In-orbit Computing: An Outlandish thought Experiment?

Debopam Bhattacherjee*, Simon Kassing*, Melissa Licciardello, Ankit Singla
ETH Zürich

ABSTRACT

Space industry upstarts are deploying thousands of satellites to offer global Internet service. These plans promise large improvements in coverage and latency, and could fundamentally transform the Internet. But what if this transformation extends beyond network transit into a new type of computing service? What if each satellite, in addition to serving as a network router, also offers cloud-like compute, making the new constellations not just global Internet service providers, but at the same time, a new breed of cloud providers offering "compute where you need it"?

We examine, qualitatively and quantitatively, the opportunities and challenges of such 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.

CCS CONCEPTS

• **Networks → Cloud computing**.

KEYWORDS

Low Earth orbit satellite, LEO, in-orbit computing, edge computing, satellite edge computing

ACM Reference Format:
Debopam Bhattacherjee*, Simon Kassing*, Melissa Licciardello, Ankit Singla. 2020. In-orbit Computing: An Outlandish thought Experiment?. In *Proceedings of the 19th ACM Workshop on Hot Topics in Networks (HotNets '20), November 4–6, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3422604.3425937

1 INTRODUCTION

"New Space" companies are gearing up to offer global broadband Internet using satellites. At least 3 of these proposals — Starlink [17], Kuiper [34], and Telesat [50] — envision constellations with more than 1,000 satellites each. Starlink's plans are the most ambitious, with tens of thousands of satellites, and by far the most mature, with them having already deployed more than 400 satellites [15].

Unlike today's satellite Internet services [30, 53], which operate in geostationary orbit (GEO) at 35,786 km above the Earth's surface, the proposed constellations will operate in low Earth orbit (LEO), below 2,000 km. This will allow the new networks to offer low latency, with round-trip times between satellites and ground stations potentially in single-digit milliseconds. Further, with their scale, these networks can provide truly global Internet coverage.

This promise of a new breed of global, low-latency, high-bandwidth Internet service providers has generated tremendous interest in both the popular press [14, 25, 43, 52], and among networking researchers [8, 9, 23, 26, 27, 32].

However, we posit that there is a potentially overlooked opportunity here beyond just network service: What if mega-constellations also offer in-orbit compute as a service, much like cloud computing from today's terrestrial data centers? What if we have thousands of networked satellite-servers, with some of them accessible from anywhere on Earth within milliseconds at almost all times? We explore the ramifications of this thought experiment.

Given that such in-orbit compute would be much more limited and pricier than today's cloud compute, it should be obvious that this is a terrible idea for most cloud applications. However, we find that several new and interesting use cases might be well-served by such an unusual infrastructure.

In-orbit compute would 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 [18, 19, 24, 45]. 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. There is clear demand for such anywhere-compute: Amazon's recently announced Snowcone [4] is targeted explicitly at edge computing in environments without accessible compute. Snowcone is a ruggedized small form-factor server, that provides cloud synchronization by shipping it back and forth. In-orbit compute would alleviate the long delays for such data movement, especially from regions with poor transport connectivity.

The flip-side of LEO constellation 'omnipresence' is that a large amount of LEO network infrastructure is largely idle at any given time: with most of Earth's surface being very sparsely populated,

---

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

HotNets '20, November 4–6, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8145-1/20/11…$15.00
https://doi.org/10.1145/3422604.3425937

many satellites are 'useless' at any time, and will not even be transiting data most of the time. Making this infrastructure more continually useful could thus be attractive. One way of doing this is to use it for computing, particularly in cases where the data computed upon is space-native, like for remote sensing and aerial imagery. For such applications, the amount of actually interesting or actionable data is often a minute fraction of the data gathered, but the volume of data generated can overwhelm the down-links from satellites [22, 39]. In-orbit processing can help save the limited bandwidth between satellites and ground stations. The ample satellite-satellite bandwidth can also allow collective data processing across satellites.

Perhaps the most interesting use cases arise for multi-party interactive applications, like games, music collaboration, virtual reality immersion in the same environment with friends, etc. Such applications require a "meetup server", which provides low latency to all involved clients; in some applications, such as gaming, it may also be necessary to have this latency be as uniform across clients as possible. If in-orbit compute is available, a meetup server can be picked specifically for the set of users involved, guaranteeing the (nearly) lowest-possible latency for them, with a more consistent latency experience across users, compared to terrestrial servers.

We flesh out these possibilities in greater detail, quantifying some of the relevant aspects, where possible. We also ruminate on the challenges and downsides of in-orbit compute, including limits on the types of applications that may benefit, and the weight, power budget, life-cycle, cost, and space-readiness of the computing hardware used. While these are substantive concerns, thus far, surprisingly, none of these have seemed entirely prohibitive.

A key technical challenge arises from the high orbital velocity of satellites: unlike GEO satellites, which appear stationary from Earth, any ground station sees a particular LEO satellite for only a few minutes, followed by a hand-off to another satellite. For stateful applications, *e.g.,* a multiplayer game, having such ephemeral satellite-servers is challenging. Presenting applications the needed persistence requires planning ahead to pick a suitable satellite-server, and timely state migration to a suitable successor. *If* this can be done, large LEO constellations can offer a powerful abstraction: GEO-like stationarity, simultaneously above all terrestrial locations, with ~65× lower latency than GEO orbits. We thus explore the constraints on such state migration, and how suitable servers and successors may be picked to relax these constraints as much as possible.

We fully acknowledge how unusual this proposal is: although there have been computers in space for many decades now, the idea of in-orbit compute as a service *is*, very literally, outlandish. Nevertheless, we found this to be an interesting thought experiment with non-trivial use cases and challenges, and hope that the community's engagement with it will either raise killer objections, or a different and interesting set of applications that we did not foresee.

2 LEO NETWORK 101

Low Earth orbit satellites operate at an altitude of less than 2,000 km above the Earth's surface. Given the altitude, a satellite's velocity and orbital period are determined by orbital mechanics. For instance, for an altitude of 550 km, used by the Starlink satellites SpaceX has thus far placed in orbit, the satellites travel at 27,306 km/h, completing each orbit in 95 min 39 sec.

LEO satellites offer an extremely different type of service compared to geostationary satellites. GEO satellites are stationary with respect to the Earth, and achieving this stationarity requires operating at 35,786 km altitude. By flying at a lower altitude than GEO satellites, LEO satellites offer much lower propagation latency — 65× for the 550 km example — but at the cost of losing stationarity. Another distinction is that the higher altitude of GEO satellites provides them a larger cone of coverage, with a handful of satellites sufficient to cover the entirety of the globe. In contrast, LEO satellites necessarily cover much smaller areas, and achieving global coverage requires the use of *many* more satellites. The coverage and velocity characteristics of LEO satellites imply that a ground station sees a particular LEO satellite only for a few minutes. After this time, if continuous connectivity is desired, the ground station must execute a connection hand-off to another LEO satellite that becomes reachable.

To provide such continuous, low-latency connectivity, several companies have proposed large LEO constellations comprising hundreds to tens of thousands of satellites. In particular, SpaceX Starlink, Amazon Kuiper, and Telesat, are all planning constellations with more than a thousand satellites. SpaceX's plans are the most ambitious, with 42,000 planned satellites. SpaceX has already launched more than 400 satellites, making Starlink the largest-ever satellite constellation.

Besides a large number of satellites connecting to ground stations, most of the proposed networks also feature inter-satellite links (ISLs), such that a connection between two distant ground stations traverses an uplink, a series of ISLs, and a downlink. The up- and down-links are planned to be radio, and more limited in bandwidth, on the order of 10 Gbps, while for ISLs higher bandwidths may be achievable [2, 36].

This design approach enables LEO mega-constellations to offer low-latency broadband Internet connectivity. With suitably designed satellite orbits, an LEO constellation can provide truly global coverage: from any location on Earth, at all times, one or more satellites are reachable.

3 IN-ORBIT COMPUTING AS A SERVICE

The premise of our thought experiment is simple: what if instead of restricting themselves to offering network connectivity, the planned LEO constellations also offer computing as a service? Thus instead of being only ISPs, what if these operators also become "in-orbit computing providers" much like cloud computing providers, but above the clouds?

We first attempt to make a positive case for the above proposition by examining which applications may benefit from the unique characteristics of in-orbit computing.

3.1 CDN and edge computing

Content distribution networks invest heavily in building points of presence close to users to minimize latency as much as possible. Nevertheless, the reach of CDNs is limited by several factors: in some geographies, there is no suitable support and power infrastructure for edge sites, or operations are made difficult by political or jurisdictional issues. As a result, in large parts of the world, CDN

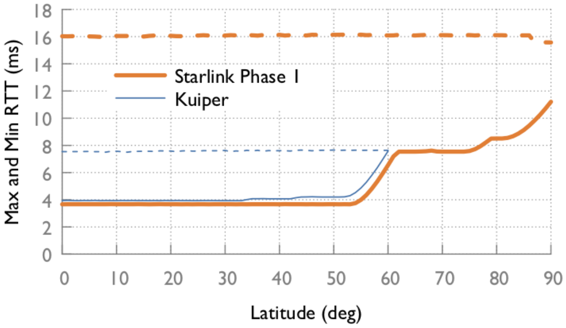

Fig. 1: *In-orbit compute is accessible from everywhere at low latency. The dotted lines are maximum latency i.e., to the farthest reachable satellite-server, while the solid lines are minimum latency i.e., to the nearest satellite-server.*

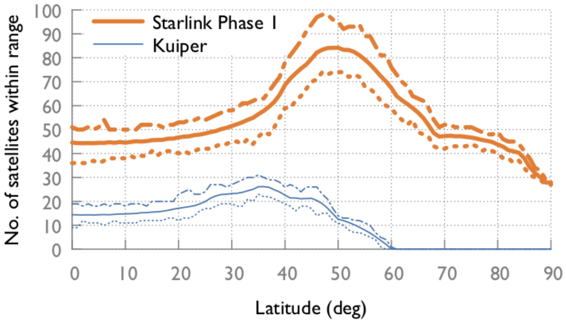

Fig. 2: *A substantial number of satellite-servers would be reachable from everywhere at all times. The solid lines show the average number across time, with the range over time shown by dotted and dashed lines. The peak at certain latitudes (less pronounced for Kuiper) arises from orbital geometry of the planned constellations.*

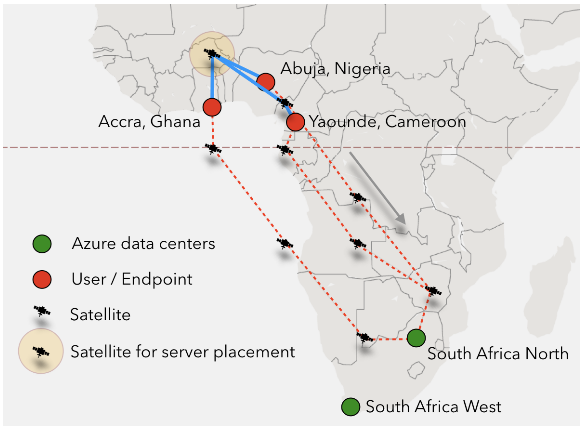

Fig. 3: *A satellite-based meetup server can be significantly closer to the endpoints in West Africa. The paths for hybrid approach are marked in red (thin), and the paths for satellite-server approach are marked in yellow (thick).*

edge latencies still exceed 100 ms [18, 19, 24, 45]. As prior work has shown, even tens of milliseconds of additional round-trip latency deteriorate Web browsing page load times substantially [11].

Further, for new, upcoming applications like **a**ugmented **r**eality, latencies beyond small tens of milliseconds are prohibitive. AR is considered a key component of our future interactions with computing, with a diverse set of envisioned applications, *e.g.*, in education [56], in e-commerce [46], assisting workers by augmenting their environment with helpful information [16], assisting drivers [51], etc. More broadly, many such networked applications that depend on low latency for rich interactivity are being referred to as "Tactile Internet" applications [1, 20, 57]. Many of these applications depend on data fetched or processed from nearby edge computing infrastructure, with tight latency requirements.

In-orbit computing could serve these needs by providing low-latency compute access everywhere on Earth. Fig. 1 shows the round-trip times from ground locations at different latitudes to two planned LEO constellations, Kuiper and Starlink. For Starlink, we only use the Phase I configuration, comprising 4,409 satellites, as details for the rest of the constellation are sparse. For each constellation, we compute the RTT from a ground location every minute over two hours, and use the maximum value across these measurements. We do so for the nearest reachable satellite, as well as the farthest (directly) reachable satellite. The nearest reachable satellite is within ∼4 ms at most latitudes for both constellations. Kuiper's design does not provide service beyond 60° latitude. Starlink's nearest satellite is within 11 ms RTT across *all* possible ground locations at *all* times. Even the farthest directly reachable satellite is within 16 ms RTT. Of course, these numbers are computed accounting for only propagation delays, so true latencies will be somewhat higher.

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. 2 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 is being envisioned for "cloudlets" or edge computing sites [10, 48]. Note that given the result in Fig. 1 on the latency to the farthest directly reachable satellite, *all* of these satellites are reachable within a small RTT.

Somewhat surprisingly, if just one server were added to each of its satellites, Starlink, with 40,000 planned satellites at its full scale, would be only 7× smaller than the largest present-day CDN, Akamai [3]. Also given that today's CDN infrastructure is skewed, with larger clusters near metro areas, in-orbit compute could very well suffice to meet the needs of sparsely populated areas not covered by CDNs.

Thus, in-orbit computing over large LEO constellations could bring substantial edge computing capabilities to any location on the planet, allowing low-latency access to these resources. This can help truly bridge the digital divide: to ensure that remote or under-served areas are not left out of the next computing revolution, access to proximal compute is as essential as access to networking.

3.2 Multi-user interaction

A variety of applications involve multiple users participating in an interactive activity, such as online gaming, collaboratively playing music, learning in a shared online classroom, or more broadly, immersion in the same virtual environment. In these applications, often the multi-user group's experience is contingent on *every* participant receiving low latency connectivity to the server hosting the application, which we shall refer to as the "meetup server". For some applications, like gaming, it is also important that latency differences across users are small, so that each user's experience is somewhat consistent, and in competitive settings, no user has a significant disadvantage compared to others.

Today, these problems are side-stepped by restrictions on which users can participate together, *e.g.*, by matchmaking in online games, which typically accounts for player latencies to the game server. This is, of course, limiting, as it prevents certain sets of users from participating with their friends. With in-orbit computing, this limitation can be overcome: for a large LEO constellation, there is always a satellite that provides nearly the minimum-possible latency for the user group as a whole, instead of being limited by the location of the terrestrial game servers.

Consider the example shown in Fig. 3. Microsoft Azure, which claims to have "more global regions than any other cloud provider" [5] has two data center regions in Africa. Three users in West Africa need a meetup server for an interactive application. The nearest Azure data center that could host the meetup server is 9,200 km round-trip from the farthest of these three users, and her latency to this server will determine their experience. Even using the Starlink LEO constellation to connect the users to this terrestrial meetup server, we find that the RTT would be as high as 46 ms. On the other hand, the RTT to a meetup server hosted using in-orbit compute on the same constellation would be 16 ms, an almost 3× reduction. For applications like augmented and virtual reality, much smaller differences would suffice to make the terrestrial server entirely unworkable, while the in-orbit server would provide high quality of experience. Note that in line with Starlink's model [47], we assume that user terminals can communicate directly via satellites without any gateway intervention.

This use case is most relevant to two types of user-groups: ones where most of the users in the group are relatively far from a data center (like in the above example, but broadly in Africa, South America, and large parts of Asia), and ones where each user themselves may be near a data center, but there is no *one* data center location that provides low latency for all users, such as for a user group partitioned across the Americas and Eurasia. A particularly illustrative example of the latter type is seen on the Kuiper constellation's first phase planned deployment, where for users in 3 locations that each have an Azure data center — South Central US, Brazil South, and Australia East — the best terrestrial meetup server incurs 97 ms, while an in-orbit server can achieve 66 ms latency. This lower latency can improve quality of experience for gamers [7, 42]. For other applications, *e.g.*, ones needing real-time haptic feedback [1, 57], the upper bound on acceptable latency may be smaller still; this will limit the physical separation of users of such applications for *any* deployment, but a lower latency architecture allows greater separation between users, and thus expands the communities served by such applications.

3.3 Processing space-native data

In-orbit computing resources would necessarily be much more limited in quantity and more expensive than terrestrial cloud resources (§4). Moving data from terrestrial sources to satellites for computing would also be costly. Thus, for most cloud applications, in-orbit compute would be unattractive.

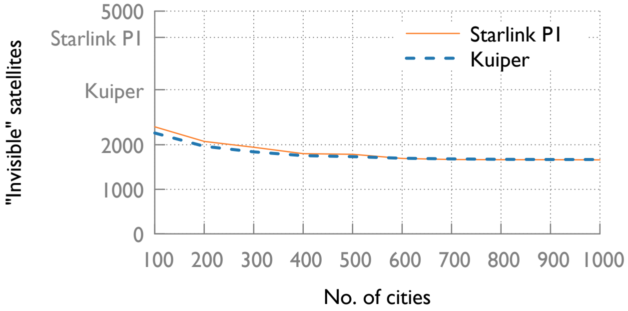

Fig. 4: *At any time, a large fraction of satellites are not directly reachable from* any *population center, and thus "invisible" to nearly all ground observers. The y-tics also mark the total number of satellites in the two constellations.*

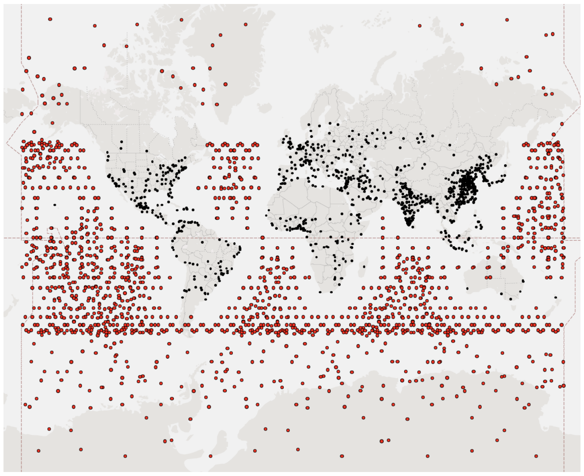

Fig. 5: *"Invisible" Starlink satellites: the* 1000 *largest population centers are shown as small (black) dots, and the invisible satellites as larger (red) circles. Other satellites are not shown.*

However, many applications *generate* their data in space, *e.g.*, Earth imagery, weather observations, and remote sensing. Data analytics on "Big Earth Observation Data" [12, 13, 35, 55] is seen as a lucrative and expanding market [41], so much so, that Amazon recently launched its "AWS Ground Station" service targeting this segment. At least one of the LEO networks planned, Starlink, has hinted that its satellites may additionally produce such sensing data [40]. Would processing this data at satellite-servers, in line with the "near-data processing" paradigm [6], be fruitful?

Satellites performing imagery and sensing are capable of multi-Gbps data production, but their sensing time is limited by data transmission capacity to ground stations [21, 22].[1] Thus, pre-processing this data using in-orbit compute could increase sensing time, and

---

[1] While the planned networks may provide on the order of 10 Gbps up/down links, given their primary objective of providing network connectivity, using a substantial fraction of this bandwidth for sensing data may require compromising one or the other function.

decrease the bandwidth cost of downloading this data and further processing it terrestrially.

A constellation with several thousand satellites could potentially perform such data processing in-orbit. The inter-satellite bandwidth is likely to be less constrained [2, 36, 37], and would allow satellites to process data cooperatively. Unlike terrestrial data centers, where latencies between servers are a few microseconds, satellite-satellite communication will incur milliseconds, but this should still be sufficient for bulk processing on the large volumes of data involved.

Another benefit of using satellites for such compute is that it allows better use of LEO infrastructure. At any time, a large fraction of a network-only constellation's resources are above areas where they do not provide useful connectivity. Opportunistic in-orbit data processing can help the operator extract useful work from this infrastructure continuously.

To illustrate the issue of low-connectivity-utility satellites, we computed the number of satellites in LEO networks that are not within reach of population centers. For this purpose, we use a snapshot of each of the Kuiper and Starlink constellations, and count the satellites that are not directly within reach from any of the largest n cities by population. Fig. 4 shows the results for $n \in \{100, 200, \ldots, 1000\}$. Even with ground stations at 1000 cities, more than a third of Starlink's and more than a half of Kuiper's satellites are "invisible" in this manner at any time. Fig. 5 shows these invisible satellites for Starlink — while some of these satellites will provide valuable connectivity using inter-satellite links, such as for transatlantic routes, the vast majority of invisible satellites are the ones South of most of the World's population, and won't have much utility for networking. (Of course, minutes later, these satellites *do* provide useful connectivity.)

4 FEASIBILITY OF IN-ORBIT COMPUTE

Space is an unusual computing environment, both in terms of establishing computing resources there, and operating and maintaining them. We thus discuss several related aspects.

Weight and volume: We compute a commodity server's weight and volume relative to that of a Starlink satellite. We use a high-end server, the HPE ProLiant DL325 Gen10 [28]: 64 cores that clock 2.4-3.35 GHz, up to 2 TB memory, and 15.6 kg weight. Compared to the latest Starlink satellites launched, the weight is 6% of a satellite's weight, and the volume is 1%. These are significant costs, but not prohibitive.

Radiation hardening: The HPE Spaceborne Computer [29] aboard the International Space Station (ISS; 408 km) is commodity hardware without any specialized hardware frame. Instead, it uses software hardening. Thus, in LEO, especially for orbits below the inner Van Allen radiation belt (outwards from 643 km), it is likely that commodity hardware is sufficient, although this is not yet a fully settled question.

Power: Based on rough estimates of a Starlink satellite's solar array size and solar efficiency numbers for the ISS, the average solar power output available is estimated to be around 1.5 kW [49]. (Satellites use batteries for continuous operation, given that substantial orbital time is spent in the Earth's shadow.) The HPE server operating at 225 W (350 W) would consume 15% (23%) of this power. This is quite large, but if necessary, lower wattage servers could be used, or a larger solar array and battery could be built-in to support this additional power requirement. This would, however, come at the cost of additional payload weight and volume.

It is also unclear how the addition of compute skews power usage over time, *e.g.,* due to spikes in communication demands coinciding with spikes in compute demands. If a satellite's power use fluctuates more due to this, it may create additional challenges in power management beyond the average output over time.

Another related problem is the increased heat generation. Heat is harder to dissipate without an atmosphere, so additional radiators, or thermoelectric harvesters [54] may be necessary. However, as noted above, such additional components will increase the mass per satellite, and hence may result in fewer satellites per launch.

Life-cycle: Starlink satellites will have a life of ~5 years [44]. This is a bit longer than the typical data center server life of 3 years [33]. Of course, if a satellite-server malfunctions before its expected life, unlike in a data center, it would not be replaced immediately. However, operators continually replenish their satellite fleet, and maintain backup satellites per orbit. Thus, even with a substantial fraction of servers failing, a large LEO constellation could continue to provide valuable in-orbit computing resources.

Cost: The relative cost of adding compute to a satellite is already accounted for in terms of weight and volume, as the server is much cheaper than the cost of launching its weight. But in absolute terms, how does in-orbit compute compare to terrestrial compute in terms of cost? Based on the per-kilogram launch cost for the Falcon 9 rockets used for Starlink launches [31], and the 15.6 kg server weight, the cost of launching the server is ~42,000 USD. The per-server total cost of ownership for a data center is estimated to be roughly 5,000 USD **per year** [33]. If we assume the satellite-server is also used for only 3 years instead of 5, then over 3 years, a coarse estimate for a satellite-server would be roughly 3× as expensive as a data center server. This low figure may be a bit deceiving: many of the data center costs are incurred for the scale (physical sites, power infrastructure, IT and support staff, etc.) but the benefits of scale are not priced here.

Summary: While there are surely other factors that would need careful accounting for a more precise feasibility study, our analysis is cautiously encouraging. A server's power-draw could be a substantial burden, and the cost of compute is several times that of terrestrial facilities. Nonetheless, *if* operators and customers perceive enough value in the use cases, these barriers may not be insurmountable.

5 VIRTUAL STATIONARITY

While applications like in-orbit data processing (§3.3) can be opportunistically used on satellite-servers, for some of the use cases for in-orbit compute, such as multi-user interactivity (§3.2), the dynamic behavior of LEO constellations poses a challenge: a particular meetup-server is only reachable by clients for a few minutes, after which a hand-off is needed to a different satellite. This hand-off is different from cellular or WiFi hand-off due to the additional mobility of the *network infrastructure* — highly dynamic yet predictable. For applications that need to persist state over a specific geography, like for games and co-immersion involving a particular set of users,

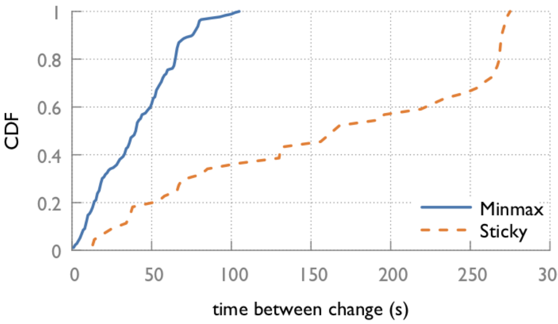

Fig. 6: *Sticky server selection significantly reduces the frequency of hand-offs compared to the MinMax approach.*

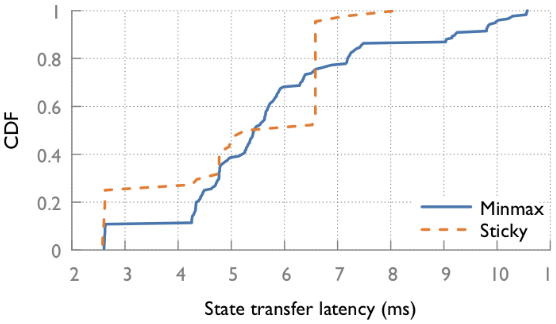

Fig. 7: *State transfer latencies with the Sticky server approach are comparable to those for the MinMax approach.*

this requires careful migration of application state before a hand-off. If this can be accomplished, a series of LEO meetup-servers expose to clients an abstraction similar to GEO satellites: stationary operation at (roughly) the same location over time.

The naive approach for selecting a meetup-server picks the latency-optimal satellite at each instant. We refer to this as "MinMax", as it minimizes the maximum latency across a set of clients connected. However, MinMax suffers from two problems: potentially frequent hand-offs to successor meetup-servers, and potentially large latency for some hand-offs. We thus propose an alternative heuristic, "Sticky", that prioritizes stationarity by planning ahead leveraging predictable satellite motions, as follows:

1. Compute the set of meetup-servers that provide latency within 10% of MinMax.
2. For each of these candidate meetup-servers, compute the time until the next hand-off. Pick the 5 candidates with the longest time until a hand-off.
3. Among these 5, pick one which would result in the least latency for hand-off to its successor.

We find that Sticky results in slightly higher latency for the user-group, *e.g.,* by 1.4 ms in the West Africa (3 users) example in Fig. 3. For this small price, Sticky substantially decreases the frequency of hand-offs, while the latency for state transfer to the successor remains similar across Sticky and MinMax.

Fig. 6 shows the distribution of times between satellite hand-offs: the median time between hand-offs is 164 sec for Sticky, *i.e.,* 4× longer than for MinMax. Further, as Fig. 7 shows, the latency incurred in migrating state to the successor server is similar and low for both approaches, with Sticky providing an advantage in the tail. Thus even simple heuristics can help decrease the satellite-churn applications see.

We acknowledge that state migration after every few minutes is still a substantial overhead. However, the high inter-satellite bandwidth could accommodate this. From the application perspective, it may be beneficial to separate session-specific state from generic application state, *e.g.,* the player and game state versus the virtual world of a game, and perform live migration only for the session-specific state, while generic state is replicated even further ahead.

6 DISCUSSION & CONCLUSION

The premise of in-orbit computing as a service may certainly seem outlandish. Our goal is not to push a strongly affirmative case for it; we only suggest that it might be worth not dismissing casually, as it has unusual and interesting properties that some applications may benefit from.

In-orbit computing could offer low-latency access to compute from anywhere on Earth, and enable a new abstraction: GEO-like stationarity without the GEO latency penalty. These properties open up potentially new use cases, including for edge computing, meetup servers for multi-user interaction, and opportunistic data analytics for space-native data.

However, in-orbit compute would be many times more expensive than terrestrial data center compute, and would only be useful in settings where the latter is limited due to either access latency, or bottlenecks in downloading satellite-generated data. Terrestrial infrastructure could potentially also expand to address the latency issue. For instance, an ambitious effort is currently exploring the use of off-shore container-based data centers [38]; coincidentally, a key barrier for such data centers is also power provisioning.

The types of applications that could benefit from in-orbit LEO compute are also limited on another front: for some settings where terrestrial data center infrastructure is limiting, GEO satellites are perfectly acceptable, because latency is not an issue. One such example is video broadcast, a common application of GEO satellites. It is unlikely that serving video through LEO satellites would be worthwhile.

Our analysis of the issues of weight, volume, space hardening, and life-cycle, indicates that these are not prohibitive problems. Power is perhaps the biggest impediment, and it is unclear if the $\sim20\%$ overhead from a beefy server would be acceptable, given the substantial modifications it could require to a satellite's power provisioning.

Weather, which we did not analyze yet, also poses limitations on availability: LEO network interruptions due to weather attenuation on the ground-satellite links would make in-orbit compute temporarily unavailable from the affected locations.

Lastly, while we attempted to inform ourselves broadly of the challenges and downsides, there may be "unknown unknowns" that could potentially make in-orbit compute entirely infeasible. We hope that our readers will either raise such objections, or suggest use cases that we did not foresee.

REFERENCES

[1] Adnan Aijaz, Mischa Dohler, A Hamid Aghvami, Vasilis Friderikos, and Magnus Frodigh. 2016. Realizing the tactile Internet: Haptic communications over next generation 5G cellular networks. In *IEEE Wireless Communications*.

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

[3] Akamai. 2020. Facts & Figures. https://www.akamai.com/uk/en/about/facts-figures.jsp.

[4] Amazon. 2020. AWS Snowcone. https://aws.amazon.com/snowcone/.

[5] Microsoft Azure. 2020. Azure locations. https://azure.microsoft.com/en-us/global-infrastructure/locations/.

[6] Rajeev Balasubramonian, Jichuan Chang, Troy Manning, Jaime H Moreno, Richard Murphy, Ravi Nair, and Steven Swanson. 2014. Near-data processing: Insights from a MICRO-46 workshop. In *IEEE Micro*.

[7] Tom Beigbeder, Rory Coughlan, Corey Lusher, John Plunkett, Emmanuel Agu, and Mark Claypool. 2004. The effects of loss and latency on user performance in unreal tournament 2003®. In *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*. 144–151.

[8] 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*.

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

[10] Kashif Bilal, Osman Khalid, Aiman Erbad, and Samee U Khan. 2018. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. In *Computer Networks*. Elsevier.

[11] 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*.

[12] Catherine Burton. 2016. Earth Observation and Big Data: Creatively Collecting, Processing and Applying Global Information. https://eijournal.com/print/articles/earth-observation-and-big-data-creatively-collecting-processing-and-applying-global-information.

[13] Gilberto Camara, Luiz Fernando Assis, Gilberto Ribeiro, Karine Reis Ferreira, Eduardo Llapa, and Lubia Vinhas. 2016. Big earth observation data analytics: Matching requirements to system architectures. In *ACM SIGSPATIAL international workshop on analytics for big geospatial data*.

[14] 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/.

[15] 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/.

[16] Mathilde R Desselle, Ross A Brown, Allan R James, Mark J Midwinter, Sean K Powell, and Maria A Woodruff. 2020. Augmented and virtual reality in surgery. In *Computing in Science & Engineering*. IEEE.

[17] 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/.

[18] Rodérick Fanou, Gareth Tyson, Eder Leao Fernandes, Pierre Francois, Francisco Valera, and Arjuna Sathiaseelan. 2018. Exploring and analysing the African Web ecosystem. In *ACM Transactions on the Web (TWEB)*.

[19] Reza Farahbakhsh, Angel Cuevas, Antonio M Ortiz, Xiao Han, and Noel Crespi. 2015. How far is Facebook from me? Facebook network infrastructure analysis. In *IEEE Communications Magazine*.

[20] G Fettweis, H Boche, T Wiegand, E Zielinski, H Schotten, P Merz, S Hirche, A Festag, W Häffner, M Meyer, et al. 2014. The tactile internet-itu-t technology watch report. In *Int. Telecom. Union (ITU), Geneva*.

[21] Dirk Giggenbach, Bernhard Epple, Joachim Horwath, and Florian Moll. 2008. Optical satellite downlinks to optical ground stations and high-altitude platforms. In *Advances in Mobile and Wireless Communications*. Springer.

[22] Dirk Giggenbach, Joachim Horwath, and Markus Knapek. 2009. Optical data downlinks from earth observation platforms. In *Free-Space Laser Communication Technologies XXI*. International Society for Optics and Photonics.

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

[24] Utkarsh Goel, Mike P Wittie, and Moritz Steiner. 2015. Faster web through client-assisted CDN server selection. In *IEEE ICCCN*.

[25] 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.

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

[27] Mark Handley. 2019. Using ground relays for low-latency wide-area routing in megaconstellations. In *ACM HotNets*.

[28] Hewlett Packard Enterprise. 2020. HPE ProLiant DL325 Gen10 Server: QuickSpecs (Version 20). https://h20195.www2.hpe.com/v2/getdocument.aspx?docname=a00045498enw.

[29] HPE Press Release. 2018. HPE delivers first above-the-cloud supercomputing services for astronauts to advance space exploration. https://www.hpe.com/us/en/newsroom/press-release/2018/11/hpe-delivers-first-above-the-cloud-supercomputing-services-for-astronauts-to-advance-space-exploration.html.

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

[31] Harry Jones. 2018. The recent large reduction in space launch cost. In *48th International Conference on Environmental Systems*.

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

[33] Jonathan Koomey, Kenneth Brill, Pitt Turner, John Stanley, and Bruce Taylor. 2007. A simple model for determining true total cost of ownership for data centers. In *Uptime Institute White Paper*. Citeseer.

[34] Kuiper Systems LLC. 2019. TECHNICAL APPENDIX: 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.

[35] Christopher Lynnes and Thomas Huang. 2018. Future of Big Earth Data Analytics.

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

[37] Mynaric. 2020. Welcome to Terabit Speed. https://mynaric.com/technology/high-speed/.

[38] Natick. 2020. Microsoft Research: Project Natick. https://natick.research.microsoft.com/.

[39] Jeng-Shyang Pan, Pei-Wei Tsai, and Hsiang-Cheh Huang. 2017. Advances in intelligent information hiding and multimedia signal processing. In *Conference proceedings IIH-MSP*. Springer.

[40] United States Patent and Trademark Office. 2017. Trademark serial no. 87576978 (STARLINK).

[41] Laurent Probst, Bertrand Pedersen, and Lauriane Dakkak-Arnoux. 2017. Digital Transformation Monitor - Big Data in Earth Observation. https://ec.europa.eu/growth/tools-databases/dem/monitor/sites/default/files/DTM_Big%20Data%20in%20Earth%20Observation%20v1.pdf.

[42] Peter Quax, Patrick Monsieurs, Wim Lamotte, Danny De Vleeschauwer, and Natalie Degrande. 2004. Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game. In *Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games*. 152–156.

[43] 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.

[44] Michael Sheetz. 2019. SpaceX launches another 60 Starlink satellites while setting two rocket reuse records. https://www.cnbc.com/2019/11/11/watch-spacex-livestream-launching-second-starlink-internet-mission.html.

[45] Rachee Singh, Arun Dunna, and Phillipa Gill. 2018. Characterizing the deployment and performance of multi-cdns. In *Proceedings of the Internet Measurement Conference 2018*.

[46] Alan Smithson, Jason Yim, Mohamed Rajani, Richard Hess, and Eden Chen. 2019. SUPERPOWERED CONSUMER ENGAGEMENT. https://www.awexr.com/usa-2019/agenda/428-superpowered-consumer-engagement.

[47] spacewatch.global. 2020. SpaceX Receives FCC Approval To Deploy One Million Starlink 'UFO' User Terminals In US. https://spacewatch.global/2020/03/spacex-receives-fcc-approval-to-deploy-one-million-starlink-ufo-user-terminals-in-us/.

[48] StackPath. 2020. What are Micro Data Centers? https://www.stackpath.com/edge-academy/micro-data-centers/.

[49] starcraftre (reddit user). 2019. Starlink satellite power (discussion thread). https://www.reddit.com/r/Starlink/comments/e6zo4a/starlink_satellite_power/f9uc6zn/?utm_source=share&utm_medium=web2x.

[50] Telesat. 2020. APPLICATION FOR MODIFICATION OF MARKET ACCESS AUTHORIZATION. https://fcc.report/IBFS/SAT-MPL-20200526-00053/2378318.pdf.

[51] Cuong Tran, Karlin Bark, and Victor Ng-Thow-Hing. 2013. A left-turn driving aid using projected oncoming vehicle paths with augmented reality. In *Proceedings of the 5th international conference on automotive user interfaces and interactive vehicular applications*.

[52] 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/.

[53] Viasat. 2020. https://www.viasat.com/.

[54] Marian Von Lukowicz, Elisabeth Abbe, Tino Schmiel, and Martin Tajmar. 2016. Thermoelectric generators on satellites—An approach for waste heat recovery in space. In *Energies*. Multidisciplinary Digital Publishing Institute.

[55] Xiaochuang Yao, Guoqing Li, Junshi Xia, Jin Ben, Qianqian Cao, Long Zhao, Yue Ma, Lianchong Zhang, and Dehai Zhu. 2020. Enabling the Big Earth Observation Data via Cloud Computing and DGGS: Opportunities and Challenges. In *Remote*

Sensing. Multidisciplinary Digital Publishing Institute.

[56] Steve Chi-Yin Yuen, Gallayanee Yaoyuneyong, and Erik Johnson. 2011. Augmented reality: An overview and five directions for AR in education. In *Journal of Educational Technology Development and Exchange (JETDE)*.

[57] Qi Zhang, Jianhui Liu, and Guodong Zhao. 2018. Towards 5G enabled tactile robotic telesurgery. In *arXiv preprint arXiv:1803.03586*.