Exploring the "Internet from space" with Hypatia

Simon Kassing*, Debopam Bhattacherjee*, André Baptista Águas, Jens Eirik Saethre, Ankit Singla
ETH Zürich

ABSTRACT

SpaceX, Amazon, and others plan to put thousands of satellites in low Earth orbit to provide global low-latency broadband Internet. SpaceX's plans have matured quickly, such that their under-deployment satellite constellation is already the largest in history, and may start offering service in 2020.

The proposed constellations hold great promise, but also present new challenges for networking. To enable research in this exciting space, we present Hypatia, a framework for simulating and visualizing the network behavior of these constellations by incorporating their unique characteristics, such as high-velocity orbital motion.

Using publicly available design details for the upcoming networks to drive our simulator, we characterize the expected behavior of these networks, including latency and link utilization fluctuations over time, and the implications of these variations for congestion control and routing.

CCS CONCEPTS

• **Networks** → **Network simulations**; **Network performance analysis**; *Network dynamics*; *Topology analysis and generation*; *Packet-switching networks*.

KEYWORDS

Low Earth orbit satellite, LEO, Internet broadband constellation, LEO network simulation, LEO network visualization

ACM Reference Format:
Simon Kassing*, Debopam Bhattacherjee*, André Baptista Águas, Jens Eirik Saethre, Ankit Singla. 2020. Exploring the "Internet from space" with Hypatia. In *ACM Internet Measurement Conference (IMC '20), October 27–29, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3419394.3423635

1 INTRODUCTION

The Internet is potentially taking "one giant leap" into space, with plans afoot for large satellite constellations to blanket the globe with low-latency broadband Internet. Numerous competitors have disclosed efforts along these lines, including SpaceX [70], Amazon [8], and Telesat [74]. With 400+ satellites already in orbit, and an increasing launch cadence, SpaceX's Starlink constellation is promising limited availability of its Internet service already in 2020 [23]. It is thus unsurprising that these ambitious plans for an "Internet from space" have captured the public imagination [10, 28, 55, 62, 75].

While the use of satellites for Internet connectivity is as old as the Internet itself[1], the under-construction constellations differ fundamentally from past efforts. The distinctions are rooted in the recent improvements in enabling technologies, as well as the goals, but manifest themselves deeply in the design. Unlike existing satellite networks [35–37], the new ones are targeting not only traditional niches such as shipping, satellite telephony, and limited connectivity for rural areas, but also mass market broadband that not only addresses these global coverage issues, but also competes with current terrestrial networks in many markets.

The first design manifestation of this goal is scale: to provision enough access bandwidth for their larger target user population, the new systems need many more satellites than past ones. Starlink, with its hundreds of satellites, is already the largest ever satellite fleet in space history, but eventually, the largest planned constellations will each comprise thousands of satellites [8, 70]. This has only become possible due to favorable trends in space technology, primarily, satellite miniaturization, and reduced launch costs.

The goal of competing outside traditional niches has another important design consequence: operation in low Earth orbit (LEO), at most 2,000 km above Earth's surface. This is essential for latencies to be comparable to terrestrial networks instead of the hundreds of milliseconds that geostationary orbits (GEO) incur. LEO operation, in turn, further reinforces the need for large scale: from GEO, each satellite is visible to a large terrestrial area, but bringing satellites closer to the Earth necessarily reduces each satellite's coverage.

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*. In such constellations, each satellite orbits the Earth every ~100 minutes, traveling at ~27,000 kmph. 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.

At HotNets 2018, three position papers [5, 29, 44] highlighted some of the networking challenges that could potentially arise in LEO networks, *e.g.,* in end-end congestion control [5] and intra-constellation routing [29]. However, progress in precisely fleshing out these challenges 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

*A coin toss decided the order of the first two authors.

IMC '20, October 27–29, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8138-3/20/10…$15.00
https://doi.org/10.1145/3419394.3423635

[1] An early satellite network, SATNET, formed an initial segment of the Internet, and in fact, provided *the* key motivation for Cerf-Kahn's work on the foundational Internet Protocol: interconnecting networks as different as ARPANET, PRNET, and SATNET to each other [1].

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**[2], an analysis framework with simulation 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 [13], that serve to provide intuition about such networks.

We use Hypatia to analyze the three largest proposed LEO networks: Starlink, Kuiper, and Telesat. Our analysis uses the regulatory information these companies have filed with governing bodies like the Federal Communications Commission (FCC) in the United States, and the International Telecommunication Union (ITU). These filings [47–49, 68, 69, 72] disclose the orbital parameters that describe the structure of the planned constellations. Our simulations of these networks reveal the impact of LEO dynamics on varying path RTTs and packet reordering, as well as fluctuations in available bandwidth along end-end paths. We discuss the implications of these observations for congestion control and routing.

In summary, 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 [40], together with our visualizations [7].

2 BACKGROUND & RELATED WORK

LEO mega-constellations being a new problem area, we include relevant background to aid our readers.

2.1 What makes an LEO satellite network?

A large LEO constellation may comprise hundreds to thousands of satellites. These satellites are organized into a number of orbits. An orbit is described by: (a) its inclination, i, the angle its plane makes with the Equator traveling northward; and (b) its height above sea-level, h < 2,000 km. Satellites within one orbit are uniformly spaced out. A set of orbits with the same i and h, and crossing the Equator at uniform spacing from each other, is called an orbital shell. (Typically orbits in a shell vary their elevation around h to avoid collisions — these minor differences are largely immaterial to networking.) Large constellations may deploy one or more such shells. This description only captures a sub-class of LEO constellations, but all recently proposed constellations fit this sub-class.

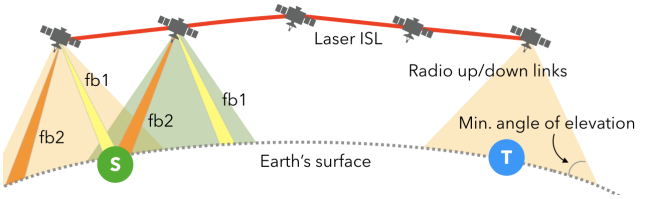


Fig. 1: *Each satellite covers a cone defined by the minimum angle of elevation, l. A satellite uses steerable beams of different frequency bands (e.g., fb1, fb2) to connect to different GSes.*

Each satellite uses radio up/down links to communicate with ground stations (GSes), as shown in Fig. 1. A satellite can only connect to GSes from which it can be seen at sufficiently high elevation in the sky, as defined by the minimum angle of elevation, l. A satellite directly overhead a GS is at elevation 90°, while one at the horizon is at 0°. If the minimum elevation $l = 40°$, only GSes that see a satellite at elevation 40° or higher in the sky can communicate with it. Thus, smaller l values allow GSes to talk to satellites closer to the horizon, while larger l values are more restrictive. However smaller l values also have a downside: connections from lower elevation experience reduced antenna gain and signal quality due to beam contour widening and increased attenuation.

According to Kuiper's FCC filings [46], each satellite will have multiple antennas, with each antenna supporting multiple steerable beams; 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 [46].

Satellites also connect to each other, using laser inter-satellite links (ISLs). An end-end path between two GSes comprises a radio up-link from the source GS to the ingress satellite, followed by zero or more laser ISLs, and then the egress radio down-link to the destination GS.

2.2 The largest proposed constellations

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

SpaceX Starlink: Table. 1 details the first phase of Starlink, with 4,409 satellites planned across 5 orbital shells [66–69]. SpaceX is currently deploying S1, with 1,584 satellites (72 orbits, each with 22 satellites), h = 550 km, and i = 53°. The minimum elevation, l = 25°. S1 will cover most of the world's population, but will not extend service to less populated regions at high latitudes. This

[2]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.

	shell	*h (km)*	*Orbits*	*Sats/orbit*	*i*
Starlink	S1	550	72	22	53°
	S2	1,110	32	50	53.8°
	S3	1,130	8	50	74°
	S4	1,275	5	75	81°
	S5	1,325	6	75	70°
Kuiper	K1	630	34	34	51.9°
	K2	610	36	36	42°
	K3	590	28	28	33°
Telesat	T1	1,015	27	13	98.98°
	T2	1,325	40	33	50.88°

Table 1: *Shell configurations for Starlink's first phase of deployment, Kuiper, and Telesat. We shall frequently refer to the first shell for each constellation, S1, K1, and T1, in the text.*

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 [34].

Amazon Kuiper: Kuiper plans three shells, with a total of 3,236 satellites at slightly different operating heights [47–49]. Kuiper entirely eschews connectivity near the poles, with all its shells having inclinations under 52°. The FCC filings mention a few possible values of l: "20(min)/30/35/45" [47].

Telesat: Telesat plans two shells with a total of 1,671 satellites [73], 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 $l = 10°$, but the feasibility of this is unclear — unlike Starlink and Kuiper, whose filings detail how to address beam contour and antenna gain changes for $l \geq 25°$, Telesat's filings thus far omit such information.

2.3 Unique dynamics of LEO networks

A satellite's height, h, determines its orbital velocity, and thus, orbital period [56]. At $h = 550$ km, the orbital velocity is more than 27,000 km/hr, and satellites complete an orbit around the Earth in ~100 minutes [6]. 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 coming closer at higher latitudes. This results in a continuous change in their relative positions and hence the ISL lengths and latencies.

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.

2.4 Large LEO networks need new research

Commercial satellite networks already provide varied network services. HughesNet [35] and Viasat [76] 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 [36, 37] 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, 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. One of the upcoming constellations, Starlink, already has more than 400 satellites operational, and expects a public launch of their service as soon as 2020 [16, 23]. 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 [2, 4, 14, 15, 18, 24, 43, 50, 53, 71, 78–80, 82], several position papers [5, 29, 44] have highlighted the *new* opportunities and challenges of mega-constellations, *e.g.,* in intra-constellation routing [29] and inter-domain routing [44], and end-end congestion control [5]. Followup work has since laid out novel proposals for topology design [6] and Internet inter-domain routing [26] in this context.

2.5 We are missing the right analysis tools

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 [65] models GEO satellite communication channels, but does not support LEO satellites or inter-satellite connectivity. Another simulation effort [33] 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 [81]. A satellite mobility model is available for ns-3 [61], 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 [6] evaluated topologies only in terms of path hop-counts and distances, not packet simulations. Likewise, work on inter-domain routing [26] only modeled the network control messages and path distances. Another effort [21] 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 [12, 22, 30, 32], most of these do not focus on networking concepts such as the evolution of paths, utilization, and congestion. The closest related work [29, 30] does not simulate packet-level behavior, and does not provide source code for its path-granularity computations or visualizations. NASA's *GMAT* [57] can be used to visualize trajectories of objects in space; *SaVi* [81] 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.

3 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 [60]. 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 [13] to render views of the trajectories, GS-perspective on overhead satellites, end-end routes, evolving link utilization, and available bandwidth on routes.

3.1 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 [41]. For existing satellites and orbital debris, NORAD [59] regularly publishes TLEs [42]. 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 [54] from the FCC or ITU filings made by the operators. Table. 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 [47, 68], 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 [25]. 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: Hypatia implements what we believe to be a reasonable pattern of connectivity between satellites, but modifying this to support arbitrary alternative interconnects is easy. Our default implementation draws on past literature in satellite networking, and information in the regulatory filings of the newly proposed constellations.

The proposed mega-constellations hint at building 4 ISLs per satellite. Starlink's filings [68], for example, mention 4 silicon-carbide components, and as recent work [6, 29] notes, these are typically components for ISLs. We thus use 4 ISLs per satellite in our default implementation.

Further, a large body of work in satellite networking indicates a typical connectivity pattern for a satellite with 4 ISLs: two links to the immediate neighbors in the orbit, and two links to satellites in adjacent orbits, forming a mesh-like network [20, 21, 29, 30, 45, 51, 52, 63, 64, 77]. Recent work [6] has called the resulting mesh-like connectivity "+Grid". We use +Grid as the default ISL interconnect.

Hypatia also supports constellations eschewing ISLs entirely [31]; experiments demonstrating this are included in Appendix A. Further, alternative ISL interconnects can be trivially supported in Hypatia, *if* they do not involve dynamic ISLs, *i.e.,* with satellites connecting to *different* satellites over time. This is a realistic assumption in most cases: ISL setup times can be tens of seconds, so reconfiguring ISLs dynamically is avoided [6]. Nevertheless, if such dynamic connectivity is desired, it will require modifying Hypatia.

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 [38]. 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 [47, 68] 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 §5.3.

For every time interval, we use a *networkx* [58] 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.

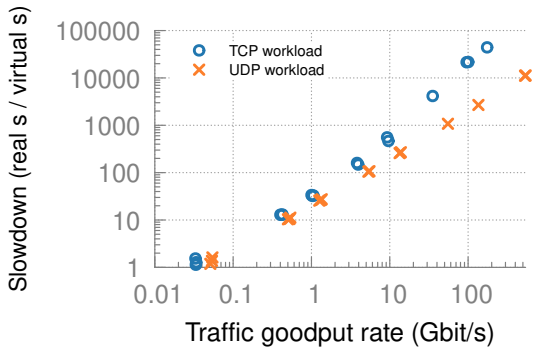


Fig. 2: ***Scalability.*** *Running experiments resulting in a 9.2 Gbit/s network-wide goodput with TCP for 1 second takes ~555 seconds. UDP simulations are faster, with 13.8 Gbit/s goodput in ~269 seconds.*

3.2 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 [61]. 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.

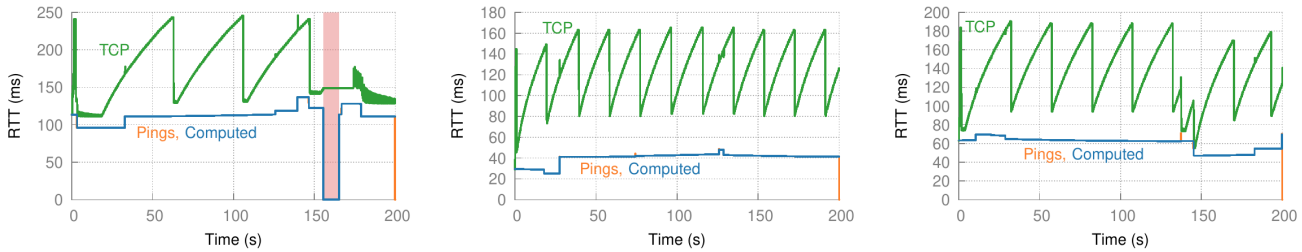
3.3 Post-processing and visualizations

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 paper, 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.

3.4 Simulator scalability

We briefly assess the scale at which Hypatia can run simulations in reasonable time. We use Kuiper's K1 shell as the LEO network, and the 100 most populous cities as the GSes. The traffic is a random permutation between the GSes. We test both TCP and UDP traffic. For TCP, each GS-pair sends each other a long running TCP flow, and we calculate network-wide goodput as the total rate of all acknowledged data, counting only packet payloads and excluding headers. For UDP, each GS-pair sends each other constant-rate,



(a) Rio de Janeiro to St. Petersburg (b) Manila to Dalian (c) Istanbul to Nairobi

Fig. 3: *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.*

paced UDP traffic at the line rate, and goodput is calculated as the total rate of network-wide payload arrivals. For both types of traffic, to control the traffic rate of the simulation, we vary the line rate of each network link, all assumed to be uniform; we test line rates of 1, 10, 25, 100 and 250 Mbit/s, and 1 and 10 Gbit/s. We run the simulations on a single core on a 2.26 Ghz Intel Xeon L5520.

We quantify 'slowdown': if Hypatia takes b real seconds to simulate a virtual seconds, slowdown is $\frac{b}{a}$. The results, shown in Fig. 2, allow an experimenter to directly answer the question: if I want to run the system at C Gbit/s goodput and obtain data for a virtual seconds, how long will it take? If slowdown for x-axis = C is y, then the answer is $y \cdot a$ seconds. The results show that, *e.g.*, to simulate ~10 Gbps network goodput for 10 virtual seconds, Hypatia would need ~33 minutes for UDP traffic and ~100 minutes for TCP. The goodput alone determines the slowdown $\frac{b}{a}$, implying a simple trade-off: given a fixed real time budget, b, one can either simulate at high goodput for a short virtual time, or simulate at lower goodput for a longer virtual time. For the same experiment setup in terms of a constellation and traffic matrix, setting the line-rate of links allows control over goodput (up to a point; as utilization increases, goodput plateaus even with increasing load).

The simulation is bottlenecked at per-packet event processing. As satellite network paths often comprise a large number of hops, each end-end packet delivery results in many more events than, *e.g.*, for a data center network simulation. The satellite network's scale is also not a significant factor, at least for tens of thousands of satellites; the simulation setup costs, which increase with network scale, are only incurred at the start, while the packet events dominate the real time incurred for simulation.

We find that opportunities for speeding simulations up may be limited. Beyond ns-3's per-packet processing time, Hypatia only adds a minor overhead for calculating the per-packet delay through the mobility model. Forwarding tables are pre-computed, and MAC tables are pre-filled to prevent ARP activity. We are exploring to what extent ns-3's distributed mode affords speedup, but for a variety of use cases, Hypatia's current simulation pace will suffice, as our later analysis shows.

4 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.

4.1 RTT fluctuations

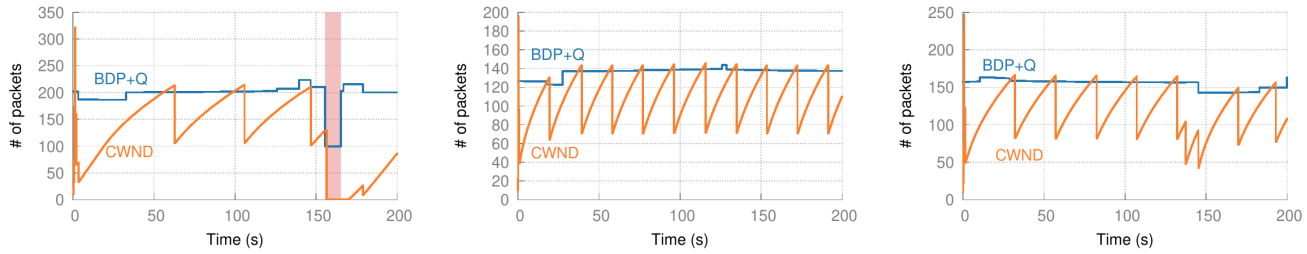
We examine how the end-end RTTs vary over time. These experiments use the Kuiper K1 shell. We run the analysis for 200 seconds, as for Kuiper-scale networks this is sufficient to show nearly the full range of variations.

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 already appeared in recent work [5, 29]; 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. 3 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. 3(a) at t = 32.9 s the path changes, which causes the RTT to rise from 96 ms to 111 ms. Occasionally, like in Fig. 3(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*.

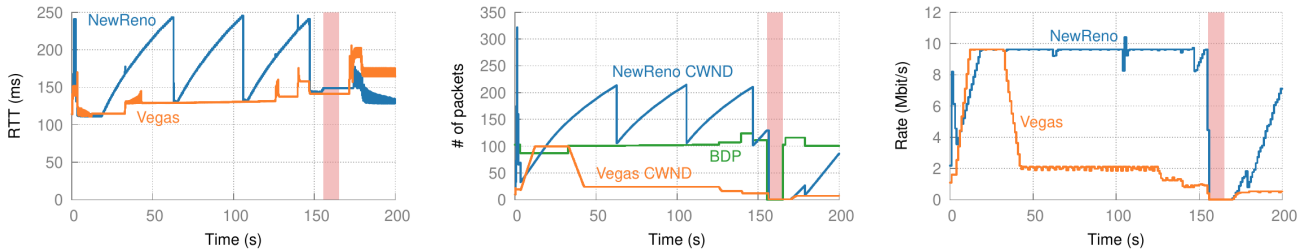
The path from Rio de Janeiro to St. Petersburg sees a disruption around 150 seconds into the simulation, shown as the shaded region in all related plots. We found that for this period, St. Petersburg does not have any visible Kuiper satellites at sufficiently high angle of elevation, which, obviously, 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.



(a) Rio de Janeiro to St. Petersburg (b) Manila to Dalian (c) Istanbul to Nairobi

Fig. 4: *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.*



(a) NewReno causes high RTTs. (b) Vegas decreases CWND on RTT increase. (c) Vegas' throughput collapses.

Fig. 5: *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.*

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.

4.2 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. 3 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. 4 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 (assuming 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. 4(a), but additionally, we can see drops in the congestion window for the other connections too, *e.g.,* in Fig. 4(c), around 140 s, TCP drops the congestion window because of packet reordering. At this time, as the path is shortened by ~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. 3(a) and 5(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 [3, 11, 27]. 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 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(a), the congestion

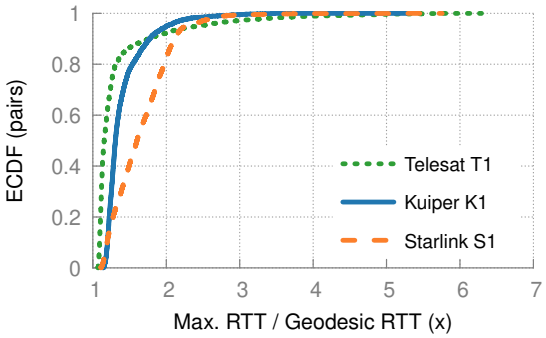


Fig. 6: *RTT vs. geodesic.* *Even the maximum RTT (over time) over LEO networks is close to the geodesic RTT for most connections, especially for Telesat and Kuiper. However, some connections see several times higher maximum RTTs.*

window in Fig. 5(b), and the achieved throughput averaged over 100 ms intervals in Fig. 5(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. 3(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(b)), and achieves very poor throughput (Fig. 5(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 [11] 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 [17, 39], 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 [26]. Solutions like splitting the transport connection are also becoming difficult to support with transport 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 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 1 first. Telesat's deployment plan is more complex [72]; 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.

5.1 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.

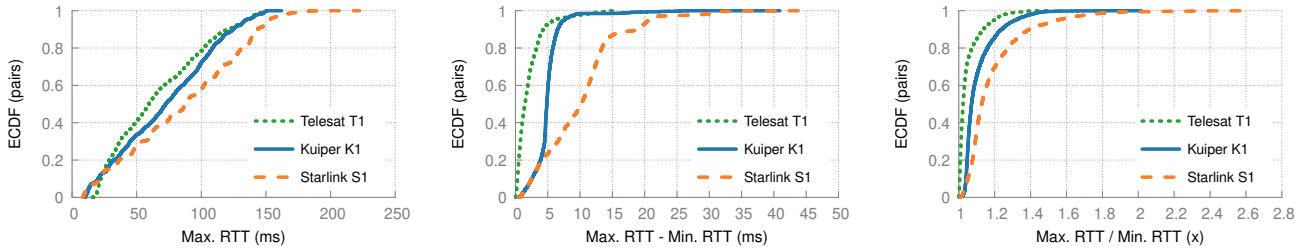
For each connection, we compute the ratio of its maximum RTT over time to the geodesic RTT between its end-points. Fig. 6 shows this ratio as a CDF across connections. For all three constellations, more than 80% of connections see a maximum RTT less than 2× the geodesic. Given that terrestrial fiber paths are often longwinded, and the speed of light in fiber is roughly $2c/3$ [9], 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 [5, 6, 29, 44]. 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.

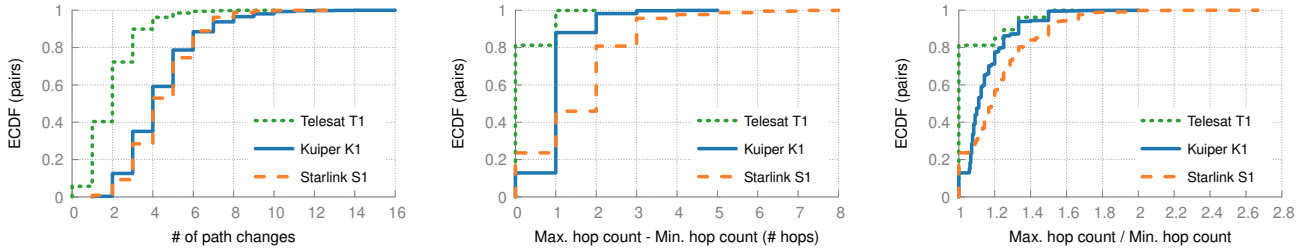
The Starlink-Kuiper differences are not due to the angle of elevation, which is similar, but the orbital structure. Both constellations use a minimum angle of elevation that is much higher than Telesat's. This means that typically, GSes can see fewer satellites. This restricts the GS-satellite connectivity, 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. 7 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



(a) CDF across pairs: max. RTT over time. (b) CDF across pairs: max RTT - min RTT. (c) CDF across pairs: max RTT / min RTT.

Fig. 7: *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.*



(a) Distribution of path changes. (b) Distribution of path hop-count differences. (c) Distribution of max hop-count / min hop-count.

Fig. 8: *Path structure changes.* *Telesat's paths change less than Kuiper's and Starlink's.*

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.

5.2 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. 8(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.

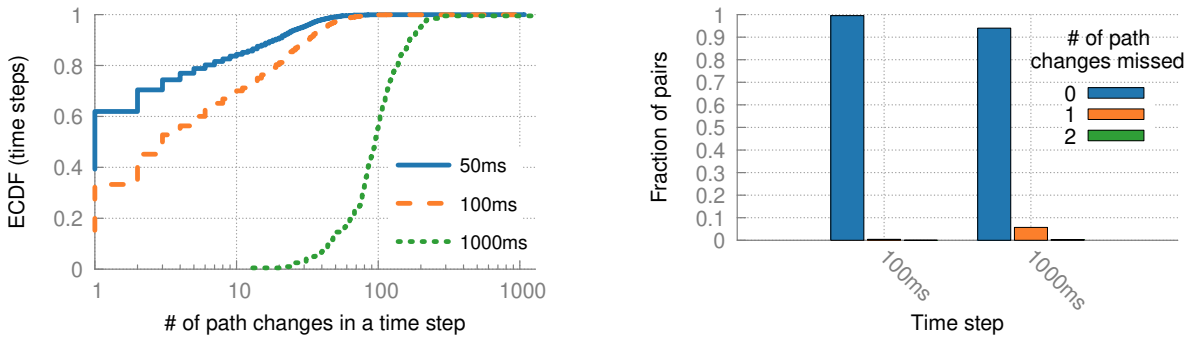
Fig. 8(b) shows how these different paths differ in terms of their hop count. For Telesat, paths do not typically change in terms of hop count. This is explained by Telesat being sparser: there are simply fewer options for end-end paths, and with farther-apart satellites, one hop of change would already 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. 8(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.

5.3 Granularity of time-step updates

Hypatia converts a continuous process of satellite movement and the resulting path changes into a discrete one. While latencies along paths are continuously updated, the forwarding state is only recomputed at fixed time-steps. We thus test how this affects our observations on path changes.



(a) (b)

Fig. 9: *Time granularity for forwarding state updates.* *(a) As expected, compared to* 50 *ms time-steps,* 100 *ms time-steps see roughly* 2× *the path changes, while* 1000 *ms see roughly* 20× *path changes. (b) The* 1000 *ms time-step misses a substantial number of path changes for some pairs, while for* 100 *ms, missed changes are negligible.*

We compute the network's forwarding state at different time-steps of 50, 100, and 1000 ms. For each configuration, we calculate: (a) how many path changes per second occur in a time-step; (b) how many path changes are missed at coarser time-steps compared to 50 ms.

We only include results for Kuiper K1, but the conclusions hold broadly. Fig. 9(a) shows the distribution of the number of path changes network-wide across time-steps. Ideally, the 100 ms time-step would have 2× the number of changes compared to the 50 ms one, and 1000 ms would have 20×. This is almost always the case for 100 ms, but for 1000 ms a significant fraction of path changes are simply missed because multiple changes happened entirely within that interval. Fig. 9(b) shows the distribution of these missed changes for both the 100 ms and 1000 ms time-steps compared to the 50 ms one. The 100 ms time-step misses for a negligible fraction (0.4%) of pairs one or more path changes, while 1000 ms misses for 6% of pairs one or more path changes.

Note that finer granularity of time-steps requires expensive shortest-path computations for the entire large network. Based on our results, 100 ms is a good compromise. Further, given that path changes occur over tens of seconds, the 100 ms time-step can only be inaccurate and not provide the actual shortest paths for at most 1% of the time.

5.4 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 §4, *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. 10 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. 10 shows the unused bandwidth, measured at a 1 s granularity, for the same connection we examined in §4, 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

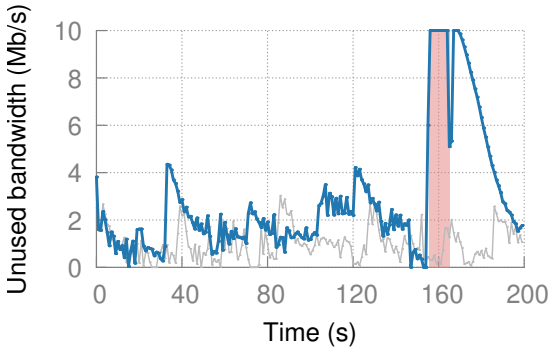


Fig. 10: *Unused bandwidth.* *For the same connection from Rio de Janeiro to St. Petersburg as in Fig. 5(c), when tested with cross-traffic, transport is often unable to use the available bandwidth. This is with a fixed set of long-running TCP flows, and a fixed routing policy. The gray line is if the satellite network is frozen at $t = 0$, effectively being a static network.*

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. 8. 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.

6 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 discuss some of the visualizations Hypatia provides. While these are best appreciated online in video and interactive Javascript [7], 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. 11 shows snapshots of the first shells of Starlink, Kuiper, and Telesat — S1, K1, and T1 in Table 1. A live 3D version of this figure is available online [7]; it is interactive and allows one to change the camera perspective in order to better see the spatial variations. Telesat covers the 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 [19], 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.

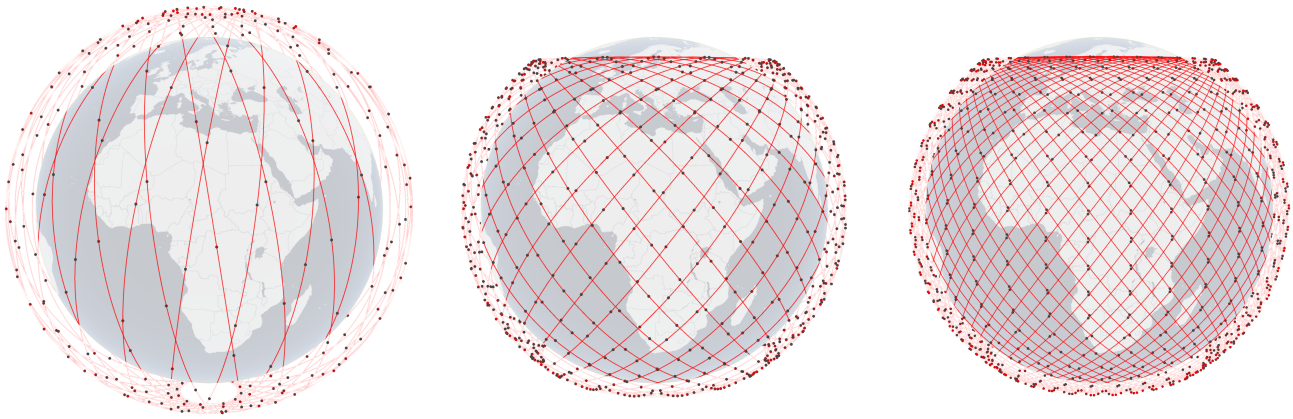
Besides coverage, inclination also has other implications for connectivity: Telesat's almost north-south orbits may offer more direct paths for routes like between Europe and Africa, while the other constellations will do so for east-west routes like between North America and Europe.

We include satellite trajectory visualizations primarily for completeness: there are a variety of other beautiful visualizations of similar nature online [6, 12, 22, 30, 32]. 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 [7] provides video of the ground observer's perspective.

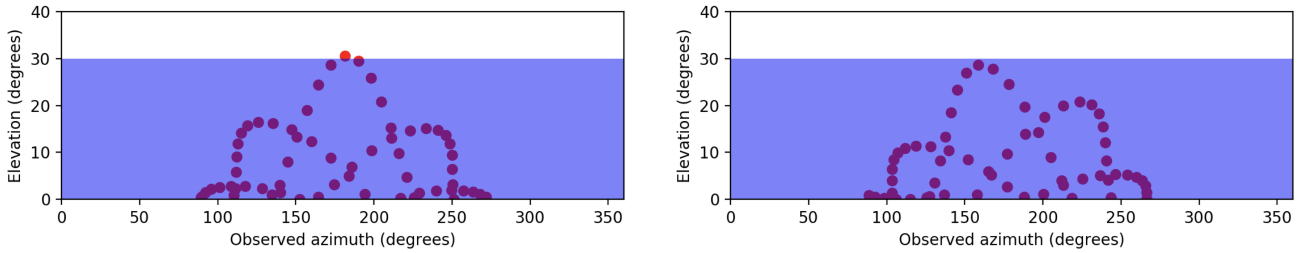
Fig. 12 shows two snapshots of Kuiper's K1 seen from St. Petersburg. The azimuth along the x-axis is the panoramic view of the sky (0° is due North, 90° is due East). The y-axis is the angle of elevation, 0° for the horizon, and 90° for directly overhead. Satellites in the shaded region are above the 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. 12(a), while at other times, it looses connectivity, as in Fig. 12(b). This explains the results for Rio de Janeiro to St. Petersburg between 155-165 s in Fig. 3(a), Fig. 4(a), and Fig. 5.

End-end paths: In §4.1, 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. 13 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 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.



(a) Telesat T1 (b) Kuiper K1 (c) Starlink S1

Fig. 11: *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.*



(a) Connectivity is possible at this time. (b) No satellites are reachable.

Fig. 12: *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.*

Link utilization: In §5.4, 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. 14, for the same experiment across Kuiper described in §5.4. The thicker / warmer-colored ISLs are more congested.

We can also visualize network-wide bottlenecks as shown in Fig. 15. 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 that there will be substantial value in using non-shortest path and multi-path routing across such busy regions.

7 LIMITATIONS & FUTURE WORK

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

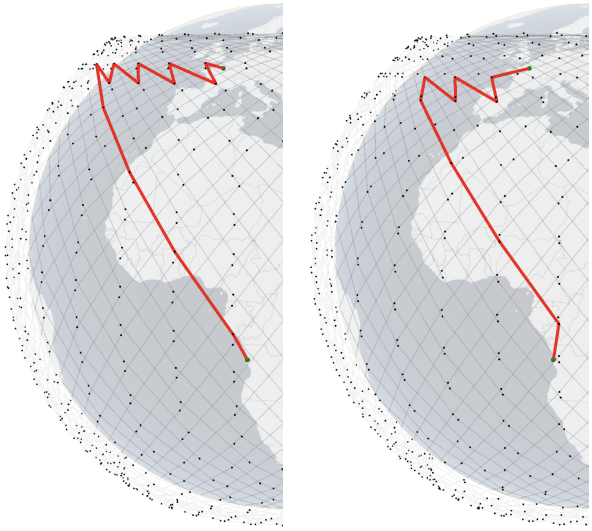


Fig. 13: *Shortest path changes over time.* *On Starlink's S1, the Paris-Luanda RTT varies between* 117 *ms (left) and* 85 *ms. The zig-zags stem from the nature of ISLs in the topology — satellites which seem visually close to each other are not necessarily connected directly.*

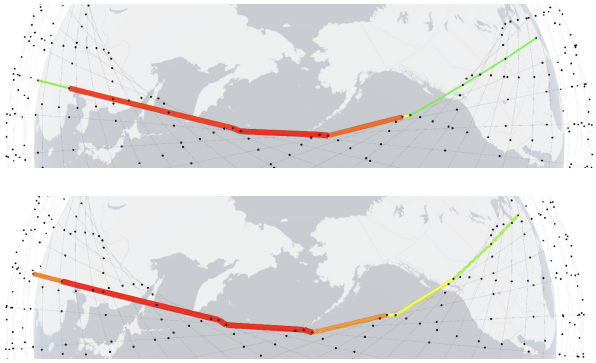


Fig. 14: *Congestion shifts over time.* *An example path, Chicago-Zhengzhou, shows how the link utilizations change over time, even with the input traffic being static. The top and bottom views are at* 10 *s and* 150 *s respectively.*

Hypatia. However, we believe it provides a useful starting point for such work.

Importantly, all the takeaways we have highlighted throughout the paper 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.

8 CONCLUSION

We present Hypatia, a framework for simulating and visualizing large LEO networks. We demonstrate Hypatia's utility in understanding the behavior of such networks, especially the temporal variations in the structure of paths and their latencies. We draw out some of the implications of this LEO network dynamism for congestion control, and routing and traffic engineering. Our work not only adds quantitative support to recent position papers drawing out the challenges of LEO networking, but also provides a sorely missing infrastructure for making progress on these challenges.

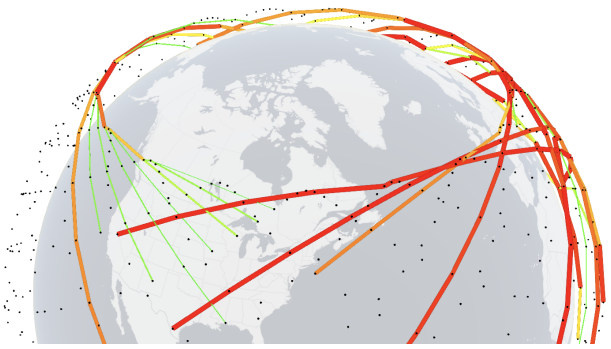


Fig. 15: *Constellation-wide utilization.* *On Kuiper, the trans-atlantic paths are highly congested for our tested traffic matrix. The red / thick ISLs are heavily utilized, while green / thin ISLs have minimal traffic. ISLs with no traffic are excluded.*

ACKNOWLEDGMENTS

We are grateful to our shepherd Eric Wustrow and the anonymous reviewers for their helpful feedback.

REFERENCES

[1] Janet Abbate. 2004. Robert ("Bob") Elliot Kahn. https://amturing.acm.org/award_winners/kahn_4598637.cfm.

[2] Riza Akturan and Wolfhard J Vogel. 1997. Path diversity for LEO satellite-PCS in the urban environment. In *IEEE Transactions on Antennas and Propagation*.

[3] Venkat Arun and Hari Balakrishnan. 2018. Copa: Practical Delay-Based Congestion Control for the Internet. In *USENIX NSDI*.

[4] Jason H Bau. 2002. *Topologies for satellite constellations in a cross-linked space backbone network*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[5] Debopam Bhattacherjee, Waqar Aqeel, Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. 2018. Gearing up for the 21st century space race. In *ACM HotNets*.

[6] Debopam Bhattacherjee and Ankit Singla. 2019. Network topology design at 27,000 km/hour. In *ACM CoNEXT*.

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

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

[9] Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. 2017. Why is the Internet so slow?!. In *International Conference on Passive and Active Network Measurement*. Springer.

[10] David Canellis. 2020. 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/.

[11] Neal Cardwell, Yuchung Cheng, C Stephen Gunn, Soheil Hassas Yeganeh, and Van Jacobson. 2017. BBR: congestion-based congestion control. In *Communications of the ACM*.

[12] Celestrak. 2020. 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.

[13] CesiumJS. 2020. https://cesium.com/cesiumjs/.

[14] Vincent WS Chan. 1999. Optical space communications: a key building block for wide area space networks. In *IEEE LEOS Annual Meeting Conference Proceedings*.

[15] Vincent WS Chan. 2000. Optical space communications. In *IEEE Journal of Selected Topics in Quantum Electronics*.

[16] Stephen Clark. 2020. 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/.

[17] Mark Claypool, Jae Won Chung, and Feng Li. 2018. 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.

[18] Gary Comparetto and Neal Hulkower. 1994. Global mobile satellite communications-A review of three contenders. In *15th International Communications Satellite Systems Conference and Exhibit*. AIAA.

[19] Datagraver. 2020. WORLD POPULATION DISTRIBUTION BY LATITUDE AND LONGITUDE - 2020. https://datagraver.com/case/world-population-distribution-by-latitude-and-longitude-2020.

[20] Olivier L De Weck, Richard De Neufville, and Mathieu Chaize. 2004. Staged deployment of communications satellite constellations in low earth orbit. In *Journal of Aerospace Computing, Information, and Communication*.

[21] Inigo del Portillo, Bruce G Cameron, and Edward F Crawley. 2019. A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. In *Acta Astronautica*. Elsevier.

[22] Elias Eccli. 2020. Starlink Constellation (2019/11/17 - 2020/05/21). https://www.youtube.com/watch?v=857UM4ErX9A.

[23] Sandra Erwin. 2019. SpaceX plans to start offering Starlink broadband services in 2020. https://spacenews.com/spacex-plans-to-start-offering-starlink-broadband-services-in-2020/.

[24] John V Evans. 1998. Satellite systems for personal communications. In *Proceedings of the IEEE*.

[25] GeoRepository. 2020. WGS72. https://georepository.com/crs_4985/WGS-72.html.

[26] Giacomo Giuliari, Tobias Klenze, Markus Legner, David Basin, Adrian Perrig, and Ankit Singla. 2020. Internet Backbones in Space. In *ACM SIGCOMM CCR*.

[27] Prateesh Goyal, Akshay Narayan, Frank Cangialosi, Deepti Raghavan, Srinivas Narayana, Mohammad Alizadeh, and Hari Balakrishnan. 2018. Elasticity Detection: A Building Block for Delay-Sensitive Congestion Control. In *ACM ANRW*.

[28] Loren Grush. 2020. 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.

[29] Mark Handley. 2018. Delay is Not an Option: Low Latency Routing in Space. In *ACM HotNets*.

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

[31] Mark Handley. 2019. Using Ground Relays for Low-Latency Wide-Area Routing in Megaconstellations. In *ACM HotNets*.

[32] Mark Handley. 2019. Using ground relays with Starlink. https://www.youtube.com/watch?v=m05abdGSOxY.

[33] Thomas Henderson and Randy Katz. 2000. Network simulation for LEO satellite networks. In *18th International Communications Satellite Systems Conference and Exhibit*. AIAA.

[34] Mark Holmes. 2020. 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/.

[35] HughesNet. 2020. https://www.hughesnet.com/.

[36] Iridium Communications Inc. 2020. Iridium NEXT. https://web.archive.org/web/20200718205415/https://www.iridiumnext.com/.

[37] Iridium Communications Inc. 2020. Iridium Satellite Communications. https://www.iridium.com/.

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

[39] Vivek Jain, Viyom Mittal, and Mohit P Tahiliani. 2018. Design and implementation of TCP BBR in ns-3. In *Proceedings of the 10th Workshop on ns-3*. ACM.

[40] Simon Kassing, Debopam Bhattacherjee, André Baptista Águas, Jens Eirik Saethre, and Ankit Singla. 2020. Hypatia source code. https://github.com/snkas/hypatia.

[41] Amiko Kauderer and Kim Dismukes. 2011. NASA HSF: Definition of Two-line Element Set Coordinate System. https://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html.

[42] T.S. Kelso. 2020. CelesTrak: Current NORAD Two-Line Element Sets. https://www.celestrak.com/NORAD/elements/.

[43] John D Kiesling. 1990. Land mobile satellite systems. In *Proceedings of the IEEE*.

[44] Tobias Klenze, Giacomo Giuliari, Christos Pappas, Adrian Perrig, and David Basin. 2018. Networking in Heaven as on Earth. In *ACM HotNets*.

[45] Mark Krebs. 2016. Satellite Constellation. https://patents.google.com/patent/US20170005719A1/en.

[46] Kuiper Systems LLC. 2019. 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.

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

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

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

[50] Kenneth Chun Hei Kwok. 2001. *Cost optimization and routing for satellite network constellations*. Ph.D. Dissertation. Massachusetts Institute of Technology.

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

[52] Jiulong Ma, Xiaogang Qi, and Lifang Liu. 2017. An Effective Topology Design Based on LEO/GEO Satellite Networks. In *International Conference on Space Information Network*. Springer.

[53] Gérard Maral. 1994. The ways to personal communications via satellite. In *International journal of satellite communications*. Wiley Online Library.

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

[55] Florian Meyer. 2019. 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.

[56] NASA. 2008. Orbits and Kepler's Laws. https://solarsystem.nasa.gov/resources/310/orbits-and-keplers-laws/.

[57] NASA. 2020. General Mission Analysis Tool. https://software.nasa.gov/software/GSC-17177-1.

[58] NetworkX developers. 2020. NetworkX. https://networkx.github.io/.

[59] North American Aerospace Defense Command. 2020. https://www.norad.mil/.

[60] ns-3 Network Simulator. 2020. https://www.nsnam.org/.

[61] Pedro Silva. 2016. ns3-satellite. https://gitlab.inesctec.pt/pmms/ns3-satellite.

[62] Saheli Roy Choudhury. 2019. 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.

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

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

[65] SNS3. 2020. https://www.sns3.org/content/home.php.

[66] SpaceX. 2016. SPACEX NON-GEOSTATIONARY SATELLITE SYSTEM. https://fcc.report/IBFS/SAT-LOA-20161115-00118/1158350.pdf.

[67] SpaceX. 2019. SPACEX NON-GEOSTATIONARY SATELLITE SYSTEM. https://fcc.report/IBFS/SAT-MOD-20190830-00087/1877671.

[68] SpaceX FCC filing. 2017. SpaceX V-BAND NON-GEOSTATIONARY SATELLITE SYSTEM. https://licensing.fcc.gov/myibfs/download.do?attachment_key=1190019.
[69] SpaceX FCC update. 2018. SPACEX NON-GEOSTATIONARY SATELLITE SYSTEM. https://licensing.fcc.gov/myibfs/download.do?attachment_key=1569860.
[70] SpaceX Starlink. 2017. https://www.spacex.com/webcast.
[71] David E Sterling and John E Hatlelid. 1991. The IRIDIUM system-a revolutionary satellite communications system developed with innovative applications of technology. In *MILCOM 91-Conference record*. IEEE.
[72] Telesat. 2018. Telesat's responses - Federal Communications Commission. http://licensing.fcc.gov/myibfs/download.do?attachment_key=1205775.
[73] Telesat. 2020. APPLICATION FOR MODIFICATION OF MARKET ACCESS AUTHORIZATION. https://fcc.report/IBFS/SAT-MPL-20200526-00053/2378318.pdf.
[74] Telesat. 2020. Telesat: Global Satellite Operators. https://www.telesat.com/.
[75] Liam Tung. 2020. 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/.
[76] Viasat. 2020. https://www.viasat.com/.
[77] Tanya Vladimirova and Kawsu Sidibeh. 2007. Inter-Satellite Links in LEO Constellations of Small Satellites. http://www.ee.surrey.ac.uk/m_ssc/research/vlsi/intersatellite.html.
[78] Markus Werner, Axel Jahn, Erich Lutz, and Axel Bottcher. 1995. Analysis of system parameters for LEO/ICO-satellite communication networks. In *IEEE Journal on Selected areas in Communications*.
[79] ROBERT WIEDEMAN, ALLEN SALMASI, and Dennis Rouffet. 1992. Globalstar-Mobile communications where ever you are. In *14th International Communication Satellite Systems Conference and Exhibit*. AIAA.
[80] Lloyd Wood. 2001. *Internetworking with satellite constellations*. Ph.D. Dissertation. University of Surrey.
[81] Lloyd Wood. 2017. Satellite constellation visualization (SaVi). https://savi.sourceforge.io/.
[82] William W Wu, Edward F Miller, Wilbur L Pritchard, and Raymond L Pickholtz. 1994. Mobile satellite communications. In *Proceedings of the IEEE*.

A GROUND STATION RELAYS

Hypatia easily accommodates experimentation with constellations that eschew inter-satellite connectivity. In this scenario, ground station relays provide "bent pipe" connectivity with long-distance communication going up and down through satellites and GSes [31].

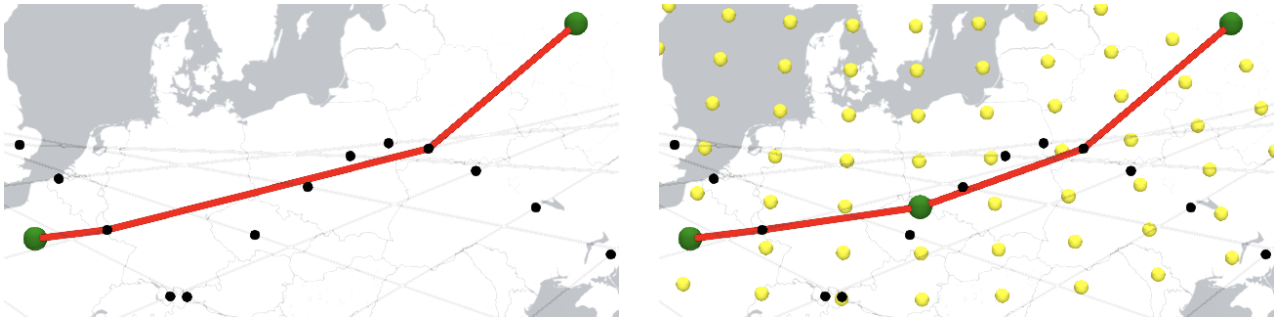
To demonstrate this capability of Hypatia, we simulate a long-lived TCP flow from Paris to Moscow over the first shell of Kuiper's K1 shell (Table 1). We compare the behavior of the flow in two scenarios: (a) the constellation model used in the rest of the paper, *i.e.,* with one GS-satellite link followed by a series of ISLs, followed by a satellite-GS link; and (b) without any ISLs, using only bent-pipe connectivity through GS relays. For the latter case, we add a grid of ground stations between Paris and Moscow such that there are multiple relays that can be chosen from.

Fig. 16(a) and Fig. 16(b) show the situation at the start ($t = 0$) for the two scenarios. Fig. 18(c) compares the path RTT over time for the two scenarios absent any traffic. As expected, the bent-pipe connectivity results in higher latency, typically by around 5 ms. Fig. 17 shows the paths at a different time (around $t = 159$ s), near the "peak RTT" in Fig. 18(c).

While a variety of configurations are possible for network devices, for this experiment, we configured the network devices at the satellites to have a fixed total up-bandwidth, and to use the same queue for all outgoing traffic. This implies that for the bent-pipe connection, an on-path satellite's GSL uplink queue is shared by the TCP packets traveling in one direction (from a GS to the satellite) and the corresponding ACKs traveling in the opposite direction (from a *different* GS to the satellite). Our experiment shows the impact of this for a uni-directional TCP NewReno flow from Paris to Moscow. Both the estimated RTT in Fig. 18(b) and the TCP congestion window in Fig. 19(b) show the effect of the aforementioned sharing of the network device between the data packets and the ACKs. The throughput over the bent-pipe connection is also modestly lower for the same reason, as seen in Fig. 19(c).

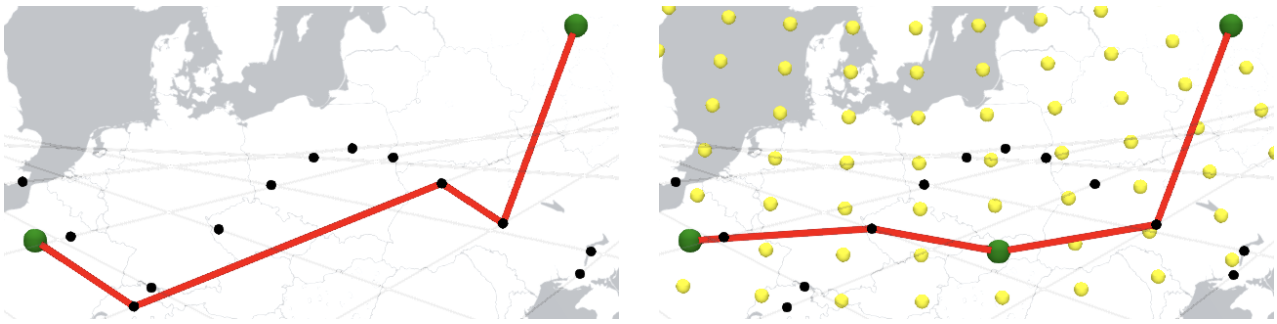
We also see, from comparing Fig. 19(a) and Fig. 19(b), that at least in this instance, TCP behavior is markedly different for the bent-pipe connection, with many more fluctuations in the congestion window. The reason is that ISLs and bent-pipe connectivity engender different bottleneck behavior in this case. The bottleneck with ISLs, is the outgoing network device from the first (source) GS. For the bent-pipe, due to the sharing of the path across data packets and ACKs, the bottleneck is the first satellite-to-ground link. Now as satellites in the path change, specifically the first satellite, the queue size at this satellite varies substantially over time. Re-ordering events that trigger TCP to interpret loss occur at $t = 52.9, 86.3, 92.5, 147.2,$ and 162.1 s, the precise points in Fig. 19(b), where the congestion window drops.

Note that this low-level behavior depends on how the network devices are configured, *e.g.,* multiple separate queues for uplink traffic would change the behavior substantially. With cross-traffic, the ISL case may also exhibit similar behavior, with more frequent reordering as the bottleneck moves away from the first up-link device. We thus caution readers against drawing deeper conclusions on TCP behavioral differences in bent-pipe versus ISL connectivity from this small experiment; this experiment is merely meant to demonstrate Hypatia's ability to support bent-pipe constellations.



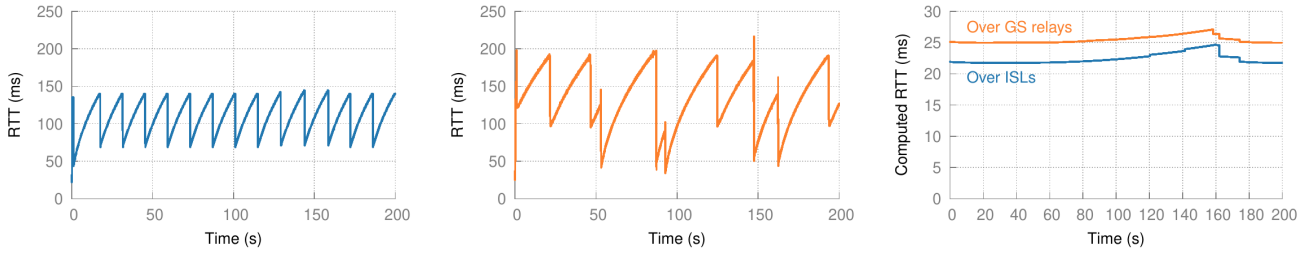
(a) Connectivity over ISLs. (b) Bent-pipe connectivity. The grid shows the candidate GS relays.

Fig. 16: *Paris to Moscow connectivity over Kuiper at the start, $t = 0$. Black dots represent satellites, yellow dots are candidate GS relays, and green dots represent endpoint GSes and chosen GS relays.*



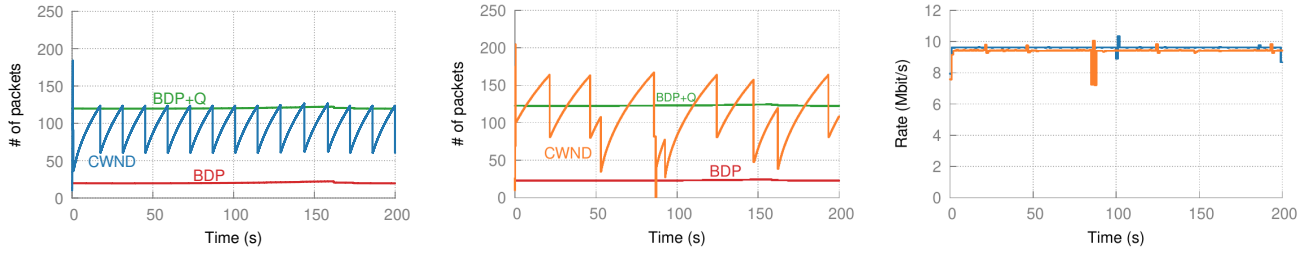
(a) Connectivity over ISLs at $t = 159.4$ *s*. (b) Connectivity over GS relays at $t = 158.3$ *s*.

Fig. 17: *Paris to Moscow connectivity over Kuiper at around $t = 159$.*



(a) TCP estimated RTT with ISLs (b) TCP estimated RTT with bent-pipe (c) Bent-pipe has higher RTT.

Fig. 18: *RTT over time. In both cases, queueing delay at the low bandwidth of 10 Mbit/s inflates RTT far beyond the computed RTT in (c).*



(a) TCP congestion window with ISLs (b) TCP congestion window with bent-pipe. ACKs can queue behind data packets, causing an artificial inflation in congestion window. (c) Bent-pipe achieves modestly lower rate.

Fig. 19: *TCP behaves differently for the ISL and bent-pipe cases, primarily due to different bottleneck behavior as the ACKs share the same bottleneck in our configuration for bent-pipe connectivity.*