LATENCETECH

every millisecond counts

MEASURING THROUGHPUT OVER SATELLITE LINKS

Document Control

Field	Information
Document Title	[MEASURING_THROUGHPUT_SAT_LEO_EN.docx]
Version	0.1
Status	Draft
Owner	MS / BG
Approved By	BG
Approval Date	06/07/2026
Review Cycle	Annual
Classification	Commercial in Confidence

Revision History

Revision	Date	Author	Description
v1	06/07/2026	MS / BG	Satellite link without traffic

Table of Contents

Introduction:

The purpose of this document is to answer these 3 questions:

- Why measure throughput?
- How to measure? The 2 methods: iPerf3 & LIFBE
- On a satellite link without traffic, are iPerf3 (UDP) and LIFBE (UDP) expected to converge?

Why measure throughput?

On a LEO satellite link, the available bandwidth is not a constant — it fluctuates continuously due to beam load, satellite handovers, weather conditions, and the operator's traffic regulation policy (Token Bucket). Measuring throughput therefore means answering a critical operational question:

→ Can my application send its data right now, at the required quality level?

→ Is my network SLO being met right now, or have I already crossed the degradation threshold?

→ Does the degradation I am experiencing come from the satellite link, the core network, or the ground station?

Without throughput measurement, the satellite network is a black box: we know it works when the application runs, and we know it fails when the application crashes. The purpose of measurement is to anticipate this degradation — ideally 30 to 120 seconds before it impacts users.

Two main use cases justify throughput measurement on LEO satellite links:

Commissioning

Characterise the maximum link capacity (Ce) under best conditions: what is the theoretical performance of this Starlink or OneWeb antenna, with a given subscription, in this geographic area, at this time, with this satellite elevation? This reference measurement serves as the basis for contractual SLAs.

Production Monitoring (Continuous Operation)

Monitor in real time the effectively available bandwidth (Ab) while business applications are running. Detect degradations before they become incidents. Correlate throughput drops with beam handovers, load peaks, or radio conditions (RSRP/RSRQ).

How to measure? The 2 methods: iPerf3 & LIFBE

General Principle of iPerf3

iPerf3 is an open-source network throughput testing tool, operating in client/server mode. It generates a data stream between two IP nodes to measure the theoretical maximum link capacity (Ce) — i.e. the maximum throughput the network can carry when deliberately saturated.

Measurement Architecture

→ A TCP control session is established on port 5201 between client and server

→ One or more data streams are opened according to the chosen protocol (TCP or UDP)

→ The server records bytes received per time interval (default: 1 second)

→ Throughput is calculated: Throughput (bps) = Volume (bytes) × 8 ÷ Duration (s)

Fundamental Characteristic

iPerf3 measures the theoretical maximum link capacity (Ce) — not the available bandwidth (Ab). It deliberately disrupts existing traffic during the test. It is a one-off commissioning tool, incompatible with continuous production use.

iPerf3 TCP

In TCP mode (default), iPerf3 lets the TCP protocol drive throughput via its congestion window mechanism (CWND). The throughput is not set in advance — it emerges from the negotiation between TCP and the network.

Mechanism — TCP Slow Start

TCP starts with a minimal congestion window (1 MSS = 1,460 bytes) and doubles it at every round trip (RTT) as long as no packet loss is detected. This ramp-up phase (Slow Start) precedes the progressive saturation of the link.

RTT 1: CWND = 1 MSS → low throughput

RTT 2: CWND = 2 MSS → throughput × 2

RTT 3: CWND = 4 MSS → throughput × 4

...: CWND = N MSS → link saturation (Ce reached)

Limitations in LEO Satellite Context

→ High RTT (40–80 ms LEO, 600 ms GEO): very slow Slow Start — TCP may not reach saturation within 10 seconds, underestimating Ce

→ Operator Token Bucket: TCP fills the allowed burst then gets throttled — the displayed average blends burst and contractual throughput

→ No latency measurement: the server uses only its local clock, with no synchronisation with the client

iPerf3 UDP

In UDP mode (option -u), the sending throughput is set in advance by the operator via the -b parameter. There is no Slow Start, no flow control: iPerf3 sends at the specified rate regardless of network conditions.

Metrics Measured in UDP

→ Throughput actually received (verification that -b is reached)

→ Packet loss (%) — primary metric in UDP

→ Jitter — calculated from timestamps embedded in each UDP datagram

→ Out-of-order packets

Key Difference with LIFBE

iPerf3 UDP and LIFBE share the same transport protocol (UDP) and the same per-packet timestamp principle. But iPerf3 UDP asks: 'What is the link quality if I inject X Gbps?' — by saturating the network. LIFBE asks: 'What is the available bandwidth without imposing anything?' — with only 20 to 50 lightweight probes.

Limitation on LEO Satellite in UDP Mode

The -b parameter must be calibrated in advance, which requires approximate knowledge of Ce. Too low: no congestion, non-representative measurement. Too high: saturation and massive packet loss. Moreover, the operator Token Bucket can distort the measurement if -b does not exceed the throttling activation threshold.

General Principle of LIFBE

LIFBE (Low Intrusive Fast Bandwidth Estimation) est une méthode brevetée (EP 2 862 324 B1, Orange Innovation / inventeur : Laurent Mussot, industrialisée par LatenceTech) d'estimation de la bande passante disponible (Ab) sur tout lien IP, filaire ou sans-fil.

Objective

Measure Ab = Ce − Rx in real time, without disturbing existing traffic, without prior knowledge of link capacity.

4-Step Principle

→ Step 1: send a single train of 20 to 50 lightweight UDP probes (≈ 73 KB total)

→ Step 2: the receiver timestamps each probe on arrival and calculates cumulative received delays $T(n\text{-}1) = t\_n^{o} - t\_1^{o}$

→ Step 3: detection of atypical jitter values using the box plot method

→ Step 4: piecewise regression on cumulative delays to identify the breakpoint — the moment queues start forming, revealing Ab

Two Modes Depending on Ce Availability

Mode 1 (Ce known): constant probe rate, linear regression. Ce is obtained via radio MCS scan or AT 3GPP command AT+CGEQNEG. Formula: Ab = Ce(1 − a/L × sendrate) + sendrate.

Mode 2 (Ce unknown): increasing probe rate, non-linear LMA regression. Ce and Ab are derived simultaneously from the breakpoint — with no prior knowledge of the infrastructure.

Advantage on LEO Satellite

The measurement takes less than one millisecond (duration of the probe train), making LIFBE near-instantaneous in the face of rapid satellite channel variations. It can be triggered every 30 seconds in production with no impact on application traffic — unlike iPerf3, which disrupts the network for at least 10 seconds.

Known Limitation on LEO Satellite

Operator Token Bucket: the 50-probe train may pass entirely within the allowed burst window before throttling activates, leading to an overestimate of Ab. Workaround: use a parallel saturating flow beforehand, or increase the number of probes to exceed the burst window.

This is a precise question that goes to the heart of both methods. **The answer is yes, but with important nuances.**

The Ideal Case: Empty Satellite Link, No Cross-Traffic

If Rx = 0 (no other traffic), then:

Ab = Ce − Rx = Ce − 0 = Ce
The available bandwidth equals the maximum link capacity. Both tools should therefore converge on the same value — the Ce capacity of the satellite link.

But 3 Physical Factors Specific to Satellite Create Divergences

Factor 1 — Incompressible Propagation Latency

On a LEO link (Starlink): RTT ≈ 40–80 ms. On GEO: RTT ≈ 600 ms. This high latency impacts the two tools differently:

iPerf3 UDP:
→ sends at the rate set by -b, independently of RTT
→ not affected by latency for raw throughput

LIFBE:
→ cumulative received delays integrate propagation latency (Dn)
→ LIFBE models it as a constant in the regression
→ the breakpoint is not affected by Dn — it is a variation
in delay that identifies it, not its absolute value

Factor 2 — The Satellite Operator Token Bucket

This is the most likely divergence in practice:

Physical transponder capacity: 200 Mbps

Operator Token Bucket: burst allowed at 200 Mbps, throttled to 50 Mbps contractual rate

iPerf3 UDP at -b 200M for 10s:
→ fills the burst in the 1st second
→ detects throttling from the 2nd second
→ displays an average between burst and throttled rate
→ result: ~80 Mbps (misleading)

LIFBE (50 UDP packets):
→ may pass entirely within the burst window
→ never triggers the throttling mechanism
→ displays 200 Mbps (overestimate)

This is the identified known limitation of LIFBE on satellite — on an empty link with Token Bucket, both tools give different results AND both are partially wrong.

Factor 3 — Satellite Channel Variability

Even on an 'empty' link, the satellite is never perfectly stable:

Beam handovers (Starlink: every 15–30s)
→ micro-outages of a few ms
→ iPerf3 UDP: counts packet losses during handover
→ LIFBE: may detect a false breakpoint if the handover occurs during the 50-probe train

Weather conditions (rain, wind)
→ signal attenuation → instantaneous reduction of Ce
→ both tools capture it but at different time scales
iPerf3: average over 10s | LIFBE: snapshot in <1ms

Summary Table

Condition	iPerf3 UDP	LIFBE	Convergence
Empty link, no Token Bucket	Ce measured correctly	Ce measured correctly	**Yes**
Empty link + Token Bucket	Burst/throttled average	Overestimate (burst)	**No**
Handover during measurement	Losses counted	False breakpoint possible	**No**
Slightly loaded link (Rx > 0)	Saturates and overrides Rx	Measures Ab = Ce − Rx	**No — by design**

Conclusion :

On a perfectly empty satellite link without a Token Bucket, both tools converge on Ce.

But this case virtually never exists in real conditions — a satellite link is always subject to an operator Token Bucket, beam handovers, and radio variability.

This is precisely why the combination of both tools is complementary: iPerf3 UDP at commissioning to characterise Ce and Token Bucket behaviour, LIFBE in continuous production to monitor Ab in real time without disrupting traffic.

7. OneWeb Measurements Sample

The following section presents a real-world measurement campaign conducted on a OneWeb LEO satellite link using the LatenceTech LIFBE-based QoSAgent. It illustrates where to position the active probes, how the measurement architecture is deployed, the packet flow through the satellite network, and the throughput results obtained over a one-week period (June 13–19, 2026).

7.1 How to Measure — Active Probe Positioning

LIFBE requires two software components to be deployed at each end of the measured IP path:

- → QoSAgent — the active probe. Deployed on the user-side terminal (vessel, remote site, vehicle). It sends the UDP probe train at regular intervals (e.g. every 30 seconds) and timestamps each packet at emission.
- → Reflector (Target Reflector) — the passive target. Deployed close to the application endpoint (cloud server, data centre). It receives the probes, timestamps them at arrival, and returns the computed Ab value to the QoSAgent and Analyzer.

The Analyzer collects the results in real time, computes throughput KPIs (Ab, Ce, Rx), and delivers dashboards, alerts, and AI-based predictions through its open REST API.

7.2 Packet Flow — Architecture Diagram

The diagram below describes the end-to-end measurement path over a LEO satellite constellation. The numbered steps illustrate how UDP probe packets travel from the QoSAgent on the vessel through the satellite link to the Reflector near the ground station teleport, and how results are returned.

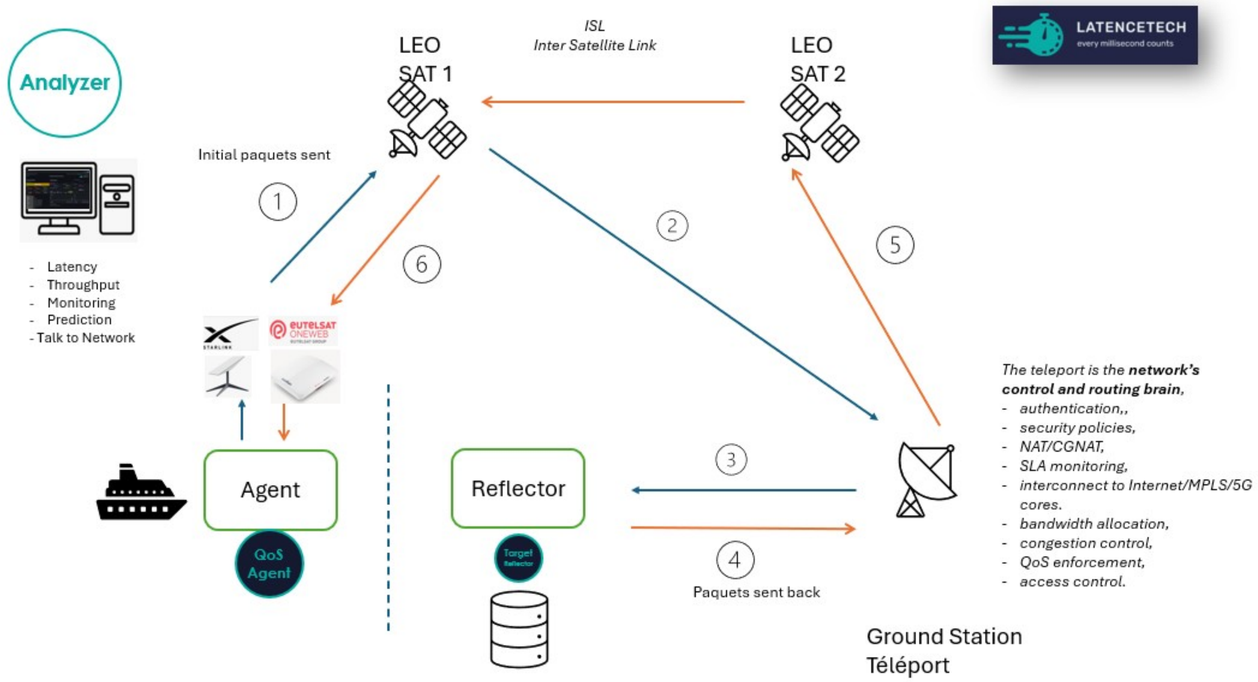

Figure 1 — LatenceTech LIFBE measurement architecture on LEO satellite (Starlink / Eutelsat OneWeb). ① Initial probes sent from QoSAgent → ② LEO SAT 1 → ③ Ground Station Teleport → ④ Packets sent back to Reflector → ⑤ via LEO SAT 2 → ⑥ Result returned to Agent. The Teleport acts as the network control and routing brain (authentication, QoS enforcement, bandwidth allocation).

7.3 Outcome Results — OneWeb Throughput Measurement

The chart below shows LIFBE throughput measurements (Available Bandwidth Ab) recorded continuously on a OneWeb OW10HL antenna (profile ONEWEB_10HLOW) over the period June 13–19, 2026.

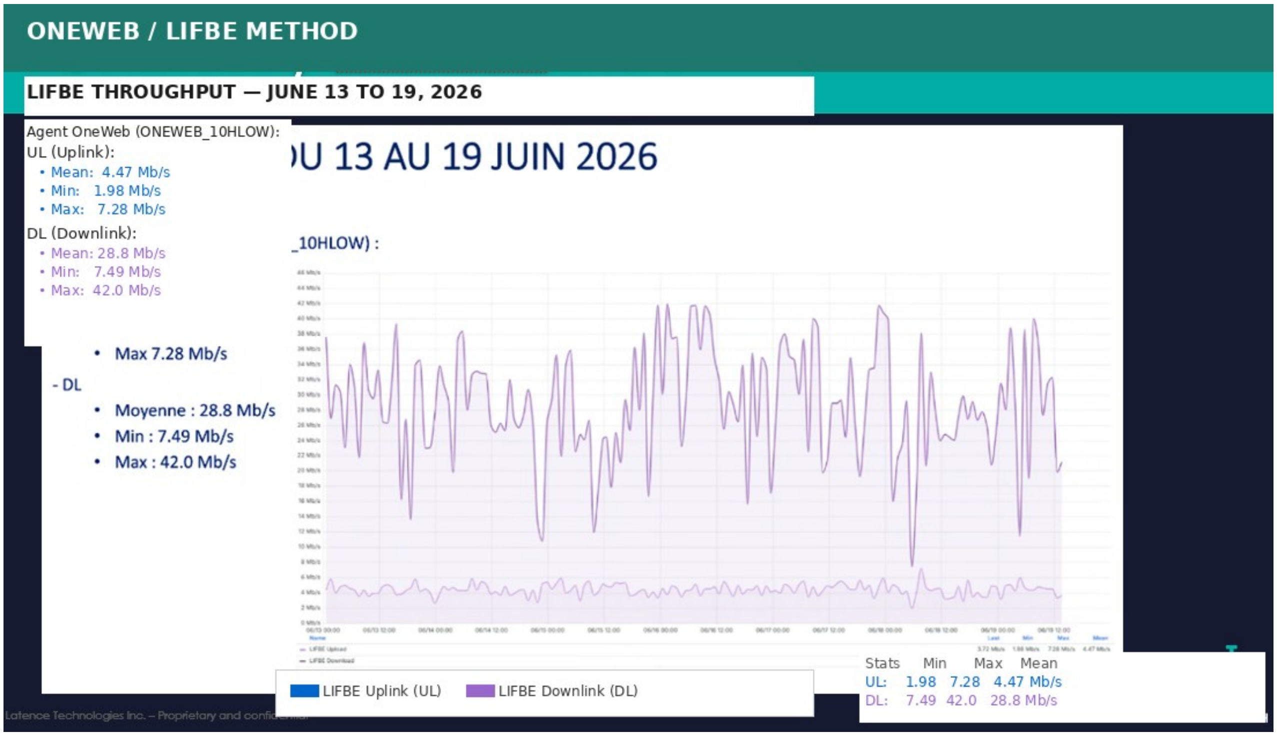

Figure 2 — LIFBE Throughput Measurements on OneWeb LEO link, June 13–19, 2026 (Agent profile: ONEWEB_10HLOW).

7.4 Analysis of Results

Uplink (UL) — LIFBE Available Bandwidth:

→ Mean: 4.47 Mb/s | Min: 1.98 Mb/s | Max: 7.28 Mb/s

Downlink (DL) — LIFBE Available Bandwidth:

→ Mean: 28.8 Mb/s | Min: 7.49 Mb/s | Max: 42.0 Mb/s

Key observations from this measurement campaign:

→ The DL/UL asymmetry is structural for OneWeb OW10HL (approximately 6:1), reflecting the beam allocation policy of the operator.

→ Ab variability is significant — DL ranges from 7.49 to 42.0 Mb/s within the same week, illustrating the dynamic nature of LEO satellite cross-traffic (beam load, handovers, Token Bucket effects).

→ Periodic DL peaks visible on the graph correspond to periods of low beam occupancy (off-peak hours, night), confirming the Token Bucket burst behaviour when the bucket is full at rest.

→ UL remains constrained throughout the week (1.98–7.28 Mb/s), consistent with the OW10HLOW profile's uplink allocation and the operator's asymmetric capacity planning.

This real-world dataset validates that LIFBE, deployed as a continuous production monitor on a LEO satellite link, provides actionable visibility into available bandwidth — enabling SLO verification, anomaly detection, and predictive alerting that no commissioning-only tool (iPerf3) can deliver.

ANNEX

THE TOKEN BUCKET

The Token Bucket is a traffic-shaping mechanism applied by the satellite operator to each terminal — independently of the physical capacity of the transponder or beam.

The Bucket-and-Tokens Analogy

The bucket fills continuously at a fixed rate

→ e.g. 50 Mbps contractual = 50 million bits per second added to the bucket

The bucket has a maximum size (allowed burst)

→ e.g. bucket capacity = 200 Mb = 4 seconds of fill

When you send data:

→ each bit consumes one token from the bucket

→ as long as tokens remain → passes at maximum physical rate (200 Mbps)

→ bucket empty → throughput throttled to contractual rate (50 Mbps)

On a Starlink Link Concretely

Contrat : 100 Mbps

Allowed burst: 200 Mbps for ~2 seconds (bucket full at rest)

Scenario:

t=0s: terminal at rest → bucket full

t=0→2s: you send data → passes at 200 Mbps (tokens available)

t=2s: bucket empty → throttling → drops to 100 Mbps contractual

t=...: bucket refills progressively during quiet periods

Why Does the Token Bucket Exist on LEO Satellite?

Three operator reasons:

The first is beam sharing — a Starlink beam serves hundreds of terminals simultaneously. Without regulation, a single terminal could monopolise capacity at the expense of others.

The second is the contractual guarantee — the operator commits to an average throughput (e.g. 100 Mbps), not an instantaneous rate. The Token Bucket smooths consumption to honour this commitment across all subscribers.

The third is priority management — professional maritime terminals, defence terminals, and residential subscribers have different Token Bucket profiles. This is the mechanism that implements this differentiation.

CROSS-TRAFFIC

Cross-traffic on LEO satellite is unique because it takes **three simultaneous and mutually invisible forms**.

Form 1 — Beam Sharing Between Terminals

This is the most structural cross-traffic. A Starlink beam covers a geographic cell and serves hundreds of terminals simultaneously. Each terminal sees its available bandwidth reduced by the activity of other terminals in the same cell — without ever seeing them.

Beam capacity: 2 Gbps (example)

Active terminals: 200

Invisible cross-traffic: cumulative traffic of the 199 other terminals

Terminal A's Ab: 2 Gbps − traffic of the 199 others = variable

This cross-traffic is completely invisible at the IP level — LIFBE measures its effect (queuing delays at the ground station) without being able to identify its source.

Form 2 — Ground Station Congestion

The ground station concentrates traffic from all beam terminals towards the terrestrial IP backbone. This is often the real bottleneck:

Beam → ground station → terrestrial link → Internet

Form 3 — Signalling and Management Traffic

The satellite itself generates permanent traffic invisible to the user:

Orbital telemetry → consumes control band capacity

Firmware updates → punctual bandwidth spikes

Beam handover → inter-satellite signalling

NTP synchronisation → permanent but low volume

What LIFBE Measures and Does Not Measure

LIFBE measures:

→ Ab = Ce − Rx_total

where Rx_total = form 1 + form 2 + form 3 cumulated

→ regardless of the cross-traffic source

→ LIFBE observes the global effect on its probe delays

LIFBE does not measure:

→ the distribution between the 3 forms

→ the identity of competing terminals

→ the ground station load in isolation

The Temporal Dynamics Specific to LEO

This is what makes LEO satellite cross-traffic even more complex than on wired or cellular networks:

t=0s: satellite A at zenith → strong signal → high Ce

t=15s: handover to satellite B → micro-outage → Rx spike

t=30s: 50 new terminals connect in the cell → Rx increases

t=45s: traffic peak (local peak hour) → Rx max → Ab min

t=60s: satellite B degrades at the horizon → Ce decreases AND Rx increases

In 60 seconds, Ab can vary by a factor of ten — a 10-second iPerf3 measurement captures only a partial window of this dynamic.

What the LatenceTech Analyzer Delivers on LEO Satellite

This is precisely why **continuous measurement** makes full sense:

Measurement every 30s over 24h:

→ temporal cross-traffic profile (peak hours, night)

→ detection of beam handovers (P99 latency spikes)

→ characterisation of operator Token Bucket (Ab distribution)

→ RSRP/RSRQ correlation with Ab (weak signal = reduced Ce = reduced Ab)

→ prediction of degradations 30–120s before application impact

In summary:

cross-traffic on LEO satellite is multi-layered, dynamic, and entirely invisible from the terminal. LIFBE does not see its causes — it continuously measures its effects, enabling the LatenceTech Analyzer to statistically characterise a link whose very nature is to vary continuously.

LIFBE

LIFBE Objective

LIFBE (Low Intrusive Fast Bandwidth Estimation) aims to estimate in real time the effectively available bandwidth (Ab) on an IP link — i.e. what an application can actually use at instant t, after deducting existing traffic — on any wired or wireless network type, without disrupting production traffic.

Measurement Principle

LIFBE sends a single train of 20 to 50 lightweight UDP probes between a sender and a receiver, analyses cumulative received delays packet by packet via piecewise regression, and detects the breakpoint at which queues start forming — this change in delay behaviour directly reveals the available bandwidth Ab, and in its advanced mode, the maximum link capacity Ce, with no prior knowledge of the infrastructure and without saturating the network.

The Central Relationship

$Ab = Ce - Rx$

Three Variables, Three Distinct Physical Realities:

Ce — Effective Link Capacity

The maximum pipe size — what the physical medium can carry under best conditions, with no other traffic. It depends solely on the physical medium and modulation.

Rx — Cross-traffic

All IP traffic already present on the link, emitted by sources other than LIFBE probes. LIFBE never sees it directly — it only measures its effect on its own probes.

Ab — Available Bandwidth

The only variable that truly matters to the application. What it can use right now, at this precise instant, without disturbing existing traffic.

How Does LIFBE Calculate Ab Without Knowing Ce or Rx?

This is the core of the invention — LIFBE observes neither Ce nor Rx directly. It sends a single train of N UDP probes and analyses the cumulative received delays:

In Three Sentences:

Ce is the pipe size — physical, fixed on fibre, variable on radio and satellite. Rx is what others are already consuming inside it — invisible, never measured directly. Ab is what remains — the only value that matters to your application, which LIFBE deduces by observing how its 50 UDP probes are delayed, without ever needing to know Ce or Rx separately.

Starlink & OneWeb Template Configuration for LIFBE:

Throughput values (maximum under ideal conditions) for various antennas available for LIFBE template calibration:

- Starlink Mini : 150 Mbps DL, 20 Mbps UL
- Starlink Standard : 250 Mbps DL, 25 Mbps UL
- Starlink High Performance : 350 Mbps DL, 40 Mbps UL
- Starlink Enterprise: 500 Mbps DL, 50 Mbps UL
- OneWeb OW10HL : 50 Mbps DL, 10 Mbps UL

OOOoooOOO