Geographic Low-Earth-Orbit Networking *without* QoS Bottlenecks from Infrastructure Mobility

Lixin Liu*, Hewu Li†, Yuanjie Li‡, Zeqi Lai‡, Yangtao Deng*, Yimei Chen§, Wei Liu*, Qian Wu†
*†‡Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing 100084, China
Email: {*liulx19, *dengyt21, *lw21}@mails.tsinghua.edu.cn, {†lihewu, †wuqian}@cernet.edu.cn,
{‡yuanjiel, ‡zeqilai}@tsinghua.edu.cn, {§cym990915}@gmail.com

***Abstract*—Low-earth-orbit (LEO) satellite mega-constellations promise broadband, low-latency network infrastructure from space for terrestrial users in remote areas. However, they face new QoS bottlenecks from *infrastructure mobility* due to the fast-moving LEO satellites and earth's rotations. Both cause frequent space-ground link churns and challenge the network latency, bandwidth, and availability at the global scale. Today's LEO networks mask infrastructure mobility with fixed anchors (ground stations) but cause single-point bandwidth/latency bottlenecks. Instead, we design LBP to remove the LEO network's QoS bottlenecks from infrastructure mobility. LBP removes remote terrestrial fixed anchors via geographic addressing for shorter latencies and more bandwidth. It adopts local, orbit direction-aware geographic routing to avoid global routing updates for high network availability. LBP further shortens the routing paths by refining handover policies by satellites' orbital directions. Our experiments in controlled testbeds and trace-driven emulations validate LBP's 1.64× network latency reduction, 9.66× more bandwidth, and improve network availability to 100%.**

I. Introduction

Space is the next frontier for networking. Low-earth-orbit (LEO) satellite mega-constellations networks are recently under rapid deployments by SpaceX Starlink [4], Amazon Kuiper [1], Telesat [5], to name a few. They extend terrestrial networks' coverage to the remaining unconnected 3 billion users [22] in rural areas, oceans, airplanes, and space. Compared to classical geostationary satellite communications, LEO satellite networks at lower altitudes (≤2,000 km) promise competitive bandwidth and latency to terrestrial networks.

Unfortunately, the potential of low latency and high bandwidth in today's LEO networks has not been fulfilled due to *infrastructure mobility.* LEO satellites move much faster (27,000 km/h) than geostationary satellites[1], airplanes, or terrestrial mobile nodes. Moreover, as we will analyze in Section II-B, the earth's rotation complicates the relative motions between LEO satellites and terrestrial infrastructure nodes. This triggers frequent space-ground link churns and topology changes, which propagate to upper layers and challenge the network availability, bandwidth, and latency.

At first glance, satellite mobility is a well-known and easy-to-solve issue. Classical mobility management (e.g., mobile IP [27], [28], cellular handovers [6], [7], and carrier-grade NAT [1], [17]) relies on the fixed infrastructure (gateways or base stations) as anchors to track mobile users' locations, migrate links via handovers, and redirect traffic through the anchors via triangular routing. While feasible for *user mobility*, this method limits QoS in LEO *infrastructure mobility*. Clearly, the fast-moving LEO satellites cannot be the fixed anchors. While terrestrial ground stations can still play as anchors (which is indeed the case in today's LEO networks), they cause global detours (e.g., 26,630 km detours and 88.77 ms delays on average in Starlink and today's ground station distributions if serving global users based on World Bank's statistical distributions [32]) and bandwidth bottlenecks due to space-terrestrial asymmetry (Section II-C). Replacing fixed anchors with distributed routing may resolve both bottlenecks, but would significantly lower the network usability due to frequent global routing updates and re-convergence (≤10% network availability, detailed in Section II-C).

To this end, this paper asks on the following question: *Can we remove network latency and bandwidth bottlenecks without lowering the network availability from the LEO infrastructure mobility?* Our study yields positive answers. We design LBP, a **L**ocation **B**ased **P**rotocol that explores *orbit-aware geographic* addressing, routing, and handovers to remove the QoS bottlenecks from LEO infrastructure mobility. LBP eliminates terrestrial fixed anchors for shorter latencies and more bandwidth. To mask infrastructure mobility, it locates each terrestrial node by its geographic location rather than its transient serving satellite's logical interface. LBP next retains high network availability with its domain-specific, *orbit direction-aware geographic* routing. Unlike the classic logical routing, LBP only relies on local geographic and orbit direction information to guarantee global reachability, thus eliminating the repetitive global routing updates and low network availability. LBP further helps shorten routing paths and reduces jitters by offering handover optimizations based on orbital plane directions.

We showcase LBP's feasibility with an IPv6-based implementation in Quagga and customized Linux kernels. Our experiments in controlled testbed and trace-driven emulations show that, by eliminating single-point bottlenecks from terrestrial ground stations, LBP saves 1.64× network latencies and increases 9.66× average bandwidth. Meanwhile, LBP retains ≈100% network service availability in LEO mobility without global routing updates, re-convergence, or signaling costs.

[1]The Kepler's third law implies that satellites at lower altitudes move faster.

978-1-6654-6824-4/22/$31.00 ©2022 IEEE

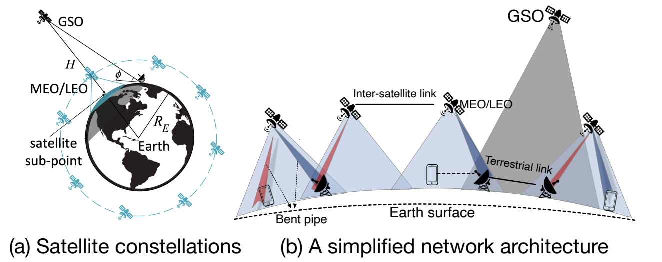


Fig. 1: Satellite constellations.

	Total satellites	Satellites per orbit M	Num. orbits N	Altitude (km)	Inclination angle (°)
Starlink	1584	22	72	540/550	53
Kuiper	1156	34	34	630	51.9
Telesat	351	13	27	1015	98.98
Iridium	66	11	6	780	86.4

TABLE I: Representative LEO satellite mega-constellations.

II. Motivation

A. LEO Satellites and Mega-Constellations Primer

A satellite can operate at the geosynchronous orbit (GSO, at ≈35,786 km altitude) or non-geosynchronous orbits. Classical satellite communications usually work at the geosynchronous orbit, which offers stable connectivity and good coverage at high altitudes (Fig 1) at the cost of long space-ground RTT (≥200ms) and low bandwidth (≤10Mbps). Instead, recent efforts seek to adopt satellites at lower non-geosynchronous orbits, with a special interest in low earth orbits (LEO at ≤2,000 km altitudes). Each LEO satellite promises high bandwidth and low latency [8], [10], [19], but at the cost of low coverage. To this end, an LEO *mega-constellation* is necessary to retain global coverage (Fig 1). Table I lists some recent LEO satellite mega-constellations. Each LEO constellation comprises N orbits and there are M satellites uniformly distributed in each orbit. All the orbits have the same inclination α and are evenly spaced along the equator.

B. Complex, Dynamic Mobility of LEO Network Infrastructure

Unlike terrestrial networks or geosynchronous satellite communications, LEO satellite networks exhibit unavoidable network *infrastructure mobility*. At the first glimpse, LEO satellite mobility seems regular due to its orbital movements. Unfortunately, it turns complex for three reasons:

1) LEO satellites move 1–2 orders-of-magnitude faster (27,000 km/h) than airplanes (≤3,000 km/h), or terrestrial mobile nodes (≤500 km/h for high-speed trains);
2) Due to the earth's rotations, the non-geosynchronous nature of low-earth-orbits complicates the relative motions between satellites and terrestrial nodes;
3) Modern LEO mega-constellations in Table I adopt *inclined orbits* (rather than polar orbits) that further complicate the space-terrestrial relative motions.

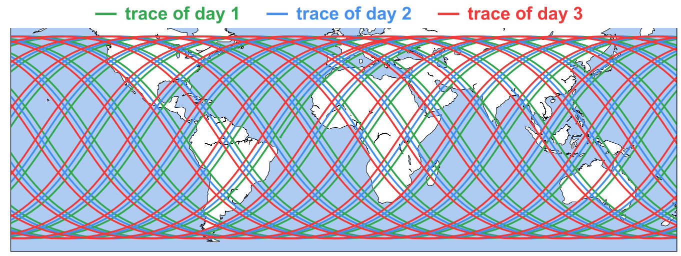


Fig. 2: Trajectory of a Starlink LEO satellite in 3 days.

As a result, the runtime location of an LEO satellite's terrestrial projection (i.e., sub-point in Fig 1) at time t is

$$\varphi_t = \arcsin(\sin\alpha \cdot \sin u_t) \quad (1)$$

$$u_t = u_0 + 2\pi/T_S \quad (2)$$

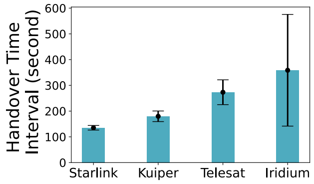
$$\lambda_t = \xi(u_t) + L_0 - 2\pi/T_E \cdot t \quad (3)$$

$$\xi(u_t) = \begin{cases} \arctan(\cos\alpha \cdot \tan u_t), & \text{if } u_t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ \arctan(\cos\alpha \cdot \tan u_t) + \pi, & \text{otherwise} \end{cases} \quad (4)$$

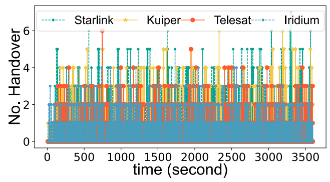
where φ_t and λ_t are the latitude and longitude of the satellite's sub-point at time t, T_S is the cycle of satellite constellation, T_E is the cycle of earth's rotation, α is the orbit inclination and $u_t \in [-\pi, \pi]$ is the satellite phase angle from its ascending node at time t, which describes the satellite position in orbit. u_0 is the satellite phase angle from its ascending node at time 0. When $u_t \in [-\pi/2, \pi/2]$, the satellite is in the ascending plane and flies towards northeast, while $u_t \in [-\pi, \pi/2) \cup (\pi/2, \pi]$ means the descending plane towards southeast. $\xi(u_t)$ represents the longitude difference from the satellite to its ascending node, which varies with the satellite phase. L_0 is the initial longitude of the orbit ascending node, which is an absolute parameter determining the orbit plane position. Fig 2 exemplifies an LEO satellite's sub-point trajectory for 3 days. In reality, this relative motion is further exacerbated by the orbital perturbations by moon, sun, and other planets. Moreover, with many satellites in the LEO mega-constellation (each having different initial positions), the relationship between satellites and ground stations become more dynamic and complex.

Impact: Frequent space-ground link churns. The above complex and dynamic LEO infrastructure mobility causes frequent space-ground link churns and thus network topology changes. Fig 3 plots the space-ground link churns for LEO mega-constellations in Table I and ground stations in Fig 4. Users in Iridium should switch from one satellite to another satellite about 10 minutes, but in current representative LEO networks, the handover time interval decreases to 3∼5 minutes. On average, each ground station switches its serving LEO satellite every 135.21s, 179.77s, 273.25s and 358.61s in Starlink, Kuiper, Telesat, and Iridium, respectively.

Infrastructure Mobility vs. User Mobility At first glance, LEO satellite network mobility seems a well-known and easy-to-solve problem. Mobility management has been a classical topic with mature solutions such as mobile IP [27], [28], cellular handovers [6], [7], carrier-grade NAT [1], [17], and many more. Despite their diversity, these solutions are primarily



(a) Handover interval per station.



(b) No. Handover per second.

Fig. 3: Frequent space-ground link churns in LEO networks.

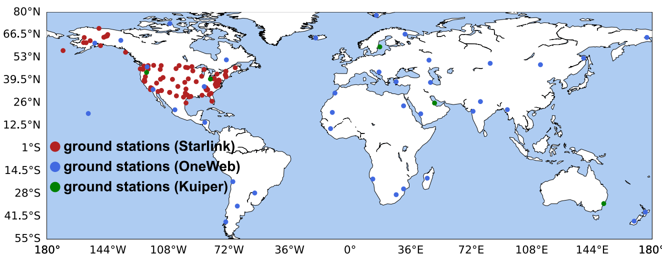


Fig. 4: Distribution of ground stations.

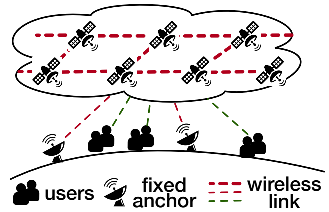
designed for *user mobility* and require fixed infrastructure as anchors to track mobile users' locations and network services. Instead, LEO satellite mega-constellations challenge this basic assumption: Some infrastructure nodes themselves (i.e., LEO satellites) are moving.

Therefore, most LEO networks today use ground stations as anchors and network gateways for terrestrial nodes. They track LEO satellites' movements, update the routing paths, and redirect all terrestrial nodes traffic from/to satellites. Terrestrial nodes associate to a ground station as gateways. LEO satellites do not directly forward traffic between terrestrial terminals. Instead, they redirect each terrestrial terminal's traffic to its serving gateway, which further forwards traffic to the destination via satellites. These tasks are complex due to the dynamic space-terrestrial relative motions. Moreover, they form new network QoS bottlenecks as we detail below.

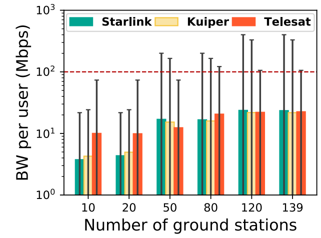
C. Impacts on Network Quality-of-Services (QoS)

The dynamic, complex satellite infrastructure mobility and frequent space-ground link churns cause new network QoS bottlenecks for LEO mega-constellations. We next empirically analyze LEO infrastructure mobility's impact on three QoS metrics: Network bandwidth, latency, and availability.

Impact on network bandwidth: As discussed in Section II-B, today's LEO networks mask satellite mobility by using ground stations as the fixed anchor. Each ground station is a carrier-grade NAT or gateway for terrestrial nodes. Satellites redirect all network traffic to remote ground stations for further processing. Since remote ground stations are fewer than LEO satellites in mega-constellations, they become the single-point bandwidth bottleneck. It has been reported that Starlink's ground stations have limited the LEO network's total capacity [12] due to limited space-terrestrial radio link capacity. The problem is exacerbated when LEO networks serve more global users. As shown in Fig 5, we generated hundreds of thousands of ground users based on [32] and limit each satellite to serve up to 2,000 users [2]. With 139 ground stations, their



(a) Ground stations as bottlenecks.



(b) Average bandwidth.

Fig. 5: Bandwidth bottlenecks from ground stations.

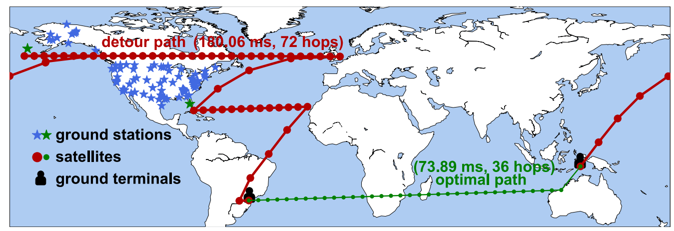
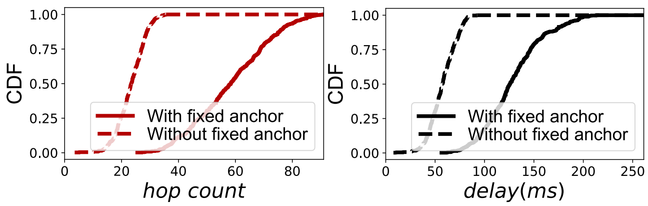


Fig. 6: A showcase of detour routing from ground stations.

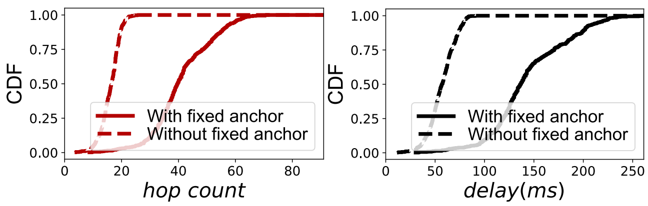
average bandwidth will be limited to 23.50 Mbps, 21.73 Mbps, 22.50 Mbps in Starlink, Kuiper, and Telesat due to the ground stations as the bottleneck. Because Iridium has only 66 satellites, the number of satellites is smaller than the number of ground stations, and the probability of ground stations becoming a bottleneck is reduced, so Iridium results are not shown here.

Impact on network latency: Binding network services to remote ground stations also causes triangular routing (thus detours and delays) at the global scale. As exemplified in Fig 6, terrestrial nodes' traffic should be first redirected to the remote ground stations (green stars), and then redirected to the destinations. In this case, the ground station is located in the U.S., while the sender and receiver are located in Africa and Asia. Such traffic detour causes 31,851 km detours and 106.17 ms extra delays. Fig 7 compares the hop counts ($|path|$, where *path* records the nodes (satellites or ground stations) included in the routing path from sender to receiver) and propagation delays ($\sum_{i=1}^{|path|-1} delay(path[i], path[i+1])$) between 1000 users with/without traffic detours by fixed ground stations. On average, the ground station causes 2.20$\times$ (2.71$\times$) and 2.40$\times$ (2.57$\times$) more delays (hop counts) in Starlink and Kuiper, respectively. Kuiper tends to have fewer hops than Starlink because the curve in Fig 7 (b) is steeper, but the trend of delay curves is similar.

Impact on network availability: To eliminate the above latency and bandwidth bottlenecks, a straightforward approach is to eliminate the ground stations as fixed anchors and gateways. However, the challenge is how to tackle the frequent space-ground link churns and topology changes. Without ground stations as anchors, frequent link churns will incur repetitive updates of routing paths. For state-of-the-art distributed routing, this implies repetitive global re-consensus on data paths (i.e., re-convergence). In distributed routing, satellites should exchange topology information, locally compute the routing tables, and achieve a global consensus on the routing paths.

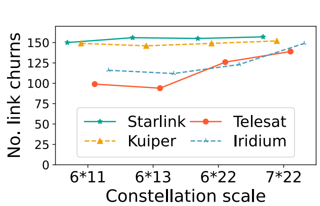


(a) Starlink Phase I (1584 satellites).

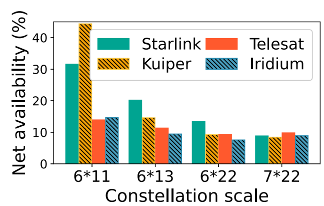


(b) Kuiper Phase I (1156 satellites).

Fig. 7: Detour latency and hop caused by fixed ground anchors.



(a) Number of link churns.



(b) Network availability.

Fig. 8: Low network availability in distributed LEO routing.

Before global routing convergence, there is no guaranteed network reachability. However, each fast-moving LEO satellite only covers terrestrial nodes for a short period (≤3 minutes in Starlink, ≤10 minutes in Iridium). Frequent topology updates could cause repetitive routing re-convergence and thus low network availability (i.e., $\frac{T_{up}}{T_{total}}$ where T_{up} is the period with global routing consensus, and T_{total} is the total network lifecycle). For intra-domain routing (e.g., OSPF, IS-IS), Fig 8 shows all complete mega-constellations would suffer from $\leq 10\%$ availability. The availability decreases with more satellites and ground stations. For inter-domain routing (e.g., BGP), [18], [25] show frequent logical topology changes cause BGP re-peering, thus sharpening the instability of global routing.

III. LBP: Removing QoS Bottlenecks from LEO Network Infrastructure Mobility

We design LBP to remove network QoS bottlenecks from LEO satellite infrastructure mobility. As shown in Section II, existing LEO networks suffer from bandwidth and latency bottlenecks from fixed terrestrial anchors (ground stations) due to space-terrestrial asymmetry. Therefore, LBP removes ground stations and shifts to fully distributed LEO satellite networks for shorter latencies and more bandwidths. The challenge, however, is how to retain high availability in frequent topology updates from LEO infrastructure mobility without fixed anchors.

To this end, LBP shifts from classic *logical* networking to *geographic, orbit direction-aware* addressing, routing, and handovers in LEO satellite networks. As shown in Fig. 9, LBP

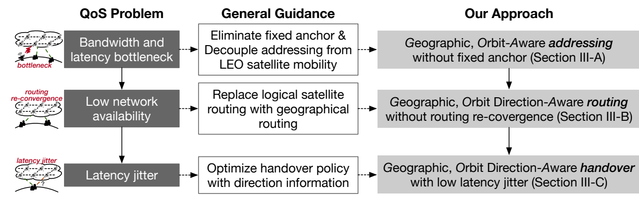


Fig. 9: Roadmap of LBP's QoS bottleneck elimination.

TABLE II: Notations and definitions.

Notation	Definition
N	Number of orbit planes
M	Number of satellites in each orbit
α	Inclination of orbit
(a_s, b_s)	satellite s's orbit number and number in orbit
$(\varphi_{s,t}, \lambda_{s,t})$	latitude and longitude of s's SSP
T_S, T_E	orbit period and earth rotation period
u_t	satellite phase angle from its ascending node at time t
$p_{s,t}$	s's plane ID (moving direction)
$r_{s,t}$	satellite s's geographical region ID in time t
r_u	user u's geographical region ID
L_r	latitudes of region boundaries

decouples addressing from LEO satellite mobility (Section III-A). It locates each terrestrial node by its geographic location rather than its serving satellite's logical interface or fixed anchors. With geographic information, LBP replaces logical satellite routing with geographic routing to find physically short paths without global routing updates/re-convergence (thus high availability). Unlike traditional geographic routing, LBP routing is aware of satellite orbital movement directions to guarantee reachability and mitigate detours (Section III-B). Its orbit direction-aware handover policy further shortens routing paths and mitigates delay jitters (Section III-C). We next detail each component.

A. Geographic, Orbit-Aware Addressing without Fixed Anchor

LBP departs from the LEO networks today by removing fixed anchors and latency/bandwidth bottlenecks from space-terrestrial asymmetry. The issue is how to locate terrestrial nodes' locations for data delivery in LEO infrastructure mobility *without* fixed anchors. The legacy *logical* addressing schemes (e.g., IP, switching labels, or cellular IDs) fail to do so since they rely on fixed anchors. They assign each user an address/ID based on its serving network node (LEO satellite in our context), which should be updated when switching to a new one. With each LEO satellite's extreme mobility in Section II-A and transient coverage (less than 3 minutes for each terrestrial node in Starlink), such logical addressing without fixed anchors would incur frequent address updates and disrupt users' network services.

Instead, LBP adopts *geographic* addressing to stabilize LEO networking in dynamic infrastructure mobility without fixed anchors. It divides the earth surface into disjoint geographic regions and assigns each terrestrial node an address based on its geographic region (rather than serving satellite's logical interface). Unless the terrestrial node moves to a new region (which is rare due to the region size as exemplified in Fig. 10b

Algorithm 1 Region boundaries computation.

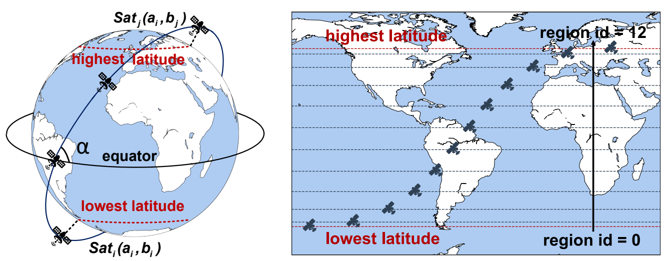
Input: orbit inclination α; number of satellites per orbit M;
Output: a list including latitudes of regional boundaries

L_r=$[-90]$;
for i = 0 to $\lceil \frac{M}{2}+1 \rceil - 1$ **do**
  L_r.append($arcsin\left(sin\alpha \cdot sin(-\frac{\pi}{2}+i\cdot\frac{2\pi}{M})\right)\cdot\frac{180}{\pi}$);
end for
L_r.append(90);
return L_r;

Algorithm 2 Geographic region ID computation.

Input: Satellite S's SSP=($\varphi_{s,t}$, $\lambda_{s,t}$) or User U's latitude $\varphi_{u,t}$;
Output: Geographic region ID of S or U

for i = 0 to $\lceil \frac{M}{2}+1 \rceil$ **do**
  if $L_r[i] \le \varphi_{s,t} < L_r[i+1]$ **then return i; end if**
end for



(a) 3D view. (b) 2D view.

Fig. 10: Satellite orbit projection. (a) 3D view. (b) 2D view. The surface of the earth is divided into 13 disjoint and ringlike geographic regions (for Starlink phase I [4]). The partition is static and each ground terminal can compute the region it within by Algorithm 2.

is huge), this address remains invariant regardless the serving satellite's mobility. Its geographic location helps LEO satellites deliver data through the physically short paths as we will detail soon.

A key step in LBP's addressing is to divide the earth surface into disjoint geographic regions. Unlike traditional geographic region divisions (e.g., by latitude/longitude, geohash, hexagon cells, or space-filling curves), LBP adopts a *satellite orbit-aware* approach to facilitate routing and handovers. Algorithm 1 shows this procedure and Fig 10b exemplifies it in Starlink. As shown in Fig 10, Sat_i's terrestrial projection (sub-point) latitude is around the lowest latitude that orbit can reach and Sat_j's sub-point latitude is around the highest latitude. LBP uses latitude circles passing ascending satellites in the same orbit with Sat_i and Sat_j to divide the earth surface into several disjoint regions. For a constellation with M satellites per orbit, the number of geographic region is $\lceil \frac{M}{2}+1 \rceil + 1$. As shown in Algorithm 2, LBP computes ground user's region ID based on its geographic latitude, and computes satellite's region ID based on satellite's runtime sub-point latitude in time t.

QoS analysis: LBP's geographic, orbit-aware addressing replaces fixed anchors to combat LEO infrastructure mobility. It avoids network bandwidth and latency bottlenecks in Section II-C and Fig. 5–7 from space-terrestrial asymmetry. We will quantify these improvements in Section V-B.

B. Orbit Direction-Aware Geographic Routing for High Availability

After eliminating the bandwidth and latency bottlenecks, the next step for LBP is to guarantee network traffic delivery in mobile LEO satellites *without* fixed anchors. Due to the frequent link churns by LEO satellite mobility (Section II-B), the space-terrestrial network topology unavoidably updates with frequent routing path changes. To cope with it, traditional logical network routing has to incur frequent global routing updates and lowers the network availability (Section II-C).

LBP seeks to retain high network service availability with *geographic* routing enabled by its addressing in Section III-A. Geographic routing is well-recognized for its efficiency and scalability. It takes the physically short path to forward the traffic. It is mostly performed locally without global routing updates and thus avoids network availability issues in Fig. 8b. However, geographic routing also suffers from data unreachability in 3D spaces due to the local minimum problem [14].

To this end, we design a domain-specific *orbit direction-aware geographic routing* to guarantee data reachability and high network availability, while still approximating the optimal shortest path routing. In the context of LEO satellite networks, establishing stable link between satellites moving in different direction is difficult due to their relative motions. In practice, there is usually no links between satellites moving in different direction even their physical distance is close. In inclined LEO satellite constellation, such as Starlink, satellites move in ascending plane and descending plane periodically and two planes are overlapping. Therefore, source and destination users may access into satellites moving in different direction (this is common while switching satellite based on distance). Considering a local minimum scenario that packets are routed to satellite S_i moving in different direction with destination satellite S_d. Even S_i is the closest satellite with S_d, it can not route packets to S_d because there is no link between them. To guarantee reachability for scenario like introduced above, LBP introduces direction information into routing, $p_{i,t}$ denote the plane that Sat_i moves in time t,

$$p_{i,t} = \begin{cases} 1, & u_t \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

The above scenario can be avoided by first routing packets along orbit that source satellite within via intra-orbit inter-satellite-links (ISL) until packets are routed to a satellite moving in same direction with destination satellite. This operation certainly can be guaranteed because satellites of each orbit nearly even distribute in two planes, packets route along the orbit definitely can reach a satellite moving in same direction with destination. Then packets can route via intra-orbit ISLs or inter-orbit ISLs, when packets arrive in destination region, there exists direct link or indirect links between current satellite and destination satellite, so local minimum scenario introduced above can be eliminated by introducing direction information. Fig 11a exemplifies LBP's orbit direction-aware geographic routing. Red satellite is the source satellite, green satellite is the destination satellite and they moves in different direction, packets firstly route via intra-orbit ISLs until reaching satellite moving in the same direction with destination. Then packets route via inter-orbit ISLs until reaching satellite locating in destination orbit. Finally, packets route via intra-orbit ISLs to destination satellite.

After introducing direction, LBP exploits orbit information to further guarantee reachability. To cover better, satellites' coverage areas are overlapped, which makes that more than one satellite and more than one orbit are visible for a user in anywhere. This overlapping causes routing ambiguity when packets arrive in destination region because all visible satellites of a user have the possibility to serve it. Finding the satellite serving destination user is the problem need to be solved. Paging is a straightforward way to locate user. Paging request messages are flooded between satellites and satellites receiving request will broadcast paging message to locate destination user. This way is straightforward, yet inefficient because the number of visible satellite for each user is not a few (more than 10 satellites can be visible for a Starlink user) . Bandwidth cost generated by paging will destroy network performance. LBP propose to use orbit ID to mitigate ambiguity without signaling cost. After routing packets to satellite moving in same direction with destination satellite, LBP then route packets via inter-orbit ISLs based on the comparison of current orbit ID and destination orbit ID. Inter-orbit ISL is relatively stable between satellites moving in same direction, so packets can be routed to destination orbit no doubt. Due satellite orbit is a closed circle, packets can definitely route to destination satellite by routing along the destination orbit via intra-orbit ISLs.

Algorithm 3 illustrates LBP's orbit direction-aware geographic satellite routing. It takes three major steps:

(1) Compare the moving direction of current satellite and destination satellite. If moving direction of current and destination satellite is same, goto step (2). Otherwise, current satellite will select up/down neighbor satellite as next hop and route packets to next hop via intra-orbit ISL. The selection is based on the comparison of region distance. Region distance includes *North* distance and *South* distance, it reflects the total deviation of current region ID and destination region ID relative to North Pole and South Pole. Intra-orbit neighbor which is nearer to the pole that has the shorter distance is selected as next hop.

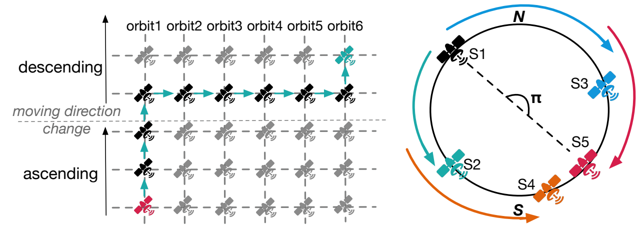
(2) Compare the orbit ID of current satellite and destination satellite. If current orbit is the destination orbit, goto step (3). Otherwise, current satellite select left/right neighbor satellite as next hop and route packets to next hop via inter-orbit ISL. The selection is based on the comparison of orbit number.

(3) Compare the region ID of current satellite and destination region. If current satellite's region ID is equal to the destination region ID, current satellite will route packets to destination user if it serves destination user, else it will select satellite that is advance in the moving direction (up neighbor) as the default next hop, unless the packets are routed from up neighbor. If packets are routed from up neighbor, the down neighbor is selected as next hop (avoid loop). If current region ID is not equal to the destination region ID, current satellite will select up/down neighbor satellite as next hop and route packets to next hop via intra-orbit ISL. The selection is based on the comparison of region ID.

Algorithm 3 Orbit Direction-Aware Geographic Routing.

Input: Current satellite Sat_s=(a_s, b_s, $p_{s,t}$, $r_{s,t}$) and destination D's $address$=(a_d, $p_{d,t}$, r_d)
Output: Sat_s's next hop for packets destined to D

$\Delta a = a_d - a_s$, $\Delta r = r_d - r_{s,t}$, $r_{max} = \lceil \frac{M}{2} + 1 \rceil + 1$;
if Sat_s is servicing D **then return**; **end if**
if $p_{s,t} = p_{d,t}$ **then**
  if $a_s = a_d$ **then** ▷ *Forward to intra neighbor*
    if $\Delta r = 0$ **then** next_hop=search(D's $address$);
    else if ($\Delta r > 0$ and $p_{s,t} = 1$) or ($\Delta r < 0$ and $p_{s,t} = 0$) **then** next_hop=$Sat_s.up$;
    else then next_hop=$Sat_s.down$; **end if**
  else ▷ *Forward to inter neighbor*
    if ($a_d > a_s$ and $\Delta a > \frac{N}{2}$) or ($a_d < a_s$ and $-\Delta a \leq \frac{N}{2}$) **then** next_hop=$Sat_s.left$;
    else then next_hop=$Sat_s.right$; **end if**
  end if
else ▷ *Forward to intra neighbor*
  $distance_n = 2 \cdot \lceil \frac{M}{2} + 1 \rceil - r_{s,t} - r_d$; $distance_s = r_{s,t} + r_d$;
  if ($distance_n = distance_s$) or ($distance_n < distance_s$ and $p_{s,t} = 1$) or ($distance_n > distance_s$ and $p_{s,t} = 0$) **then** next_hop=$Sat_s.up$;
  else then next_hop=$Sat_s.down$; **end if**
end if
return next_hop;



(a) A showcase of routing path by LBP. (b) Vertical projection.

Fig. 11: (a) A showcase of routing path in LBP. (b) The result of the projection of the intra-orbit routing path perpendicular to the orbital plane, and the arrows in different colors indicate the route direction to the satellite in the corresponding color.

QoS analysis: We next analyze how LBP's routing retains high network availability and approximates the theoretically optimal routing paths.

∘ *High network availability:* LBP solves the network availability issues in Section II-C for two reasons. Firstly, LBP computes routes locally without global routing updates and thus avoids network availability issues in Fig 8b. Secondly, LBP exploits direction information to avoid local minimum problem caused by link absence, and then LBP further introduce orbit ID to mitigate ambiguity caused by overlapping coverage. Orbit direction-awareness make LBP can retain ≈100% network availability.

○ *Closeness to optimal routing:* Besides high network availability, LBP approximates the theoretically optimal routing paths. We prove that the hop count difference of LBP and the optimal is less than $M - H_{intra}$, where H_{intra} and H_{inter} denote the number of intra-orbit ISLs and inter-orbit ISLs in the optimal path respectively. path$_{OPT}$ and path$_{LBP}$ denote the routing path computed by shortest path algorithm and Algorithm 3.

Theorem 1. $|path_{LBP}| - |path_{OPT}| \leq M - 2 \cdot H_{intra}$

Proof. Firstly, $|\text{path}_{OPT}| = H_{inter} + H_{intra}$. As shown in Algorithm 3 line 9-10, LBP selects inter-orbit neighbor in the left/right direction with minimal difference of orbit ID, that is same with the optimal path, thus $H_{inter,LBP} = H_{inter}$. Now we need to proof that $H_{intra,LBP} - H_{intra} \leq M - 2 \cdot H_{intra}$, that is $H_{intra,LBP} \leq M - H_{intra}$. Fig 11b shows the selection of intra-orbit ISLs from black satellite $S1$, satellite $S2, S3, S4, S5$ denote access satellites of destination users. Without loss of generality, we assume that destination user's region is as same as its access satellite ($r_{S*,t} = r_d$). Intra-orbit ISL selection can be classified into two categories based on whether current satellite moves in the same direction with destination satellite. *(1) same direction: ($S1$, $S2$).* (a) If $S1$ and $S2$ within same orbit, $S1$ selects intra-orbit neighbor nearer destination region. Due to satellite orbit is a closed cycle with M hops, it is apparent that $H_{intra,LBP} = H_{intra}$ (best case) or $H_{intra,LBP} = M - H_{intra}$ (worst case), and $H_{intra} \leq M - H_{intra}$. Therefore, we got $H_{intra,LBP} \leq M - H_{intra}$. (b) If they in different orbit, LBP route packets to satellite in the same orbit with $S2$ after H_{inter} hops. Then same with (a), we got $H_{intra,LBP} \leq M - H_{intra}$. *(2) different direction: ($S1$, $S3(S4, S5)$).* Packets firstly route via intra-orbit ISLs to satellite moving in same direction with $S3(S4, S5)$ (line 13-15). Because the orbit is a closed cycle, it is obvious that this selection can guarantee minimal hop count, that $H_{intra,LBP} = H_{intra} \leq M - H_{intra}$. To sum up, $|\text{path}_{LBP}| - |\text{path}_{OPT}| \leq M - 2 \cdot H_{intra}$.

□

C. Orbit Direction-Aware Handover for Latency Optimization

Besides solving all issues in Section II-C, LBP's orbit-aware geographic approach can further help optimize LEO satellite handover strategies for shorter routing paths and fewer delay jitters. A serving satellite should migrate its terrestrial users to a new satellite when they are out of its coverage. For a LEO satellite mega-constellation, multiple candidate satellites are available as the target. The analysis in [11] shows that the selection of access satellite has a significant influence on the hop count of the path. A straightforward satellite handover strategy chooses the nearest satellite (i.e., the satellite with maximal elevation) and accesses to it until it is invisible [13]. However, this strategy may cause detours between satellites in the opposite moving directions due to the lack of inter-satellite links (Section III-B).

To this end, LBP suggests selecting the target satellites in ascending plane. Unless there is no ascending satellite can be visible, users select satellites in descending plane to guarantee network access. We devise a handover policy to further reducing latency jitter after optimizing latency. Latency jitter here is defined as the variance of latency over the entire period. This policy includes two key points.

(1) (Initial access) Communication users had better connect to the satellites moving in the same direction. Based on the proof of Theorem 1, connecting to satellites moving in the same direction can experience nearly optimal latency. Therefore, LBP suggests selecting the target satellite in ascending plane. LBP devise a 0/1 identifier to label if a user can be covered by ascending plane. This identifier helps satellite deciding accept/reject user's access request.

(2) (Switching) Users had better switch to the satellite within the same orbit and moves in the same direction with old service satellite. Switching to this kind of satellite can optimize latency jitter, because the orbit ID difference of source and destination user's service satellite can keep relatively stable. Thus inter-orbit ISL hop count can keep relatively stable accordingly. Because the rare change of region ID, intra-orbit path latency rarely changes.

QoS analysis: LBP's orbit direction-aware handover policy can help optimizing latency by accessing into satellites moving in the same direction, and reducing latency jitter via stabilizing inter/intra orbit hop counts.

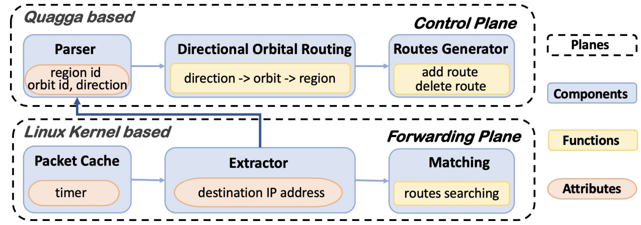
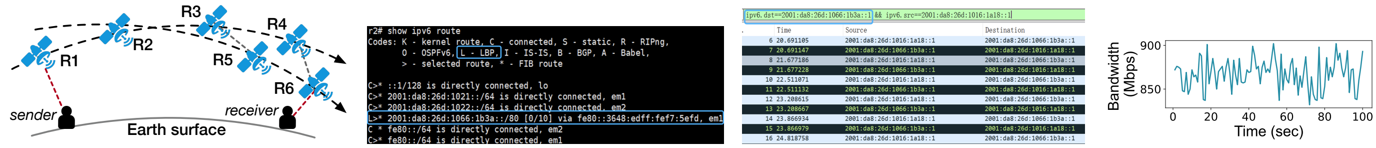


Fig. 12: IPv6-based LBP implementation.

IV. IMPLEMENTATION

We showcase how to implement LBP in LEO satellite networks with an IPv6 router prototype. Note LBP can also be realized with other choices such as layer-2 label switching or cellular networks. We choose IPv6 since it has been the *de facto* standard in terrestrial Internet, experimentally tested by Cisco in satellites [31], and used by Telesat LEO satellite constellation [5]. Fig. 12 illustrates our IPv6 satellite router prototype based on Quagga [3] in a customized Linux kernel. At the control plane, we package LBP as a Quagga protocol daemon lbpd (similar to ospf6d and bgp6d in Quagga). At the data plane, LBP reuses the Linux kernel's prefix matching-based IPv6 packet forwarding based on routing decisions from the control plane. We next elaborate on how to implement LBP's addressing, routing, and handovers in this prototype.

• **Orbit-aware geographic IPv6 addressing:** We renovate IPv6 addressing semantics with LBP's geographic, orbit-aware locator (Section III-A). Each terrestrial node's IPv6 address is a concatenation of the ISP's network prefix, the terrestrial node's geographic region in Fig 10b, and a unique suffix within



(a) Small-scale testbed setup. (b) LBP's routing table in R2. (c) Packets received by the receiver. (d) Throughput in 1Gbps links.

Fig. 13: Testbed setup, routing reachability and throughput measurement results in our IPv6-based LBP implementation.

this region. This geographic IPv6 address is decoupled from the destination's serving satellite and thus is stable regardless of LEO mobility. To support directional orbital routing in Section III-B, we further embed the terrestrial node's serving satellite's moving direction into the IPv6 hop-by-hop optional header. In this way, the serving satellite's time-varying moving direction will not affect the stability of the IPv6 address.

• **Orbit direction-aware geographic routing:** Upon receiving a new IPv6 packet, our prototype first checks if its address matches any FIB entries. If yes, it forwards this packet based on the FIB entry with the longest prefix match and hop-by-hop optional header's orbit direction information. Otherwise, LBP temporally caches this packet (timer is configurable) and duplicates its IPv6 header to our Quagga-based control plane. The control plane parses the geographic region (from destination IPv6 address) and orbit information (from hop-by-hop optional header), runs Algorithm 3 in Section III-B to compute the next hop, and installs this new routing entry to the data-plane FIB. Then this cached packet can be forwarded based on this new FIB entry.

Each FIB entry is associated with a liveness timer so that it can be updated periodically in LEO satellite mobility, LBP defines this *temporary FIB entries* as *temporary route tables*, that can avoid route computation with per packet and support satellite mobility. Only the first few table mismatching packets will trigger route computation and be cached, the rest packets having same destination address can be routed according to the matched FIB entry directly. This way can further support high performant inter-satellite routing, because it can reduce computation cost and highly efficient table matching can better support QoS, reducing the probability of single satellite bottleneck. Moreover, temporary route tables can be computed in advance according to the prediction of satellite direction, orbit and geographic region information. Different from methods that need to predict precise geographical location of satellites, LBP only need to predict simple moving direction and coarse geographic region information.

• **Orbit direction-aware handover:** Each terrestrial node follows Section III-C to select its serving satellite. Our testbed uses a centralized controller to emulate this handover process. It computes each satellite and terrestrial node's locations, follows Section III-C to decide each terrestrial node's serving satellite, and reconfigures the connectivity between nodes.

V. EVALUATION

We validate LBP's functionality in our prototype, and quantify LBP's QoS merits with large-scale trace-driven emulations based on real LEO mega-constellations and ground stations.

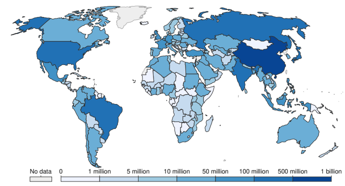


Fig. 14: Global Internet user distribution in our tests [32].

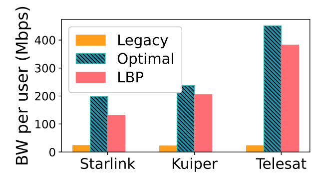


Fig. 15: LBP's improvement on network bandwidth.

A. Small-Scale Prototype Validation of LBP's Functionality

We first test LBP in a controlled testbed in Fig. 13a. This testbed consists of six DELL R740 servers (each emulating an LEO satellite and running our IPv6-based LBP prototype in Section IV). There are two orbits, one with satellites R1–R4 and the other with satellites R5–R6. We set the moving direction of R1 and R2 is ascending, which means that they are moving nearing the north pole. The moving direction of the rest is set as descending direction. Each inter-satellite link has 1 Gbps total capacity. All nodes are connected by a central controller, which tracks each LEO satellite's locations and reconfigure their connectivity upon topology changes.

In this experiment, we test LBP's basic network traffic delivery under LEO satellite mobility without fixed anchors (i.e., QoS bottlenecks). We randomly choose the source-destination pairs in Fig. 13a and validate if the destination can receive all packets based on LBP's orbit-aware geographic routing in Section III-B. Fig. 13b and Fig. 13c confirm LBP's routing delivers data in presence of LEO satellite mobility, without reliance on terrestrial ground stations. Fig. 13d shows that LBP's routing almost saturates the total link capacity since its data forwarding engine follows the lightweight IPv6 longest prefix matching-based forwarding, thus feasible for resource-constrained LEO satellites.

B. Large-Scale Emulations on LBP's QoS Improvements

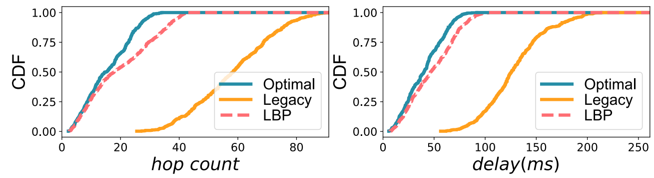
We next evaluate LBP's QoS with large-scale trace-driven emulations in LEO mega-constellations in Table I and ground stations in Fig. 4. We choose 1,000 terrestrial users based on the distribution of global Internet users from World Bank [32] (Fig. 14), randomly generate source-destination pairs between them, and route their traffic in a 24-hour period, and compute each pair's network latency, bandwidth, and availability. Under the same experimental setting, we compare LBP with two solutions: (1) **Legacy LEO networking:** It uses ground stations as fixed anchors. As shown in Section II-C, this solution suffers from network latency and bandwidth bottlenecks from space-terrestrial asymmetry; (2) **Optimal LEO networking:** This is an ideal solution. It eliminates ground stations, has an oracle with global knowledge to compute the shortest paths, and synchronously updates all satellites' routing upon topology changes. Clearly, it cannot be realized in practice. We use it to assess LBP's closeness to optimal routing.

Network bandwidth improvement: Fig. 15 compares the average user bandwidth of LBP, the legacy LEO networking with fixed anchors, and the ideal optimal networking. We have two observations. First, by removing the ground stations and thus space-terrestrial asymmetry, LBP increases 107.83 Mbps (4.59×), 182.89 Mbps (8.42×), and 359.49 Mbps (15.98×) average bandwidth in Starlink, Kuiper, and Telesat compared to the legacy solution, respectively. Second, LBP's average network bandwidth is close to the oracle optimal solutions with 0.28× differences.

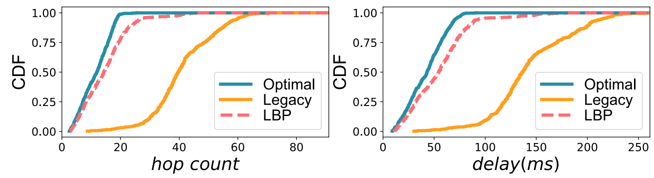
Network propagation latency reduction: Fig. 16 and 17 compares the routing propagation latency in LBP, the legacy LEO networking with fixed anchors, and the ideal optimal networking. Compared with the legacy LEO networking, LBP eliminates the fixed anchors and thus routing detours. This reduces 80.90 ms (1.68×) and 88.63 ms (1.60×) average latency in Starlink and Kuiper, respectively. It also reduces 38.55 (1.93×) and 26.35 (1.77×) average routing hop counts in Starlink and Kuiper, respectively. Moreover, LBP's orbit-aware geographic routing also approximates the ideal shortest paths with ≤7.87 ms (≤13.03 ms) and ≤4.22 (≤3.33) differences in propagation delays and hop counts in Starlink (Kuiper). This confirms our theoretical analysis in Theorem 1.

Network delay jitter reduction: As explained in Section III-C, LBP's orbit direction-aware handovers help shorten routing paths and mitigate the propagation latency jitters. Fig. 18 compares the variance of routing latency (defined as $\frac{\sum_{t=1}^{T}(latency-mean\ latency)^2}{T}$) in the 24-hour period in LBP's orbit-aware handover policies and the orbit-oblivious handovers by choosing the physically closest serving satellite. Compared to orbit direction-oblivious handovers, LBP reduces the hop count variance by 93.12% (36.11%), and reduces the propagation delay variance by 52.11% (2.1%) in Starlink (Kuiper).

High network availability: Without ground stations as fixed anchors, LBP still retains high network availability and approximates the optimal oracle LEO networking. Fig. 19a compares the network availability (i.e., $\frac{T_{up}}{T_{total}}$ where T_{up} is the period with global routing consensus, and T_{total} is the total network lifecycle)) between LBP and the legacy distributed routing (OSPF). As explained in Section II-C, the legacy distributed routing suffers from low network availability due to repetitive routing re-convergence. Instead, LBP retains 100% network availability since it adopts local geographic routing without global routing updates. Moreover, unlike traditional geographic routing that suffers from local minimum (thus unreachability), LBP's orbit direction awareness also guarantees routing reachability between any source-destination pairs (shown as Fig. 19b).

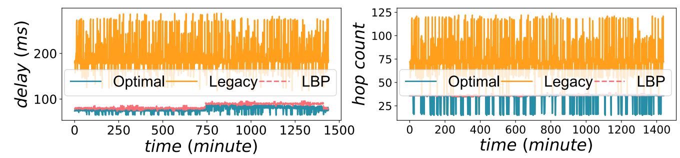


(a) Starlink Phase I (1584 satellites).

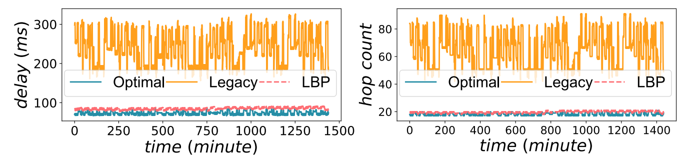


(b) Kuiper Phase I (1156 satellites).

Fig. 16: Improvement on network propagation latency.



(a) Starlink Phase I (1584 satellites).



(b) Kuiper Phase I (1156 satellites).

Fig. 17: A showcase of improvement on propagation latency (Users' location is shown in Figure 6).

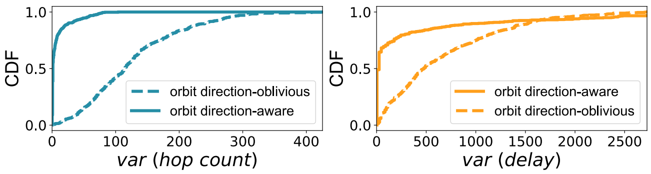
VI. Related Work

In recent years, LEO networks have attracted academia and industry's interests due to the rapid launches of satellite mega-constellations. Recent studies have explored diverse aspects of the LEO networking, including the topology design [10], routing [15], [16], congestion control [30], [33], and emerging applications like navigation [26], in-orbit computing [9] and content delivery [18]. Our work complements them by systematically studying the QoS bottlenecks in LEO networks.

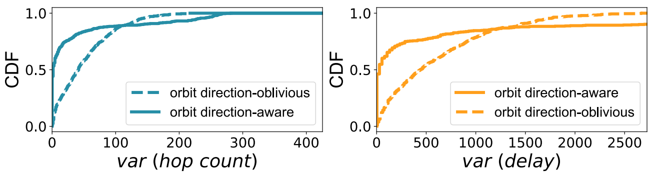
For QoS optimizations in satellite networks, most research efforts are based on LEO networks with ground stations as fixed anchors. They showcase LEO networks' potentials for low-latency routing [8], [25], enhance satellite routing with ground stations' assistances [1], [17], [20], improve bandwidth with multi-path satellite routing and congestion control [23], [29], to name a few. Instead, this paper shows ground stations (anchors) have become QoS bottlenecks in LEO infrastructure mobility. We devise LBP to remove these bottlenecks with orbit-aware geographic network addressing, routing, and handover. LBP differs from traditional geographic routing [21], [24] since it is orbit-aware to guarantee reachability, shorten routing paths, and mitigate routing latency jitters.

VII. Conclusion

This paper studies the QoS bottlenecks from infrastructure mobility in low-earth-orbit networks. It shows the classic

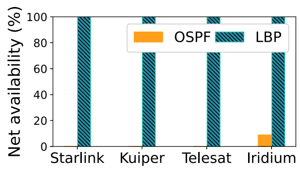


(a) Starlink Phase I (1584 satellites).

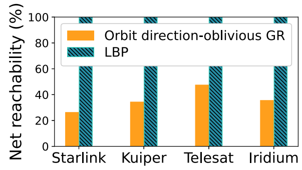


(b) Kuiper Phase I (1156 satellites).

Fig. 18: LBP's propagation latency jitter.



(a) Network availability.



(b) Routing reachability.

Fig. 19: Improvement on network availability & reachability.

fixed anchor-based mobility solutions have caused bandwidth and latency bottlenecks due to space-terrestrial asymmetry in today's LEO networks. The root cause is that, these solutions are designed for *user mobility* rather than *infrastructure mobility*. We thus propose LBP to remove global network QoS bottlenecks without lowering the network availability. LBP exploits geographic and orbital information to eliminate terrestrial fixed anchors for low latencies and high bandwidth. It decouples network addressing from the serving satellites' logical interfaces, routes terrestrial traffic based on orbit direction-aware geographic routing with high network availability, and optimizes its handovers with orbital directions for shorter paths and fewer jitters. Our IPv6-based prototype and trace-driven emulations demonstrate LBP's feasibility and benefits of latency reduction, bandwidth improvements, and high network availability.

Acknowledgment

This work was supported by National Key Research and Development Plan of China (2018YFB1800301). Thanks to China Mobile and Zhongguancun National Laboratory for their support to this work. Hewu Li and Yuanjie Li are corresponding authors.

References

[1] Amazon Kuiper satellite constellation. https://tinyurl.com/bs7syjnk.
[2] How many internet users can one SpaceX Starlink satellite support at once? https://www.quora.com/How-many-internet-users-can-one-SpaceX-Starlink-satellite-support-at-once/, 2021.
[3] Quagga software routing. https://www.quagga.net/, 2021.
[4] SpaceX Starlink. https://www.starlink.com/, 2021.
[5] Telesat LEO satellites. https://www.telesat.com/leo-satellites/, 2021.
[6] 3GPP. TS25.301: Radio Interface Protocol Architecture, 2008.
[7] 3GPP. TS36.331: 5G NR: Radio Resource Control (RRC), Mar. 2021.
[8] D. Bhattacherjee, W. Aqeel, I. N. Bozkurt, A. Aguirre, B. Chandrasekaran, P. B. Godfrey, G. Laughlin, B. Maggs, and A. Singla. Gearing up for the 21st century space race. In *ACM HotNets*, 2018.
[9] D. Bhattacherjee, S. Kassing, M. Licciardello, and A. Singla. In-orbit computing: An outlandish thought experiment? In *ACM HotNets*, 2020.
[10] D. Bhattacherjee and A. Singla. Network Topology Design at 27,000 km/hour. In *ACM CoNEXT*, 2019.
[11] Q. Chen, G. Giambene, L. Yang, C. Fan, and X. Chen. Analysis of inter-satellite link paths for leo mega-constellation networks. *IEEE Transactions on Vehicular Technology*, 70(3):2743–2755, 2021.
[12] I. Del Portillo, B. G. Cameron, and E. F. Crawley. A Technical Comparison of Three Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband. *Acta Astronautica*, 2019.
[13] A. Donner, M. Berioli, and M. Werner. Mpls-based satellite constellation networks. *IEEE JSAC*, 22(3):438–448, 2004.
[14] S. Durocher, D. Kirkpatrick, and L. Narayanan. On routing with guaranteed delivery in three-dimensional ad hoc wireless networks. In *International Conference on Distributed Computing and Networking*, pages 546–557. Springer, 2008.
[15] E. Ekici, I. F. Akyildiz, and M. D. Bender. Datagram routing algorithm for leo satellite networks. In *Proceedings IEEE INFOCOM 2000.*, volume 2, pages 500–508. IEEE, 2000.
[16] E. Ekici, I. F. Akyildiz, and M. D. Bender. A distributed routing algorithm for datagram traffic in leo satellite networks. *IEEE/ACM Transactions on networking*, 9(2):137–147, 2001.
[17] FCC. Petition of Starlink Services, LLC for Designation as an Eligible Telecommunication Carrier. https://tinyurl.com/ury6rzw5, 2021.
[18] G. Giuliari, T. Klenze, M. Legner, D. Basin, A. Perrig, and A. Singla. Internet backbones in space. *SIGCOMM CCR*, 2020.
[19] M. Handley. Delay is not an option: Low latency routing in space. In *ACM HotNets*, pages 85–91, 2018.
[20] M. Handley. Using ground relays for low-latency wide-area routing in megaconstellations. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, pages 125–132, 2019.
[21] T. R. Henderson and R. H. Katz. On distributed, geographic-based packet routing for leo satellite networks. In *Globecom'00*, volume 2, pages 1119–1123. IEEE, 2000.
[22] ITU Publications. Measuring digital development: Facts and figures 2021. https://www.itu.int/en/ITU-D/Statistics/Documents/facts/FactsFigures2021.pdf, 2021.
[23] Z. Jiang, Q. Wu, H. Li, and J. Wu. Scmptcp: Sdn cooperated multipath transfer for satellite network with load awareness. *IEEE Access*, 6:19823–19832, 2018.
[24] B. Karp and H.-T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *ACM MobiCom*, pages 243–254, 2000.
[25] T. Klenze, G. Giuliari, C. Pappas, A. Perrig, and D. Basin. Networking in heaven as on earth. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, pages 22–28, 2018.
[26] S. Narayana, R. V. Prasad, V. Rao, L. Mottola, and T. V. Prabhakar. Hummingbird: energy efficient gps receiver for small satellites. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.
[27] E. S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil. Proxy Mobile IPv6, 2008. RFC 5213.
[28] H. Soliman, C. Castelluccia, K. ElMalki, and L. Bellier. Hierarchical Mobile IPv6 Mobility Management, 1999. RFC 5380.
[29] Z. Song, Q. Wu, H. Li, J. Liu, and Z. Lai. Hysteresis optimized multipath tcp data scheduling algorithm in predictable networks. In *2021 IEEE IWCMC*, pages 1381–1386. IEEE, 2021.
[30] Y. Wang, K. Zhao, W. Li, J. Fraire, Z. Sun, and Y. Fang. Performance evaluation of quic with bbr in satellite internet. In *2018 6th IEEE WiSEE*, pages 195–199. IEEE, 2018.
[31] L. Wood, W. Ivancic, D. Stewart, J. Northam, C. Jackson, and A. da Silva Curiel. Ipv6 and ipsec on a satellite in space. In *58th International Astronautical Congress*, 2007.
[32] World Bank. Number of Internet Users by Country. https://tinyurl.com/3u8rn4um, 2017.
[33] H. Zhang, T. Wang, Y. Tu, K. Zhao, and W. Li. How quick is quic in satellite networks. In *International Conference in Communications, Signal Processing, and Systems*, pages 387–394. Springer, 2017.