What Goes Around Comes Around: Predictive Ground Station Selection For LEO Constellations

Fan-Xue Gabriella Reidys
University of Illinois Urbana-Champaign

Mubashir Anwar
University of Illinois Urbana-Champaign

Ragini Gupta
University of Illinois Urbana-Champaign

Aryan Taneja
University of Illinois Urbana-Champaign

Deepak Vasisht
University of Illinois Urbana-Champaign

Matthew Caesar
University of Illinois Urbana-Champaign

Abstract

LEO satellite networks have grown into critical communications infrastructure serving rapidly increasing volumes of Internet traffic. Selecting which ground station (GS) each satellite uses has thus become a key traffic engineering problem. Existing systems decide using present-time conditions, with each satellite picking the best GS currently in view and holding it for the connection window. This overloads the few stations near densely populated regions, and choices made at decision time can fail as demand shifts within the window. We present INSITE, which addresses these problems by combining two complementary ideas: planning satellite-GS assignments jointly across the constellation and planning against demand predicted over each window rather than demand observed at decision time. In preliminary results, INSITE reduces the load imbalance factor by 51%, overflow traffic detoured to distant stations by 74%, and packet loss under application latency budgets by over 3×.

CCS Concepts

• **Networks** → **Traffic engineering algorithms**; **Network management**.

Keywords

Satellite Networking, Ground Station Selection, Traffic Prediction, Traffic Engineering

ACM Reference Format:
Fan-Xue Gabriella Reidys, Mubashir Anwar, Ragini Gupta, Aryan Taneja, Deepak Vasisht, and Matthew Caesar. 2026. What Goes Around Comes Around: Predictive Ground Station Selection For LEO Constellations. In *The 25th ACM Workshop on Hot Topics in Networks (HotNets '26), November 16–17, 2026, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3847290.3859943



This work is licensed under a Creative Commons Attribution 4.0 International License.
HotNets '26, Salt Lake City, UT, USA
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-3029-0/2026/11
https://doi.org/10.1145/3847290.3859943

1 Introduction

Satellite networks are becoming a core part of modern communications infrastructure. Over the past two years, Starlink's subscriber count has doubled to over 12 million [7], and the network increasingly supports critical and latency-sensitive applications [41, 44, 47, 51]. As these networks carry more traffic, they run into congestion. The root cause of this congestion is structural: user demand follows the highly uneven geography of population, concentrating traffic wherever people cluster. This is already visible operationally. Since 2024, Starlink has levied one-time, demand-based surcharges of up to $1,500 in dense regions [16]. Operators are responding with capacity expansion, proposing to deploy another 100,000 satellites [5] and upgrade ground stations (GSs) [22]. However, adding capacity does not by itself resolve congestion because GSs remain expensive and difficult to site in dense regions, making GS selection a core traffic-engineering (TE) problem for these networks.

GS selection in current LEO constellations is driven by present-time conditions. Historically, a satellite offloaded to the nearest GS in its footprint, and traffic was then tunneled through the user's nearest point of presence [19, 28]. Recent measurements of Starlink indicate a more adaptive variant: selection also considers link availability, reliability, and capacity, so traffic is sometimes routed to a farther station based on the observed conditions [19].

This design suffers from two drawbacks. First, selection is greedy in space. Current routing logic prioritizes proximity to minimize latency. Because user demand is highly concentrated, multiple satellites simultaneously converge on the handful of GSs nearest to population centers. In our trace-driven analysis (§2), the busiest 25% of GSs carry 50% of the traffic, and the imbalance persists over time. Second, selection ignores future demand. A satellite's offered load keeps shifting as it sweeps across regions of varying population,

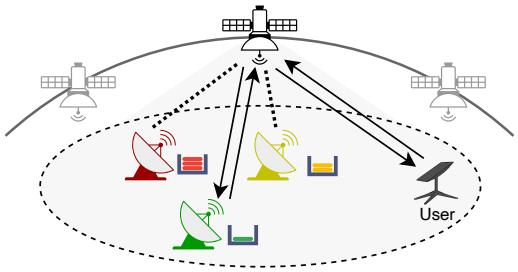


Figure 1: Packet traversal in a LEO satellite network.

yet each station assignment must hold for some interval, from 15 seconds up to several minutes [39]. Demand observed at decision time does not represent demand over that window: 71.3% of satellites see their load shift by more than 50% within a 3-minute window (Figure 4).

Therefore, we propose INSITE, a predictive, jointly planned GS selection framework. To avoid greedy selection, INSITE assigns each satellite to a station while accounting for the load that all other satellites will place on that station. To incorporate future demand, it plans using the predicted load of every satellite over the upcoming connection window rather than the load observed at commit time. Thus, INSITE can balance load across GSs, minimizing inter-satellite link (ISL) usage and reducing congestion-induced queueing. Achieving this requires solving three challenges:

The demand a satellite will serve is hard to predict from its own history. By the time a satellite must choose a station, it has moved over new geography that its recent traffic does not reflect. INSITE builds on a key observation: although a satellite's demand appears to fluctuate arbitrarily, it is *spiral-cyclostationary*—Earth's rotation shifts each successive orbit from the last, so coverage spirals across the globe rather than repeating a fixed path. Because all satellites follow this structure, INSITE predicts a region's demand from these accumulated cross-satellite observations, organized as a location-indexed demand map, rather than from any single satellite's history.

No single demand value safely represents the window. A satellite's demand is a trajectory that changes over the connection window, yet the assignment is a single choice that must hold throughout. Decide against the current demand, and the choice fails when demand climbs minutes later. Decide against the worst case, and satellites refuse stations that could have served them, pushing load elsewhere and recreating the very imbalance we set out to remove. INSITE selects against a representative demand value chosen so the assignment holds across the entire window.

Perfect predictions do not make choices independent. Knowing every satellite's demand does not by itself prevent conflict, because choices interact. The station that looks best to one satellite may be exactly the capacity a nearby satellite needs moments later. Let each satellite optimize for itself against perfect predictions, and their individually sensible choices still collide at shared stations. INSITE therefore plans assignments jointly, selecting each satellite's station against the predicted load of all satellites contending for the same stations over the window.

We evaluate INSITE using trace-driven simulations and real-world demand data. We simulate satellite orbits and GSs using CosmicBeats [33], modeling 8,137 satellites and 64 GSs across the continental US. In preliminary results, INSITE reduces load imbalance by 51%, overflow traffic that would otherwise be forced onto detours to distant stations by 74%, and packet loss under application latency budgets by over 3×.

2 Motivation

2.1 End-to-End Satellite Network

Traditional satellite networks rely on geostationary satellites that relay signals through fixed, pre-configured links to ground gateways, carrying traffic very different from that of the broader Internet. Over the last decade, megaconstellations of thousands of LEO satellites have introduced new challenges of scale and extreme mobility. Each satellite remains within view of a given user or GS for only a few minutes, moving at roughly 7.5 km/s, far faster than any user a terrestrial mobile network must track. Yet this mobility is more predictable than that of mobile users, since satellites follow known orbits. What changes drastically is the demand a satellite serves, which can shift within minutes as it sweeps from dense to sparse regions.

Figure 1 shows how traffic traverses a LEO satellite network. A user terminal connects to an overhead satellite, which relays traffic down to a GS; from there it is tunneled to a Point of Presence (PoP) and enters the terrestrial Internet [19, 28]. Traffic bound for the user takes the same path in reverse, entering at a PoP and being forwarded to the GS that can reach the user's serving satellite. This user-bound *forward* direction carries most of the volume, since Internet traffic is download-dominant [18], and it is the direction whose demand we predict. Because forward and return traffic traverse the same satellite-GS contacts and face the same choice of GS, a single selection algorithm can serve both directions. A satellite's GS thus determines where its traffic enters and exits. When a nearby GS cannot absorb the load, the fallback is to carry traffic over ISLs to a distant station, but ISL capacity is scarce and the detour can incur substantial latency [19]. Assignments also should not be re-decided continuously, since each reassignment disrupts traffic in flight [28]. We call the interval a satellite remains assigned to a GS the *connection window*, bounded by the

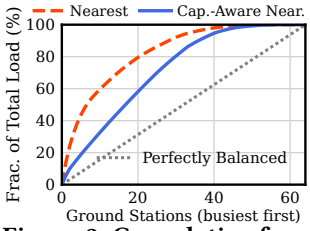


Figure 2: Cumulative fraction of total load across 64 GSs, busiest first.

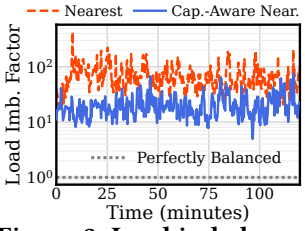


Figure 3: Load imbalance factor (busiest divided by 10th-percentile load).

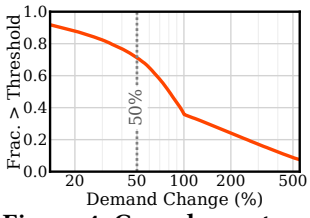


Figure 4: Complementary CDF of demand change over 3-minute windows.

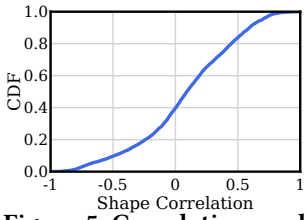


Figure 5: Correlation analysis of successive satellite orbits.

few minutes a satellite stays within a station's 25° elevation view [39].

Three recent shifts motivate revisiting GS selection. First, ground segments now offer a choice. Starlink's US gateway network has grown from six gateway applications in 2019 [35] to over 100 sites by 2025 [37], and even the 64 stations in our simulation leave a satellite a median of 6 stations in view. Second, constellations are now dense enough that every location is observed frequently, enabling prediction. In our September 2025 snapshot of 8,137 satellites, a populated location is revisited every 42 s at the median (Fig. 6). The constellation has since grown past 10,000 satellites [34], so revisits only become more frequent. Third, the demand keeps growing. Starlink grew from 1 million subscribers in December 2022 to over 12 million by 2026 [7]. Because users cluster around population centers, this growth concentrates on the GSs that serve them.

2.2 Failure Modes of Current GS Selection

We quantify two failure modes using a simulation of 8,137 satellites and 64 GSs, with each satellite's offered demand derived from population density (see §4). Deployed operators do not disclose their selection logic, but measured behavior is consistent with a capacity-aware, proximity-first policy [19, 28]: traffic goes to the closest station with spare capacity, spilling to the next closest otherwise. All variants of this class select against presently observed conditions, reacting to current demand rather than future demand.

Greedy in space. Figure 2 plots the cumulative share of offered load under two present-time selection policies over a two-hour trace. Under *Nearest*, where each satellite offloads to the closest visible station, the top 7 stations absorb 50% of all traffic, while ~25% of the stations carry essentially no load. This is either because they are far from concentrated demand or because they are rarely the nearest one. Making satellites capacity-aware offers only a minor improvement. Under *Capacity-Aware Nearest*, where each satellite picks the closest station whose remaining capacity can cover its demand, the top 17 stations still carry 50% of load.

Figure 3 shows this skew is not transient. Across the trace, even the capacity-aware policy holds a median load imbalance factor, the busiest GS's load divided by the 10th-percentile GS's, of 18.3. This concentration indicates the achievable ground-segment capacity is determined by a few hot GSs rather than aggregate capacity. It also shows why capacity expansion alone is unlikely to help. New capacity deployed uniformly inherits the skew, as demand grows with the subscriber base it attracts, so the imbalance persists at scale.

Blind in time. Even a well-chosen station becomes suboptimal within a single connection window. Figure 4 plots the fraction of windows whose demand changes by more than a given amount within 3 minutes. 71.3% of windows see offered load shift by more than 50%. A lightly loaded satellite offloads to the station with the most spare capacity in view. Moments later, another satellite sweeps the same region under heavy load, and with that capacity partly consumed, no station in view can absorb it, forcing an ISL detour to a distant station. The first choice was capacity-aware at commit time, yet still created congestion minutes later, because nothing in it accounted for the demand about to arrive.

2.3 Spiral-Cyclostationary Demand

A satellite's orbit fixes its ground track, and Earth's rotation shifts that track westward by roughly 22° per orbit, so coverage spirals around the globe rather than retracing a fixed path. The demand at every location beneath it follows a recurring daily pattern. Each pass sweeps geography the last pass did not see, so the demand a satellite experiences does not reflect these typical patterns. Figure 5 shows that successive orbits are barely correlated (Pearson correlation [46]).

This demand, however, can be reconstructed from the constellation's view. Demand observed at nearby points is similar, differing by less than 10% at the median within roughly 0.5° (~50 km). As shown in Figure 6, some satellite passes within that radius of a given populated area every 42 seconds at the median, generating frequent demand observations. Pooling these observations at a fixed location over an eight-hour trace recovers a smooth signal (Figure 7) with the same diurnal, recurring shape that real cellular demand exhibits [4]. We call this structure, a recurring demand field resampled by a spiraling constellation, the *spiral cyclostationarity* of LEO demand. We use the term descriptively: INSITE

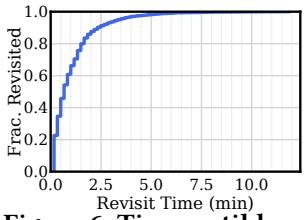


Figure 6: Time until location is revisited by a satellite within 0.5°.

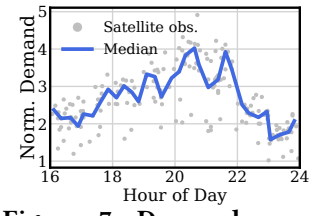


Figure 7: Demand over Chicago formed using aggregated observations.

performs no cyclostationary signal analysis, but builds its design directly on this structure, in the form of a constellation-pooled, location-indexed demand map that any satellite can read and every satellite writes.

3 Design

INSITE has two components: a *demand predictor* that estimates the traffic each satellite will serve over an upcoming interval, and a *constellation-aware GS selector* that assigns satellites to GSs using those estimates. We first formalize the GS-selection problem, then present the observation that motivates the predictor and selector, and their design details.

3.1 Problem formulation

Setup. We consider a LEO constellation serving a geographic region through bent-pipe egress, with a set $\mathcal{S}$ of satellites and $\mathcal{G}$ of GSs. Operators reconfigure egress at discrete intervals; within one interval $[t, t{+}W]$, demand shifts as satellites move, so we discretize the interval into steps $\tau \in \mathcal{T} = \{t, t{+}\Delta, \ldots, t{+}W\}$. Each satellite $s \in \mathcal{S}$ serves a downlink-dominated demand $d_s(\tau) \geq 0$ at step τ, determined by the users in its footprint. Because egress is bent-pipe, s can offload only to GSs in view, which orbital geometry determines per step, $\mathcal{R}(s, \tau) \subseteq \mathcal{G}$. Each GS g has finite capacity C_g, bounding the traffic it can absorb at each step.

Decision. At the start of each interval, the operator's controller chooses an assignment $a : \mathcal{S} \to \mathcal{G}$, held fixed over the interval, mapping each satellite to a single GS in view. We model one active GS per satellite per interval, abstracting the two gateway-facing antennas of current hardware as a single active link (see §6). Each satellite's demand lands on its assigned GS, so the load on GS g at step τ, $\ell_g(\tau)$, is simply the total demand of the satellites assigned to it.

Objective. Because user demand mirrors uneven population density, load concentrates on the GSs near dense regions while distant capacity sits underused. INSITE chooses the assignment a that balances load across the interval, minimizing the peak utilization any GS reaches: $\min_a \ U(a)$, where $U(a) = \max_{g \in \mathcal{G}, \tau \in \mathcal{T}} \ell_g(\tau; a)/C_g$. Demand beyond its assigned GS's capacity must detour over an inter-satellite link to a farther GS, a higher-latency path. We call this *forced overflow*, and we aim to minimize its total across all GSs and steps. Keeping peak utilization below one eliminates overflow, and when overflow is unavoidable, the min-max objective bounds the excess at the most overloaded GS.

Challenge. The difficulty is temporal. The assignment for $[t, t{+}W]$ is committed at time t, but the demand $d_s(\tau)$ for $\tau > t$ has not yet materialized. A *present-time* policy substitutes the instantaneous demand $d_s(t)$ for the future demand and optimizes for a snapshot that may no longer hold by the end of the interval. An assignment that fits at commit time can produce congestion minutes later, once a satellite has swept into a denser region. INSITE instead plans against a *predicted* demand $\widehat{d}_s(\tau)$ over the whole interval, estimated from observations available up to t. The problem is therefore online, requiring a fresh assignment each interval from past observations only, and it decomposes into two subproblems. INSITE must first estimate $\widehat{d}_s(\tau)$ for every satellite and step in the interval (§3.3), and then choose the assignment minimizing the peak utilization under the predicted demand (§3.4). The subproblems are coupled, since any prediction error can harm the resulting assignment.

3.2 Limits of per-satellite history

Estimating $\widehat{d}_s(\tau)$ is hard because the satellite's motion over the rotating Earth denies it a usable history, in two ways. First, a satellite samples any given location sparsely due to the *spiral-cyclostationary* nature of the orbit (§2.3). Second, a satellite's own demand series rarely repeats. Even if the satellite passes over the same geography, it may do so at a different time of day, so its load may show little self-similarity across orbits. A satellite's spiral-cyclostationary orbit makes it difficult to predict future demand from its own past demand.

However, the demand a satellite is about to serve has usually been observed already by other satellites. As major constellations such as Starlink now cover nearly every populated region [36], each such region is passed over by many different satellites over time. Whichever satellite last passed over a region has already sampled its recent demand. INSITE therefore predicts by location rather than by satellite. Pooling every satellite's observations of a geographic region turns many sparse, non-repeating per-satellite series into one dense per-location series, and that series is predictable because demand typically recurs on a daily cycle (e.g., diurnal patterns). A satellite entering a region can thus inherit a full history without collecting it itself.

3.3 The demand predictor

Representation. INSITE maintains a single location-indexed view of demand, used by all satellites. The serving region's surface is discretized into cells, and each cell accumulates the

demand observations made by satellites whose coordinates fall within the cell's range. A satellite about to enter a cell can use that cell's pooled observations to predict its demand.

To predict future demand, we use two approaches. First, we use a simple linear regression that fits the observations within a lookback window, and extrapolates the future demand at each step in the upcoming window. We use the 12 most recent observations as the lookback window, which represent about 8 minutes of past demand, since a satellite typically passes through any given cell every 40 seconds (see §2.3). The linear regression exploits the fact that the total demand for the cell typically changes gradually, and allows simple per-cell predictions with minimal overhead.

We also evaluate a more complex prediction approach based on a gated recurrent unit (GRU) trained across observations from all cells. The inputs are the cell and the recent demand observations from the lookback window. Its output is the predicted demand for each step in the upcoming window. Training on many cells lets the GRU learn the general structure of demand dynamics (e.g., diurnal patterns). It is retrained every 2 weeks by default to account for gradual changes in demand patterns over time.

3.4 Constellation-aware egress selection

Given predicted demand $\widehat{d}_s(\tau)$ of every satellite, INSITE assigns each satellite to a reachable GS to minimize predicted peak utilization. It does so jointly, making each satellite's assignment in view of the demand all others will place on the same GSs, rather than each satellite individually. Without joint decisions, a satellite could offload to the nearest GS with spare capacity, blind to a second satellite that will sweep the same region moments later. The following satellite finds that capacity already consumed, and its traffic is forced onto a farther GS, or worse, cannot find a suitable GS in range. Even when every station has spare capacity at the moment each choice is made, such independent choices can collectively overload the stations near dense regions.

As a result, INSITE uses a constellation-aware assignment, building on three decisions. First, it orders satellites by predicted peak demand, with the heaviest satellites first. For each satellite in turn, it scores every reachable GS by the peak utilization that station would reach over the interval if this satellite's demand were added to the demand already placed on it, and assigns the station whose peak is lowest. It then records the satellite's predicted demand on the chosen station, so that every later satellite is scored against it.

Each decision answers one part of the problem. Processing the heaviest satellites first places the hardest-to-place load while the most capacity is still free, since a large demand fits few stations later in the ordering. Scoring on the interval peak rather than the commit-instant load keeps the assignment viable across the window, since a station that looks empty at t may be heavily loaded later. And recording each satellite's predicted demand before the next is chosen coordinates the choices: every assignment is made against the load the rest of the fleet has already claimed, preventing suboptimal per-satellite decisions.

4 Experimental Evaluation

Setup. We obtain satellite orbits from public Two-Line Elements (TLEs) as of September 2025 (8,137 satellites) [9] and the locations of 64 US ground stations [38]. We feed both into CosmicBeats [33] for orbital and visibility simulation. We restrict user demand and GSs to the continental US; satellites outside it carry no load. Similarly to prior work [48], we place our users according to population density. Each satellite's offered demand combines the census population [43] under its footprint, a diurnal modulation [27], and a per-user demand rate [3]. Each GS has a capacity of 320 Gbps, matching the maximum antenna count in FCC filings [12] at 8 Gbps per antenna [40]. We calibrate the per-user rate so that aggregate demand sits at 60% of total GS capacity. In this regime, GSs experience congestion without load balancing, but headroom remains to relocate traffic. The final dataset comprises 8,137 satellites, 64 ground stations, and eight hours of simulated demand. Our motivation experiments (§2) use a representative two-hour segment. To validate our predictors, we also use one week of real-world cellular demand from Milan [4].

We do not model inter-satellite links. Satellites offload only to GSs in view, so every policy, including ours, must absorb demand within that geometry. Demand exceeding the capacity of every visible GS counts as overflow, the traffic a deployed system would carry over ISLs to farther stations.

Baselines. We compare INSITE against the present-time selection policies detailed in §2: *Nearest* and *Capacity-Aware Nearest*, as defined there. We further evaluate three variants of INSITE, which differ only in how they determine demand. *INSITE-Present* runs our joint selection algorithm against currently observed demand, isolating the algorithm's contribution without any prediction. *INSITE-Predict* is the full design, selecting against demand forecast by our map-based predictor. *INSITE-Oracle* replaces the predictor with ground-truth future demand, an upper bound that isolates the impact of prediction error. The gap between Predict and Oracle measures the headroom our preliminary predictor leaves unexploited, while the gap between Present and Oracle measures the potential benefit of including predictions.

Prediction accuracy. INSITE uses two constellation-pooled predictors: a location-indexed demand map using per-cell linear regression (LR), or a single learned GRU, shared across all locations. We validate each separately, first on the real Milan data, shown in Figure 8. The GRU trained on 8,000

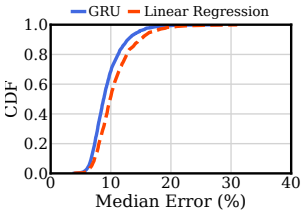


Figure 8: Prediction error on the Milan dataset.

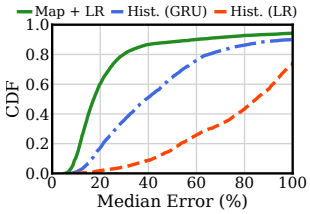


Figure 9: Prediction error using a satellite's own history.

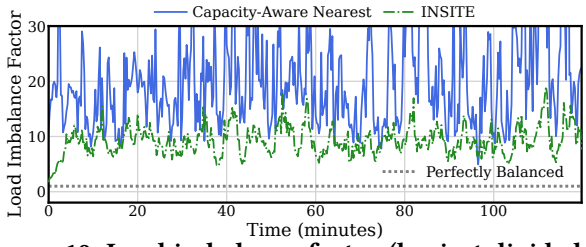


Figure 10: Load imbalance factor (busiest divided by P10 station load) over the two-hour trace.

Milan cells, evaluated on 2,000 unseen cells and days, yields 8.8% median per-cell error, a minor improvement over Linear Regression, which has 9.9% median error. Figure 9 compares our accuracy in simulation against predicting from each satellite's own history. We predict over 2,422 held-out (satellite, window) instances over a 300-second window. We show only linear regression (Map + LR), since the two predictors perform similarly on the Milan data and LR has lower overhead. Map + LR yields 17% median error, against 39% for Own-history (GRU) and 85% for Own-history (LR). This gap shows the effectiveness of using the aggregated demand map for handling spiral-cyclostationary demand.

Load balance. Figure 10 plots the load imbalance factor over time under INSITE and Capacity-Aware Nearest. We omit Nearest, which has 58.8× median imbalance, to better show the difference between the other two. INSITE holds the constellation at a median imbalance of 9.0×, against 18.3× under Capacity-Aware Nearest. The remaining imbalance is due to visibility geometry rather than headroom INSITE fails to use. With no ISLs, demand over dense regions can spread only across the stations in view, and once those saturate, further flattening requires capacity no policy can see.

Overflow. Five of the 64 stations face aggregate demand above their capacity that cannot be offloaded. Since no selection policy can relieve them without ISLs, we omit them from the figures, but report the results for the full set of GSs in the text. Figure 11 shows the total overflow per second over the two-hour trace. INSITE-Present reduces total overflow by 70.0% against Capacity-Aware Nearest (2,448 to 734 Tb). INSITE-Predict, using linear regression, cuts it further to 645 Tb, a 73.7% reduction overall and 12.1% on top of INSITE-Present. INSITE-Oracle reaches 483 Tb, a further

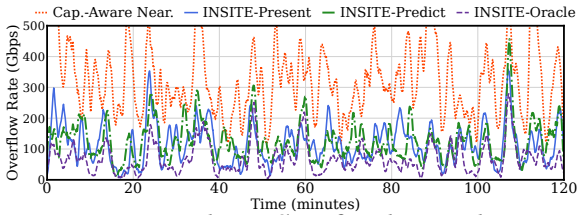


Figure 11: Demand overflow for the two-hour trace.

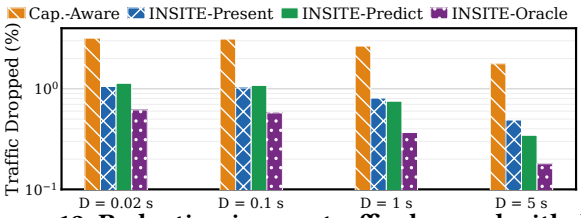


Figure 12: Reduction in user traffic dropped with different queue depths (D).

34.2% reduction on top of INSITE-Present, showing headroom for prediction. Across all 64 stations, INSITE-predict reduces overflow by 27.6%.

Packet Loss Under Application Latency Budgets. We model each GS as a finite-buffer FIFO queue served at 320 Gbps and report the fraction of user traffic dropped. We report packet drops rather than tail latency since the lack of ISLs can cause unbounded delays at congested GSs. We evaluate the 59 stations that remain unsaturated under INSITE-present across four queue depths spanning real-time (20 ms) to elastic (5 s) traffic deadlines [1, 13, 29], shown in Figure 12.

Coordination is the first-order win, and it holds at every budget. INSITE-present cuts loss roughly 3× against Capacity-Aware Nearest, and the gain survives the tightest real-time budget, where the buffer is too shallow to absorb sustained contention, yet coordination alone removes two-thirds of drops. The benefit of prediction grows with the latency budget. At 20 and 100 ms, INSITE-predict is slightly worse than INSITE-present (1.14% vs. 1.05%; 1.08% vs. 1.02%) since our forecast predicts demand per interval and cannot see small bursts. Prediction instead helps by avoiding stations that stay overloaded for a long time, and with such small buffers that benefit is not yet visible. As the buffer grows, avoiding long-lasting overload matters more. INSITE-predict overtakes at 1 s (0.75% vs. 0.81%) and at 5 s cuts loss by roughly 30% (0.49% to 0.35%), closing about 45% of the remaining gap to INSITE-oracle. The oracle's 0.18–0.62% marks the floor our selection algorithm reaches with perfect knowledge, 1.7–2.7× below INSITE-Present across budgets.

5 Related Work

Traffic engineering in wide-area networks is mature, precomputing candidate paths and periodically optimizing traffic over them [17, 20, 23, 24, 42, 45, 49], even reacting to sub-second bursts [15]. But these systems assume fixed egress

and optimize only paths. LEO constellations instead face continuous topology change. Recent work targets this by characterizing and reducing path churn and routing instability [6, 26], planning ISL topology at scale [10], and speeding up TE computation [48]. These efforts optimize how traffic traverses the network and take GSs as given endpoints. When a GS is congested, the remedy at this layer is to detour traffic over ISLs to a farther station, at the cost of longer paths [19]. Better GS selection reduces how much traffic must take these detours in the first place.

Ground station selection via mixed-integer optimization [2, 30] solves an *offline* problem with demand given as input, not an online decision reacting as a satellite's demand shifts within its window. OrbitTransit [50] assigns satellites to GSs against the same nearest and nearest-available baselines we use, but targets delay-tolerant Earth-observation data that can be buffered in orbit. NEO-GNN [21] targets bent-pipe broadband but jointly plans each snapshot against currently observed demand. INSITE instead treats GS selection as a *predictive* decision over unknown, time-varying demand. A solver such as NEO-GNN could select against INSITE's forecast in place of our constellation-aware selector (§3.4).

Demand prediction for satellite networks has been studied on its own, but with two limits our setting exposes. Prior work models demand as a function of orbital position [11], capturing a recurring spatial field rather than how load evolves within the window, or relays a neighbor satellite's recent load as a proxy [14] that Earth's rotation quickly makes stale. More broadly, these deliver a predictor in isolation: even where GS scheduling is cited as motivation, none jointly optimizes the selection decision against the forecast. INSITE instead anchors demand to each satellite's actual sub-satellite geography, pooling constellation-wide observations into a single time-varying forecast that drives selection.

6 Discussion and Open Questions

GS selection as one layer of load management. INSITE balances load at the ground segment, but selection interacts with other layers. Our current design leaves several questions open. First, we evaluate GS selection without inter-satellite links, isolating the selection decision so that dropped traffic directly reflects its quality. In an ISL-enabled network, how does the chosen GS shape the way overflow traverses ISLs? Second, INSITE selects on GS load, treating each gateway's exit to the terrestrial network as uniform. How much do terrestrial conditions (e.g., ISP congestion, peering, and backbone-path capacity from the point of presence onward) affect egress quality? Third, selection cannot create capacity if GSs are overloaded under any policy. There, operators must rely on slower levers such as admission control, demand-based pricing, or new GS deployment. Can INSITE's demand map help here as an offline testbed, letting operators compare alternative selection strategies or candidate GS sites against predicted demand before deployment?

Exploiting prediction beyond selection. INSITE produces two things present-time selection cannot produce—a committed schedule of GS assignments and a forecast of per-GS load over the window. How can these help existing satellite TE? The schedule fixes the endpoints that in-space routing takes as input [48], and tells handover-aware transport [8, 25] when satellite–GS links will change. The forecast lets ISL routing reserve capacity for predicted overflow [48], lets operators schedule bulk transfers into predicted spare GS capacity [24], and gives demand-predictive WAN TE an ingress traffic matrix for satellite traffic [31].

Adapting to heterogeneous user needs. In LEO networks, the infrastructure itself moves, so unlike cellular networks, handovers occur at the granularity of a satellite instead of a user. When a satellite switches ground stations, all user flows it carries move together. Since these users may differ in traffic type, latency sensitivity, and destination, a ground station that suits one user may be poor for another. How can we extend INSITE to account for the heterogeneous needs of users that share a satellite?

Multi-gateway allocation. We model each satellite as offloading to a single station, abstracting the two GS-facing antennas of current hardware [32] as one active assignment. Letting a satellite split its load across simultaneous GSs turns selection into a fractional allocation, which could further relieve contention.

Prediction accuracy. We use a basic ML model to demonstrate our design. Several approaches could improve its accuracy: 1) de-biasing each cell's observations by the satellite-contention factor before pooling, 2) tuning cell granularity to trade spatial precision against sample density, and 3) modeling deployment factors absent from our simulator (e.g., hardware, weather, terrain) to improve real-world accuracy.

7 Conclusion

We presented INSITE, a ground station selection framework that jointly plans across the constellation using predicted demand over the connection window. In preliminary results, INSITE substantially reduces load imbalance, forced overflow, and packet loss.

References

[1] 5G-ACIA. 2019. *5G for Connected Industries and Automation.* White Paper. 5G Alliance for Connected Industries and Automation (5G-ACIA).

[2] Yuma Abe, Fabio Ortiz, Eva Lagunas, Victor Monzon Baeza, Symeon Chatzinotas, and Hiroyuki Tsuji. 2024. Optimizing Satellite Network Infrastructure: A Joint Approach to Gateway Placement and Routing. In *Proceedings of the IEEE Vehicular Technology Conference*

(VTC2024-Spring). 1–6. https://doi.org/10.1109/VTC2024-Spring62846.2024.10683455

[3] Astound Broadband. 2025. What Internet Speed Do You Need for Streaming? https://www.astound.com/learn/internet/internet-speed-streaming/. Accessed: 2025-09-01.

[4] Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. 2015. A Multi-source Dataset of Urban Life in the City of Milan and the Province of Trentino. *Scientific Data* 2, 1 (2015), 150055. https://doi.org/10.1038/sdata.2015.55

[5] BASENOR Editorial Desk. 2026. SpaceX Files for 100,000 Gen 3 Starlink Satellites in Massive FCC Bid. Basenor. https://www.basenor.com/blogs/news/spacex-files-for-100-000-gen-3-starlink-satellites-in-massive-fcc-bid.

[6] Vaibhav Bhosale, Ahmed Saeed, Ketan Bhardwaj, and Ada Gavrilovska. 2023. A Characterization of Route Variability in LEO Satellite Networks. In *Passive and Active Measurement: 24th International Conference, PAM 2023, Virtual Event, March 21–23, 2023, Proceedings*. Springer-Verlag, Berlin, Heidelberg, 313–342. https://doi.org/10.1007/978-3-031-28486-1_14

[7] Aveek Bhowmik. 2026. SpaceX's Starlink Surpasses 12M Customers Across 160 Countries As Growth Accelerates. Yahoo Finance. https://finance.yahoo.com/markets/stocks/articles/spacex-starlink-surpasses-12m-customers-000951284.html.

[8] Xuyang Cao and Xinyu Zhang. 2023. SaTCP: Link-Layer Informed TCP Adaptation for Highly Dynamic LEO Satellite Networks. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. 1–10. https://doi.org/10.1109/INFOCOM53939.2023.10228914

[9] CelesTrak. [n. d.]. NORAD Two-Line Element Set Format. https://celestrak.org/NORAD/documentation/tle-fmt.php. Accessed: 2025-08-10.

[10] Sheng Cen, Qiying Pan, Yifei Zhu, and Bo Li. 2024. SatFlow: Scalable Network Planning for LEO Mega-Constellations. arXiv:2412.20475 [cs.NI] https://arxiv.org/abs/2412.20475

[11] Quan Chen, Yi Zhang, Jianming Guo, Lei Yang, Chengguang Fan, Yong Zhao, and Xiaoqian Chen. 2019. Traffic Prediction Based on Surrogate Model in Satellite Constellation Networks. In *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*. 126–130. https://doi.org/10.23919/WMNC.2019.8881472

[12] Federal Communications Commission. 2025. Satellite Communications Services: Actions Taken. Public Notice, Report No. SES-02782, Space Bureau. https://docs.fcc.gov/public/attachments/DOC-412437A2.pdf, accessed Sept. 2025.

[13] Tongtong Feng, Qi Qi, Bo He, and Jingyu Wang. 2024. Deadline and Priority Constrained Immersive Video Streaming Transmission Scheduling. arXiv:2408.17028 [cs.NI] https://arxiv.org/abs/2408.17028

[14] Lizeng Gong, Quan Chen, Lei Yang, Zhenglong Yin, and Yi Wang. 2025. Autonomous Traffic Prediction for LEO Satellite-Based IoT Based on Satellite Spatiotemporal Features Mapping. *IEEE Internet of Things Journal* 12, 14 (2025), 27021–27032. https://doi.org/10.1109/JIOT.2025.3562631

[15] Fei Gui, Songtao Wang, Dan Li, Li Chen, Kaihui Gao, Congcong Min, and Yi Wang. 2024. RedTE: Mitigating Subsecond Traffic Bursts with Real-time and Distributed Traffic Engineering. In *Proceedings of the ACM SIGCOMM 2024 Conference* (Sydney, NSW, Australia) *(ACM SIGCOMM '24)*. Association for Computing Machinery, New York, NY, USA, 71–85. https://doi.org/10.1145/3651890.3672231

[16] Alejandro Gutierrez Prieto. 2026. What Is Starlink's Demand Surcharge? HighSpeedInternet.com. https://www.highspeedinternet.com/resources/what-is-starlinks-demand-surcharge.

[17] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. 2013. Achieving high utilization with software-driven WAN. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (Hong Kong, China) *(SIGCOMM '13)*. Association for Computing Machinery, New York, NY, USA, 15–26. https://doi.org/10.1145/2486001.2486012

[18] ITU-R. 2015. *IMT Traffic Estimates for the Years 2020 to 2030*. Report Report ITU-R M.2370-0. International Telecommunication Union, Radiocommunication Sector. https://www.itu.int/pub/R-REP-M.2370-2015 Accessed: 2026-07-17.

[19] Liz Izhikevich, Manda Tran, Katherine Izhikevich, Gautam Akiwate, and Zakir Durumeric. 2024. Democratizing LEO Satellite Network Measurement. *Proc. ACM Meas. Anal. Comput. Syst.* 8, 1, Article 13 (Feb. 2024), 26 pages. https://doi.org/10.1145/3639039

[20] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jonathan Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. 2013. B4: Experience with a Globally Deployed Software Defined WAN. In *Proceedings of the ACM SIGCOMM Conference*. Hong Kong, China. http://cseweb.ucsd.edu/~vahdat/papers/b4-sigcomm13.pdf

[21] Aruna Jayarajan, N. Cameron Matson, and Karthikeyan Sundaresan. 2026. LEO Satellite Network Orchestration with Heterogeneous Graph Neural Networks. In *IEEE INFOCOM 2026 - IEEE Conference on Computer Communications*. 1–10. https://doi.org/10.1109/INFOCOM59046.2026.11571289

[22] Michael Kan. 2026. SpaceX Preps Next-Gen Gateway to Accelerate Starlink Speeds. https://www.pcmag.com/news/spacex-preps-next-gen-gateway-to-accelerate-starlink-speeds PCMag, April 23, 2026.

[23] Srikanth Kandula, Dina Katabi, Bruce Davie, and Anna Charny. 2005. Walking the tightrope: responsive yet stable traffic engineering. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Philadelphia, Pennsylvania, USA) *(SIGCOMM '05)*. Association for Computing Machinery, New York, NY, USA, 253–264. https://doi.org/10.1145/1080091.1080122

[24] Srikanth Kandula, Ishai Menache, Roy Schwartz, and Spandana Raj Babbula. 2014. Calendaring for wide area networks. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (Chicago, Illinois, USA) *(SIGCOMM '14)*. Association for Computing Machinery, New York, NY, USA, 515–526. https://doi.org/10.1145/2619239.2626336

[25] Zeqi Lai, Zonglun Li, Qian Wu, Hewu Li, Jihao Li, Xin Xie, Yuanjie Li, Jun Liu, and Jianping Wu. 2025. LeoCC: Making Internet Congestion Control Robust to LEO Satellite Dynamics. In *Proceedings of the ACM SIGCOMM 2025 Conference*. 129–146.

[26] Yuanjie Li, Lixin Liu, Hewu Li, Wei Liu, Yimei Chen, Wei Zhao, Jianping Wu, Qian Wu, Jun Liu, and Zeqi Lai. 2024. Stable Hierarchical Routing for Operational LEO Networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., DC, USA) *(ACM MobiCom '24)*. Association for Computing Machinery, New York, NY, USA, 296–311. https://doi.org/10.1145/3636534.3649362

[27] Zikun Liu, Fan-Xue Gabriella Reidys, Sarah Tanveer, and Deepak Vasisht. 2025. Vivisecting Starlink Throughput: Measurement and Prediction. *Proc. ACM Netw.* 3, CoNEXT4, Article 24 (Nov. 2025), 23 pages. https://doi.org/10.1145/3768971

[28] Nitinder Mohan, Andrew E. Ferguson, Hendrik Cech, Rohan Bose, Prakita Rayyan Renatin, Mahesh K. Marina, and Jörg Ott. 2024. A Multifaceted Look at Starlink Performance. In *Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 2723–2734. https://doi.org/10.1145/3589334.3645328

[29] Alireza Monfared. 2024. Understanding Network Latency and Its Impact on Industrial Applications. https://www.omnitron-systems.com/blog/understanding-network-latency-and-its-impact-

on-industrial-applications. Omnitron Systems. Accessed: 2026-01-17.

[30] Nils Pachler, Edward F. Crawley, and Bruce G. Cameron. 2025. Mixed Integer Linear Programming for Ground Infrastructure and Inter-Satellite Link Design in Satellite Constellations. *IEEE Transactions on Vehicular Technology* (2025).

[31] Yarin Perry, Felipe Vieira Frujeri, Chaim Hoch, Srikanth Kandula, Ishai Menache, Michael Schapira, and Aviv Tamar. 2023. DOTE: Rethinking (Predictive) WAN Traffic Engineering. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. USENIX Association, Boston, MA, 1557–1581. https://www.usenix.org/conference/nsdi23/presentation/perry

[32] Mike Puchol. 2022. Modeling Starlink Capacity. https://mikepuchol.com/modeling-starlink-capacity-843b2387f501. Accessed July 2026.

[33] Jayanth Shenoy, Om Chabra, Tusher Chakraborty, Suraj Jog, Deepak Vasisht, and Ranveer Chandra. 2024. CosMAC: Constellation-Aware Medium Access and Scheduling for IoT Satellites. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., DC, USA) *(ACM MobiCom '24)*. Association for Computing Machinery, New York, NY, USA, 724–739. https://doi.org/10.1145/3636534.3690657

[34] Clovelly Software. 2026. satellitemap.space. https://satellitemap.space/constellation/starlink.

[35] SpaceX Services, Inc. 2019. Application for Gateway Earth Station. FCC File No. SES-LIC-20190402-00451. https://fcc.report/IBFS/SES-LIC-20190402-00451/1640758. Filed April 2, 2019. Accessed: 2026-10-19.

[36] Starlink. 2025. Starlink | Availability Map. https://starlink.com/map. Accessed: 2026-07-17.

[37] Starlink. 2025. Starlink Network Update. https://starlink.com/updates/network-update. Accessed: 2026-10-19.

[38] Starlink Insider. 2023. Starlink Ground Station Locations: An Overview. https://starlinkinsider.com/starlink-gateway-locations/. Accessed: 2025-08-10.

[39] Hammas Bin Tanveer, Mike Puchol, Rachee Singh, Antonio Bianchi, and Rishab Nithyanand. 2023. Making Sense of Constellations: Methodologies for Understanding Starlink's Scheduling Algorithms. In *Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies* (Paris, France) *(CoNEXT 2023)*. Association for Computing Machinery, New York, NY, USA, 37–43. https://doi.org/10.1145/3624354.3630586

[40] Telecompetitor. 2024. Starlink Speeds Hit 8.2 Gbps Download in Test With "No Land in Sight". https://www.telecompetitor.com/starlink-speeds-hits-8-2-gbps-download-in-test-with-no-land-in-sight/. Reporting statements by M. Nicolls and E. Musk, SpaceX. Accessed Sept. 2025.

[41] The Health Wagon. 2025. Cutting-Edge Starlink Technology Brings High-Speed Telehealth to Remote Virginia Homes. https://thehealthwagon.org/cutting-edge-starlink-technology-brings-high-speed-telehealth-to-remote-virginia-homes/.

[42] Slavica Tomovic and Igor Radusinovic. 2019. Toward a Scalable, Robust, and QoS-Aware Virtual-Link Provisioning in SDN-Based ISP Networks. *IEEE Transactions on Network and Service Management* 16, 3 (2019), 1032–1045. https://doi.org/10.1109/TNSM.2019.2929161

[43] U.S. Census Bureau. 2020. 2020 Census Population Counts. https://www2.census.gov. Accessed: 2025-08-10.

[44] USAID Office of Inspector General. 2025. Ukraine Response: USAID Did Not Fully Mitigate the Risk of Misuse of the Starlink Satellite Terminals It Delivered to Ukraine. https://oig.usaid.gov/node/7845.

[45] Jason Min Wang, Ying Wang, Xiangming Dai, and Brahim Bensaou. 2019. SDN-Based Multi-Class QoS Guarantee in Inter-Data Center Communications. *IEEE Transactions on Cloud Computing* 7, 1 (2019), 116–128. https://doi.org/10.1109/TCC.2015.2491930

[46] Wikipedia. [n. d.]. Pearson correlation coefficient. https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

[47] Mike Wilson. 2026. U.S. State Department Signs Two-Year Disaster Response Deal With Starlink. Drive Tesla. https://driveteslacanada.ca/news/starlink-disaster-response-state-department-mou/.

[48] Hao Wu, Yizhan Han, Mohit Rajpal, Qizhen Zhang, and Jingxian Wang. 2025. SaTE: Low-Latency Traffic Engineering for Satellite Networks. In *Proceedings of the ACM SIGCOMM 2025 Conference* (São Francisco Convent, Coimbra, Portugal) *(SIGCOMM '25)*. Association for Computing Machinery, New York, NY, USA, 896–916. https://doi.org/10.1145/3718958.3750524

[49] Zhiying Xu, Francis Y. Yan, Rachee Singh, Justin T. Chiu, Alexander M. Rush, and Minlan Yu. 2023. Teal: Learning-Accelerated Optimization of WAN Traffic Engineering. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA) *(ACM SIGCOMM '23)*. Association for Computing Machinery, New York, NY, USA, 378–393. https://doi.org/10.1145/3603269.3604857

[50] Haoyuan Zhao, Long Chen, Yi Ching Chou, Hao Fang, and Jiangchuan Liu. 2026. [Emerging Ideas] OrbitTransit: Traffic Delivery and Diffusion for Earth Observation via Satellite Mobility. In *Proceedings of the 24th Annual International Conference on Mobile Systems, Applications and Services* (University of Cambridge, Cambridge, United Kingdom) *(MobiSys '26)*. Association for Computing Machinery, New York, NY, USA, 202–215. https://doi.org/10.1145/3745756.3809199

[51] Haoyuan Zhao, Hao Fang, Feng Wang, and Jiangchuan Liu. 2023. Realtime Multimedia Services over Starlink: A Reality Check. In *Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video* (Vancouver, BC, Canada) *(NOSSDAV '23)*. Association for Computing Machinery, New York, NY, USA, 43–49. https://doi.org/10.1145/3592473.3592562