots of traffic over a desired time period, picking the one that provides the highest performance over time.

4.4 Richer use of motifs

Our simple motif approach achieves a substantial improvement over +Grid connectivity. Next, we show that if a small (configurable and controlled) amount of dynamism in connectivity is permissible, this can further improve

performance.

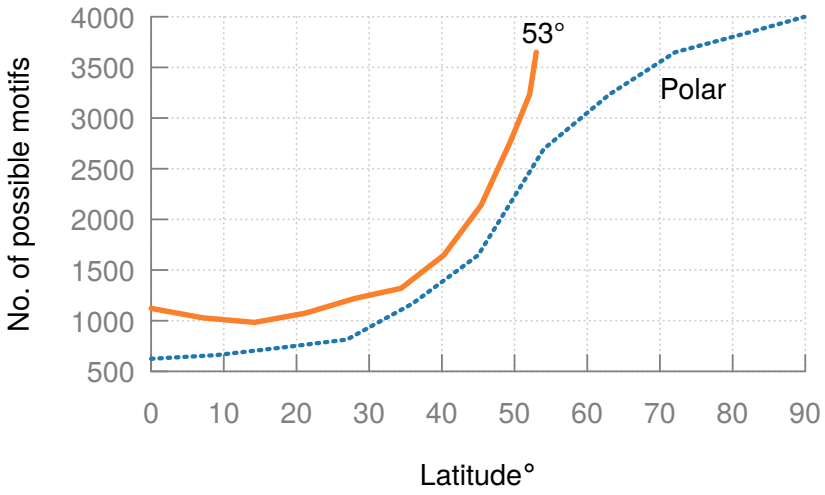


Figure 4.9: *The number of motifs possible increases with increasing latitude because of smaller inter-satellite distances.*

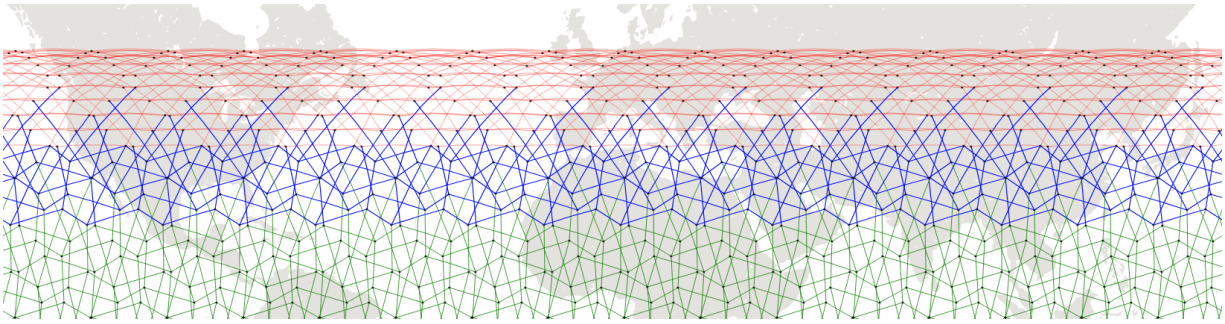


Figure 4.10: *Multi-motif for a* 40^2 *constellation with three* $18°$ *latitude zones. Connectivity South of the Equator is similar, and thus omitted; the Pacific region and higher latitudes are not shown either. This particular combination optimizes for* Φ_1*, and uses a maximum ISL length of only* 2,000 *km, much lower than the visibility limit of* 5,014 *km.*

4.4.1 Non-uniform satellite distances

For both polar and non-polar constellations, the density of satellites is not uniform across latitudes, as is clear from Fig. 2.1. This, in turn implies that at some latitudes, a larger set of satellites is reachable within a fixed ISL range, and consequently, more potential motifs. This observation is quantified in Fig. 4.9, which shows the number of motifs possible for satellites at different latitudes at an arbitrary time snapshot for two 40^2 constellations, one polar, and one with 53° inclination. The ISL range is fixed at 5,014 km throughout. The number of motif possibilities generally increase away from the Equator up until each constellation's limiting latitude. For the 53° constellation, the motif options increase from ∼1,100 at the Equator to ∼3,600 at 53°. Note that this density variation is not merely a temporal effect that is present in some time snapshots and absent in others – it is a persistent feature of these constellations stemming from their satellite trajectories, with minor temporal variations around this broader trend.

Using one uniform motif constellation-wide, as in §4.3, ignores the increased potential ISL choices at higher latitudes. We next explore how this observation may be exploited.

4.4.2 Exhaustive multi-motif search?

The success of exhaustive search across possibilities for constellation-wide uniform motifs (§4.3) prompts us to consider whether the same approach can be used to identify the best *combination* of motifs at different latitudes.

We can logically think of a circular satellite orbit as four quadrants, and using symmetry, restrict our exhaustive search to one quadrant. For a 40^2 constellation, 10 satellites are expected to be in such a quadrant at any

time. Per Fig. 4.9, each satellite has on the order of 10^3 motif choices. A combinatorial search through all combinations for all 10 satellites would thus involve on the order of 10^{30} multi-motif combinations (modulo some reduction due to consistency constraints across connections). Assessing a single combination requires calculating shortest paths and hop counts across its topology for large numbers of city pairs, incurring tens of seconds of compute with the *networkx* library [145]. While the problem is embarrassingly parallel, it is simply too large: we estimate that such an exhaustive search would require $\sim 10^{20}$ days even if 1 million compute cores were in use.

4.4.3 A coarser, iterative search

To overcome the complexity of exhaustive search, we use an iterative heuristic. We consider zones of a width, W, of a few latitude degrees. For instance, for a 53° constellation, if $W = 18°$, we have 3 zones: 0-18°, 18-36°, 36-53°. While we refer to only the positive latitudes for simplicity, the 18-36° zone (for instance) also covers latitudes from $-18°$ to $-36°$.

We consider each zone separately, starting with the first. Within a zone, the latitudes closest to the Equator determine the motifs possible, as satellites are farthest apart there. For each motif possible in the first zone, we evaluate its Φ_1 by populating the *entire* constellation with it, and keep the best.

To move to the next zone, we remove all ISLs from the constellation, except those connected to any satellite within the preceding zone. We again identify all possible motifs for the zone, and exhaustively evaluate performance with each motif populating the *rest* of the constellation, leaving links from previous zone(s) fixed. We repeat this process until each zone has been

used to augment the motif combination. One caveat is that, as we must adhere to the constraint of 4 links per satellite, this occasionally leaves some connectivity unused at some satellites near zone boundaries.

Fig. 4.10 shows the motif combination generated using $W = 18°$, and a maximum ISL length of 2,000 km.

4.4.4 Performance and link churn

Notice that as satellites traverse across latitude zones, now their connectivity changes, unlike with a single motif. But by design, the above procedure accommodates configuration of the degree to which such dynamic ISL changes are permitted, by setting the zone width, W. With $W = 18°$, a satellite changes its ISLs every ∼12 minutes, making an ISL setup overhead of a few or even small tens of seconds tolerable. Lower W (more zones) could potentially improve performance for static snapshots, but increases link churn, thus suffering in practice from link setup overheads. However, we find that performance improvements start to saturate after 3 zones ($W = 18°$): for 3 or more zones, Φ_1 improves by 7% over the single motif (one zone) while with 2 zones, the improvement is 5.6%. For a 40^2 polar constellation (instead of 53° inclination), multi-motifs with $W = 18°$ improve over a single motif by 9.5%. We remind readers that in the context of high-value investments like the satellite constellations, these improvements, while not as large as going from +Grid to the best single-zone motif, are nevertheless substantial.

We also find that multi-motifs cut variability over time. We examine minute-by-minute snapshots of the constellation built using the multi-motif approach with $W = 18°$ for a period of 2 hours. Fig. 4.11 shows that the multi-motif improves both median Φ_1, as well as the variation in Φ_1 across

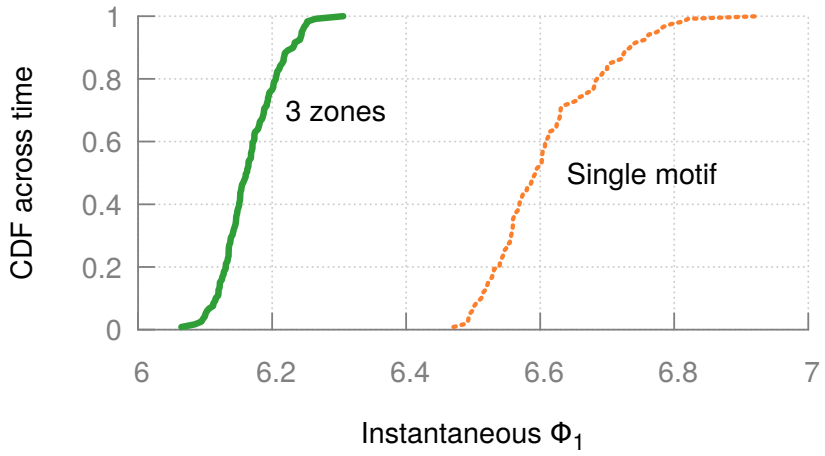


Figure 4.11: *Multi-motifs cut variance across time compared to one motif.*

time.

4.4.5 Performance for different metrics

An operator may prioritize stretch (S) or capacity (hop count, B). We thus evaluate our approach for M_1, M_5, M_{10}, (and correspondingly, Φ_1, Φ_5, Φ_{10}) where $M_\alpha = S\alpha + B$ (see §4.2.1).

Fig. 4.4 also shows, in addition to the average population-weighted stretch and hop count for the single motifs and the random graph, the same metrics for 3 multi-motifs (marked mm_1, mm_5, and mm_{10}), each optimized towards the three Φ_α's mentioned above. As the results show, the improvements are significantly beyond the Pareto frontier achievable with single motifs. For Φ_1, Φ_5, and Φ_{10}, the improvements are respectively 48%, 30%, and 20% over +Grid.

Base metrics: Through most of our discussion, to be able to work with one optimization criterion and for sake of brevity, we have compressed the stretch and hop-count objectives into the joint Φ_α metrics. It is however

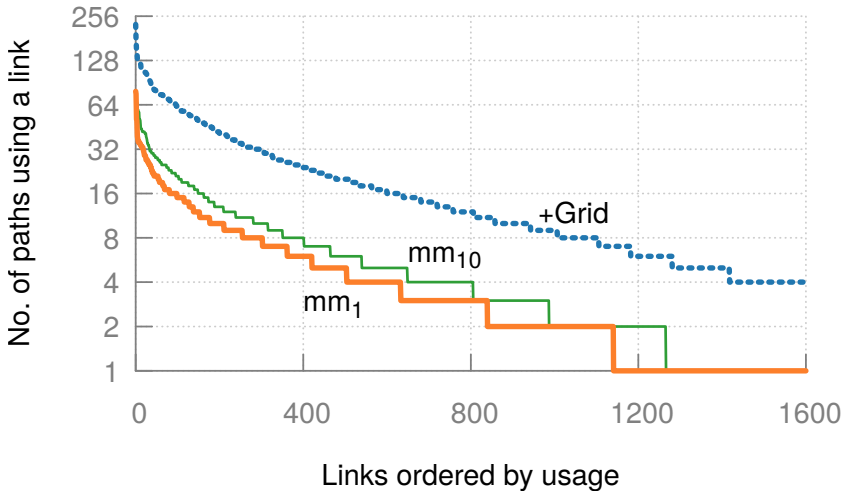


Figure 4.12: *+Grid congests ISLs more due to the higher hop counts.*

worth noting that higher α values understate the improvements, because changes in stretch, which is given a large weight, are smaller across a wide range of topology designs compared to hop-count, which varies more. For instance, the 20% improvement in Φ_{10} with mm_{10} is composed of 53% reduction in hop count, while increasing stretch by only 9%, compared to +Grid. As noted earlier in §4.3.3, even small compromises in stretch (*e.g.,* 2%) yield a large reduction in hop count (respectively 32%). The broader point worth emphasizing is that we can pick motifs or multi-motifs that optimize for a wide range of stretch and hop-count objectives.

4.4.6 Hop counts and congestion

Hop counts are an intuitive measure of network capacity, with fewer hops implying lower in-network capacity utilization per end-end connection. They are also easier to evaluate efficiently, and incorporate into optimization, than, *e.g.,* network throughput as measured with some routing scheme. Nevertheless, to alleviate concerns about the potential gap between hop

counts and congestion, we analyze congestion under shortest path routing.

For the same 3 multi-motifs as in §4.4.5, we analyze the levels of congestion assuming routing on lowest-latency paths. For each of 5,000 randomly selected city pairs, we compute lowest-latency paths between them, and count the frequency of appearance of each ISL in such paths. (This is referred to as "edge betweenness centrality" in graph theory.)

Fig. 4.12 shows the top quartile of ISLs sorted by their frequency of use (x-axis) against the number of paths using each ISL (y-axis). Compared to the multi-motifs, +Grid uses each ISL in many more paths, indicative of higher congestion. The 75^{th} and 90^{th} percentile link-use frequency for +Grid is 4× and 5× that of mm_1 respectively. Across multi-motifs, as we prioritize stretch more (mm_{10}), congestion increases; mm_5 lies between mm_1 and mm_{10}, and is omitted for clarity.

4.5 Optimizing Starlink & Kuiper

Our design methods are meant to be general enough for application to arbitrary future constellations, but we assess their effectiveness for the first shell deployments of SpaceX's Starlink ($S1$) and Amazon's Kuiper ($K1$) constellations.

SpaceX Starlink $S1$: In the context of this work, we stick to the FCC filing [48] according to which $S1$ will have 1,584 satellites, in 24 orbits ($i = 53°$, $h = 550$ km), each with 66 satellites. We evaluate our approach using both (a) the maximum ISL range, with only the visibility constraint, which is 5,014 km (§4.3.6); and (b) the minimum range necessitated by +Grid, 2,006 km. The latter represents worst-case power-limited ISLs.

Amazon Kuiper $K1$: Kuiper's first phase will be a 34^2 constellation

	Visible range ISLs		Min range ISLs	
	median M_1	Φ_1	median M_1	Φ_1
Starlink $S1$	52%	54%	**37%**	**40%**
Kuiper $K1$	38%	45%	1%	4%
$\mathbf{40^2}$	45%	48%	9%	7%

Table 4.1: *%-values are improvements over +Grid achieved by a 3-zone multi-motif. For Starlink $S1$, even min range ISLs yield large gains. Note that for $S1$, $o \times n = 24 \times 66$ following Starlink's past FCC filing [48].*

(51.9°, 630 km). In this case, the maximum and minimum ISL ranges are 5,440 km and 1,761 km respectively.

Results: Table 4.1 shows improvements in the M_1 (median across city-pairs) and Φ_1 (weighted M_1 sum) for Starlink, Kuiper, and a 40^2 constellation. The largest reductions in the metrics are for Starlink, where even with worst-case assumptions about ISL range, improvements of 37% (median M_1) and 40% (Φ_1) are achievable. This is due to Starlink's structure with fewer orbits and more satellites in each orbit. Fewer orbits imply that satellites in adjacent orbits are farther from each other, necessitating a higher minimum ISL range for +Grid, while a larger number of satellites per orbit imply an increased number of candidates for longer ISLs.

4.6 Limitations

The secretiveness of the industry, and several technology and market unknowns, pose challenges unique to such work:

- The satellites' ISL range and speed of link setup depend on a complex calculus involving non-networking factors like satellite weight and launch cost, making it hard to zero in on the inputs for topology design.

- Market conditions, regulator oversight, and terrestrial connectivity, will together drive workloads, making it impossible to evaluate for *the* right traffic matrix.
- Space endeavors are prone to setbacks and changes, so the setting we are studying is evolving. For instance, over the course of our work, SpaceX updated its Starlink plans to use lower and different orbits to address concerns about space debris. The plans we use were up to date as of July 2019.

However, these uncertainties are poor arguments for not addressing the technical challenges. There is, after all, a potentially short and closing window for influencing the design of the planned constellations. We thus attempted to address the uncertainties by accounting for a broad set of likely inputs, *e.g.,* best-case to worst-case ISL range, slower or faster link setups, and two intuitive traffic models. At the very least, even with the most conservative assumptions (*i.e.,* worst-case ISLs, single motifs with no link changes), our work shows significant promise beyond the +Grid strategy widely assumed to be the default.

4.7 Brief introduction to LEO trajectory design

While we thus far assumed trajectory design parameters to be given, following FCC and ITU filings by the constellation providers, and focused on topology design, we should note that trajectory design in itself is an interesting optimization problem. If one can model the complex physical constraints (including launch sites, deployment, de-orbiting, collision avoidance, etc.), which is difficult to achieve at this time given the scarcity of publicly available information, one could try to tune the trajectories of

satellites for improving network performance. Here I present a brief, first-cut analysis of LEO trajectory design.

In order to understand the impact of different trajectory design parameters on LEO network performance, we pick a fixed budget of satellites (1,600 similar to Starlink $S1$'s budget of 1,584 and the uniform 40^2 constellation) to deploy in a single shell and vary the number of orbits (o; and hence also the number of satellites per orbit n), inclination (i), and the minimum angle of elevation (e) and perform a grid search over the entire design space. As discussed in the background in Chapter. 2, we assume height to be fixed (as operating height depends on complex non-networking factors) at 550 km following Starlink $S1$, and phase offset between adjacent orbits, p, to be fixed at 0.5 for uniformity across time. Uniformity maximizes the coverage over time for each constellation, which is a desirable characteristic.

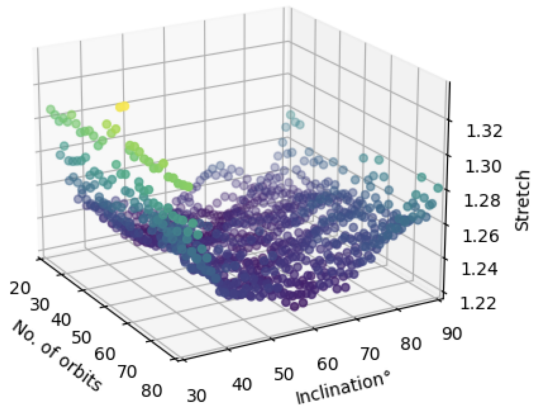
Ranges of parameters. Following the discussions in Chapter. 2, i varies between 26° (Boca Chica at 25.99°N) and 90° (prograde orbits) at a granularity of 2°, and e varies between 10° (Telesat's minimum angle of elevation) and 80° (angles higher than this are impractical due to very narrow cones of coverage per satellite) at a granularity of 5°. The number of orbits, o, varies between 20 and 80 in order to offer continuous coverage along great circles (along the Equator and also along any orbital plane). The number of satellites per orbit, n, varies accordingly, such that the budget does not overshoot beyond 1,600. For each configuration, we assume the topology to be the default +Grid, and compute the following performance metrics.

Metrics. For each configuration, we compute **weighted average stretch**, as discussed above, across all city-pairs, the weights being population products. For each city, we also compute the number of satellites visible from that location at epoch. In order to take into account path loss, for each

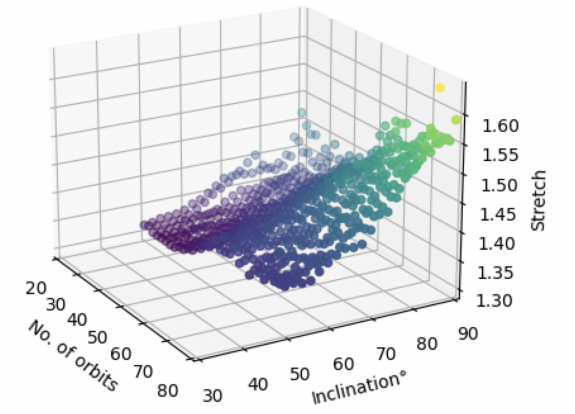
satellite we first find out all cities (γ) within its cone of coverage, divide the up-link (1 unit) equally among all cities (equals $1/\gamma$). The final up-link share assigned to a ground station at a line-of-sight distance of d_s from a satellite operating at a height h is modeled as $(1/\gamma) * (h^2/d_s^2)$. The **available upload** per city is the aggregated up-link share for the location across all visible satellites. For each configuration, we compute the weighted aggregated upload across cities, with the weight being population. Note that hop-count is uninteresting for the default +Grid connectivity, and hence is not included in the trajectory analyses.

Stretch varies across configurations. Fig. 4.13(a) plots weighted average stretch across different values of o and i when $e = 10°$ (Telesat proposal). It is interesting to see stretch values to be lower for i in the range [50 - 70]. For higher inclinations, satellites are sparser at lower and mid latitudes thus offering sub-optimal paths to cities in these locations. For very low inclinations, many city-pairs at mid-latitudes need to first connect to the constellation further south (in the northern hemisphere) thus inflating path latencies. The minimum stretch of 1.22 is achieved for $i = 54°$ and $o = 28$. For e fixed at 25° (similar to Starlink $S1$), Fig. 4.13(b) also shows a similar trend, with stretch increasing for higher inclinations. The minimum stretch of 1.3 is attained for $i = 52°$ (similar to Starlink $S1$'s 53°) albeit for $o = 28$ (for Starlink $S1$, currently $o = 72$ [56], while a past proposal had $o = 24$ [48]).

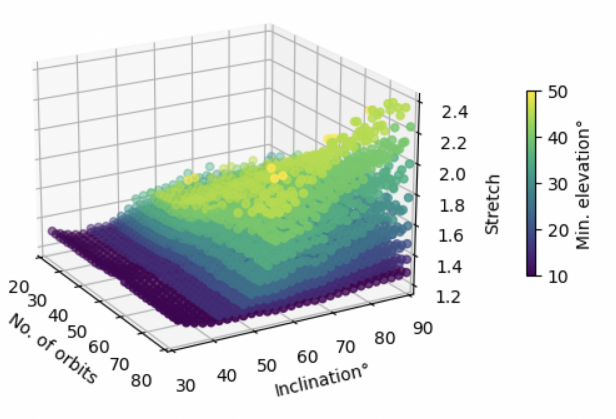
Also note that stretch increases as we move from $e = 10°$ (Fig. 4.13(a)) to $e = 25°$ (Fig. 4.13(b)). This is more evident in Fig. 4.13(c) where e is color-coded. The layers with different colors indicate how stretch increases with e. Interestingly, the layer structure in the plot also highlights how, for a fixed e, stretch increases with o as well as i. The impact of higher o



(a) Min. angle of elevation fixed at 10°



(b) Min. angle of elevation fixed at 25°



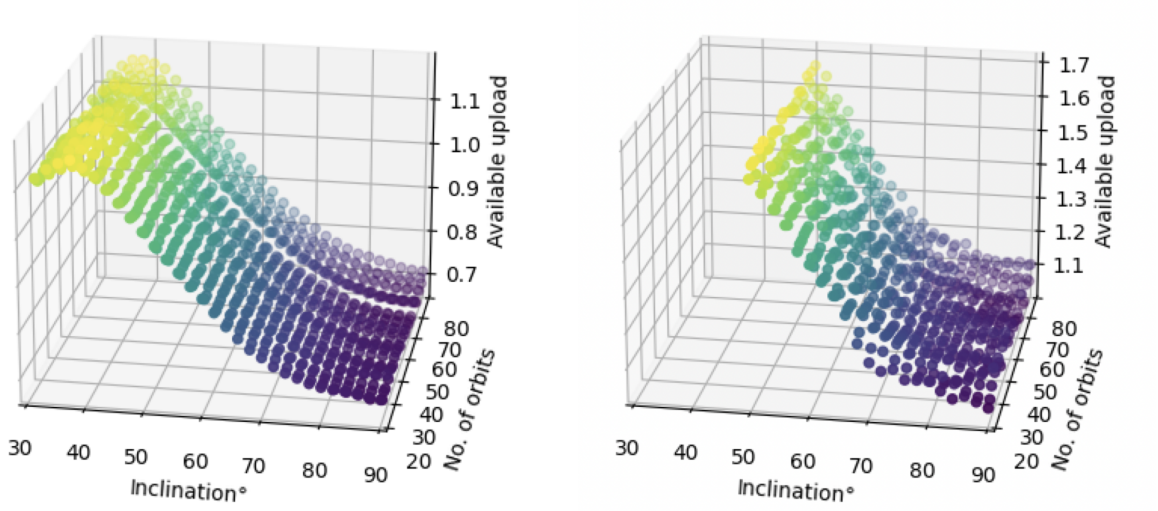
(c) Min. angle of elevation color-coded

Figure 4.13: *Stretch varies for different values of o, e, and i. For (a) and (b) stretch is color-coded, while for (c) e is color-coded. Warmer colors represent higher values.*

(or o/n ratio) with a fixed budget of satellites is also quantified in the next chapter in §5.4.1.

Let us re-iterate the takeaways. For the 100 most populous cities, low stretch is achieved by operating at a low minimum angle of elevation, inclination varying between 50° and 70°, and having lower number of orbits (o) and higher number of satellites per orbit.

Available upload varies across configurations. It is interesting to see



(a) Min. angle of elevation fixed at 10°

(b) Min. angle of elevation fixed at 25°

Figure 4.14: *Available upload varies with inclination and min. angle of elevation.*

the changes in available upload for different values of i and e. Fig. 4.14(a) and Fig. 4.14(b) plots available upload for e fixed at 10° and 25° respectively. We observe that this metric does not depend on the ratio of the number of orbits o and the number of satellites per orbit n. For $e = 10°$ the peak upload is achieved at $i = 38°$ while for $e = 25°$ (similar to Starlink $S1$) the peak is achieved at $i = 50°$ (similar to Starlink $S1$'s 53°).

We further explore in Fig. 4.15 how the available upload varies across different values of i and e for Starlink $S1$'s latest 72×22 configuration. We see that neither a very low $e = 10°$ nor a high $e = 40°$ offers available upload as high as with $e = 25°$ as in Starlink $S1$. Also note how the coverage across 100 most populous cities varies for different values of i and e. For offering coverage to all 100 cities, a lower i (32°) has to be combined with a low e (10°), while a higher i (56°) could be combined with a high e (40°). As also evident in Fig. 4.15, certain combinations of i and e (like $i = 50°, e = 40°$) do not offer full coverage.

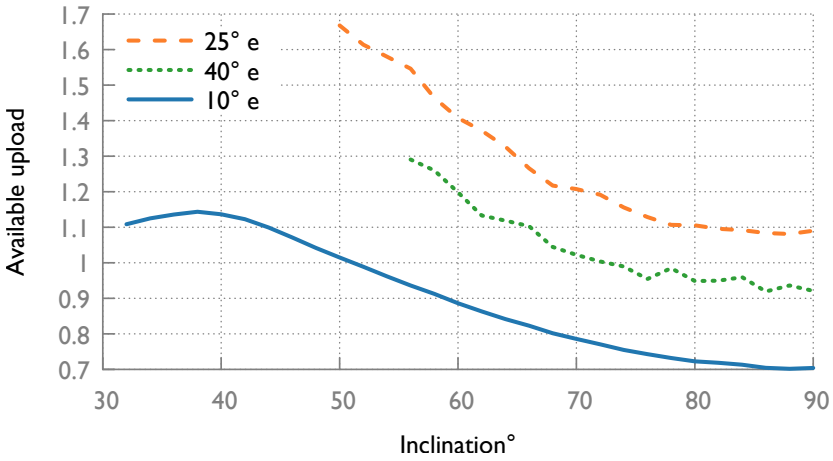
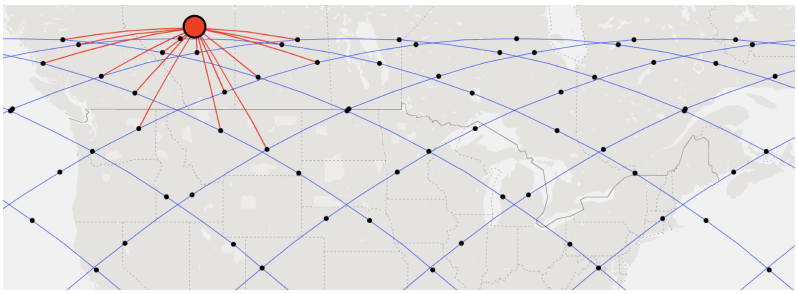


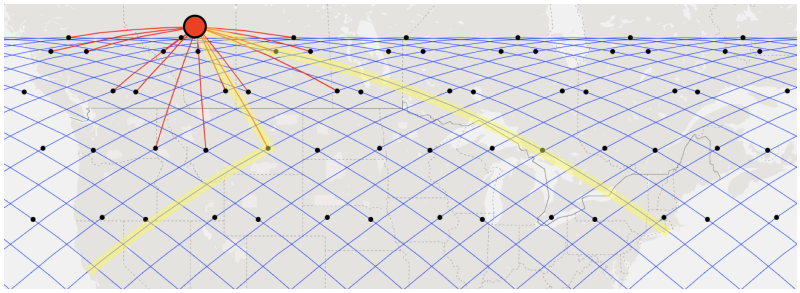
Figure 4.15: *Different coverage and available upload for different combinations of inclination, i, and min. angle of elevation, e. Given the geographical distribution of* 100 *cities, some of which are at higher latitudes, some combinations of i and e does not offer full coverage.*

For the 100 most populous cities, high available upload is achieved by operating at a moderate minimum angle of elevation ($e = \sim 25°$) and inclination within a range that depends on e (usually the lower the better, modulo any coverage constraint). In our analyses, we did not find any significant impact of the o/n ration on available upload.

Why is Starlink $S1$ 72×22? Starlink recently updated their $S1$ shell configuration ($o \times n$) from 24×66 [48] to 72×22 [56]. This might seem to be counter-intuitive given our results show lower o/n is better for stretch. We speculate on one possibility that might have driven Starlink towards this design. Constellations might [146] need to turn-off inter-orbit ISLs at higher latitudes closer to the inclination i due to higher relative velocities between satellites in adjacent orbits. Fig. 4.16 depicts the influence of this constraint. With a 24×66 shell, Edmonton can connect to 16 satellites spread across only 6 orbits. With a 72×22 shell instead, Edmonton still



(a)



(b)

Figure 4.16: *With more orbits than shells, higher latitudes can experience greater path diversity. Edmonton, Canada can connect to (a) only* 6 *Starlink*-1 *orbits for* $o \times n = 24 \times 66$ *compared to (b)* 17 *orbits for* $o \times n = 72 \times 22$.

connects to a similar number of satellites over time but spread across 17 orbits. Considering the connectivity from Edmonton to either the west coast or the east coast of the US, we find that the 72×22 Starlink-1 shell offers higher path redundancy than that possible with the 24×66 shell. A more in-depth path-redundancy analysis is left to future work.

4.8 Related past work

Topology design. A prior wave of interest in satellite networking in the 1990s spurred substantial academic work [62, 147–159], of which we only discuss the most relevant. Wood (2001) [62] discusses trajectory design and +Grid connectivity, which we compare against. Gavish and Kalvenes [146]

discuss 4 hand-designed variants of +Grid connectivity. These address a specific problem in polar "π constellations", where across a continuous half of the Equator, all satellites travel northward, and in the other half, all southward, with connectivity challenges at the boundaries between the two halves ("seams"). In contrast, the modern constellations we tackle are all uniform ("2π") and do not present this particular topology problem. Further, the density of the new constellations presents many more design choices than are open to manual design. As we point out (§4.3.5), the nature of the problem changes entirely for dense constellations, where far from feasibility of connectivity being an issue, the choices for connectivity become overwhelming, making such design-by-hand prohibitive.

A recent effort [63] compares the design of 3 recently proposed constellations — Starlink, Telesat, and OneWeb, discussing how the number of GSes would affect aggregate throughput offered by such systems. The analysis assumes +Grid ISL connectivity, and can be viewed as making a case for our work: the authors show that if ISL link capacity were increased (via technology changes), it would boost system throughput substantially for each constellation. Note that the same positive outcomes can be achieved by making efficient use of ISLs of a given capacity/cost, which is our approach to the problem. Another effort [160] tackles connectivity between pairs of small, resource-constrained CubeSats; this is a different problem addressing only pairwise CubeSat-CubeSat connectivity, not Internet connectivity *via* satellites.

Mobile connectivity has been studied in other contexts, including high-speed rail [161], drones [162], and planes [93]. But none of these simultaneously feature all the peculiarities of our setting: **(1)**predictable motion of **(2)**thousands of **(3)**high-speed systems **(4)**connected to each other with

multiple links to $^{(5)}$provide connectivity between fixed endpoints. The problem itself differs fundamentally from past work, and naturally drives us to different solutions as well.

In graph theory, motifs and "graphlets" are well studied, but from a different perspective: *identifying* repetitive patterns of connectivity in given graphs [163–167]. Of course, even in terms of design, the utility of repetitive patterns is long understood, *e.g.,* in areas like visual and graphic design [168]. To the best of our knowledge, we are the first to systematically apply such ideas to satellite network design.

Trajectory design. While past work [169, 170] explored LEO trajectory design, the optimization goals during the late nineties were different – to achieve continuous coverage over an intended geographic area, to maximize the time for which inter-satellite connectivity was feasible, etc. The new scale of deployments today makes these goals almost trivial (still needs the right trajectories though) to achieve, given the large number of LEO satellites that could be deployed due to much lower deployment costs, thus allowing one to focus more on network performance rather than connectivity.

4.9 Related publications

The plots on LEO topology design and the corresponding discussions in this chapter have been taken from the following publication:

- D. Bhattacherjee and A. Singla, "Network topology design at 27,000 km/hour," in *ACM CoNEXT*, 2019

This work received IRTF's Applied Networking Research Prize, 2020.

5

Enabling LEO research with Hypatia

Large LEO constellations promise global coverage at low-latency and high-bandwidth. However, realizing the full potential of these networks requires addressing new research challenges posed by their unique dynamics. The high-velocity movement of satellites creates not only high churn in the ground to satellite links, but also fluctuations in the structure of end-end paths as the satellites comprising the paths move. Progress in precisely fleshing out networking challenges that arise due to this unforeseen dynamicity and addressing them faces a substantial roadblock: lack of network analysis tools that incorporate the dynamic behavior of LEO networks. This creates a substantial risk that instead of networking research laying out the potential future trajectories for the industry, research will rather lag the industry's rapid strides. Thus, to help accelerate research on LEO networks, we developed **Hypatia**[1], an analysis framework with simulation

[1]The name is a tribute to an early leader in astronomy and mathematics, who is better recognized as a commentator and teacher, rather than for her new inventions, in line with the spirit of this work.

and visualization modules. Hypatia provides a packet-level LEO network simulator based on ns-3, as well as several types of network visualizations based on Cesium [172], that serve to provide intuition about such networks.

More precisely, we make the following contributions:

- We lay out the case for building network analysis tools for upcoming LEO networks. As a first step towards meeting this need, we develop Hypatia, an analysis framework capturing the orbital dynamics of LEO networks.
- We use regulatory filings by the largest three planned LEO networks to evaluate and visualize their networks.
- Using packet-level simulations, we analyze the behavior of individual end-end connections across such networks in terms of their changing latencies and path structure, and show how this impacts congestion control negatively, even in the absence of any competing traffic.
- Further, by simulating traffic constellation-wide, we show that the changes in path structure result in a difficult problem for routing and traffic engineering, as the utilization of paths and links is highly dynamic.
- Hypatia's visualizations aid intuition about the structure of satellite trajectories and their impact on a constellation's behavior, and pin-point traffic hotspots in the network and show their evolution over time.

Satellite networking played an important role in laying the foundations of the Internet, and may again provide the impetus for substantial and exciting changes. We hope that Hypatia will serve as an enabler for that work. Hypatia's source code is available online [173], together with our visualizations [174]. This chapter addresses the following goal:

Goal C: Enabling broader research by building the right analysis tools.

5.1 Related work & the dearth of analysis tools

Commercial satellite networks already provide varied network services. HughesNet [12] and Viasat [13] primarily serve areas poorly connected by terrestrial fiber, as well as aircrafts and ships. These are both GEO satellite constellations, and operating at 35,786 km, they incur hundreds of milliseconds of latency. Besides, their performance and service goals being different, their GEO satellites are, by definition, stationary with respect to the Earth, and thus do not feature LEO dynamics. Iridium [20, 21] operates in LEO, but primarily offers satellite telephony rather than broadband Internet. Iridium, with 82 satellites in operation, is the largest of the networks that pre-date the new LEO mega-constellations.

Thus, as already discussed in Chapter. 1 & 2, no prior networks have all the features of the new LEO networks, the largest of which are planned to operate thousands of satellites instead of tens, and provide mass market low-latency broadband Internet, rather than niche services. Starlink already has 1,600+ satellites operational and has started offering limited beta-service since 2020 [6, 175]. Over the long-term, such networks have the potential to fundamentally change the Internet, making it crucial for research to keep pace with the hectic pace of industry developments.

The networking community, recognizing this need, is indeed ramping up research in this direction. While there is a large body or earlier work from the 1990s on GEO and small LEO networks [62, 147–159], several position papers [51, 94, 176] have highlighted the *new* opportunities and challenges of mega-constellations, *e.g.,* in intra-constellation routing [51] and inter-

domain routing [176], and end-end congestion control [94]. Followup work has since laid out novel proposals for topology design [54] and Internet inter-domain routing [177] in this context.

Unfortunately, the networking community lacks the right tools to attack many of the LEO networking challenges recent work has pointed out. We need software to simulate the behavior of such networks, so that we can deeply understand the problems, and new research ideas can be evaluated. Understanding the packet-level behavior of a network is obviously important for congestion control research, but ultimately, practitioners also want to evaluate routing and topology work in terms of how it impacts network packets, *e.g.,* do some routing schemes cause more packet reordering, and thus, ultimately result in poor performance?

Unfortunately, there is no simulator that fully addresses these needs. SNS3 [178] models GEO satellite communication channels, but does not support LEO satellites or inter-satellite connectivity. Another simulation effort [179] focused on the polar constellations of interest in the nineties, and the problems of interest therein, *e.g.,* connectivity across "seams" that result from satellites traveling northward in one (longitudinal) hemisphere and southward in the other. While we could have extended this work for our study of modern LEO networks, we based Hypatia on the ns-3 platform to benefit from its more active development and support. Note that this prior work also did not analyze congestion control and traffic engineering, nor did it provide visualizations beyond the below-discussed SaVi tool [180]. A satellite mobility model is available for ns-3 [181], which can convert satellite trajectories in a specific format into a coordinate system compatible with ns-3. This capability is useful, and we build on it by adding models for inter-satellite and GS-satellite connections. Recent work on LEO inter-satellite

topologies [54] evaluated topologies only in terms of path hop-counts and distances, not packet simulations. Likewise, work on inter-domain routing [177] only modeled the network control messages and path distances. Another effort [63] estimates the throughput of new LEO networks using statistical methods, and minimizes the number of GSes needed to support the throughput. It does not account for network routing and transport dynamics.

We also need visualizations that help build sorely missing intuition for these new networks. While there are many beautiful visualizations, at least for Starlink [108, 182–184], most of these do not focus on networking concepts such as the evolution of paths, utilization, and congestion. The closest related work [51, 108] does not simulate packet-level behavior, and does not provide source code for its path-granularity computations or visualizations. NASA's *GMAT* [185] can be used to visualize trajectories of objects in space; *SaVi* [180] can additionally render coverage of a satellite. However, neither provides the ability to define the topology, model network links, or run network-centric measurements.

While we expect that eventually the community will collect measurements from real clients on LEO networks, this will not alleviate the need for simulation and analysis tools. For a variety of network contexts, such tools continue to be valuable to understand existing phenomena, and to devise novel, hard-to-evaluate-in-the-wild techniques.

5.2 Hypatia architecture

To address the urgent need for tools that enable research on LEO networks, we built Hypatia. Hypatia provides a packet-level simulator that incorporates LEO dynamics, and a visualization module to aid intuition. The

packet simulator is implemented as a module for ns-3 [186]. It takes into account satellite trajectories, coverage constraints for GS-satellite connectivity, and the structure of inter-satellite connectivity. It can be used to implement and evaluate novel ideas for satellite trajectory design, inter-satellite topology, routing, and congestion control. The visualization component uses Cesium [172] to render views of the trajectories, GS-perspective on overhead satellites, end-end routes, evolving link utilization, and available bandwidth on routes.

5.2.1 Setting up a simulated LEO network

At its simplest, Hypatia allows users to specify satellite trajectory parameters and ground station locations. From these, it automatically generates the state of each satellite over time in a space-industry standard data format, the GS-satellite and ISL connectivity, and time-varying forwarding state that decides the paths packets take. We discuss what parts users need to modify for more complex simulations.

TLE generation: A **t**wo-**l**ine **e**lement is a standard format for representing the trajectory of an Earth-orbiting object [187]. For existing satellites and orbital debris, NORAD [188] regularly publishes TLEs [189]. These TLEs are an input dependency for the satellite mobility model we build on. This arrangement has thus far sufficed for ns-3's limited use in this setting: studying connectivity with one existing satellite.

However, this meant that we needed to ourselves generate TLEs for satellites that are not yet in orbit, but for which we know orbital parameters in terms of the Keplerian orbital elements [50] from the FCC or ITU filings made by the operators. Table. 2.1 shows the values we obtained from these filings. We only include a simplified subset of the parameters in the table;

the remaining ones can be easily derived from the symmetries in play, *e.g.,* only using circular orbits [36, 58], satellites in one orbit being uniformly spaced out, and orbits being uniformly spread across the Equator.

We built a utility that accepts Keplerian orbital elements as input, and outputs TLEs in the WGS72 world geodetic system standard [190]. To test that the output TLEs specify the same constellation as the input Keplerian orbital elements, we use *pyephem*, a Python library that can generate constellations from either the Keplerian elements or TLEs.

ISL connectivity: A large body of work in satellite networking indicates a typical +Grid connectivity pattern, as discussed in §2.2.1, for a satellite with 4 ISLs, forming a mesh-like network [51, 63–67, 107–110]. We also use +Grid as the default ISL interconnect in Hypatia.

Alternative ISL interconnects, like the motif-based topologies discussed in Chapter. 4, can be trivially supported in Hypatia. Nevertheless, we focus on +Grid interconnects in this Chapter without loss of generality, and highlight some critical LEO networking challenges that need to be addressed.

GS-satellite connectivity: We currently simulate only static GSes with multiple parabolic antennas, not user terminals with single phased-array antennas that can be mobile [138]. However, Hypatia can be easily extended to model such terminals. Hypatia inherits from ns-3 the ability to impose sophisticated models on the GS-satellite channel, *e.g.,* for loss. Nevertheless, Hypatia's current implementation makes several simplifying assumptions about the GS-satellite links:

- Hypatia supports multiple GSL (ground-satellite link) network devices per satellite and GS. As default in our experiments, we set one GSL network device for both satellites and ground stations. Each network device can send packets to any other GSL network device, as long as

the forwarding plan allows it. Additional connectivity restrictions can be imposed, *e.g.,* to restrict user terminals such that they can only connect to one satellite at a time.

- Across satellites and ground stations, no connections interfere with each other. While this is a strong assumption, Starlink and Kuiper mention [36, 58] that frequency management will be software-defined and done online to optimize towards this goal.
- Each GS can be configured to either: (a) connect to multiple satellites; or (b) connect to its nearest satellite.
- Since many loss-free handoff techniques are known for other mobile settings, when GS-Satellite connections are handed off, there is no loss. Hypatia delivers in-flight packets from the now out-of-reach satellite, while new packets stop arriving at it.

We make these simplifications, which relax practical constraints and are favorable to LEO networks, for two reasons: (a) this framework suffices to draw out many of the challenges; and (b) doing anything else requires substantial *design* work that is not within our scope, *e.g.,* frequency management for this setting will likely be studied extensively in future work, for which Hypatia can serve as a vehicle.

Forwarding state: We compute the forwarding state of satellites and ground stations at a configurable time granularity, with the default being 100 ms. Note that this step converts what is necessarily a continuous process of satellite motion into discrete intervals where we check and update the forwarding state. In between these intervals, the latencies are correctly calculated based on satellite motion, but the paths being used may deviate from the shortest. We discuss the implications of this experimentally in

§5.4.3.

For every time interval, we use a *networkx* [145] module to generate the network graph, accounting for satellite positions and link lengths between satellites and to ground stations. On this graph, the forwarding state for each node can be calculated based on arbitrary routing strategies. Our current implementation simply uses shortest-path routing, computed with the Floyd-Warshall algorithm. The forwarding state changes are also added into ns-3's discrete event queue: at the first time the event fires, it reads new forwarding state into static routing table entries, and then adds the next forwarding state change event at exactly the configured time interval. Any routing strategy implementable with static routes can be easily supported. This is also true for multi-path routing, but obviously, would require additional logic to be implemented for splitting traffic across these paths.

5.2.2 Running packet-level simulations

The packet-level simulator can be used to run simulations of LEO satellite networks for arbitrary satellite trajectories, GS locations, routing, congestion control, and queuing. While the generated TLEs for satellites, and routing and forwarding states are pre-computed and fed to the simulator, it is responsible for simulating the mobility of satellites and thereby accounting for varying link latencies over time. For this purpose, we adapt an available ns-3 satellite mobility model [181]. While this model adds a 1-3 km error per day to satellite trajectories, this can be ignored safely for simulations that simulate less than a few hours, as the networking implications of these distances are minimal.

5.2.3 Post-processing and visualizations

Hypatia's ns-3 module can simulate both UDP and TCP traffic and log the relevant metrics for each transport, including RTTs, congestion window, and application level flow progress over time. We use *gnuplot* to generate all plots included in this chapter, and *Cesium* for visualizations. *Cesium* is a general-purpose 3D mapping library for Javascript. We extend it to render the following interactive visualizations, writing python code that takes the outputs of our simulations and generates the visual elements for Cesium to render.

- The satellite trajectories over time.
- The ground observer view over time, showing the satellites visible in the sky at different angles of elevation.
- Changes in end-end paths over time.
- Changes in link utilization and available bandwidth on an end-end path over time.

5.2.4 Simulator scalability

For testing the scalability[2] of Hypatia, we measure the network-wide goodput (acknowledged data rate for TCP; data arrival rate for UDP) across a random permutation of GS pairs picked from the 100 most populous cities for varying line rates. We run this experiment on a single-core Intel Xeon $L5520$ operating at 2.26 GHz.

[2]This experiment was primarily conducted by my collaborator, Simon Kassing, and hence I do not include here the corresponding plot and analysis which are available in the paper [191]. Instead, I only summarize the results here for completeness.

In order to simulate 10 Gbps UDP goodput for 10 virtual seconds, the simulations need to run for ∼33 real (clock) minutes. For TCP traffic, this is ∼100 real minutes. For the same clock-time budget, a lower goodput could be simulated for longer virtual period, and vice-versa.

The constellation scale does not impact simulation scalability, as the cost to setup the network and generate forwarding states is incurred as part of pre-processing (§5.2.1). This cost is incurred only once for each constellation and routing strategy irrespective of the number of simulations. The simulations are primarily bottlenecked by ns-3's per-packet event processing. A minor overhead is incurred in order to accommodate the link latency changes over time due to satellite mobility.

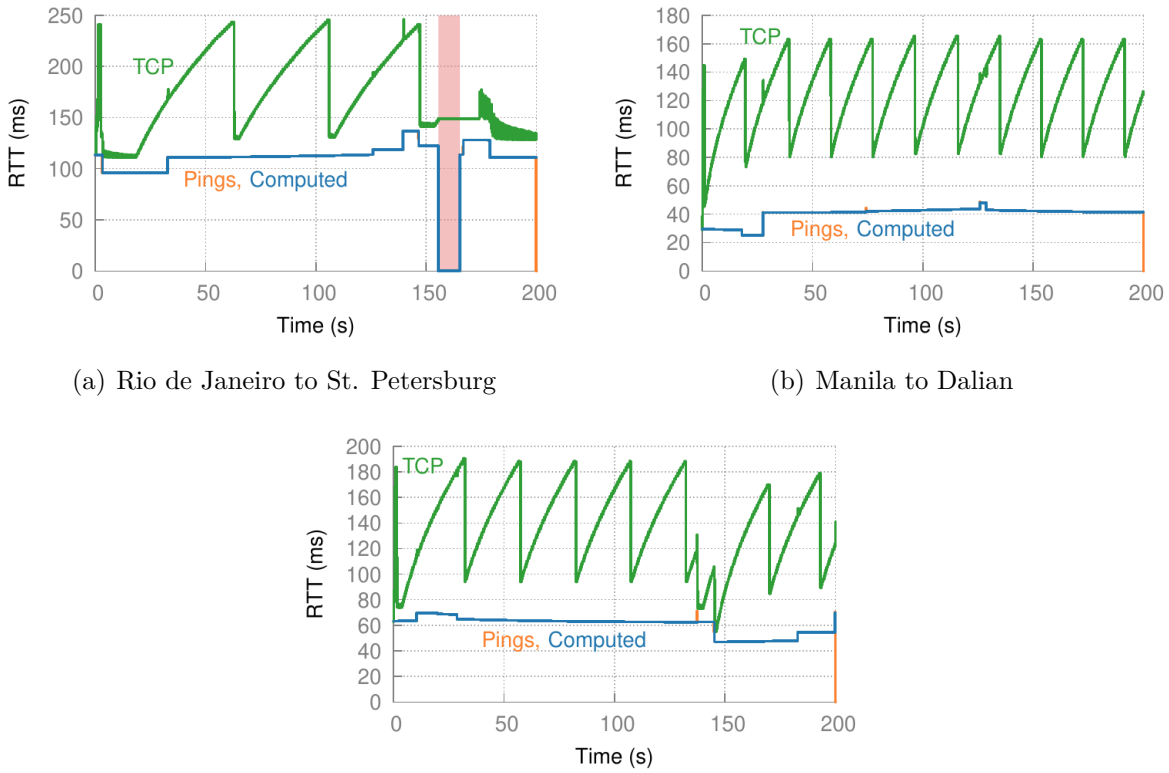
5.3 Examining a few LEO paths

We first analyze connectivity between a few GS-pairs in depth to give a view of how an end-end connection behaves.

5.3.1 RTT fluctuations

We examine how the end-end RTTs vary over time. These experiments use the Kuiper K1 shell. We run the analysis for 200 seconds, as for Kuiper-scale networks this is sufficient to show nearly the full range of variations.

For each source-destination pair, *s*-*d*, *s* sends *d* a ping every 1 ms, and logs the response time. We also compare the measured RTTs to those generated using *networkx* computations for the same end-points, and the same constellation. For these *networkx* computations, we use snapshots of the system every 100 ms, and compute the shortest paths using the Floyd-Warshall algorithm. Analysis based on such computations has al-



(a) Rio de Janeiro to St. Petersburg

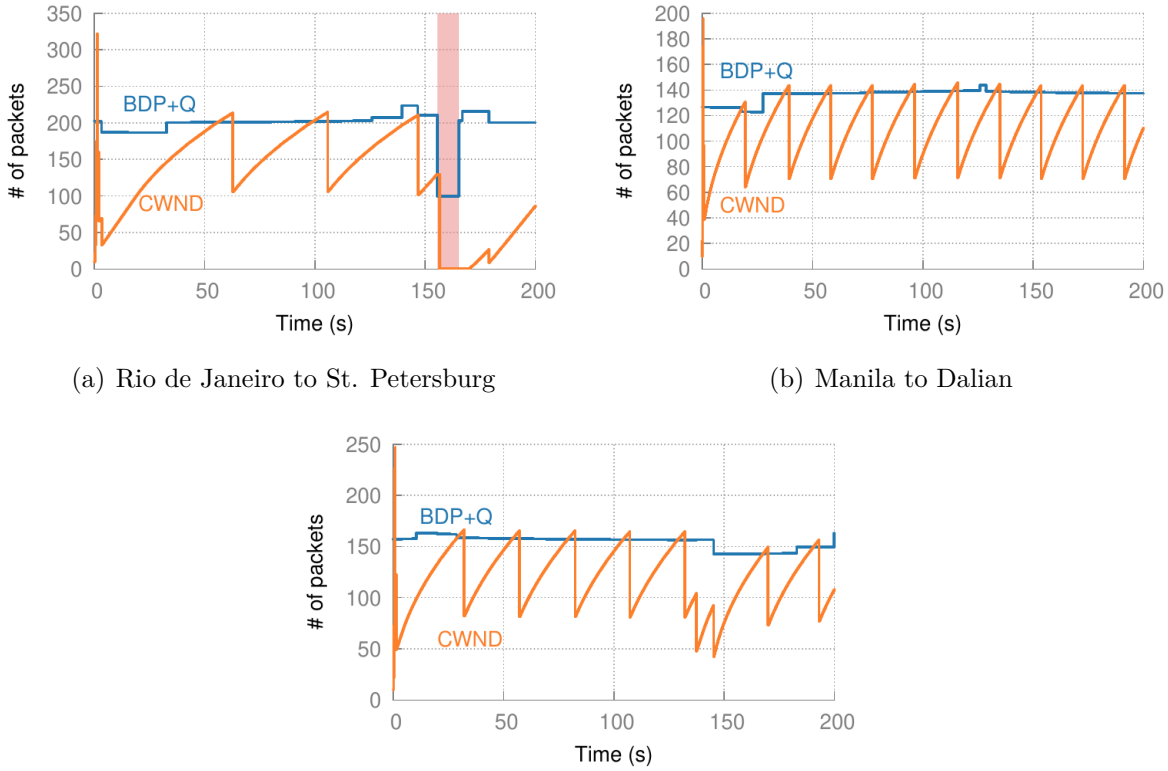
(b) Manila to Dalian

(c) Istanbul to Nairobi

Figure 5.1: ***RTT fluctuations.*** *RTTs calculated by networkx and measured in our simulator using pings match closely, with the lines almost entirely overlapping, as shown for* 3 *paths. The TCP per-packet RTTs are also shown, measured in the absence of any other traffic except the source-destination pair. The queue size is* 100 *packets, i.e., approximately* 1 *BDP for* 10 *Mbps and* 100 *ms. Note: The last few pings' RTT is shown as* 0 *due to them not yet returning back in time to give a valid RTT measurement.*

ready appeared in recent work [51, 94]; we use it both as a validation for some of our simulator's satellite-specific code, and to highlight and explain the subtle differences that actual packets sometimes experience compared to paths computed from a static snapshot.

Fig. 5.1 shows the results for three *s*-*d* pairs. The ping measurements from Hypatia ('Pings') and the snapshot computations from *networkx* ('Computed') match closely for most of the time. For instance, in Fig. 5.1(a) at $t = 32.9$ s the path changes, which causes the RTT to rise from 96 ms to



(a) Rio de Janeiro to St. Petersburg

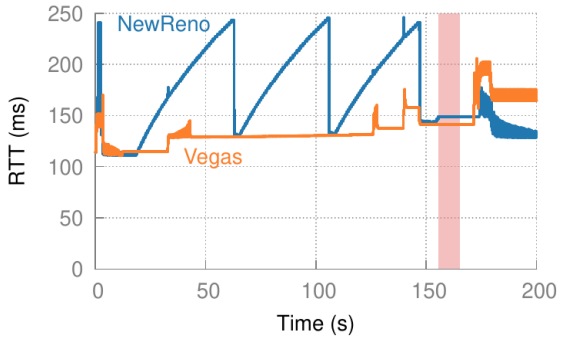
(b) Manila to Dalian

(c) Istanbul to Nairobi

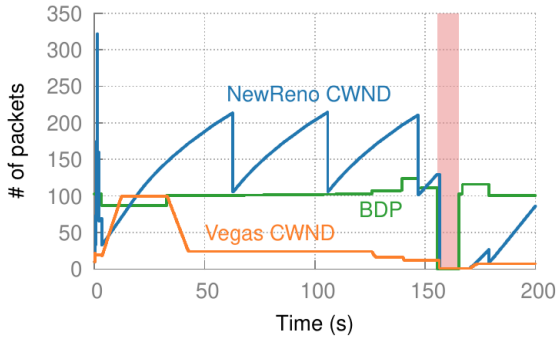
Figure 5.2: ***TCP congestion window evolution.*** *As expected, the congestion window typically fluctuates between the BDP and BDP plus queue size (*100 *packets). However, in certain cases, when the RTT gets lower, reordering happens, and even though there is no loss, the congestion window is still halved.*

111 ms. Occasionally, like in Fig. 5.1(c) around 130 seconds, we see spikes in the ping RTT compared to *networkx*. These spikes result from forwarding state changes across the path: as a packet travels on what was the shortest path when it departed the source, the packet arrives at some satellite no longer on the new shortest path, as satellites have moved. This results effectively in the packet having taken a detour compared to the instant path computation in *networkx*.

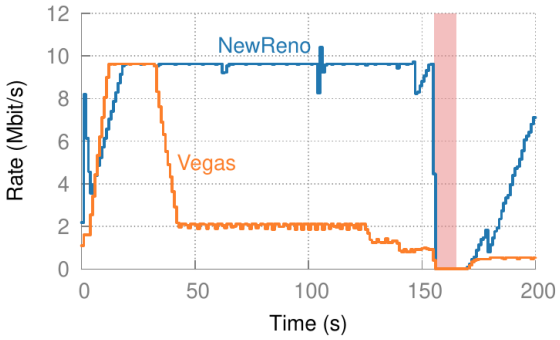
The path from Rio de Janeiro to St. Petersburg sees a disruption around 150 seconds into the simulation, shown as the shaded region in all related



(a) NewReno causes high RTTs.



(b) Vegas decreases CWND on RTT increase.



(c) Vegas' throughput collapses.

Figure 5.3: ***Both loss- and delay-based CC suffer.*** *As seen here for the connection from Rio de Janeiro to St. Petersburg, while loss-based congestion control (NewReno) fills up queues, delay-based congestion control (Vegas) infers increasing delay as congestion and collapses in throughput. This happens at* 35 *s, and from then on, throughput stays low for Vegas.*

plots. We found that for this period, St. Petersburg does not have any visible Kuiper satellites at sufficiently high angle of elevation, which, obviously, results in the satellite network path being disconnected. For Kuiper, its other two shells do not address this missing connectivity either; high-latitude cities like St. Petersburg will not see continuous connectivity over Kuiper.

For the other two paths, there are smaller but still substantial variations in the RTT over time. Across time, the Manila-Dalian path has a minimum

RTT of 25 ms and a maximum RTT of 48 ms, thus changing by nearly 2×. For the Istanbul-Nairobi path, this RTT range is 47-70 ms.

For real-time applications that care about jitter, these variations could necessitate a somewhat large "jitter buffer" to store and deliver packets to the application at an even rate. The determining latency in such cases will be the maximum latency of an end-end connection over time.

Takeaway for applications: The maximum end-end RTT over time can be much higher than the minimum, and will determine the latency for jitter-sensitive applications.

5.3.2 Congestion control, absent congestion

We also explore how congestion control works on changing satellite paths. For this, we first use a congestion-free setting: the measured end-end connection is the only one sending traffic, with the rest of the network being entirely empty.

Fig. 5.1 also includes the per-packet RTT observed by TCP (NewReno) packets. This TCP observed RTT is calculated as the time difference between sending a packet and receiving its ACK. As expected, TCP continually fills and drains the buffer, thus increasing the RTT. To make the simulations faster, the shown experiments use a 10 Mbps line-rate. The buffers are sized to 100 packets, *i.e.,* 1 bandwidth-delay product (BDP) for 100 ms. With higher rate, we expect the same trend, with a smaller increase in RTTs as queues drain faster.

Fig. 5.2 shows the TCP congestion window evolution for the same 3 connections over the same period. The instantaneous BDP, aggregated with queue capacity, *i.e.,* BDP+Q, is also shown at each point in time – this is the maximum number of packets that can be in-flight without drops (as-

suming there is one bottleneck). The network device queue size, Q, for both ISLs and GSLs is set to 100 packets. For the times when BDP+Q is stable, TCP, as expected, repeatedly hits it, incurs a drop, cuts the rate, and ramps up again. But the changes in RTT, and thus BDP+Q, result in TCP changing its behavior. The disconnection event for St. Petersburg is evident from Fig. 5.2(a), but additionally, we can see drops in the congestion window for the other connections too, *e.g.,* in Fig. 5.2(c), around 140 s, TCP drops the congestion window because of packet reordering. At this time, as the path is shortened by $\sim$16 ms, packets transmitted later use the new shorter path, and arrive first at the destination. The resulting duplicate ACKs are interpreted as loss by the sender. The TCP RTT oscillations at the right end of Fig. 5.1(a) and 5.3(a) are caused by delayed acknowledgements. We checked that disabling delayed ACKs eliminates these, but does not change the rest of the observed behavior, which is our focus.

TCP's filling up of buffers and the resulting deterioration in per-packet latency is a widely recognized problem [134, 135, 192]. For LEO networks that promise low-latency operation, this is perhaps even more undesirable. We thus also test delay-based transport by repeating the same experiments, except using TCP Vegas. Note that the algorithms are not competing with each other, rather, each transport is tested entirely separately, *i.e.,* without any competing traffic – the issue of Vegas not being aggressive enough against Reno or Cubic is entirely orthogonal and immaterial here. Any transport implementable in ns-3 can be evaluated in Hypatia.

Fig. 5.3 shows the behavior of both NewReno and Vegas for one of the paths, Rio de Janeiro to St. Petersburg, Across the 200 s simulations, the per-packet RTT is shown in Fig. 5.3(a), the congestion window in Fig. 5.3(b),

and the achieved throughput averaged over 100 ms intervals in Fig. 5.3(c). Vegas, as expected, often operates with a near-empty buffer, *e.g.,* until around 140 s, it matches the ping RTT measurements in Fig. 5.1(a) closely. Unfortunately, however, Vegas interprets the increase in latency at ∼33 s as a sign of congestion, drastically cuts its congestion window (Fig. 5.3(b)), and achieves very poor throughput (Fig. 5.3(c)) after this point.

We tested NewReno and Vegas primarily because they are two well-known algorithms using loss- and delay-based congestion detection, and are already implemented in ns-3. However, Hypatia can be used with any congestion control algorithm implemented in ns-3. For instance, once a mature implementation of BBR [134] is available, evaluating its behavior on LEO networks would be of high interest. As of this writing, while there are some BBR implementations available online [193, 194], these have not been merged into ns-3, and we did not invest effort in testing these.

Our above results highlight challenges for congestion control in LEO networks: both loss and delay are poor signals of congestion in this setting. Loss, besides suffering from its well-known problem of only arising *after* buffers are full and latencies are inflated, is additionally vulnerable to being inferred incorrectly due to reordering. On the other hand, delay is also an unreliable signal because delay fluctuations occur even without queueing. This makes congestion control in this setting a difficult problem. Of course, if the sender knows the satellite path's variations, they can "subtract" them out and adapt. However, in general, the end-points need not even be aware that they are using a satellite-path: an end-point that is directly connected to a fixed connection could have its traffic sent to the nearest ground station by its ISP, as suggested in recent work [177]. Solutions like splitting the transport connection are also becoming difficult to support with transport

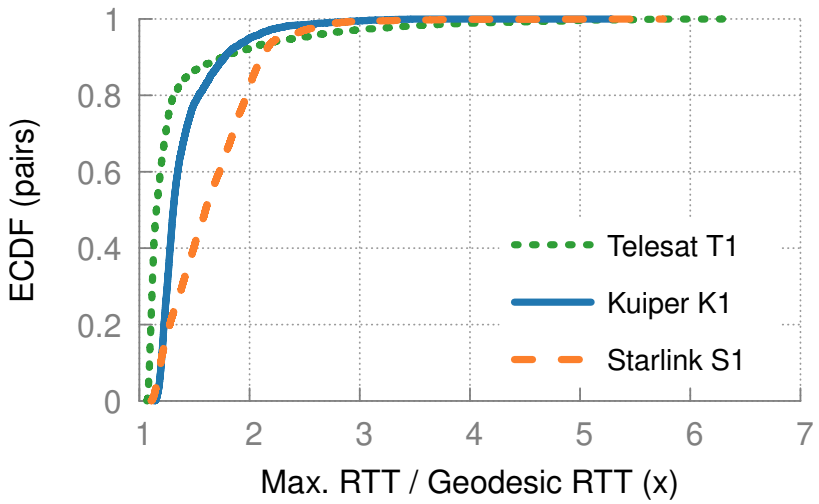


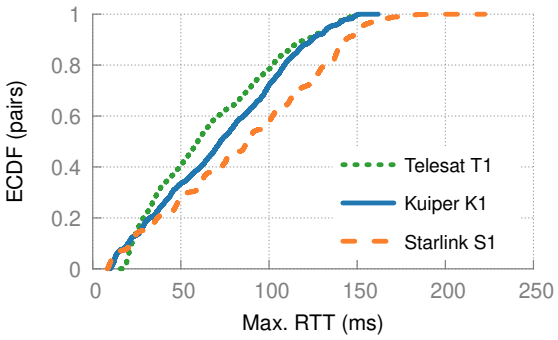
Figure 5.4: ***RTT vs. geodesic.*** *Even the maximum RTT (over time) over LEO networks is close to the geodesic RTT for most connections, especially for Telesat and Kuiper. However, some connections see several times higher maximum RTTs.*

such as QUIC, that does not support man-in-the-middle behavior.

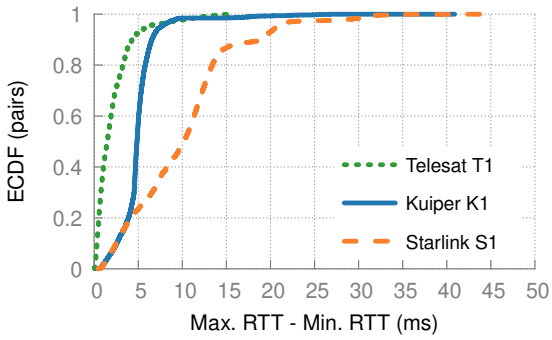
Takeaway for congestion control: Both loss and delay can be poor signals for congestion control in LEO networks.

5.4 A constellation-wide view

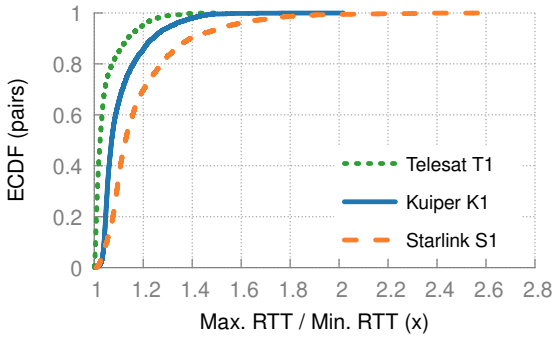
We use the first planned deployments for Starlink and Kuiper, and the first shell for Telesat to examine constellation-scale behavior. Starlink and Kuiper plan to deploy their shells S1 and K1 in Table 2.1 first. Telesat's deployment plan is more complex [142]; we simply use its first shell, T1. We use the world's 100 most populous cities as GSes, and examine connections between all pairs of GSes.



(a) CDF across pairs: max. RTT over time.



(b) CDF across pairs: max RTT - min RTT.



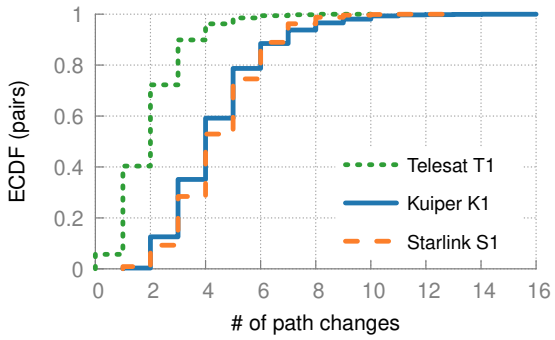
(c) CDF across pairs: max RTT / min RTT.

Figure 5.5: ***RTTs and variations therein.*** *Starlink S1 has a smaller number of satellites per orbit (22) than Kuiper K1, and thus sees both higher and more variable RTTs. Telesat sees lower and less variable RTTs despite fewer satellites because its extremely low minimum angle of elevation allows more GS-satellite connectivity options.*

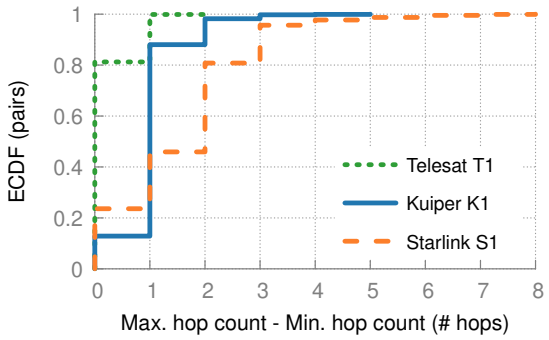
5.4.1 RTTs and variations therein

We measure the minimum and maximum RTT for each connection over the simulation duration. We also compute the "geodesic RTT" *i.e.*, the time it would take to travel back and forth between a connection's end-points at the speed of light in vacuum, c. This is thus the minimum achievable RTT.

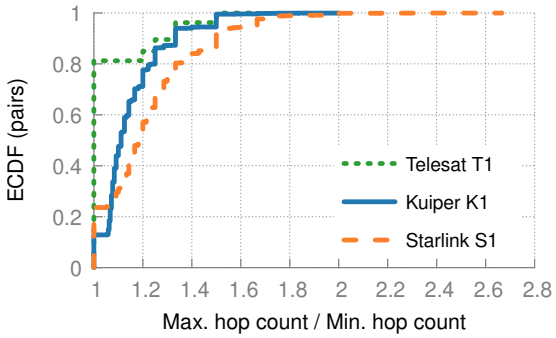
For each connection, we compute the ratio of its maximum RTT over time to the geodesic RTT between its end-points. Fig. 5.4 shows this ratio as a CDF across connections. For all three constellations, more than 80% of connections see a maximum RTT less than 2× the geodesic. Given that



(a) Distribution of path changes.



(b) Distribution of path hop-count differences.



(c) Distribution of max hop-count / min hop-count.

Figure 5.6: ***Path structure changes.*** *Telesat's paths change less than Kuiper's and Starlink's.*

terrestrial fiber paths are often longwinded, and the speed of light in fiber is roughly $2c/3$ [195], this implies that for most connections in our simulation, LEO networks will have substantially lower latencies than today's Internet. The long tail of latency inflation compared to the geodesic arises from connections between relatively nearby end-points, for which the overheads of the up-down connectivity to satellites are significant. For this reason, we already exclude end-point pairs that are within 500 km of each other from this plot and other results in this section.

Similar observations about latency in LEO networks have already been made in other work [51, 54, 94, 176]. However, a new and surprising finding here is about the comparison of the constellations. Telesat has the

fewest satellites, with less than a third of Kuiper's and less than a fourth of Starlink's, and yet it achieves the lowest latencies for most connections. Starlink's latencies are also higher than Kuiper's.

The explanations for these results lie in the connectivity parameters and the orbital structure of the constellations. Telesat claims that it will use a much lower minimum angle of elevation, 10°, compared to Starlink (25°) and Kuiper (30°). This allows GSes to see more of Telesat's satellites at any time, providing more options for end-end paths. Additionally, as these low elevation paths are closer to the horizon, the overhead of the up-down link is often smaller.

The Starlink-Kuiper differences are not due to the angle of elevation, which is similar, but the orbital structure. Both constellations use a minimum angle of elevation that is much higher than Telesat's. This means that typically, GSes can see fewer satellites. This restricts the GS-satellite connectivity, and increases the impact of satellite-satellite connectivity. Kuiper's orbital design, with 34 orbits of 34 satellites each, is more uniform than Starlink's, with 72 orbits of 22 satellites each. In particular, satellites within an orbit are much farther apart in Starlink, and paths often require zig-zagging through multiple orbits to reach the destination.

We also evaluate how much the RTT fluctuates over time across different connections. Fig. 5.5 shows the distribution across connections of: (a) the absolute value of the maximum RTT within a connection; (b) the difference between the maximum and minimum within a connection; and (c) the ratio between the maximum and minimum within a connection. The results show that while Starlink sees the largest latency changes (∼10 ms in the median), the other constellations also feature significant latency variation at the tail. Telesat's variations are the smallest again because of its low inclination: the

same satellites are reachable for longer, and result in more continuous and smaller latency changes. For Starlink, for more than 30% of connections, the maximum RTT is at least 20% larger than the minimum RTT.

For two reasons, we caution readers against concluding that 'Telesat is a better design': (a) There are downsides to using a lower minimum angle of elevation, as discussed in §2.1; and (b) We are evaluating constellations strictly from their filings, and it is unclear to us if some operators are more optimistic than others about the plausible design parameters; it is worth remembering that the filings are meant to secure radio spectrum for an operator by showing the potential utility of its network. The larger point, as far as the Hypatia framework is concerned, is that given the right input parameters, we can compare different designs along metrics like RTTs and RTT variability.

5.4.2 Path structure evolution

Besides RTTs, we also examine the structure of the underlying paths. For each connection, we measure the number of times its path changes over the simulation. If the forwarding state computed in two successive time-steps shows *any* different satellites composing the path, we count this as one path change. Across connections, we compute the CDF of these path changes. For each connection, we also calculate the maximum and minimum number of satellite hops in the path across the simulation.

Fig. 5.6(a) shows that in the median, over the 200 s simulation, Starlink and Kuiper connections see 4 path changes, while Telesat connections see 2 changes. These results are in line with our explanation of RTT variations: Telesat's use of a lower minimum angle of elevation allows remaining connected to a satellite for longer, and reduces path changes. The tail of path

changes is long as well: for Kuiper and Starlink, 10% of connections see 7 or more path changes.

Fig. 5.6(b) shows how these different paths differ in terms of their hop count. For Telesat, paths do not typically change in terms of hop count. This is explained by Telesat being sparser: there are simply fewer options for end-end paths, and with farther-apart satellites, one hop of change would already be substantial. For Starlink, with its large number of satellites, there are many more options for paths, and more than a third of connections see paths with at least 2 more hops than the minimum number.

Fig. 5.6(c) shows the same hop-count data in terms of relative change in hop-count. For Starlink, more than 10% of connections see more than 50% change in hop-count.

Unlike today's Internet, LEO network paths evolve rapidly, especially for the denser networks, with paths changing multiple times per minute, and often by a substantial number and fraction of hops. Routing within LEO networks thus features high churn. Nevertheless, given the tens of seconds between typical changes, we do not expect the setting up of desired routing state itself to be a major bottleneck.

5.4.3 Granularity of time-step updates

We quantify[3] the inaccuracies introduced in the forwarding behavior of the networks due to computing of forwarding tables at discrete time intervals in the pre-processing phase (§5.2.1). In practice, the link and path properties change continually, while Hypatia computes paths at configurable time granularities. For Kuiper $K1$ and all city-pairs (100 most populous cities

[3]These experiments were primarily conducted by my collaborator, Simon Kassing, and hence I do not include here the corresponding plots and analyses which are available in the paper [191]. Instead, I only summarize the results here for completeness.

as GSes), 100 ms (1,000 ms) time steps see 2× (20×) path changes as compared to 50 ms time steps. Also, the median number of path changes per 50 ms interval is 1. We posit that 100 ms time interval is a reasonable compromise, given it can only be inaccurate and not provide the actual shortest paths for at most 1% of the time. Note that path changes occur at a granularity of tens of seconds.

5.4.4 Bandwidth fluctuations

Beside the structure and latency of paths, and the response of individual TCP connections, we would also like to understand the result of interactions between traffic flows in such networks. Towards this goal, we conduct a simple experiment, sending long running TCP flows between pairs of GSes over their shortest paths.

We use the same LEO network as in §5.3, *i.e.,* Kuiper's K1 shell, with each link in the network set to 10 Mbps capacity to allow us to scale the experiment. Instead of just pings, we now send long running TCP NewReno flows between these GS pairs, which are still the same random permutation of the world's 100 most populous cities. From the random permutation matrix, we remove the pairs which have the same source or destination satellite as Rio de Janeiro or St. Petersburg at any point through the simulation; this prevents the first and last hops from being the bottleneck, allowing us to focus on the ISL network's behavior. We do not put this forth as a representative traffic matrix; rather, it is simply one way of sending substantial traffic through the network, and as we show next, reveals interesting network behavior. Hypatia can support arbitrary input traffic matrices.

We find that despite the traffic matrix being fixed throughout our 200 s

experiment, and the routing policy consistently being shortest path routing, the motion of satellites makes the path-level behavior highly dynamic.

Monitoring link utilization at one link is not a particularly useful way of demonstrating this in LEO networks — a particular ISL will traverse the globe in ∼100 min, seeing conditions corresponding to its location over time. We thus measure the "unused bandwidth" for each GS-pair, *i.e.,* how much unused capacity is there on the end-end path for that GS-pair over time. This is simply the path's link capacity (10 Mbps in our running scenario) minus the utilization of the most congested on-path link at any time. In a static network with fixed routing, and a fixed set of long-running TCP flows, we should expect this unused bandwidth to be small. This static-network TCP behavior is shown as the the gray line in Fig. 5.7 for the topology frozen at its $t = 0$ position.

However, we find that in an LEO network with cross-traffic, the amount of unused bandwidth is larger than that in the static case. Fig. 5.7 shows the unused bandwidth, measured at a 1 s granularity, for the same connection we examined in §5.3, from Rio de Janeiro to St. Petersburg. While there are short periods, such as around 20 s, where the full capacity of the path is used (together, by this connection and other cross-traffic), for a lot of the time, there is substantial unused capacity: 31% of the time, more than a third of the capacity is unused (excluding the unreachable period between 155-165 s), compared to 11% of the time if the satellite network were kept static at its $t = 0$ state.

The reason for this difference is the shifts in cross traffic resulting from the path changes: links constituting a GS-pair's shortest path change over time, and for each link, the set of GS-pairs it is used for changes as well. This implies that the traffic mix at any link is highly dynamic, making it

difficult for transport to adapt – the goal of TCP-like transport is, after all, to fairly share bandwidth across the flows traversing a bottleneck. In LEO networks the bottlenecks and which flows constitute the traffic mix change substantially over time. Note that this finding is not at odds with the results on infrequent path structure changes in Fig. 5.6. Although the median GS-pair's path changes only a few times over our 200-second simulation period, each end-end path has many links, and some of these links carry traffic from *many* GS-pairs. This results in a cumulative effect of changes in the cross-traffic traversing (the relatively stable) links of an individual GS-pair's path.

These observations have consequences for both traffic engineering and transport. Routing and traffic engineering could be planned ahead, such that knowing the upcoming changes in paths, traffic can be shifted *a priori* from links that will become new bottlenecks. This is a network-layer operation within the LEO network, and thus in the operator's control. A likely more difficult remedy is to attempt to make transport more responsive in adapting to changes: it is not clear that this can be done without causing *more* instability, as aggressive transport ramps up and down faster.

Takeaway for routing / TE: LEO networks present uncharted territory for routing and TE, and their interactions with transport. Traffic could potentially be moved away from links that will otherwise soon become bottlenecks due to changes in the set of end-end paths they serve.

5.5 Visualizing LEO networks

Since LEO networks are new to us, and likely to most of the networking research community, we found it extremely useful to visualize some aspects of them, and thus build our intuitions on their expected behavior. We

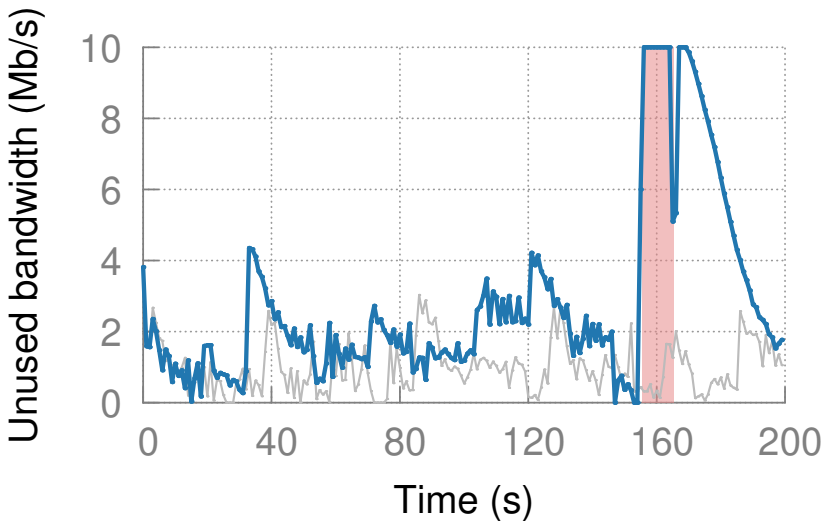
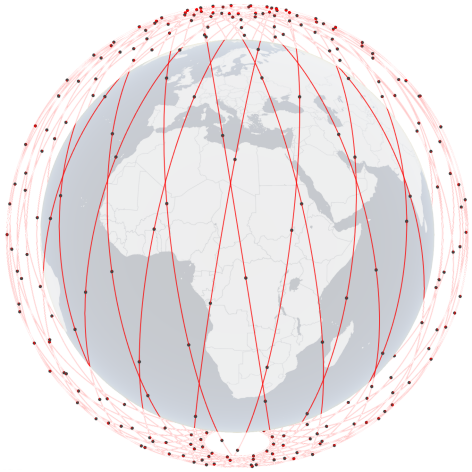


Figure 5.7: ***Unused bandwidth.*** *For the same connection from Rio de Janeiro to St. Petersburg as in Fig. 5.3(c), when tested with cross-traffic, transport is often unable to use the available bandwidth. This is with a fixed set of long-running TCP flows, and a fixed routing policy. The gray line is if the satellite network is frozen at $t = 0$, effectively being a static network.*

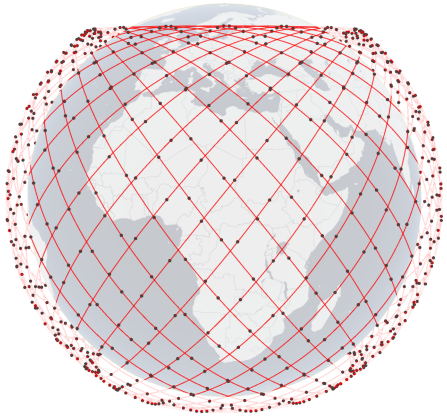
discuss some of the visualizations Hypatia provides. While these are best appreciated online in video and interactive Javascript [174], we include here snapshots discussing their utility.

Satellite trajectories: It is difficult to grasp the role of different satellite trajectory parameters (§2.1) without being able to visually see their outcomes. Visualizing the trajectories of satellites in a constellation also drives intuition about how satellites travel together, the differences between the multiple shells of some constellations, the density of satellites over equatorial and polar regions, etc.

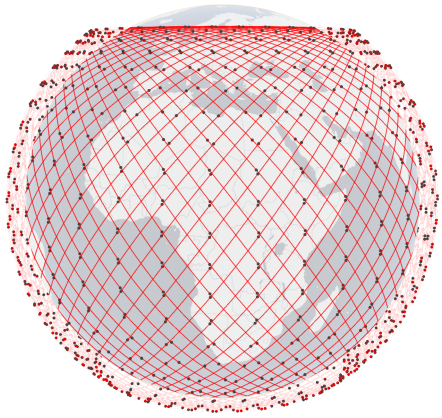
Fig. 5.8 shows snapshots of the first shells of Starlink, Kuiper, and Telesat — S1, K1, and T1 in Table 2.1. A live 3D version of this figure is available online [174]; it is interactive and allows one to change the camera perspective in order to better see the spatial variations. Telesat covers the



(a) Telesat T1



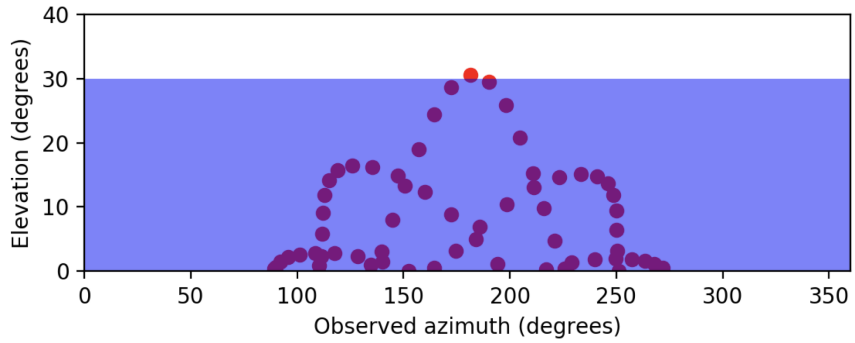
(b) Kuiper K1



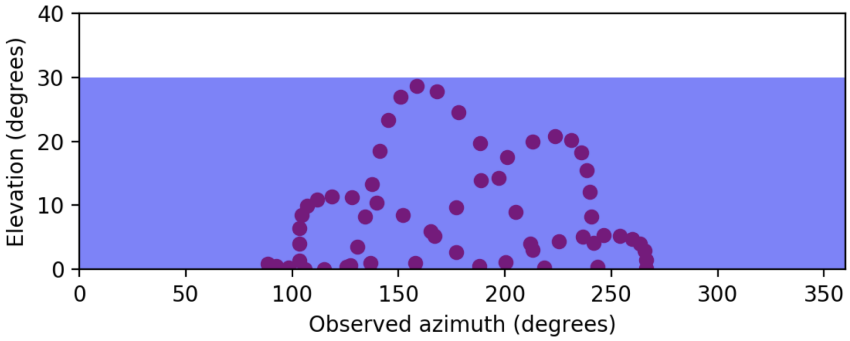
(c) Starlink S1

Figure 5.8: ***Constellation trajectories.*** *(a) Telesat* $T1$ — 27×13, 1,015 *km*, 98.98° *(b) Kuiper* $K1$ — 34×34, 630 *km*, 51.9° *(c) Starlink* $S1$ — 72×22, 550 *km*, 53°. *Satellites are black dots, while orbits are marked in red.*

polar regions by virtue of the higher inclination of its orbits (98.98°), while Kuiper and Starlink provide denser coverage at lower latitudes. Given that a vast majority of the global population resides at lower latitudes [196], lower inclination allows satellites to spend more time over densely populated areas. These design differences may imply differences in the target markets of the constellation operators.



(a) Connectivity is possible at this time.



(b) No satellites are reachable.

Figure 5.9: ***Ground observer's view.*** *The x-axis is the horizon, with* $0° = N$, $90° = E$, *while the y-axis is the angle of elevation in the sky. The shaded region includes satellites above the horizon, but have angle of elevation lower than the minimum required to connect. From St. Petersburg, Kuiper's* $K1$ *is intermittently reachable.*

Besides coverage, inclination also has other implications for connectivity: Telesat's almost north-south orbits may offer more direct paths for routes like between Europe and Africa, while the other constellations will do so for east-west routes like between North America and Europe.

We include satellite trajectory visualizations primarily for completeness: there are a variety of other beautiful visualizations of similar nature online [54, 108, 182–184]. To the best of our knowledge, no *open-source* visualization tools are available that focus on *network* behavior of LEO

constellations, aspects of which we describe next.

Ground station view: For any given constellation, and a specified location, Hypatia can show how that constellation appears in the sky to a ground station. This view helps understand the role of the minimum angle of elevation, as well as the inclination of orbits. The visualizations show that close to the horizon, there are many more satellites, but the satellites a GS can communicate with, *i.e.,* above the minimum angle of elevation, are much more limited. From high latitude cities, one can see the limits of low-inclination orbits: few satellites in such orbits are visible, with this visibility often being intermittent. The online version of this visualization [174] provides video of the ground observer's perspective.

Fig. 5.9 shows two snapshots of Kuiper's K1 seen from St. Petersburg. The azimuth along the x-axis is the panoramic view of the sky (0° is due North, 90° is due East). The y-axis is the angle of elevation, 0° for the horizon, and 90° for directly overhead. Satellites in the shaded region are above the horizon, but still at an angle of elevation lower than the minimum needed for connectivity. Over certain periods, a GS at this location can connect to Kuiper, as in Fig. 5.9(a), while at other times, it looses connectivity, as in Fig. 5.9(b). This explains the results for Rio de Janeiro to St. Petersburg between 155-165 s in Fig. 5.1(a), Fig. 5.2(a), and Fig. 5.3.

End-end paths: In §5.3.1, we discuss RTT variations due to the LEO dynamism. To better understand these, it is useful to visualize the end-end paths at different points in time. Fig. 5.10 shows an example path on Starlink, Paris-Luanda, which experiences one of the highest RTT variations. The longest (117 ms) and shortest (85 ms) RTT paths during our 200 s simulation are shown. It is typical of such north-south paths to pick an orbit and stick to it as long as possible in order to reduce latency. But in

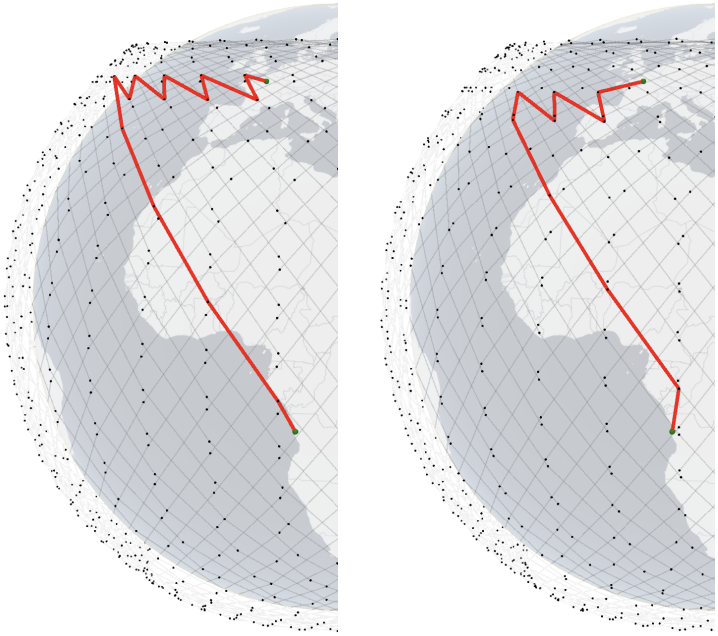


Figure 5.10: ***Shortest path changes over time.*** *On Starlink's S1, the Paris-Luanda RTT varies between 117 ms (left) and 85 ms. The zig-zags stem from the nature of ISLs in the topology — satellites which seem visually close to each other are not necessarily connected directly.*

the former case, exiting this orbit (at the north end of the illustrations) towards the destination takes 9 zig-zag hops, while in the latter case only 6 are needed.

Link utilization: In §5.4.4, we discuss how even for a static traffic matrix, LEO dynamics cause links and paths to vary in utilization over time. This is shown for one example path in Fig. 5.11, for the same experiment across Kuiper described in §5.4.4. The thicker/warmer-colored ISLs are more congested.

We can also visualize network-wide bottlenecks as shown in Fig. 5.12. For the particular traffic matrix we use, the ISLs over the Atlantic, connecting the US to Europe and parts of Asia, are highly congested. This indicates

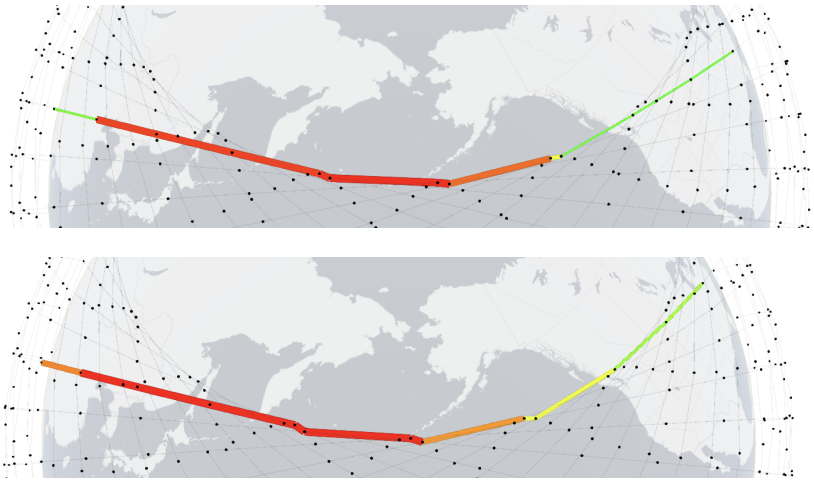


Figure 5.11: ***Congestion shifts over time.*** *An example path, Chicago-Zhengzhou, shows how the link utilizations change over time, even with the input traffic being static. The top and bottom views are at* 10 *s and* 150 *s respectively.*

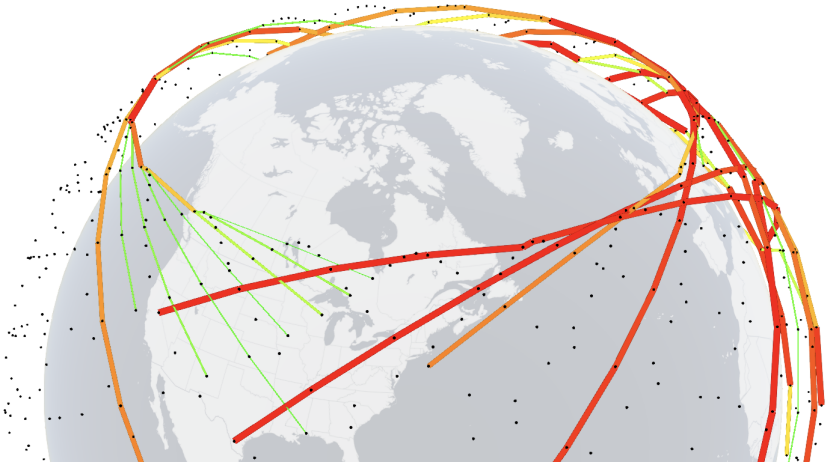


Figure 5.12: ***Constellation-wide utilization.*** *On Kuiper, the trans-atlantic paths are highly congested for our tested traffic matrix. The red / thick ISLs are heavily utilized, while green / thin ISLs have minimal traffic. ISLs with no traffic are excluded.*

that there will be substantial value in using non-shortest path and multi-path routing across such busy regions.

5.6 Limitations

Hypatia is only the first step in building up research infrastructure for a new breed of networks. It has several under-developed pieces, including some where the sparsity of publicly available information was limiting for us.

- The most under-developed aspect is the radio GS-satellite segment design. It would help to frame more realistic models of the interfaces at both satellites and GSes, and for antenna gain and interference.
- The current model of ISLs is also somewhat simplistic, and it would be useful to model the impact of the Doppler effect on the bandwidth and reliability of ISLs.
- Incorporating a weather model would enable work on reliability and rerouting around bad weather.
- Work on multi-path routing and congestion control will also require some modifications to Hypatia.
- GEO-LEO connectivity, albeit not implemented already, should be straightforward to implement if GEO coverage and minimum elevation constraints are known.
- Simulating constellations with heterogeneous satellite and ISL capabilities could be interesting as well — as satellites are gradually deployed, their capabilities may advance over time. Heterogeneity in terms of link capacities is easy to accommodate, but changes to support different numbers of ISLs across satellites will require additional work in defining topologies that appropriately use such heterogeneity.

More broadly, as is typical for simulation infrastructure, we cannot fully anticipate the needs of novel proposals for LEO networking, and it is likely that many such efforts will require modifications of Hypatia. However, we believe it provides a useful starting point for such work.

Importantly, all the takeaways we have highlighted throughout this chapter are robust to Hypatia's current limitations. Regardless of how the missing design details are filled in as more information becomes available, the RTTs are going to vary, congestion control is going to see noisy loss and delay signals, and the shifting paths pose clear challenges for routing and traffic engineering.

5.7 Related publications

The plots and the corresponding discussions in this chapter have been taken from the following publication:

- S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the 'Internet from space' with Hypatia," in *ACM IMC*, 2020

This work received the best paper award at ACM IMC, 2020.
I am joint-first author along with Simon Kassing.
While both Simon and I are responsible for the engineering and system architecture aspects of Hypatia, Simon focused more on its usability and scalability, and I was in charge of design decisions and incorporating LEO dynamics and visualizations. Figures 5.1 - 5.7 in this chapter are our joint contribution. I thank Simon and Ankit for the exhaustive brainstorming sessions and André and Jens (Master and Bachelor thesis students supervised by me) for initiating the engineering groundwork.

6

Contextualizing LEO networks in the low-latency networking space

Thus far, this dissertation has focused on LEO network performance. In this chapter, I contextualize them in the broader space of wide-area networking infrastructure for low-latency communication. We have seen terrestrial low-latency speed-of-light networks built with point-to-point radio operating in niche scenarios, like algorithmic trading corridors [198]. One can design such networks also across continents leveraging the dense deployment of towers. Such specialized networks could offer the minimum communication latencies achievable terrestrially and provide a point of comparison for the LEO counterparts. Further, such alternatives allow us to envision an ambitious low-latency network design goal – how to design a globally spanning hybrid (different media) network enabling ubiquitous low-latency communications? This chapter first fleshes out the outlines for designing a terrestrial speed-of-light network and then briefly compares the performance and cost of this network and a globally spanning LEO network

– Starlink's *S*1. Finally, it concludes by envisioning a global low-latency hybrid network design.

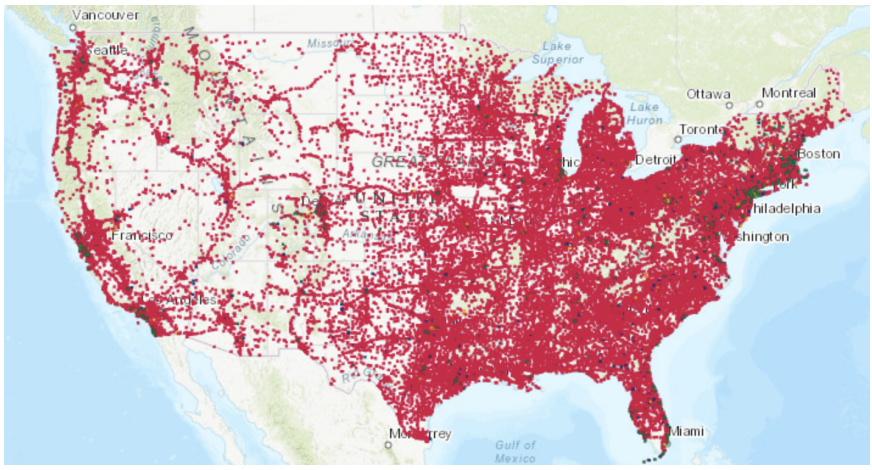


Figure 6.1: *American tower deployment as per* 5^{th} *March,* 2021.

American Tower deployment. The network design is motivated by the fact that continent-wide dense deployment of towers by a single entity (an incumbent) exist today. American Tower [199], for example, claims to have a presence at more than 42,000 tower sites across the US, as of 5^{th} March 2021. Fig. 6.1 shows their current deployment. We could not access their database due to legal bindings.

6.1 A terrestrial speed-of-light network

We explore the design of **cISP**, an Internet Service Provider that provides nearly speed-of-light (c) latency by exploiting wireless electromagnetic transmissions, which can be realized with point-to-point microwave antennae mounted on towers. This approach holds promise for overcoming the shortcomings fundamental to today's fiber-based terrestrial networks: the transmission speed in air is essentially equal to c, and the richness of existing tower infrastructure makes more direct paths possible.

We propose a hybrid design that achieves end-to-end connectivity with nearly straight-line wireless links, augmented by existing fiber infrastructure, as needed, in order to respect budget constraints. These low-latency links are used judiciously where they provide the maximum latency benefit, and only for the small proportion of Internet traffic that is latency-sensitive. We design a simple heuristic that achieves near-optimal results for the network design problem.

Technology choices. Several physical layer technologies are amenable for use in our design, including free-space optics (FSO), microwave (MW), and millimeter wave (MMW). At present, we believe MW provides the best combination of range, resilience, throughput, and cost. Future advances in any of these technologies, however, can be easily rolled into our design, and can only improve our cost-benefit analysis. While hollow fiber [200] could, in the future, also provide *c*-latency, it would still suffer from the circuitousness of today's fiber conduits.

Spectrum and licensing. We propose the use of MW communication in the 6-18 GHz frequency range. These frequencies are not very crowded, and licensing is generally not very competitive, except at 6 GHz in cities, and along certain routes, like the above mentioned HFT corridor. The licenses are given on a first-come, first-served basis, recorded in a public database, and they protect against the deployment of other links that would interfere with licensed links.

Line-of-sight and range. Successive MW towers need line-of-sight visibility, accounting for the Earth's curvature, terrain, trees, buildings, and other obstructions, and atmospheric refraction. Attenuation also limits range. For our analyses, we assume a maximum range of 100 km. Lower ranges would increase the cost while making the network more robust.

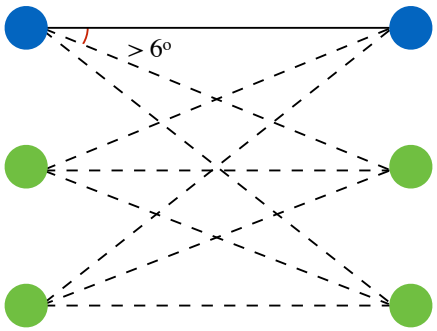


Figure 6.2: *Bandwidth augmentation: k^2 hops with $O(k)$ new towers.*

Bandwidth. Between any two towers, using very efficient encoding (256 QAM or higher), wide frequency channels, and radio multiplexing, a data rate of about 1 Gbps is achievable [201]. This bandwidth is vastly smaller than for fiber, and necessitates a hybrid design using fiber and MW. While computing the cost per GB of *c*ISP we drive the network at ∼100 Gbps aggregate bandwidth.

We can employ a simple trick to enhance the effectiveness of parallel series of towers, as shown in Fig. 6.2. Instead of k parallel series of towers providing merely a $k\times$ bandwidth improvement, connecting multiple antennae on each tower to other towers, we can obtain a $k^2\times$ improvement. Using antennae with overlapping frequencies requires an angular separation of 6° [202], as shown in Fig. 6.2. Again, the latency inflation caused by the resulting gap between parallel series of towers is small. For a tower-tower hop distance of 100 km, the minimum distance between two parallel towers should be $100 \cdot \tan(6°) = 10.6$ km, which, as noted above, has a small effect on end-to-end latency for long links.

Geographic coverage. Connecting individual homes directly to such a MW network would be cost-prohibitive. To maximize cost-efficiency, we focus on long-haul connectivity, with the last mile being traditional fiber.

At short distances, fiber's circuitousness and refraction are small overheads.

Cost model. We rely on cost estimates in recent work [203] and based on our conversations with industry participants involved in equipment manufacturing and link provisioning. The cost of installing a bidirectional MW link, on existing towers, is approximately \$75K (\$150K) for 500 Mbps (1 Gbps) bandwidth. The average cost for building a new tower is \$100K, with wide variation by terrain and across cities and rural areas. Any additional towers needed to augment bandwidth for particular links incur this "new tower" cost. The operational costs comprise several elements, including management and personnel, but the dominant operational expense, by far, is tower rent: $\$25 - 50$K per year per tower. We estimate cost per GB by amortizing the sum of building costs and operational costs over 5 years.

6.2 cISP Design

At an abstract level, given the tower and fiber infrastructure, a set of sites (*e.g.,* cities) to interconnect, and a traffic model between them, we want to select a set of tower-level connections that minimize network-wide latency while adhering to a budget. Our approach comprises the following 2 broad steps.

1. Identifying a set of links that are likely to be useful by determining, for each pair of sites (s, d), the best feasible tower-level connectivity, if s and d were to be directly connected by a series of towers.
2. Building all $O(n^2)$ direct links, connecting each site to every other, would be prohibitively expensive. Thus, a subset of site-to-site links, together with existing fiber conduits, form our network. Choosing the appropriate subset is the key algorithmic problem.

3. Provisioning capacity beyond 1 Gbps along any link involves building additional tower-level links, *e.g.,* by identifying and using links that are also nearly shortest paths, but were omitted in step 1 above.

6.2.1 Step 1: Feasible hops

We first use line-of-sight and range constraints to decide which tower pairs in the FCC database [204] can be connected. Achievable tower-to-tower hop length is limited primarily by the Earth's curvature. MW hops must clear this curvature and any obstructions in an ellipsoidal region between the sender and the receiver antennae called the Fresnel zone.

We assess hop feasibility between each pair of towers by using terrain data made available by NASA [205], which includes buildings and ground clutter, and effectively incorporates the height of the tree canopy. This NASA data set combines data from the Shuttle Radar Topography Mission (SRTM) [205] and the National Elevation Database (NED) [206], and typically yields acceptably small error ($\sim$2 m) against reference, high-accuracy LIDAR measurements.

After identifying feasible tower-to-tower hops, for each pair of sites, we find the shortest path through a graph containing these hops, which we call a *link*. In line with observations from the tower data around major population centers, we assume each site itself hosts enough towers to use as the starting point for connectivity from that site to many others.

Note: This step was primarily performed by my collaborators, Prof. Gregory Laughlin (Yale University) and Prof. Anthony Aguirre (UCSC), whose hop engineering routines have been used to design line-of-sight networks, at least 4 of which are now deployed, including ultra-low latency routes be-

tween data centers hosting financial market matching engines. I included this step here with their consent for continuity.

6.2.2 Step 2: Topology design

Picking a subset of these site-to-site links involves solving a typical network design problem. The Steiner-tree problem [207] can be easily reduced to this problem, thereby establishing hardness. However, standard approximation algorithms, like linear program relaxation and rounding, yield sub-optimal solutions, which although provably within constant factors of optimal, are insufficient in practice. We develop a simple heuristic, which, by exploiting features specific to our problem setting, obtains nearly optimal solutions.

Inputs: Our network design algorithm requires:

- A set of sites to be interconnected, $v_1, v_2, \ldots, v_n$.
- A traffic matrix H specifying the relative traffic volume $h_{ij} \in [0, 1]$ between each pair v_i and v_j.
- The geodesic distance d_{ij} between each v_i and v_j.
- The distance along the shortest, direct MW path between each pair, m_{ij}, as well as its cost, c_{ij}. This is part of the output of step 1.
- The optical fiber distance between each pair, o_{ij}, which we multiply by 1.5 to account for fiber's higher latency.
- A total budget B limiting the maximum number of bidirectional MW links that can be built.

Expected output: The algorithm must decide which direct MW links to pick, *i.e.,* assign values to the corresponding binary decision variables, x_{ij}, such that the total cost of the picked links fits the budget, *i.e.,* $\sum_{ij} x_{ij} c_{ij} \leq$

B. Our objective is to minimize, per unit traffic, the mean stretch, *i.e.,* the ratio of latency to c-latency, where c-latency is the speed-of-light travel time between the source and destination of the traffic.

Problem formulation: Expressing such problems in an optimization framework is non-trivial: we need to express our objective in terms of shortest paths in a graph that will itself be the *result*. We use a formulation based on network flows.

Each pair of sites (v_s, v_t) exchanges h_{st} units of flow. To represent flow routing, for each potential link ℓ, we introduce a binary variable $f_{stij,m}$ which is 1 *iff* the $v_s \to v_t$ flow is carried over the microwave link $v_i \to v_j$, and a binary variable $f_{stij,o}$ which is 1 *iff* the same flow is carried over the optical link[1] $v_i \to v_j$. The objective function is:

$$min \sum_{s,t} \frac{h_{st}}{d_{st}} \sum_{i,j} (o_{i,j} f_{stij,o} + m_{i,j} f_{stij,m}) \tag{6.1}$$

The h_{st} term achieves our goal of optimizing *per unit traffic*. The $\frac{1}{d_{st}}$ term achieves our goal of optimizing the *stretch*.

The constraints include: flow input and output at sources and sinks; flow conservation; total budget; and the requirement that only links that are built ($x_{ij} = 1$) may carry flow. All variables are binary, so flows are "unsplittable" (carried along a single path) and the overall problem is an integer linear program (ILP).

Note that we have decomposed the problem so that link capacity is *not* a constraint in this formulation: MW links will be built with sufficient capacity in step 3; fiber links are assumed to have plentiful bandwidth at negligible cost relative to MW costs. As a result, the objective function will

[1] A "link" between sites can use multiple physical layer hops, both for MW and fiber. The underlying multi-physical-hop distances are already captured by the inputs o_{ij} and m_{ij} so the optimization views it as a single link.

guide the optimizer to direct each $v_i \to v_j$ flow along the shortest path of built links, which is the direct MW link $v_i \to v_j$ if it happens to be built, or otherwise, a path across some mix of one or more fiber and MW links.

Solution approach: As we shall see, simply handing the ILP to a solver did not scale to beyond medium-sized networks. By exploiting our problem structure, however, we develop a simple heuristic that yields near-optimal results at smaller scales (verified against the exact ILP solution) and can solve the problem at the larger scales of interest.

The first observation we make is that a large number of variables in our formulation will never take non-zero values, allowing us to eliminate them and any resulting null constraints. Roughly stated: if, for a particular (v_s,v_t) pair, a microwave path is of higher latency than a fiber path (which we can always use, at zero expense), then it will never carry $v_s \to v_t$ flow, though other flows may still traverse it. Similar observations apply to individual "distant, off-path" fiber and MW links. This simple observation substantially reduces the problem size. Note that standard network design problems do not typically have this structure available. This is entirely due to the hybrid design using fiber, which is assumed to be cheap, where available. We benefit, in this case, from having an "oracle" that tells us *a priori* when certain flow assignments are "obviously bad" and will not be useful. Further, carefully defined, such constraints preserve optimality; this part of our solution is not an approximation.

Second, we use a fast greedy heuristic to prune out MW links that are unlikely to be chosen. The heuristic operates using a larger budget ($2\times$ in our implementation) than we are ultimately allowed. In each iteration, we add to the solution the MW city-to-city link that decreases average stretch the most, continuing until the total cost reaches the inflated budget; the

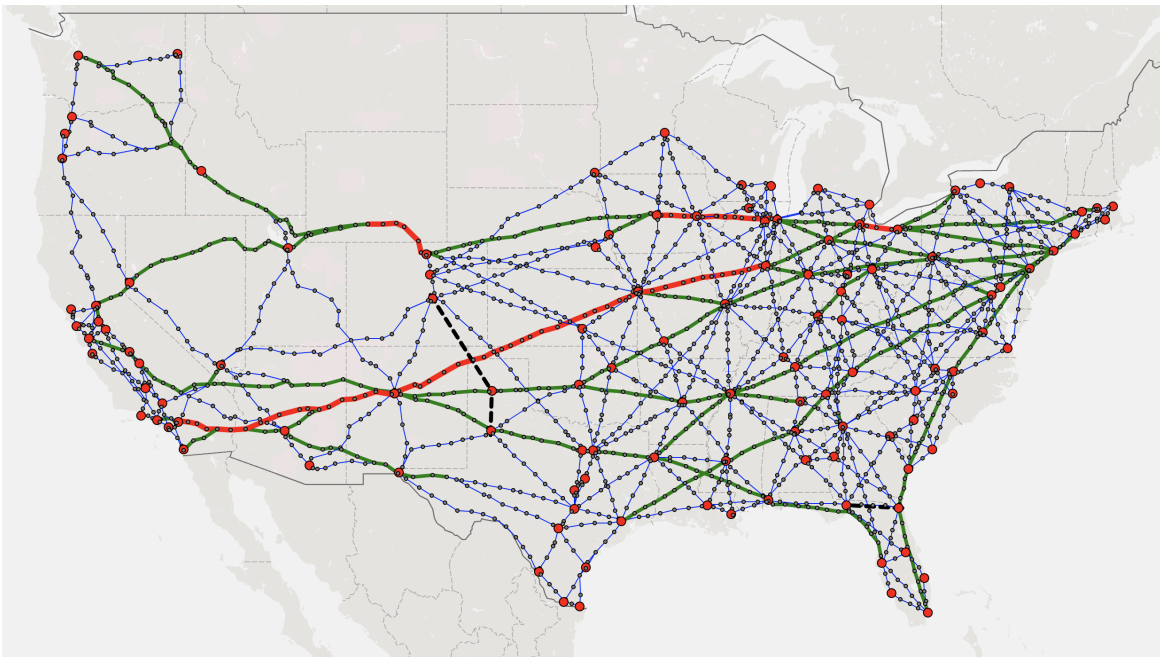


Figure 6.3: *A* 100 *Gbps,* 1.05× *stretch network across* 120 *population centers (big, red) in the US. Blue links (thin) need no additional towers. Green links (thicker) and red links (thickest) need 1 and 2 series of additional towers, respectively. The black dashed links represent fiber paths.*

chosen links are candidates given to the ILP. Intuitively, the other links are uninteresting – they are unlikely to be picked in the final optimization even when a substantially larger budget is available, and so are not presented as options to the ILP. This approach does not provide any guarantees, but we find that on small problem sizes, where the exact ILP can also be evaluated, it obtains the optimal solution.

6.2.3 Step 3: Augmenting capacity

This is achieved following the discussion under 'Bandwidth.' in the previous section (see Fig. 6.2). This approach implies that for site-to-site bandwidths under 1 Gbps, we need just one series of towers; for bandwidths between 1-4 Gbps, we need 2 series; for 4-9 Gbps, 3; etc.

6.3 A cISP for the United States

We now apply the framework above for a concrete instantiation: designing a cISP for the U.S. mainland. To assess line-of-sight connectivity between existing towers, we use fine-grained data on tower infrastructure, buildings, terrain, and tree canopy. The fiber conduit data is available from past work [74].

Defining the sites and traffic model: To maximize utility while keeping costs low, we connect only the 200 most populous cities in the contiguous United States. In addition, we coalesce suburbs and cities within 50 km of each other, ending up with 120 population centers. (Henceforth, when we refer to "cities", we refer to these population centers.) Based on population data for 2010 [208], we calculate that 85% of the US population lives within 100 km of these 120 cities. For the traffic matrix, we use demands between city pairs that are proportional to their population product.

Which city-city links are feasible? We use existing towers listed in FCC's Antenna Structure Registration [209] and databases from American Tower, Crown Castle, and several other tower companies for which we were able to download data. We cull these rather large databases of MW towers to a subset of 12,080 towers as follows: Towers from rental companies are typically suitable for use. From the FCC database, we only use towers over 100 m height. When tower-density exceeds 50 towers per 0.5° square grid cell, we randomly sample towers. (Using all towers could only improve our results, but increases compute time.)

Evaluating link feasibility across tower pairs within range of each other using the aforementioned NASA data [205], we find 261,019 tower-tower hops that satisfy line-of-sight constraints. We find that each city itself has

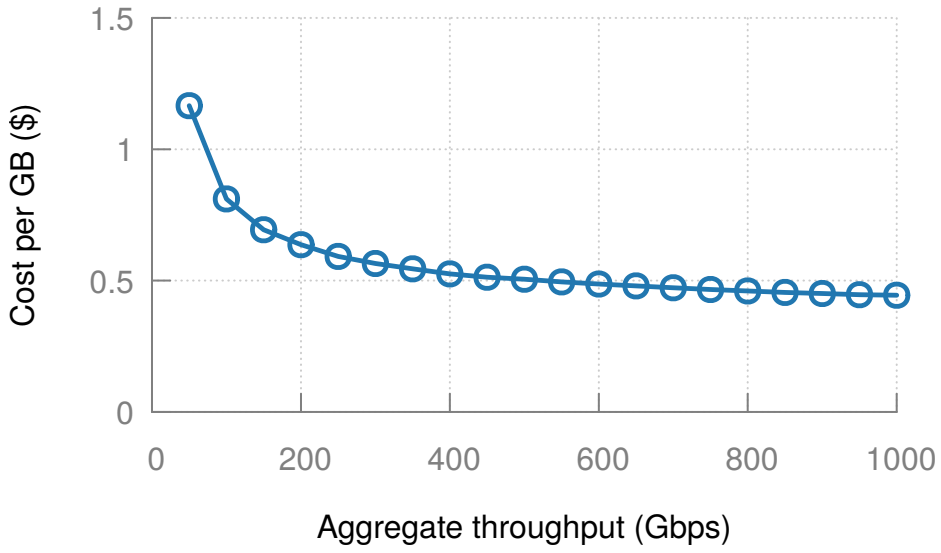


Figure 6.4: *Cost per GB for the city-city traffic model decreases with increasing aggregate throughput.*

large numbers of suitable towers in its vicinity. We run a shortest path computation on a graph comprising the cities and towers and city-tower and tower-tower hops to find the shortest city-city MW links. This yields both the cost (*i.e.,* number of towers) and latency (*i.e.,* distance along the chosen series of towers) for each city-city link.

For fiber distances, we compute the shortest paths over the InterTubes [74] dataset on US fiber conduits.

What subset of links should we build? We use the Gurobi solver [210] to solve our topology design problem. Fig. 6.3 shows an example network. Designed with a budget of 3,000 towers and maximum hop length of 100 Km, its average latency is 1.05× *c*-latency.

Augmenting capacity: We produce a target aggregate demand (*i.e.,* the sum of all site-site traffic demands) by scaling the traffic matrix H. Then, each tower-tower MW hop that would be over-utilized (given the shortest-path routing and the 1 Gbps capacity per link) is augmented with

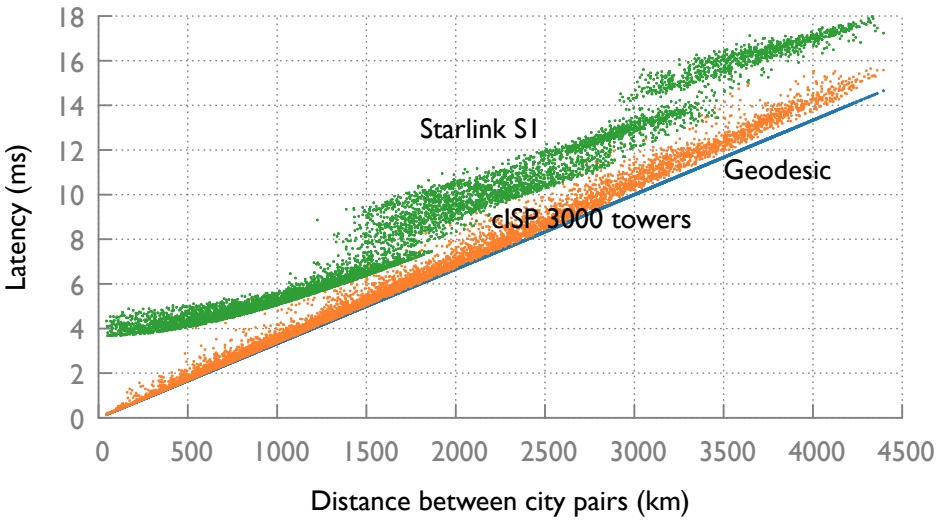


Figure 6.5: *cISP versus Starlink $S1$ latencies between city-pairs in the US. The speed-of–light geodesic latencies are marked in blue.*

additional towers at each end, as described before. Fig. 6.3's topology, when provisioned for an aggregate throughput of 100 Gbps, has 1,660 tower-tower hops that use only already built towers seen in tower databases, while 552 hops need one additional new tower at each end, and 86 hops need 2 additional towers at each end. Using the cost model described before, we find that the cost per GB for this topology, with latency within 1.05× and 100 Gbps throughput, is \$0.81. For some context, this is ∼10× the cost per GB for content delivery networks [211]. Fig. 6.4 gives the conservative cost-estimates as cISP's aggregate bandwidth increases – all additional towers for bandwidth augmentation are accounted for as new towers.

6.4 cISP versus LEO latency

Now that we have cISP as a point of comparison, we compare the achieved latency benefits and offered cost per GB with that of Starlink's $S1$ which

is already under heavy deployment at the time of writing this dissertation.

Fig. 6.5 shows that Starlink $S1$ offers one-way propagation latency close to what could be achieved with cISP built across the US with a budget of 3,000 towers. Even for city-pairs in the US which are farthest apart (4,000+ km), Starlink offers a one-way latency never higher than cISP by more than 3.4 ms. Of course for nearby city pairs (less than 500 km distance), Starlink latency is relatively higher due to the latency cost of bouncing data between the terrestrial and satellite planes.

The jumps in Starlink $S1$ latency, as evident in Fig. 6.5 around city-pair distances of 1,500 and 3,000 km, reflect increasing number of satellite hops in the end-to-end path. As the angle of elevation $e = 25°$ defines the cone of terrestrial coverage of a Starlink $S1$ satellite, even if a city lies just immediately outside this threshold, the corresponding packet needs to travel an additional ISL and satellite hop in order to reach the city.

Fig. 6.5 and Fig. 2.5 (discussed previously; quantifies longer distance latencies over LEO) together show that LEO satellite networks have capability to offer low latency connectivity both within and across continents, and are comparable in performance to the best achievable low-latency terrestrial infrastructure.

Nevertheless, within a continent, cISP offers a practical lower bound of communication latency. Also note that Starlink claims in a recent filing [212] to currently offer last-mile round-trip latency of 31 ms in practice, more than 3.8× higher than estimated, showing that the service is not yet latency optimized in the beta-testing phase.

6.5 cISP versus LEO cost per GB

As noted above, cISP built with 3,000 towers and supporting 100 Gbps throughput can offer low-latency transit for $0.81 per GB. Nevertheless, if an incumbent like American Tower [199] were to deploy it, the cost could be as low as $0.33/GB by leveraging the high density of already deployed towers available for bandwidth augmentation along different paths. This optimistic cost estimate assumes no rental cost or additional tower installation cost to augment bandwidth. We refrain from considering higher bandwidth cISP networks (Fig. 6.4; bottom right) built with the same budget in order to be conservative about the additional towers needed for augmenting capacity.

Starlink beta currently offers uncapped connectivity at $99/month [213]. At an average household consumption of 273.5 GB [214], this translates to $0.36/GB. Note that we do not take into account the equipment (user terminal) cost of $499 which a user has to pay upfront for availing Starlink beta services. We rather assume that this is orthogonal equipment cost and the data transfer costs are covered solely by the monthly subscriptions.

As seen above, the roughly estimated cost of sending data via Starlink $S1$ ($0.36/GB) is lower than what cISP can offer by default ($0.81) and also comparable to an optimistic cost model ($0.33/GB) corresponding to a cISP deployment by an incumbent (like American Tower). Nevertheless, there is no public report yet claiming Starlink profits. Hence, the Starlink cost-per-GB analysis above has to be taken with a grain of salt given they are currently in the beta-testing phase, possibly looking for a market share soon.

6.6 Ongoing efforts

There are several ongoing high-profile Internet infrastructure efforts, including X moonshot factory's project Taara [215], and Facebook connectivity's Magma [216], Rural Access [217], and Terragraph [218]. Project Taara consists of networks under deployment in India and Africa, based on free-space optics, and described as "Expanding global access to fast, affordable internet with beams of light". While Facebook's Magma and Rural Access aim to extend connectivity to rural areas by offering a software, hardware, business model, and policy framework, Terragraph aims to extend last-mile connectivity to poorly connected urban and suburbans areas by leveraging short millimeter-wave hops. Free-space networks of this type will likely become more commonplace in the future, and these works show that many of the concerns with line-of-sight networking can indeed be addressed with careful planning.

6.7 Designing low-latency hybrid networks

Given a steady increase in market demand for latency-critical Internet applications, how could we build a globally spanning network that supports such demands? Choices like terrestrial point-to-point radio, free-space optics, drones [219], LEO satellites, hollow-core [220], and solid glass-core fiber each have their pros and cons. They offer trade-offs in throughput, latency, robustness, deployment complexity, and cost. It is a grand networking challenge to design topology, routing, queuing, and transport for hybrid low-latency networks catering to varying application demands. One has to take into account the differences in speed and robustness of different media and still meet service-level agreements.

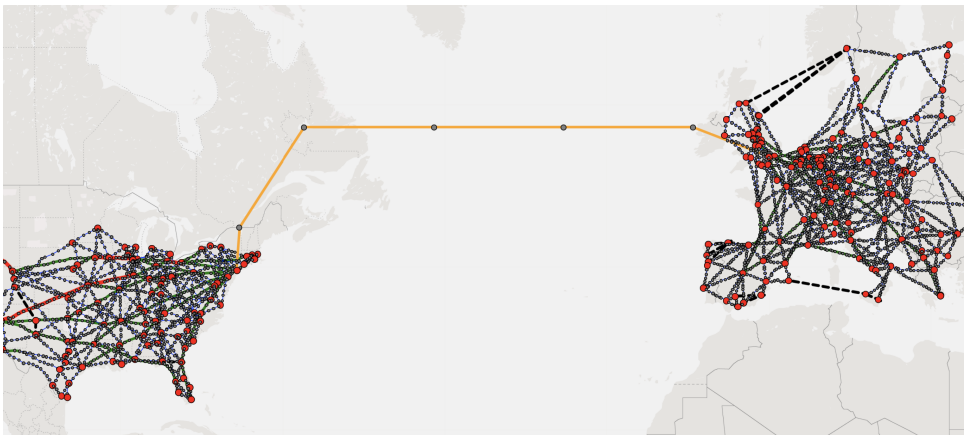


Figure 6.6: *An envisioned trans-Atlantic low-latency hybrid network consisting of LEO satellites, terrestrial radio, and fiber.*

Fig. 6.6 shows one such hybrid network – terrestrial radio and fiber offer cISP services in the US and Europe while LEO satellite paths bridge the inter-continental latency-gap. While in §2.3.2 we have already seen how LEO could lower trans-Atlantic communication latencies, in-depth analysis of the hybrid networks is left to future work.

Low latency networks, built with specialized infrastructures like terrestrial low latency radio or LEO satellites, will likely offer transit at a cost higher than today's Internet. For some context, the cost per GB of cISP and Starlink beta, as discussed in §6.5, are at least ∼4× higher than the cost per GB for content delivery networks [211]. Hence, in scenarios where both traditional fiber and low-latency infrastructures are available, traffic that is latency-critical could use the specialized low-latency network services, while the rest of the traffic continues to use default, lower-cost paths. For private WANs, like those operated by Google and Microsoft, integrating such traffic differentiation is likely straightforward, but exposing this possibility more broadly across the public Internet may require substantial changes. One possibility is for ISPs to use heuristics to pick the low-latency route

for a subset of customer traffic, such as sending all VoIP traffic up to a certain traffic volume over low-latency channels. Another is to expose the choice to customers, such that an agent on their operating system network stack decides when to send on the low-latency route. More generally, path-aware networking approaches [221] could be used to accommodate such path diversity.

6.8 Related publications

Some of the plots and the corresponding discussions in this chapter have been taken from the following publication:

- D. Bhattacherjee, W. Aqeel, S. A. Jyothi, I. N. Bozkurt, W. Sentosa, M. Tirmazi, A. Aguirre, B. Chandrasekaran, B. Godfrey, G. P. Laughlin, B. M. Maggs, and A. Singla, "cISP: A Speed-of-Light Internet Service Provider," in *USENIX NSDI*, [To be published], 2022

I am joint-first author along with W. Aqeel.
At the time of writing this dissertation, the final version is not publicly available, but a prior technical report is available on arXiv:
https://arxiv.org/pdf/1809.10897

7

Contributions & future work

We are entering an era where network elements (LEO satellites) fly at more than twice the speed of sound in air. Such incomparable dynamicity leads to both continuous and abrupt changes in the structure and properties of paths within these networks. No terrestrial network has such properties, and the few deployed geosynchronous communication satellites have either no or very limited mobility. In order to efficiently deliver traffic over such a globally spanning dynamic network, one needs to rethink all the core network design challenges. How do we place satellites and connect them together? How do we route packets through the network given the link conditions and end-to-end paths change all the time? How do communicating endpoints cope up with a highly dynamic network infrastructure?

Summary of contributions. This dissertation quantifies broadband LEO network performance opportunities, highlights the importance of having inter-satellite connectivity, and qualitatively and quantitatively identifies the topology design, routing and traffic engineering, and transport challenges. It proposes a novel *motif*-based LEO topology design scheme

achieving ∼2× higher efficiency than the state-of-the-art. We further observe that the spatial geometry of the problem admits even more efficient solutions than single motifs, *if* a limited, controlled form of dynamic interconnection is permissible. For the largest and most mature of the planned constellations, Starlink, our approach promises 54% higher efficiency under reasonable assumptions on link range, and 40% higher efficiency in even the most pessimistic scenarios. By analyzing a range of uncertain constraints in this manner, this work attempts to arrive at robust results that we hope can influence the design of the planned satellite constellations, as well as motivate further research in this exciting space. Also, this work offers an open-sourced simulation and visualization framework, Hypatia, set to lower the barrier for networking researchers to engage in LEO networking research. It demonstrates Hypatia's utility in understanding the behavior of such networks, especially the temporal variations in the structure of paths and their latencies. It draws out some of the implications of this LEO network dynamism for congestion control, and routing and traffic engineering.

While the body of work included in this dissertation is a good start at addressing LEO networking challenges and unleashing the LEO broadband potential, there are various future research directions to be explored.

7.1 LEO topology design nuances

As LEO networks get deployed, it is important to explore and analyze various nuances of topology design that are absent in terrestrial networks and previous generation LEO constellations.

Connectivity beyond intra-shell. Recent advances [27, 28] demonstrate

laser connectivity between satellites in largely different trajectories. Imagine a constellation with a polar and a non-polar shell of satellites trying to achieve laser inter-satellite connectivity across shells (Starlink recently deployed 10 polar satellites with ISL capabilities [73]). Such deployments are critical for serving polar regions, where installing terrestrial gateways is difficult.

Further, such possibilities open up a new horizon for topology design where satellites at largely different heights, like GEO, MEO, and LEO, can communicate with each other to offer broadband as well as other services like Earth monitoring. Given multi-Gbps LEO-to-GEO laser links have been demonstrated already, such a hybrid design could benefit from the high throughput and low latency offerings of LEO as well as the MEO/GEO stability due to significantly reduced orbital dynamics.

Our motif-based topology design approach could be extended to incorporate such inter-shell/layer connectivity (both polar-to-inclined and LEO-to-MEO/GEO). Instead of taking into consideration only the connectivity options for a single satellite, cohorts of nearby satellites moving together within a shell could be considered as units with optimal motifs for intra- and inter-cohort connectivity. A few links per cohort could be dedicated to inter-shell/layer connectivity. Given the vastly different speeds at different altitudes and the limited resources at each shell/layer, these links will need to change over time in a dynamic manner. The rest of the cohort's links are within the same shell, maintained continuously, and benefit from the motif abstraction. This strategy shows the generality of the 'motif' abstraction, but the development and analyses of methods customized for such settings are left to future work.

Emulating satellite-hop behavior. The topology design work in this

dissertation considers hop count as a performance metric, and demonstrates lower hop count leading to less congestion. Nevertheless, it is important to model satellite hops and emulate hop behavior in order to understand their impact in greater detail. However, accounting for queuing and processing delays at each hop would only improve our motif-based topology design results, as we achieve much lower hop counts than +Grid.

Sun-synchronous orbits. Some of these constellations plan to deploy sun-synchronous orbits [44], which allow similar spatial coverage at the same local time everyday. This calls for an in-depth analysis of the temporal and spatial variations in global Internet traffic demands, and the alignment of such demands with the supply of resources (LEO satellites) in such orbits.

A systematic exploration of the whole LEO topology design space can push these networks towards better network performance.

7.2 Co-designing the LEO network stack

In order to improve the performance of a system, it is critical to make the right design choices and have the right interfaces between the individual components. In the context of satellite networks, topology, routing, and end-to-end transport can be co-designed towards a specific operating point in the performance tradeoff space. In a static network with no cross traffic, shortest path routing can be an optimal single-path routing choice, but in a dynamic satellite network, this greedy approach might see frequent path changes affecting end-to-end transport. While compartmentalizing network functionalities was necessary for the evolution of the Internet, new age networks need special treatment. Transport techniques hidden well within the network stack have now become user-space [222–

224] and are currently seeing widespread adoption in the Internet. Also, existing work [225] has demonstrated the possibility of applications explicitly stating their demands to transport. Such advances make us optimistic about systematically improving cross-layer communications such that the entire network stack works in synchrony towards the same set of objectives as needed by the applications.

If globally deployed LEO networks start serving a large fraction of the Internet traffic, it is enticing to explore **clean-slate network design**. It does not necessarily mean reinventing the wheel, but can benefit from re-visiting promising techniques explored in the past. Although network-assisted end-to-end congestion control [226, 227] has been explored in the past, large service providers are yet to see such deployments due to the inherent complexity in today's Internet. If LEO satellite networks start offering end-to-end traffic delivery, they can be great testbeds for such techniques. While one design choice (proactive endpoints) could be to allow the endpoints to advertise their demands to the network every round-trip communication, another choice (proactive network) is to expose the physical layer latency change information to end-to-end transport. More broadly, they can open up the possibility of addressing some core networking problems without necessarily limiting the solution space due to constraints inherent in the Internet.

In this context, it is worth discussing how LEO broadband might benefit from borrowing networking techniques used in recently proposed clean-slate path-aware design approaches [228, 229] and, more specifically, SCION [221]. Interesting LEO properties include multiple low-latency paths between city-pairs which change every few tens of seconds due to LEO dynamics. Note that in LEO networks the endpoints and the network could be different enti-

ties – even for the largest constellations, ground stations could be operated by third parties [230, 231]. Given this scenario, it would be interesting to explore, as in [221], the inherent multi-path opportunities for transferring data:

- Both endpoints and the network have a say in deciding which paths could be used for transferring data. The network might decide not to expose certain paths to certain endpoints, while the endpoints can pick from the offered paths based on application demands (low latency versus high throughput).
- Multi-path communication possibilities make denial-of-service attacks more challenging. Following path-aware networking techniques, an adversary would be able to use a LEO path only for few tens of seconds before the path ceases to exist, thus significantly limiting the attack vector.

The above discussions highlight the importance of clean-slate LEO network design.

7.3 LEO networks need new measurement techniques

Various existing space and satellite geodetic techniques [232] like interferometry, laser ranging, and altimetry also allow us to precisely estimate ionospheric and tropospheric conditions, Earth's gravity field, etc. which would impact the network performance of individual satellites. Assimilating varying sources of information would allow us to predict and validate performance over time. Also, an understanding of the environment and the corresponding network performance will lead us to reconstruct proprietary

behavior like encoding and error correction. Existing work [233], for example, uses player and network state logs to reconstruct video streaming algorithms. Similarly, knowledge of the environment and access to both communicating endpoints can uncover proprietary satellite network protocols. Such reconstructions would help us explain network performance differences.

7.4 LEO edge-compute

What if LEO mega-constellations also offer in-orbit compute as a service, much like cloud computing from today's terrestrial data centers? In-orbit compute could extend the cloud's promise of computing *when* you want, to computing *wherever* you want. Current cloud data center maps are relatively sparse, with hardly any sites in many geographies, such as South America, Africa, and large parts of Asia. Even CDN *edge* locations, that offer more limited services, incur 100+ ms latencies in many places [234–237]. In contrast, a large LEO constellation can be within a few milliseconds from everywhere on Earth, including locations unsuitable for terrestrial facilities, *e.g.,* due to poor power and support infrastructure, or prohibitive political and legal concerns. In-orbit compute can thus offer ubiquitous "edge computing" capabilities, without the many hurdles in deploying terrestrial infrastructure in many locations.

One satellite may not offer a large amount of available compute, so we quantify how many satellites are reachable from a ground location at any time. As Fig. 7.1 shows, for Kuiper, 10+ satellite-servers would be reachable from any location for most latitudes Kuiper services. For Starlink, 30+ satellites are reachable from almost all locations at all times; typically more than 40 satellites are reachable. These numbers are similar to what

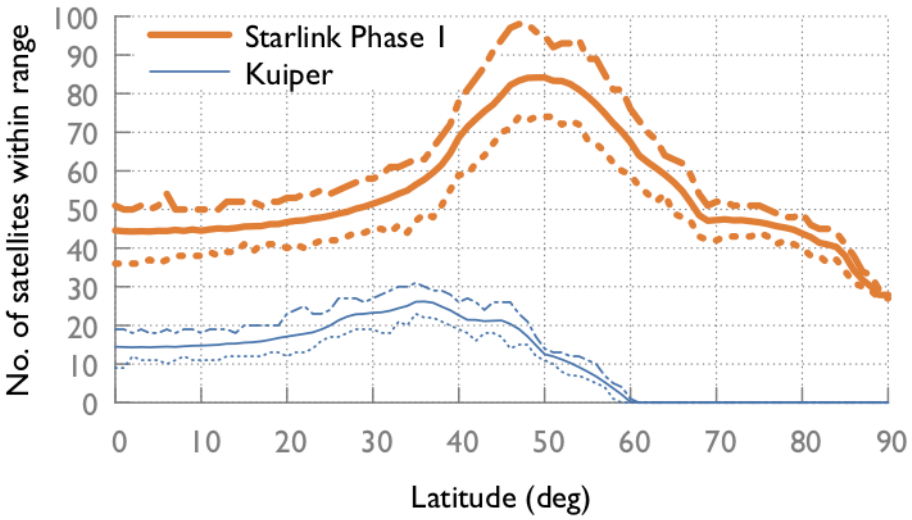


Figure 7.1: *A substantial number of satellites would be reachable from everywhere at all times. The solid lines show the average number across time, with the range over time shown by dotted and dashed lines. The peak at certain latitudes (less pronounced for Kuiper) arises from orbital geometry of the planned constellations.*

is being envisioned for "cloudlets" or edge computing sites [238, 239].

Our recently published work at HotNets'20 examines, qualitatively and quantitatively, the opportunities and challenges of in-orbit computing. Several applications could benefit from it, including content distribution and edge computing; multi-user gaming, co-immersion, and collaborative music; and processing space-native data. Adding computing hardware to a satellite does not seem prohibitive in terms of weight, volume, and space hardening, but the required power draw could be substantial. Another challenge stems from the dynamics of low Earth orbit: a specific satellite is only visible to a ground station for minutes at a time, thus requiring care in managing stateful applications. Our exploration of these trade-offs suggests that this "outlandish" proposition should not be casually dismissed, and may merit deeper engagement from the research community.

The relevant publication is:

- D. Bhattacherjee, S. Kassing, M. Licciardello, and A. Singla, "In-orbit Computing: An Outlandish thought Experiment?" In *ACM HotNets*, 2020

I am joint-first author along with S. Kassing.

7.5 Impact of solar superstorm on LEO networks

A recent work [241] sheds light on the possible impact of coronal mass ejection (CME), also known as solar superstorm, on terrestrial Internet. Very strong solar storms could generate geo-magnetically induced current (GIC) on the Earth surface, enter terrestrial infrastructure, and damage them. It is important to analyze the impact of such solar storms also on LEO satellite infrastructure. While Starlink and Kuiper plan to deploy at heights lower than the inner Van Allen radiation belt (outwards from 643 km), intense solar activities could push this belt to far lower heights, thus effectively increasing radiation significantly at LEO heights. While GIC should not be an issue for satellites, higher radiation could potentially lead to long periods of lost connectivity due to either ground and/or orbital infrastructure being systematically turned off or communication components malfunctioning, damage to the communication and compute components, and also orbital decay. The worst-case scenario would be satellite collisions during such periods of lost connectivity and orbital decay, eventually triggering Kessler syndrome, thus making the space unusable for years.

Space agencies are exploring the possibility to monitor and predict such solar superstorms days in advance [242]. Such predictions could allow the constellations to proactively maneuver satellites in time in order to place them

in orbits (slightly different heights, lower inclinations) with lower chances of collision during probable autonomous operations. Precisely modeling solar superstorms and simulating satellite maneuvers to mitigate such events are out of scope of this work.

While the previous chapter concluded by envisioning a globally-spanning hybrid low-latency network design, this chapter focused on the future of LEO network research while also touching upon the possibility of placing compute in LEO. No matter how network design evolves in the future, LEO mega-constellation deployments, with their unique opportunities, are set to change the face of global connectivity forever. This work contributes to this ongoing 'revolution' by quantifying the performance of such networks, exploring topology design, and offering a framework to enable broader community research.

Bibliography

[1] SpaceX Starlink, https://www.spacex.com/webcast, 2017.

[2] A. Boyle, *Amazon to offer broadband access from orbit with 3,236-satellite 'Project Kuiper' constellation*, https://www.geekwire.com/2019/amazon-project-kuiper-broadband-satellite/, 2019.

[3] Telesat, *Telesat: Global Satellite Operators*, https://www.telesat.com/, 2020.

[4] OneWeb, *How OneWeb is changing global communications*, http://www.oneweb.world/, 2018.

[5] J. Foust, *SpaceX launches Starlink satellites and rideshare payloads*, https://spacenews.com/spacex-launches-starlink-satellites-and-rideshare-payloads/, 2021.

[6] S. Erwin, *SpaceX plans to start offering Starlink broadband services in 2020*, https://spacenews.com/spacex-plans-to-start-offering-starlink-broadband-services-in-2020/, 2019.

[7] L. Grush, *SpaceX may spin out internet-from-space business and make it public*, https://www.theverge.com/2020/2/6/21126540/spacex-starlink-internet-space-mega-constellation-ipo-spinoff-gwynne-shotwell, 2020.

[8] F. Meyer, *A new network design for the "Internet from space"*, https://ethz.ch/en/news-and-events/eth-news/news/2019/12/a-new-network-design-for-the-internet-from-space.html, 2019.

[9] D. Canellis, *Bezos and Musk's internet-from-space race is back on*, https://thenextweb.com/hardfork/2020/03/30/oneweb-collapse-internet-space-race-leo-satellite-bezos-musk-back-on/, 2020.

[10] L. Tung, *SpaceX's public beta of internet from space service coming by fall 2020*, https://www.zdnet.com/article/elon-musk-spacexs-public-beta-of-internet-from-space-service-coming-by-fall-2020/, 2020.

[11] S. Roy Choudhury, *Super-fast internet from satellites is the next big thing in the space race*, https://www.cnbc.com/2019/07/22/fast-internet-via-satellites-is-the-next-big-thing-in-the-space-race.html, 2019.

[12] HughesNet, https://www.hughesnet.com/, 2020.

[13] Viasat, https://www.viasat.com/, 2020.

[14] J. Brodkin, *SpaceX hits two milestones in plan for low-latency satellite broadband*, https://tinyurl.com/yb9t5cf6, 2018.

[15] *GPS: The Global Positioning System*, https://www.gps.gov/, 2018.

[16] IAC, *GLONASS*, https://www.glonass-iac.ru/en/, 2018.

[17] European Commission, *Galileo*, https://tinyurl.com/ydbcrgjh, 2018.

[18] SES, https://www.ses.com/networks/, 2018.

[19] O3b Networks and Sofrecom, *Why Latency Matters to Mobile Backhaul*, https://tinyurl.com/yc4vor3e, 2017.

[20] Iridium Communications Inc., *Iridium Satellite Communications*, https://www.iridium.com/, 2020.

[21] Iridium Communications Inc., *Iridium NEXT*, https://www.iridiumnext.com/, 2020.

[22] J. Foust, *SpaceX sets booster reuse milestone on Starlink launch*, https://spacenews.com/spacex-sets-booster-reuse-milestone-on-starlink-launch/, 2021.

[23] H. Jones, "The recent large reduction in space launch cost," 48th International Conference on Environmental Systems, 2018.

[24] SpaceX, *FALCON 9*, https://www.spacex.com/vehicles/falcon-9/, 2021.

[25] SpaceX, *FALCON HEAVY*, https://www.spacex.com/vehicles/falcon-heavy/, 2021.

[26] F. Davoli, C. Kourogiorgas, M. Marchese, A. Panagopoulos, and F. Patrone, "Small satellites and CubeSats: Survey of structures, architectures, and protocols," *International Journal of Satellite Communications and Networking*, 2019.

[27] Z. Sodnik, B. Furch, and H. Lutz, "Optical intersatellite communication," *IEEE journal of selected topics in quantum electronics*, vol. 16, no. 5, pp. 1051–1057, 2010.

[28] ESA, *Alphasat Optical Communication*, https://tinyurl.com/y23vthrs, 2013.

[29] Mynaric, https://mynaric.com/, 2020.

[30] Tesat, http://www.tesat.de/en/, 2020.

[31] Mynaric, *Annual Report 2019*, https://mynaric.com/wp-content/uploads/2020/05/AR_2019_EN_Website.pdf, 2019.

[32] Airbus Defence & Space SAS, *Towards the All Optical satellite communications system*, https://nebula.esa.int/sites/default/files/neb_study/1276/C4000117594ExS.pdf, 2018.

[33] C. Carrizo, M. Knapek, J. Horwath, D. D. Gonzalez, and P. Cornwell, "Optical inter-satellite link terminals for next generation satellite constellations," in *Free-Space Laser Communications XXXII*, International Society for Optics and Photonics, 2020.

[34] J. Foust, *SpaceX adds laser crosslinks to polar Starlink satellites*, https://spacenews.com/spacex-adds-laser-crosslinks-to-polar-starlink-satellites/, 2021.

[35] A. Mauldin, *Will New Satellites End the Dominance of Submarine Cables?* https://blog.telegeography.com/will-new-satellites-end-the-dominance-of-submarine-cables, 2019.

[36] SpaceX FCC filing, *SpaceX V-band non-geostationary satellite system*, https://licensing.fcc.gov/myibfs/download.do?attachment_key=1190019, 2017.

[37] A. Rebatta, *295 Tbps: Internet Traffic and Capacity in 2017*, https://tinyurl.com/y73pq8u4, 2017.

[38] IRTF, *Applied Networking Research Prize*, https://irtf.org/anrp/, 2021.

[39] IMC 2020 Award Committee, *Award Winners*, https://conferences.sigcomm.org/imc/2020/awards/, 2020.

[40] D. Bhattacherjee and A. Singla, *LEOCONN 2021*, https://leoconn.github.io/, 2021.

[41] IETF, *SATCOM activities within IETF and IRTF*, https://trac.ietf.org/trac/ietf/meeting/wiki/111sidemeetings#point4, 2021.

[42] SatNet authors, *Polar and inclined orbits with +Grid connectivity*, https://satnet-authors.github.io/cesium_orbit_grid_demo.html, 2019.

[43] ESA, *Polar and Sun-synchronous orbit*, http://www.esa.int/ESA_Multimedia/Images/2020/03/Polar_and_Sun-synchronous_orbit, 2020.

[44] SpaceX FCC update, *APPLICATION FOR MODIFICATION OF AUTHORIZATION FOR THE SPACEX NGSO SATELLITE SYSTEM*, https://fcc.report/IBFS/SAT-MOD-20200417-00037/2274315.pdf, 2020.

[45] Telesat, *Application for modification of market access authorization*, https://fcc.report/IBFS/SAT-MPL-20200526-00053/2378318.pdf, 2020.

[46] OrbiterWiki, *Launch Azimuth*, https://www.orbiterwiki.org/wiki/Launch_Azimuth, 2019.

[47] H. Rose McLaughlin, *Boca Chica: The tiny Texas town where SpaceX is building Starship*, https://tinyurl.com/y65zb3fe, 2019.

[48] SpaceX FCC update, *SPACEX non-geostationary satellite system*, https://licensing.fcc.gov/myibfs/download.do?attachment_key=1569860, 2018.

[49] Kuiper Systems LLC, *Application of Kuiper Systems LLC for Authority to Launch and Operate a Non-Geostationary Satellite Orbit System in Ka-band Frequencies*, https://licensing.fcc.gov/myibfs/download.do?attachment_key=1773885, 2019.

[50] marine.rutgers.edu, *Keplerian Elements*, https://marine.rutgers.edu/cool/education/class/paul/orbits.html, 2001.

[51] M. Handley, "Delay is Not an Option: Low Latency Routing in Space," in *ACM HotNets*, 2018.

[52] J. Foust, *ESA spacecraft dodges potential collision with Starlink satellite)*, https://spacenews.com/esa-spacecraft-dodges-potential-collision-with-starlink-satellite/, 2019.

[53] T. Fernholz, *SpaceX's new satellites will dodge collisions autonomously (and they'd better)*, https://qz.com/1627570/how-autonomous-are-spacexs-starlink-satellites/, 2019.

[54] D. Bhattacherjee and A. Singla, "Network topology design at 27,000 km/hour," in *ACM CoNEXT*, 2019.

[55] SpaceX, *SPACEX non-geostationary satellite system*, https://fcc.report/IBFS/SAT-LOA-20161115-00118/1158350.pdf, 2016.

[56] ——, *SPACEX non-geostationary satellite system*, https://fcc.report/IBFS/SAT-MOD-20190830-00087/1877671, 2019.

[57] M. Holmes, *In the Eye of the Storm: Greg Wyler Breaks Cover to Talk OneWeb*, http://interactive.satellitetoday.com/via/oneweb-special-edition/in-the-eye-of-the-storm-greg-wyler-breaks-cover-to-talk-oneweb/, 2020.

[58] Kuiper USASAT-NGSO-8A ITU filing, https://www.itu.int/ITU-R/space/asreceived/Publication/DisplayPublication/8716, 2018.

[59] Kuiper USASAT-NGSO-8B ITU filing, https://www.itu.int/ITU-R/space/asreceived/Publication/DisplayPublication/8774, 2018.

[60] Kuiper USASAT-NGSO-8C ITU filing, https://www.itu.int/ITU-R/space/asreceived/Publication/DisplayPublication/8718, 2018.

[61] SpaceX FCC filing, *Application for approval for orbital deployment and operating authority for the spacex ngso satellite system*, https://tinyurl.com/y7mvpdvz, 2016.

[62] L. Wood, "Internetworking with satellite constellations," Ph.D. dissertation, University of Surrey, 2001.

[63] I. del Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," in *Acta Astronautica*, Elsevier, 2019.

[64] O. L. De Weck, R. D. Neufville, and M. Chaize, "Staged deployment of communications satellite constellations in low earth orbit," in *Journal of Aerospace Computing, Information, and Communication*, 2004.

[65] A. Siddiqi, J. Mellein, and O. de Weck, "Optimal reconfigurations for increasing capacity of communication satellite constellations," in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, 2005.

[66] LeoSat, *Technical Overview*, http://leosat.com/media/1114/leosat-technical-overview.pdf, 2018.

[67] T. Vladimirova and K. Sidibeh, *Inter-Satellite Links in LEO Constellations of Small Satellites*, http://www.ee.surrey.ac.uk/m_ssc/research/vlsi/intersatellite.html, 2007.

[68] Wikipedia, *Thermosphere*, https://en.wikipedia.org/wiki/Thermosphere, 2019.

[69] M. Handley, "Using Ground Relays for Low-Latency Wide-Area Routing in Megaconstellations," in *ACM HotNets*, 2019.

[70] Y. Hauri, D. Bhattacherjee, M. Grossmann, and A. Singla, "'Internet from Space' without Inter-satellite Links," in *ACM HotNets*, 2020.

[71] D. Bhattacherjee, W. Aqeel, I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Gearing up for the 21st century space race," in *ACM HotNets*, 2018.

[72] E. Ralph, *SpaceX Starlink 'space lasers' successfully tested in orbit for the first time*, https://www.teslarati.com/spacex-starlink-space-lasers-first-orbital-test/, 2020.

[73] J. Foust, *SpaceX adds laser crosslinks to polar Starlink satellites*, https://spacenews.com/spacex-adds-laser-crosslinks-to-polar-starlink-satellites/, 2021.

[74] R. Durairajan, P. Barford, J. Sommers, and W. Willinger, "InterTubes: A Study of the US Long-haul Fiber-optic Infrastructure," in *ACM SIGCOMM*, 2015.

[75] WonderNetwork, *Global Ping Statistics*, https://wondernetwork.com/pings.

[76] Fibre Atlantic, *GTT Express*, http://www.fiberatlantic.com/system/J6Qmo, 2015.

[77] D. Bhattacherjee, W. Aqeel, S. A. Jyothi, I. N. Bozkurt, W. Sentosa, M. Tirmazi, A. Aguirre, B. Chandrasekaran, B. Godfrey, G. P. Laughlin, B. M. Maggs, and A. Singla, "cISP: A Speed-of-Light Internet Service Provider," in *USENIX NSDI*, [To be published], 2022.

[78] G. Laughlin, A. Aguirre, and J. Grundfest, "Information transmission between financial markets in Chicago and New York," *Financial Review*, 2014.

[79] Econoday, *Econoday*, http://mam.econoday.com/, 2018.

[80] Louis, B., Baker, N., and McCormick, J., *Hft traders dust off century-old tool in search of market edge*, https://tinyurl.com/ycl8m3yg, 2018.

[81] J. Brutlag, *Speed Matters for Google Web Search*, http://goo.gl/vJq1lx, 2009.

[82] M. Chow, D. Meisner, J. Flinn, D. Peek, and T. F. Wenisch, "The Mystery Machine: End-to-end Performance Analysis of Large-scale Internet Services," in *USENIX OSDI*, 2014.

[83] Akamai, *Akamai "10for10"*, `https://www.akamai.com/us/en/multimedia/documents/brochure/akamai-10for10-brochure.pdf`, Jul. 2015.

[84] Amazon Web Services, Inc., *Alexa Top Sites*, `https://aws.amazon.com/alexa-top-sites/`.

[85] S. Sanfilippo, *hping*, `http://www.hping.org/`.

[86] M. Belshe, "More bandwidth doesn't matter (much)," *Google Inc*, 2010.

[87] S. Sundaresan, "Characterizing and improving last mile performance using home networking infrastructure," Ph.D. dissertation, Georgia Institute of Technology, 2014.

[88] Z. S. Bischof, J. S. Otto, and F. E. Bustamante, "Up, down and around the stack: ISP characterization from network intensive applications," *ACM SIGCOMM W-MUST*, 2012.

[89] R. Netravali, A. Sivaraman, K. Winstein, S. Das, A. Goyal, and H. Balakrishnan, "Mahimahi: a lightweight toolkit for reproducible web measurement," in *ACM CCR*, 2014.

[90] Sitespeed.io, *Welcome to the wonderful world of Web Performance*, `https://www.sitespeed.io/`.

[91] D. Bhattacherjee, M. Tirmazi, and A. Singla, "Measuring and exploiting the cloud consolidation of the Web," *arXiv:1906.04753*, 2019, Preprint.

[92] M. Harris, *SpaceX Claims to Have Redesigned Its Starlink Satellites to Eliminate Casualty Risks*, `https://tinyurl.com/y5adgwcp`, 2019.

[93] R. W. Kingsbury, "Mobile Ad hoc networks for oceanic aircraft communications," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[94] D. Bhattacherjee, W. Aqeel, I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Gearing up for the 21st century space race," in *ACM HotNets*, 2018.

[95] Greater London Authority (GLA), *Global City Population Estimates*, `https://data.london.gov.uk/dataset/global-city-population-estimates`, 2018.

[96] *Flightaware API*, https://tinyurl.com/zsgd7fq, 2018.

[97] T. Karin, *global-land-mask*, https://pypi.org/project/global-land-mask/, 2019.

[98] S. Agarwal and J. R. Lorch, "Matchmaking for online games and other latency-sensitive P2P systems," in *ACM SIGCOMM*, 2009.

[99] M. Claypool and K. Claypool, "Latency can kill: Precision and deadline in online games," in *Proceedings of the first annual ACM SIGMM conference on Multimedia systems*, 2010.

[100] S. Kassing, *floodns: temporal routed flow simulation*, https://github.com/snkas/floodns, 2020.

[101] D. Nace, N.-L. Doan, E. Gourdin, and B. Liau, "Computing optimal max-min fair resource allocation for elastic flows," in *IEEE/ACM Transactions on Networking*, 2006.

[102] I. del Portillo, *Itu-rpy: A python implementation of the itu-r p. recommendations to compute atmospheric attenuation in slant and horizontal paths.* https://github.com/iportillo/ITU-Rpy/, 2017.

[103] J. Petranovich, "Mitigating the effect of weather on ka-band high-capacity satellites," in *ViaSat Inc., Carlsbad, CA*, 2012.

[104] ETSI, *Digital Video Broadcasting (DVB) Implementation guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications*, https://dvb.org/wp-content/uploads/2019/12/a171-1_s2_guide.pdf, 2015.

[105] Water Encyclopedia, *Global Distribution of Precipitation*, http://www.waterencyclopedia.com/Po-Re/Precipitation-Global-Distribution-of.html, 2020.

[106] Zoran Sodnik, ESA, *Future and Challenges on Optical Communication Terminal Developments: ESA's view*, https://tinyurl.com/yd6vjzpg, 2020.

[107] M. Krebs, *Satellite Constellation*, https://patents.google.com/patent/US20170005719A1/en, 2016.

[108] M. Handley, *Starlink revisions, Nov 2018*, https://www.youtube.com/watch?v=QEIUdMiColU, 2018.

[109] J. Ma, X. Qi, and L. Liu, "An Effective Topology Design Based on LEO/GEO Satellite Networks," in *International Conference on Space Information Network*, Springer, 2017.

[110] K. Sidibeh, "Adaption of the IEEE 802.11 protocol for inter-satellite links in LEO satellite networks," Ph.D. dissertation, University of Surrey (United Kingdom), 2008.

[111] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *USENIX NSDI*, 2012.

[112] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira, "Xpander: Towards optimal-performance datacenters," in *ACM CoNEXT*, 2016.

[113] M. Besta and T. Hoefler, "Slim fly: A cost effective low-diameter network topology," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE Press, 2014.

[114] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM CCR*, ACM, vol. 38, 2008, pp. 63–74.

[115] D. S. Johnson, J. K. Lenstra, and A. R. Kan, "The complexity of the network design problem," *Networks*, vol. 8, no. 4, pp. 279–285, 1978.

[116] M. Werner, "A dynamic routing concept for ATM-based satellite personal communication networks," *IEEE JSAC*, 1997.

[117] R. Mauger and C. Rosenberg, "QoS guarantees for multimedia services on a TDMA-based satellite network," *IEEE Communications*, vol. 35, no. 7, 1997.

[118] J. Bai, X. Lu, Z. Lu, and W. Peng, "A distributed hierarchical routing protocol for non-GEO satellite networks," in *IEEE ICPP Workshops*, 2004.

[119] B. Jianjun, L. Xicheng, L. Zexin, and P. Wei, "Compact explicit multi-path routing for LEO satellite networks," in *IEEE HPSR*, 2005.

[120] G. Song, M. Chao, B. Yang, and Y. Zheng, "TLR: A traffic-light-based intelligent routing strategy for NGEO satellite IP networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 6, pp. 3380–3393, 2014.

[121] T. Taleb, D. Mashimo, A. Jamalipour, N. Kato, and Y. Nemoto, "Explicit load balancing technique for NGEO satellite IP networks with on-board processing capabilities," *IEEE/ACM TON*, 2009.

[122] I. F. Akyildiz, E. Ekici, and M. D. Bender, "MLSR: a novel routing algorithm for multilayered satellite IP networks," *IEEE/ACM TON*, 2002.

[123] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, *et al.*, "Detour: Informed Internet routing and transport," *IEEE Micro*, 1999.

[124] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *ACM SIGCOMM CCR*, vol. 32, no. 1, pp. 66–66, 2002.

[125] A. Akella, B. Maggs, S. Seshan, and A. Shaikh, "On the performance benefits of multihoming route control," *IEEE/ACM TON*, 2008.

[126] V. Valancius, B. Ravi, N. Feamster, and A. C. Snoeren, "Quantifying the benefits of joint content and network routing," in *SIGMETRICS*, 2013.

[127] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "The flattening internet topology: natural evolution, unsightly barnacles or contrived collapse?" In *PAM*, 2008.

[128] Y.-C. Chiu, B. Schlinker, A. B. Radhakrishnan, E. Katz-Bassett, and R. Govindan, "Are we one hop away from a better Internet?" In *ACM IMC*, 2015.

[129] G. Giuliari, T. Klenze, M. Legner, D. Basin, A. Perrig, and A. Singla, "Internet backbones in space," *ACM SIGCOMM Computer Communication Review*, 2020.

[130] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: end-to-end congestion control for wired/wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 467–479, 2002.

[131] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: a new congestion control scheme for satellite IP networks," *IEEE/ACM TON*, 2001.

[132] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting Congestion Control for Consistent High Performance.," in *USENIX NSDI*, 2015.

[133] M. Dong, T. Meng, D Zarchy, E Arslan, Y Gilad, B Godfrey, and M Schapira, "PCC Vivace: Online-Learning Congestion Control," in *USENIX NSDI*, 2018.

[134] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: congestion-based congestion control," in *Communications of the ACM*, 2017.

[135] V. Arun and H. Balakrishnan, "Copa: Practical Delay-Based Congestion Control for the Internet," in *USENIX NSDI*, 2018.

[136] Y. Hauri, D. Bhattacherjee, M. Grossmann, and A. Singla, "'Internet from Space' without Inter-satellite Links," in *ACM HotNets*, 2020.

[137] D. Bhattacherjee and A. Singla, *Network topology design at 27,000 km/hour*, `https://satnetwork.github.io`, 2019.

[138] ISS National Lab, *ISSRDC 2015 - A Conversation with Elon Musk*, `https://www.youtube.com/watch?v=ZmEg95wPiVU`, 2015.

[139] B. D. McKay and N. C. Wormald, "Uniform generation of random regular graphs of moderate degree," *Journal of Algorithms*, 1990.

[140] M. Dorigo and M. Birattari, *Ant colony optimization*. Springer, 2010.

[141] O. Montenbruck and E. Gill, *Satellite orbits: models, methods and applications*. Springer Science & Business Media, 2012.

[142] Telesat, *Telesat's responses - Federal Communications Commission*, `http://licensing.fcc.gov/myibfs/download.do?attachment_key=1205775`, 2018.

[143] S. Amiri and B. Reif, "Internet penetration and its correlation to gross domestic product: An analysis of the nordic countries," *International Journal of Business, Humanities and Technology*, 2013.

[144] Wikipedia, *"2014 est. PPP-adjusted GDP ($BN)" by Brookings Institution*, `https://en.wikipedia.org/wiki/List_of_cities_by_GDP`, 2014.

[145] NetworkX developers, *NetworkX*, `https://networkx.github.io/`, 2020.

[146] B. Gavish and J. Kalvenes, "The impact of intersatellite communication links on LEOS performance," *Telecommunication Systems*, 1997.

[147] J. H. Bau, "Topologies for satellite constellations in a cross-linked space backbone network," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[148] V. W. Chan, "Optical space communications," in *IEEE Journal of Selected Topics in Quantum Electronics*, 2000.

[149] K. C. H. Kwok, "Cost optimization and routing for satellite network constellations," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[150] J. V. Evans, "Satellite systems for personal communications," in *Proceedings of the IEEE*, 1998.

[151] M. Werner, A. Jahn, E. Lutz, and A. Bottcher, "Analysis of system parameters for LEO/ICO-satellite communication networks," in *IEEE Journal on Selected areas in Communications*, 1995.

[152] V. W. Chan, "Optical space communications: A key building block for wide area space networks," in *IEEE LEOS Annual Meeting Conference Proceedings*, 1999.

[153] W. W. Wu, E. F. Miller, W. L. Pritchard, and R. L. Pickholtz, "Mobile satellite communications," in *Proceedings of the IEEE*, 1994.

[154] J. D. Kiesling, "Land mobile satellite systems," in *Proceedings of the IEEE*, 1990.

[155] G. Comparetto and N. Hulkower, "Global mobile satellite communications-A review of three contenders," in *15th International Communications Satellite Systems Conference and Exhibit*, AIAA, 1994.

[156] D. E. Sterling and J. E. Hatlelid, "The IRIDIUM system-a revolutionary satellite communications system developed with innovative applications of technology," in *MILCOM 91-Conference record*, IEEE, 1991.

[157] R. WIEDEMAN, A. SALMASI, and D. Rouffet, "Globalstar-Mobile communications where ever you are," in *14th International Communication Satellite Systems Conference and Exhibit*, AIAA, 1992.

[158] R. Akturan and W. J. Vogel, "Path diversity for LEO satellite-PCS in the urban environment," in *IEEE Transactions on Antennas and Propagation*, 1997.

[159] G. Maral, "The ways to personal communications via satellite," in *International journal of satellite communications*, Wiley Online Library, 1994.

[160] B. Soret, I. Leyva-Mayorga, and P. Popovski, "Inter-plane satellite matching in dense LEO constellations," *arXiv preprint arXiv:1905.08410*, 2019.

[161] O. B. Karimi, J. Liu, and C. Wang, "Seamless wireless connectivity for multimedia services in high speed trains," *IEEE Journal on selected areas in communications*, vol. 30, no. 4, pp. 729–739, 2012.

[162] X. Lin, V. Yajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H.-L. Maattanen, M. Bergstrom, S. Euler, and Y.-P. E. Wang, "The sky is not the limit: LTE for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 204–210, 2018.

[163] M. Ansariola, M. Megraw, and D. Koslicki, "IndeCut evaluates performance of network motif discovery algorithms," *Bioinformatics*, vol. 34, no. 9, pp. 1514–1521, 2017.

[164] S. Purohit, L. Holder, and G. Chin, "Temporal graph generation based on a distribution of temporal motifs," in *Proceedings of the 14th International Workshop on Mining and Learning with Graphs*, 2018.

[165] H. Kugler, S.-J. Dunn, and B. Yordanov, "Formal Analysis of Network Motifs," in *International Conference on Computational Methods in Systems Biology*, Springer, 2018, pp. 111–128.

[166] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Topological generalizations of network motifs," *Physical Review E*, vol. 70, no. 3, p. 031 909, 2004.

[167] Y. Shavitt, *Network Motifs and Efficient Counting of Graphlets*, `https://tinyurl.com/y2ym2vxf`, 2013.

[168] M. Davis, *Visual design in dress*. Prentice-Hall, 1980.

[169] T. W. Beech, S. Cornara, M. B. Mora, and G. Lecohier, "A study of three satellite constellation design algorithms," in *14th international symposium on space flight dynamics*, 1999.

[170] S. Cornara, T. W. Beech, M. Belló-Mora, and G. Janin, "Satellite constellation mission analysis and design," *Acta Astronautica*, 2001.

[171] D. Bhattacherjee and A. Singla, "Network topology design at 27,000 km/hour," in *ACM CoNEXT*, 2019.

[172] CesiumJS, `https://cesium.com/cesiumjs/`, 2020.

[173] S. Kassing, D. Bhattacherjee, A. Baptista Águas, J. Eirik Saethre, and A. Singla, *Hypatia source code*, `https://github.com/snkas/hypatia`, 2020.

[174] Bhattacherjee, Debopam and Singla, Ankit, *LEO satellite networks*, `https://leosatsim.github.io/`, 2020.

[175] S. Clark, *SpaceX's Starlink network surpasses 400-satellite mark after successful launch*, `https://spaceflightnow.com/2020/04/22/spacexs-starlink-network-surpasses-400-satellite-mark-after-successful-launch/`, 2020.

[176] T. Klenze, G. Giuliari, C. Pappas, A. Perrig, and D. Basin, "Networking in Heaven as on Earth," in *ACM HotNets*, 2018.

[177] G. Giuliari, T. Klenze, M. Legner, D. Basin, A. Perrig, and A. Singla, "Internet Backbones in Space," in *ACM SIGCOMM CCR*, 2020.

[178] SNS3, `https://www.sns3.org/content/home.php`, 2020.

[179] T. Henderson and R. Katz, "Network simulation for LEO satellite networks," in *18th International Communications Satellite Systems Conference and Exhibit*, AIAA, 2000.

[180] L. Wood, *Satellite constellation visualization (SaVi)*, `https://savi.sourceforge.io/`, 2017.

[181] Pedro Silva, *ns3-satellite*, `https://gitlab.inesctec.pt/pmms/ns3-satellite`, 2016.

[182] M. Handley, *Using ground relays with Starlink*, `https://www.youtube.com/watch?v=m05abdGSOxY`, 2019.

[183] E. Eccli, *Starlink Constellation (2019/11/17 - 2020/05/21)*, `https://www.youtube.com/watch?v=857UM4ErX9A`, 2020.

[184] Celestrak, *Orbit Visualization (e-tool per StarLink)*, `https://celestrak.com/cesium/orbit-viz.php?tle=/NORAD/elements/supplemental/starlink.txt&satcat=/pub/satcat.txt&orbits=0&pixelSize=3&samplesPerPeriod=90&referenceFrame=1`, 2020.

[185] NASA, *General Mission Analysis Tool*, `https://software.nasa.gov/software/GSC-17177-1`, 2020.

[186] ns-3 Network Simulator, `https://www.nsnam.org/`, 2020.

[187] A. Kauderer and K. Dismukes, *NASA HSF: Definition of Two-line Element Set Coordinate System*, `https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html`, 2011.

[188] North American Aerospace Defense Command, `https://www.norad.mil/`, 2020.

[189] T. Kelso, *CelesTrak: Current NORAD Two-Line Element Sets*, `https://www.celestrak.com/NORAD/elements/`, 2020.

[190] GeoRepository, *WGS72*, `https://georepository.com/crs_4985/WGS-72.html`, 2020.

[191] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the "Internet from space" with Hypatia," in *ACM IMC*, 2020.

[192] P. Goyal, A. Narayan, F. Cangialosi, D. Raghavan, S. Narayana, M. Alizadeh, and H. Balakrishnan, "Elasticity Detection: A Building Block for Delay-Sensitive Congestion Control," in *ACM ANRW*, 2018.

[193] M. Claypool, J. W. Chung, and F. Li, "BBR' an implementation of bottleneck bandwidth and round-trip time congestion control for ns-3," in *Proceedings of the 10th Workshop on ns-3*, ACM, 2018.

[194] V. Jain, V. Mittal, and M. P. Tahiliani, "Design and implementation of TCP BBR in ns-3," in *Proceedings of the 10th Workshop on ns-3*, ACM, 2018.

[195] I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla, "Why is the Internet so slow?!" In *International Conference on Passive and Active Network Measurement*, Springer, 2017.

[196] Datagraver, *World population distribution by latitude and longitude - 2020*, `https://datagraver.com/case/world-population-distribution-by-latitude-and-longitude-2020`, 2020.

[197] S. Kassing, D. Bhattacherjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the 'Internet from space' with Hypatia," in *ACM IMC*, 2020.

[198] D. Bhattacherjee, W. Aqeel, G. Laughlin, B. M. Maggs, and A. Singla, "A Bird's Eye View of the World's Fastest Networks," in *ACM IMC*, 2020.

[199] American Tower Global Wireless Solutions, `https://www.americantower.com/us/`, 2004.

[200] DARPA, *Novel Hollow-Core Optical Fiber to Enable High-Power Military Sensors*, `http://www.darpa.mil/news-events/2013-07-17`, 2013.

[201] J. Hansryd and J. Edstam, "Microwave capacity evolution," *Ericsson review*, vol. 1, pp. 22–27, 2011.

[202] T. Manning, *Microwave Radio Transmission Design Guide*. Artech House, 2009.

[203] G. Laughlin, A. Aguirre, and J. Grundfest, "Information transmission between financial markets in chicago and new york," *Financial Review*, 2014.

[204] F. C. Commission, *Universal Licensing System*, `http://wireless2.fcc.gov/UlsApp/UlsSearch/searchLicense.jsp`.

[205] NASA Jet Propulsion Laboratory, *U.S. Releases Enhanced Shuttle Land Elevation Data*, `https://www2.jpl.nasa.gov/srtm/`, 2015.

[206] USGS, *National Elevation Dataset (NED)*, https://lta.cr.usgs.gov/NED, 2018.

[207] M. R. Garey and D. S. Johnson, "The Rectilinear Steiner Tree Problem is NP-Complete," *SIAM Journal on Applied Mathematics*, 1977.

[208] Center for International Earth Science Information Network (CIESIN), Columbia University; United Nations Food and Agriculture Programme (FAO); and Centro Internacional de Agricultura Tropical (CIAT), *Gridded Population of the World: Future Estimates (GPWFE)*, http://sedac.ciesin.columbia.edu/gpw, Accessed: 2014-01-12, 2005.

[209] Federal Communications Commission, *Antenna Structure Registration Database*, http://goo.gl/3OIFDT, 2018.

[210] I. Gurobi Optimization, *Gurobi optimizer reference manual*, 2016.

[211] Microsoft Azure, *Content Delivery Network pricing*, https://azure.microsoft.com/en-us/pricing/details/cdn/, 2021.

[212] Starlink Services, *PETITION OF STARLINK SERVICES, LLC FOR DESIGNATION AS AN ELIGIBLE TELECOMMUNICATIONS CARRIER*, https://ecfsapi.fcc.gov/file/1020316268311/Starlink%20Services%20LLC%20Application%20for%20ETC%20Designation.pdf, 2021.

[213] M. Sheetz, *SpaceX prices Starlink satellite internet service at $99 per month, according to e-mail*, https://www.cnbc.com/2020/10/27/spacex-starlink-service-priced-at-99-a-month-public-beta-test-begins.html, [Online; accessed 11-March-2021], 2020.

[214] J. Engebretson, *Broadband Data Usage Report: Internet-only Homes Use Almost Twice as Much Data as Bundled Homes*, https://www.telecompetitor.com/broadband-data-usage-report-internet-only-homes-use-almost-twice-as-much-data-as-bundled-homes/, [Online; accessed 11-March-2021], 2019.

[215] X, the moonshot factory, *Taara – Expanding global access to fast, affordable internet with beams of light*, https://x.company/projects/taara/, 2018.

[216] Facebook connectivity, *Magma*, https://connectivity.fb.com/magma/, 2021.

[217] Facebook connectivity, *Rural Access*, https://connectivity.fb.com/rural-access/, 2021.

[218] Facebook connectivity, *Terragraph*, https://connectivity.fb.com/terragraph/, 2021.

[219] M. J. Lum, D. C. Friedman, H. H. King, R. Donlin, G. Sankaranarayanan, T. J. Broderick, M. N. Sinanan, J. Rosen, and B. Hannaford, "Teleoperation of a surgical robot via airborne wireless radio and transatlantic internet links," in *Field and service robotics*, Springer, 2008.

[220] Y. Chen, Z. Liu, S. R. Sandoghchi, G. T. Jasion, T. D. Bradley, E. N. Fokoua, J. R. Hayes, N. V. Wheeler, D. R. Gray, B. J. Mangan, *et al.*, "Multi-kilometer long, longitudinally uniform hollow core photonic bandgap fibers for broadband low latency data transmission," *Journal of Lightwave Technology*, 2016.

[221] A. Perrig, P. Szalachowski, R. M. Reischuk, and L. Chuat, *SCION: a secure Internet architecture*. Springer, 2017.

[222] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017.

[223] A. Narayan, F. Cangialosi, P. Goyal, S. Narayana, M. Alizadeh, and H. Balakrishnan, "The case for moving congestion control out of the datapath," in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, 2017, pp. 101–107.

[224] A. Narayan, F. Cangialosi, D. Raghavan, P. Goyal, S. Narayana, R. Mittal, M. Alizadeh, and H. Balakrishnan, "Restructuring endpoint congestion control," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, pp. 30–43.

[225] P. S. Schmidt, T. Enghardt, R. Khalili, and A. Feldmann, "Socket intents: Leveraging application awareness for multi-access connectivity," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, 2013, pp. 295–300.

[226] D. Katabi, M. Handley, and C. Rohrs, "Congestion control for high bandwidth-delay product networks," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, 2002, pp. 89–102.

[227] N. Dukkipati, *Rate Control Protocol (RCP): Congestion control to make flows complete quickly*. Citeseer, 2008.

[228] B. Trammell, J.-P. Smith, and A. Perrig, "Adding path awareness to the internet architecture," *IEEE Internet Computing*, vol. 22, no. 2, pp. 96–102, 2018.

[229] S. Dawkins, "Path aware networking: A bestiary of roads not taken," *IETF Internet Draft*, 2018.

[230] Microsoft Azure, *New Azure Orbital, ground station as a service, now in preview*, https://azure.microsoft.com/fr-fr/updates/new-azure-orbital-ground-station-as-a-service-now-in-preview/, 2020.

[231] aws, *AWS Ground Station*, https://aws.amazon.com/ground-station/, 2020.

[232] G. Seeber, *Satellite geodesy: foundations, methods, and applications*. Walter de gruyter, 2008.

[233] M. Grüner, M. Licciardello, and A. Singla, "Reconstructing proprietary video streaming algorithms," in *USENIX ATC*, 2020.

[234] R. Singh, A. Dunna, and P. Gill, "Characterizing the deployment and performance of multi-cdns," in *Proceedings of the Internet Measurement Conference 2018*, 2018.

[235] U. Goel, M. P. Wittie, and M. Steiner, "Faster web through client-assisted CDN server selection," in *IEEE ICCCN*, 2015.

[236] R. Farahbakhsh, A. Cuevas, A. M. Ortiz, X. Han, and N. Crespi, "How far is Facebook from me? Facebook network infrastructure analysis," in *IEEE Communications Magazine*, 2015.

[237] R. Fanou, G. Tyson, E. L. Fernandes, P. Francois, F. Valera, and A. Sathiaseelan, "Exploring and analysing the African Web ecosystem," in *ACM Transactions on the Web (TWEB)*, 2018.

[238] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," in *Computer Networks*, Elsevier, 2018.

[239] StackPath, *What are Micro Data Centers?* https://www.stackpath.com/edge-academy/micro-data-centers/, 2020.

[240] D. Bhattacherjee, S. Kassing, M. Licciardello, and A. Singla, "In-orbit Computing: An Outlandish thought Experiment?" In *ACM HotNets*, 2020.

[241] S. A. Jyothi, "Solar Superstorms: Planning for an Internet Apocalypse," in *ACM SIGCOMM*, 2021.

[242] ESA, *Monitoring space weather*, https://www.esa.int/Safety_Security/Monitoring_space_weather2, 2021.