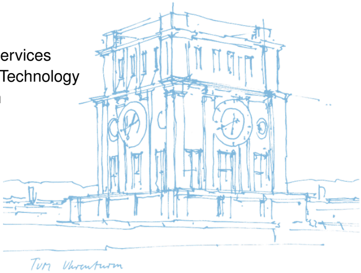
Development of a Framework for Retrieval of Parameters of the Starlink Dish

Final talk for the IDP by

Roberto Castellotti
advised by Leander Seidlitz, Johannes Zirngibl

Monday 27th November, 2023

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich



What is Starlink?

- Starlink is a Low Earth Orbiting (LEO) Satellite Constellation
- Brings Internet connection to remote areas
- 4000+ satellites with plans to launch more
- End users have a Dish to connect to satellites in sight
- Performance is higher compared to geostationary satellites (GEOSAT) based connections
- Satellites are orbiting at 550 km in height vs GEOSAT’s 35,000 km
- Average latency is 35 ms
- Download bandwidth is $>$ 100 MBps

Starlink 101

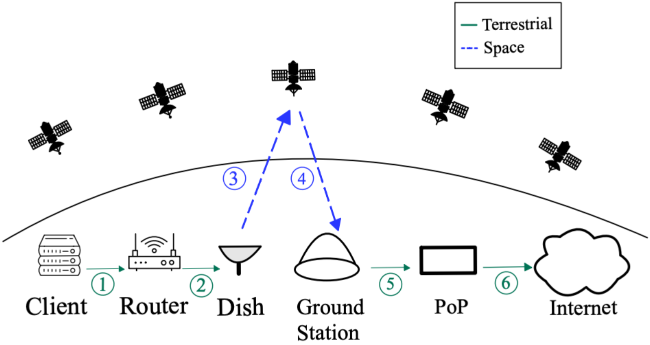


Figure 1: Starlink in a nutshell (ignoring Inter Satellite Links), from [1]

Our Dish

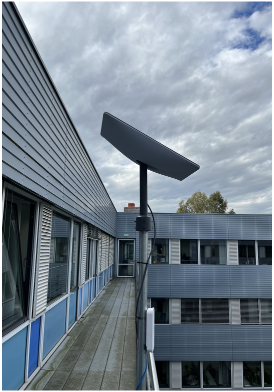


Figure 2: Our Starlink Dish

Problem Statement

should I remove this slide?

- Documenting gRPC API
- Measuring Latency to Point of Presence
- Understanding Routing Decisions
- Visualize Visible Satellites and patterns in their appearance
- Physical Layer Influences on Performance
- Retrieval of Obstruction Maps
- Satellite Handovers detection based on Obstruction Maps
- Correlation between Satellite Handovers and Bandwidth Drops

Documenting the gRPC API

We started by documenting the gRPC API running on the Dish

- We started by documenting the gRPC[1] API running on the dish
- API is reachable at `192.168.100.1:9200`
- The dish exposes a gRPC API with "Server Reflection"[2]
- 55 Endpoints are available
  - Majority of them do not work
  - 2 categories of errors: `Uninmplemented`, `PermissionDenied` and a some other errors
- Most interesting working Endpoints [3]:
  - `reboot`
  - `get_status`
  - `get_obstruction_map`

[1] `https://grpc.io` is a RPC framework from Google

[2] "Runtime construction of requests without having stub information precompiled into the client." `https://github.com/grpc/grpc/blob/master/doc/server-reflection.md`

[3] `https://gist.github.com/rcastellotti/e20630366dfeaeada6cc2680f562f6ac`

Measuring Latency to the Point of Presence

Can we measure Latency to the Point of Presence (PoP)?

- In short: we don't have to
- The `get_status` endpoint contains a `pop_ping_latency_ms` field
- We started polling the endpoint to gather latency to the Point of Presence
- Latency is pretty stable, averaging 35 ms with some irregular peaks
- Data reported by the API seems to be consistent with traceroutes

Measuring Latency to the Point of Presence

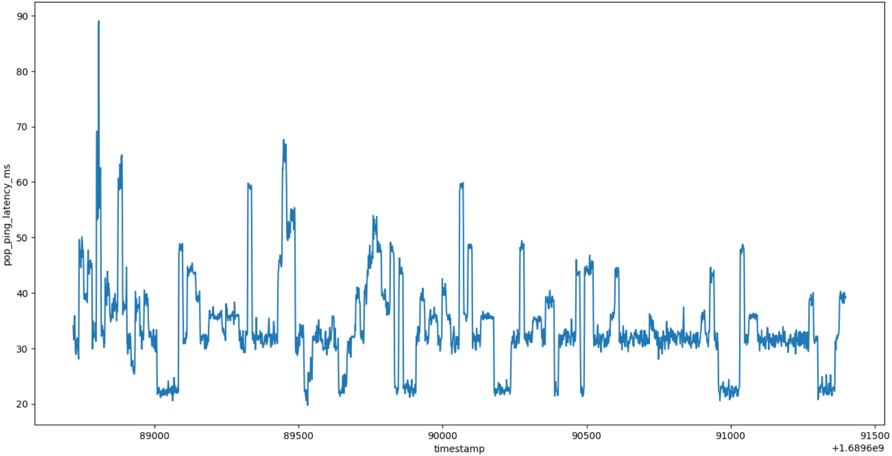


Figure 3: Visualizing latency to the Point of Presence.

Understanding Routing Decisions

Can we gather additional information about routing inside AS14593 [4]?

- Retrieved IP address blocks from major cloud providers (AWS, Azure, Oracle), position already known [5]
- Chose five geographically sparse targets around the globe, i.e., for aws:
  - `ap-northeast-2` Asia Pacific (Seoul)
  - `us-east-1` US East (N. Virginia)
  - `ap-south-1` Asia Pacific (Mumbai)
  - `sa-east-1` South America (São Paulo)
  - `me-south-1` Middle East (Bahrain)
- Tracerouted the targets over several days and saved data to later visualize it

[4] The Autonomous System SpaceX opearates; https://www.peeringdb.com/net/18747

[5] does not necessarily mean the last hop will be in that area (little information around what happens inside data centers), but a good enough approximation

Understanding Routing Decisions

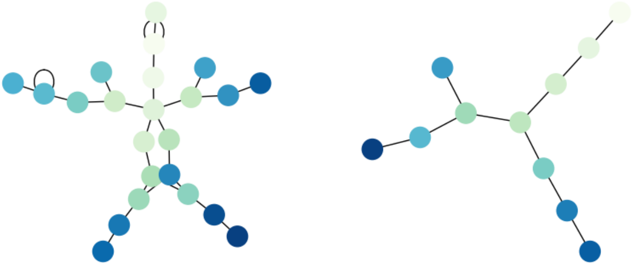


Figure 4: First seven hops of traceroutes to 5 AWS datacenters using ICMP; left is Starlink, right is cabled connection

Visualize Visible Satellites

Can we visualize visible satellites?

- Two-line element sets (TLE)[6]:
  - A data format encoding information abbout Earth-orbiting object for a given point in time
  - Allow for estimation of position and velocity at any point in past or future
- Downloaded a list of Starlink's satellites TLEs from celestrak.org
- Wrote a Python script to calculate visible satellites[7]
- Gathered information about satellite position
- Visualized patterns in satellite appearances.

[6] https://en.wikipedia.org/wiki/Two-line_element_set

[7] https://gitlab.lrz.de/netintum/teaching/tumi8-theses/idp-castellotti/-/blob/main/common.py?ref_type=heads#L132

Visualizing Patterns in Visible Satellites

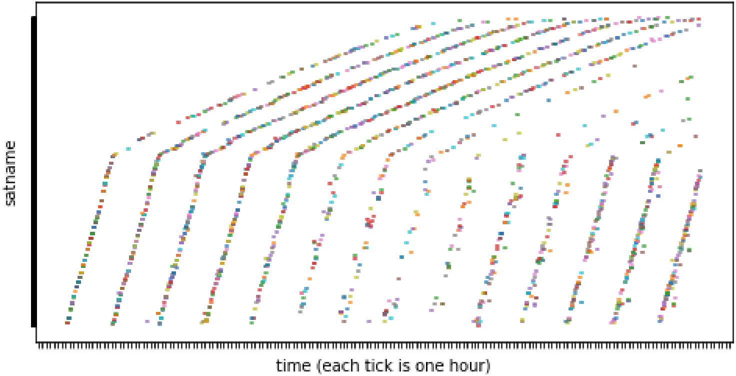


Figure 5: Visualizing patterns in Visible Satellites, we add a dot whenever we see a satellite

Physical Layer Influences on Performance

Does the physical layer have any influences on performance?

- We wanted to understand whether the physical layer influences RTT
- The Dish may decide to send a packet only when it fills a buffer
- Sent packets to a host we control with iPerf to create traffic on the interface, varying the payload size
- Downloaded Debian ISOs from 5 different mirrors (to neutralize upload speed differences)
- Measured RTTs

Physical Layer Influences on Performance

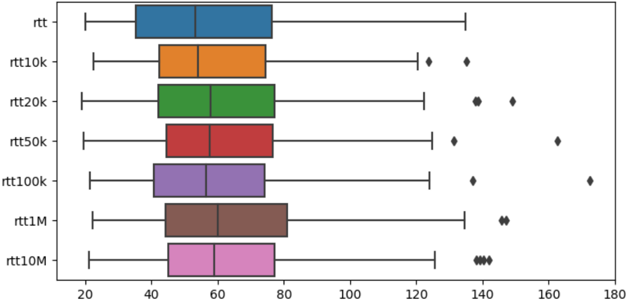


Figure 6: Physical Layer Influences on Performance

the dish_get_obstruction_map Endpoint

Can we use a side channel to extract satellite handovers from obstruction maps?

- An Obstruction Map captures the position where the Dish has seen satellites
- Designed to provide a way to report whether the dish positioning is optimal
- Following the approach described by Izhikevich et al. [1], we retrieved maps
  - Reboot the Dish to clear the Obstruction Map (`api.reboot`)
  - Poll the endpoint frequently enough to see satellite traces (`api.get_obstruction_map`)
  - Save the maps to visualize them later

Querying the `dish_get_obstruction_map` Endpoint

```
"apiVersion":"9",
"dishGetObstructionMap":{
    "minElevationDeg":10.0,
    "numCols":123,
    "numRows":123,
    "snr":[-1.0,-1.0,-1.0,-1.0,...,1.0,1.0,-1.0,-1.0]
}
```

- Interpret the `"snr"` field as a matrix, it contains 15129 (123*123) items
- Export this as images using Matplotlib, adding a timestamp for each map

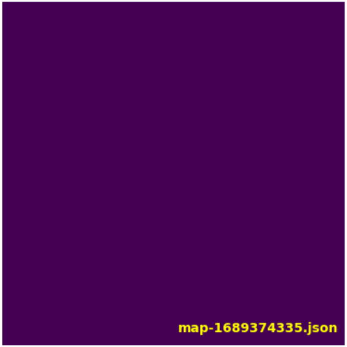


Figure 7: An Obstruction Map

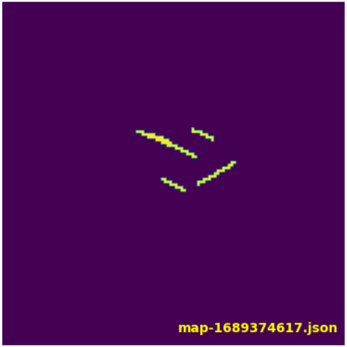


Figure 8: An Obstruction Map

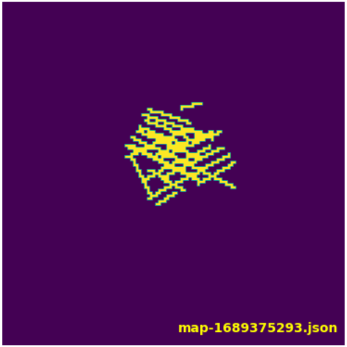


Figure 9: An Obstruction Map

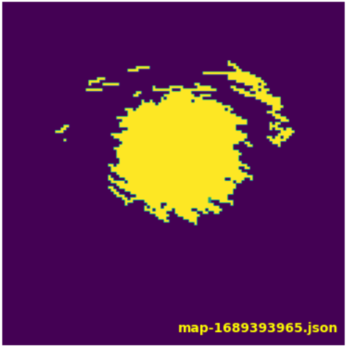


Figure 10: An Obstruction Map

Detecting Handovers Algoritmically

- Going through visualizations frame by frame is not feasible (1000s of frames)
- Following the map matrix interpretation, we assume:
  - "1" means a satellite was detected in that position
  - "-1" means no satellite was detected in that position
- Iterate through matrices two by two and sum them
- Check if, in the sum matrix, a "0" entry is "near" a "2" entry (inside a 3*3 matrix)
  - If it is "near" no handover was performed
  - If it is in complete different position an handover must have been performed

Obstruction Maps as Matrices (No Handover)

$$\begin{bmatrix} -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -2 & 2 & -2 & -2 \\ -2 & -2 & 2 & -2 \\ -2 & -2 & -2 & 0 \\ -2 & -2 & -2 & -2 \end{bmatrix}$$

Obstruction Maps as Matrices (Handover)

$$\begin{bmatrix} -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -2 & -2 & 2 & -2 \\ -2 & -2 & -2 & 2 \\ -2 & -2 & -2 & -2 \\ 0 & -2 & -2 & -2 \end{bmatrix}$$

Correlating Satellite Handovers and Bandwidth Drops

Is there any correlation between satellite handovers and drops in bandwidth?

- We now have an algorithm to detect handovers
- We run two scripts in parallel:
  - 1st: Retrieves an obstruction map every second
  - 2nd: Gathers Bandwidth data
- We visualize Bandwidth data and Satellite Handovers

Correlating Satellite Handovers and Bandwidth Drops

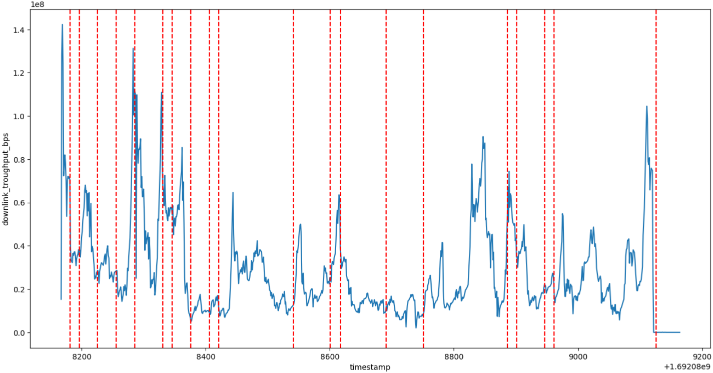


Figure 11: Correlating Satellite Handovers and Bandwidth Drops, blue is bandwidth, red vertical dashes are satellite handovers

Similar Technologies

- Eutelsat operates a LEO constellation, Oneweb [8], mainly serving Enterprises ar ments
- Amazon is currently launching its own LEO constellation: Project Kuiper [9], wit launch 3,236 satellites, with secured lanches (BlueOrigin)
- The China Aerospace Science and Technology Corporation plans to deploy satellite constellation [10]
- The increasing number of LEO satellites might cause problems, mainly:
  - Overall sky brightness increases (impacts astronomical observation)
  - Proliferation of debris

[8] https://oneweb.net

[9] https://www.aboutamazon.com/what-we-do/devices-services/project-kuiper

[10] https://spacenews.com/china-to-begin-constructing-its-own-megaconstellation-later-this-year/

estiamation

Final Remarks

We conclude that:

- Majority of the Endpoints on the gRPC API are not accessible
- PoP Latency is remarkably stable
- Physical layer has no remarkable influences on Performance
- Bandwidth is sustained consistently, even in the presence of satellite handovers
- Getting insights into inner routing is hard

Bibliography

[1] L. Izhikevich, M. Tran, K. Izhikevich, G. Akiwate, and Z. Durumeric.
Democratizing leo satellite network measurement, 2023.