



# Multi-Timescale Evaluation of Starlink Throughput

Johan Garcia  
Icomera AB  
Karlstad University  
Karlstad, Sweden  
johan.garcia@kau.se

Simon Sundberg  
Karlstad University  
Karlstad, Sweden  
simon.sundberg@kau.se

Giuseppe Caso  
Karlstad University  
Karlstad, Sweden  
giuseppe.caso@kau.se

Anna Brunstrom  
Karlstad University  
Karlstad, Sweden  
anna.brunstrom@kau.se

## ABSTRACT

Although Starlink has been rolled-out for several years, there is still a lack of knowledge regarding system details and performance characteristics. To address this, we perform a network layer measurement campaign utilizing precise timestamping to analyze throughput variations at multiple time scales, and infer system timing details. On larger timescales we quantify the diurnal variations where the throughput varies with the time of day. On the medium timescales we establish the likely frequency allocation and beam switching period to be 15 seconds. The associated connectivity disturbances contribute to severe link underutilization for single long-lived TCP flows, which typically reach only 46% of the estimated link capacity. On the sub-millisecond timescale our network layer measurements corroborate recent physical layer investigations of the Starlink frame timing, which is confirmed to be 1.33 ms.

## CCS CONCEPTS

• **Networks** → **Network measurement**; *Network performance analysis*.

## KEYWORDS

Diurnal Variation, Frame Timing, LEO

### ACM Reference Format:

Johan Garcia, Simon Sundberg, Giuseppe Caso, and Anna Brunstrom. 2023. Multi-Timescale Evaluation of Starlink Throughput. In *Workshop on Low Earth Orbit Networking and Communication (LEO-NET '23)*, October 6, 2023, Madrid, Spain. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3614204.3616108>



This work is licensed under a Creative Commons Attribution International 4.0 License.

*LEO-NET '23, October 6, 2023, Madrid, Spain*

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0332-4/23/10.

<https://doi.org/10.1145/3614204.3616108>

## 1 INTRODUCTION

Low-Earth orbit (LEO) systems, such as Starlink [19], enable high throughput Internet connectivity across a significant portion of Earth's landmass. In addition to providing Internet connectivity to end users, the Starlink network also opens up for many additional use cases. Providing passengers onboard vehicles such as trains and buses with Internet connectivity when traversing areas of no, or poor, cellular connectivity is one such use case, as is the use of Starlink for 5G backhaul connectivity [1, 2]. Other ancillary use cases include using the Starlink signals for positioning [14, 20], or as a passive radar [3]. In contrast to traditional Geostationary Equatorial Orbit (GEO) satellites, the dense constellations of LEO systems extend system coverage, resulting in significant throughput and latency benefits. As of July 2023, Starlink is the largest LEO satellite-based ISP, operating around 3500 satellites and being commercially available in the US from 2020 and in the EU from 2021.

Several works have performed empirical analyses of Starlink performance. In [13] TCP and QUIC protocols were used to study Starlink throughput, latency, packet loss, and web browsing performance which compared favorably to a GEO solution. Similarly, [8] characterized Starlink web performance via multiple vantage points and a dedicated browser extension, showing slightly better web performance compared to WiFi and cellular ISPs. Performance bottlenecks were also discussed, showing that packet loss, throughput, and latency were negatively affected by bad weather conditions, inter-satellite handovers, and the bent-pipe architecture.<sup>1</sup> The work in [10] also analyzed Starlink performance by executing experiments in urban/rural scenarios and under mobility, showing that Starlink exhibited slightly higher latency and lower throughput compared to a cable-based ISP, with performance affected by several environmental factors, in agreement with previous studies. The work in [22] investigated the use of Starlink for realtime multimedia services, showing that Starlink can support video-on-demand and live-streaming services if proper system configurations are used. However, the service quality decreased when Starlink

<sup>1</sup>Starlink currently uses a bent-pipe architecture, where the connected LEO satellite acts as a relay between the user and the closest Starlink ground station. The inter-satellite link mode [4] is expected to be available soon on a large scale. Initial and unconfirmed evidence of its use was reported in [22].

was used for interactive video conferencing in bad weather conditions.

Within the above context, we focus on downlink throughput performance and, differently from all previous work, we perform a fine-grained multi-timescale evaluation capable of detecting system-specific timing structures. Hence, rather than focusing on external factors (e.g., weather conditions), we aim at improving understanding of internal system features. Such improved understanding can be beneficial when creating realistic traces for trace-driven network simulations or emulations [6, 15], validating space-specific network emulators such as Starrynet [9], or devising tailored optimization solutions (e.g., upper-layer protocol modifications).

To the authors' best knowledge, our work is the first study exploiting network layer receiver-side measurements to infer details of the Starlink system in terms of frequency scheduling and beam switching periods, as well as frame timings, which has previously only been examined with specialized physical layer measurement setups [3, 7, 14].

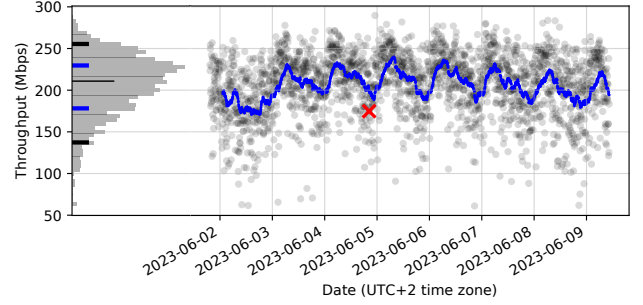
In this work we provide three main contributions:

- a network layer measurement campaign, and related study of multi-timescale throughput variations,
- the novel use of high-precision network timestamps for throughput variation and timing studies,
- a corroboration of beam switching and frame times recently deduced from physical layer measurements.

## 2 MEASUREMENT SETUP

The Starlink deployment includes a Gen-2 Starlink kit, mainly comprised of: a) a satellite dish with an electronic phased antenna; b) a motorized base for self-orientation of the dish; c) a WiFi router with an Ethernet adapter. The dish antenna is installed on the roof of the main building of the Department of Computer Science at Karlstad University, and connected to our measurement machine via a 1 Gbit Ethernet network, see [17] for further details.

We perform our measurement campaign using the Ookla speedtest command line tool [16] to generate traffic over the dedicated Starlink access to one specified speedtest server located in Stockholm, which is also where IP-geolocation places the Starlink exit node. Speedtest employs multiple parallel TCP connections and each measurement has a duration of 9 to 15 seconds [11]. We set up a cron-job to execute a test every 5 minutes over approximately a week, for a total of 2195 tests. The traffic generated by the tests is captured using tcpdump. Additionally, we use eBPF [21] to implement a tool to capture inter-packet delay (IPD) data. The IPD-tool collects timestamps from all packets at the tc-eBPF hook and calculates the link-wide IPDs, which it then stores along with packet sizes in a compact format using just 4 bytes per



**Figure 1: Average run throughput over time.**

packet. The IPD data is well suited for studying link throughput variations and has a much smaller storage footprint than pcap files.

The tests are run from a physical machine with a 4-core Intel i7-6700 CPU and 16 GB of memory, running Ubuntu 22.04 with a 6.3.2 Linux kernel. We use an Intel 10G X550T Ethernet network interface card (NIC), which can supply hardware timestamps for all received packets with a resolution of tens of nanoseconds. We disable Large Receive Offload (LRO) and Generic Receive Offload (GRO) so each individual network packet can be monitored, rather than the merged 64 KiB superpackets that the offloads may produce.

## 3 MULTI-TIMESCALE EVALUATION

### 3.1 Diurnal variation

We initially consider the data at the longest time scales, and quantify the existence of diurnal variation in throughput performance. The average throughput per measurement run for each of the 2195 runs is shown in Figure 1, together with the distribution of the values in the left part of the figure. Furthermore, we apply an exponential moving average (EWMA) with a halflife of 18 observations (i.e., 3 hours), and show this as the blue line in the graph. In the distribution to the left, the thick marks show the 5th and 95th percentiles of the throughput values (black) and the EWMA values (blue), while the tall thin line indicates the throughput median of 211 Mbps. While it is not easy to discern the diurnal variation from the raw measurement data, with the EWMA moving average, we can observe an apparent daily pattern with the maximum throughput obtained around 5 am local time (UTC+2). The diurnal variation expressed as the 5th and 95th percentile of the EWMA is between 178 and 230 Mbps. For the overall throughput measurements the corresponding percentiles are 137 and 255 Mbps. These results are consistent with the diurnal variation results reported in [8] for a shorter and sparser measurement run. The red mark in the plot indicates an example measurement run, which we now examine in more detail.

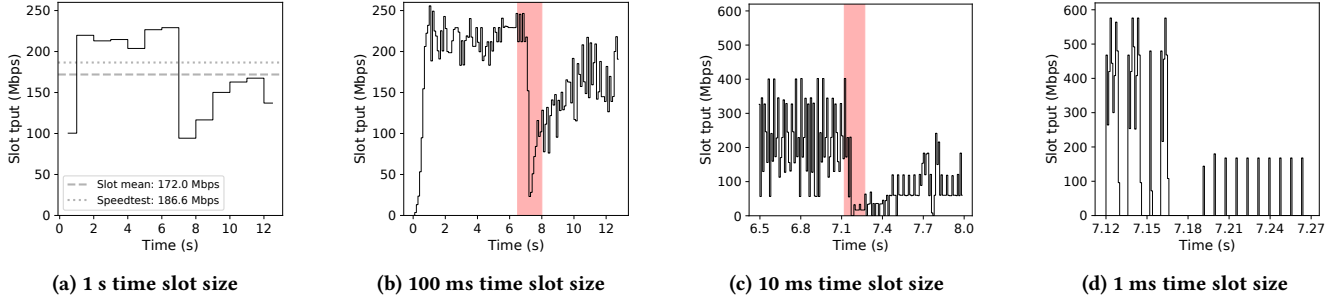


Figure 2: Multi-timescale view of one measurement run. Note the difference in y-axis scales.

### 3.2 Multi-timescale view of measurements

A multi-timescale representation of the throughput evolution of the example run is provided in Figure 2. The four subgraphs show the throughput evolution of the measurement run when the throughput is averaged over time slots of varying durations.

Looking at the coarse-grained throughput evolution in Figure 2a, we can see that there is a considerable drop in throughput in the 7th second which is then not completely recovered during the remainder of the measurement run. The graph also shows the throughput reported from the speedtest tool, and the throughput mean calculated across the displayed time slots. In the 100 millisecond time slots graph in Figure 2b, a more granular representation of the same time interval is provided. It can be observed that the decrease in throughput goes almost all the way down to zero when considering the average over a 100 millisecond slot.

Figure 2c shows a zoom-in of the red-marked region in Figure 2b that corresponds to the throughput drop. Here we can observe that there is a very large variation at the 10 millisecond time scale. In the left part, before the drop, the observed throughput regularly varies between 50 and 400 Mbps within a few 10 ms time slots of each other.

Finally, in Figure 2d we consider a further zoom-in of the critical part where the throughput decrease occurs. At millisecond resolution the variation is even more pronounced with frequent time slots having zero throughput. We can also see clearly the change in behavior when going from the higher throughput down to the lower throughput. The left part of the plot has longer bursts, with high throughputs, leading to the average throughput of roughly 220 Mbps apparent in the corresponding time region of Figure 2c. This is followed by a silent period of almost 26 milliseconds.

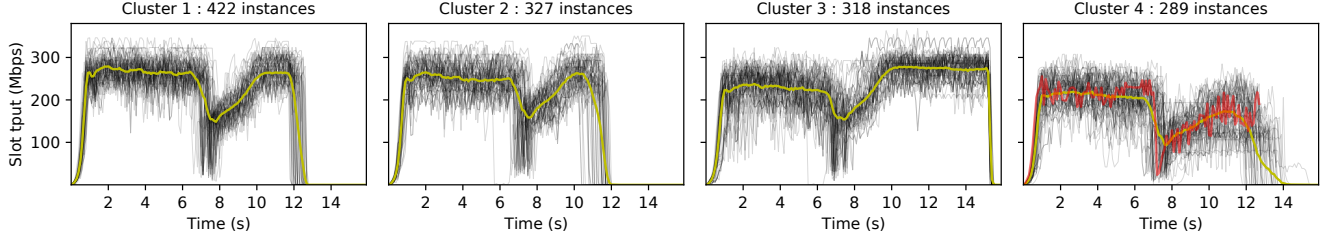
After that, there is low throughput while the longer-term throughput starts to recover. In the right half of the 1 ms scale plot we observe small millisecond spikes that are regularly placed with an identical time distance between them. In this example run, the sudden fall in throughput just before

7.2 seconds is an unexpected behavior. Judging from a single run it is impossible to determine if such behavior is typical, or what the underlying reason might be. Thus, we next consider the 100 ms time slot behavior of the entire data set.

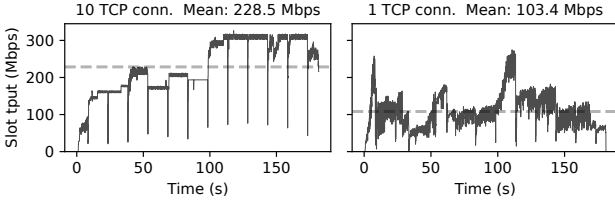
### 3.3 Clustering of per-run throughput

To examine the presence of sudden throughput reductions, or other consistent differences in the results, we perform time series clustering to identify typical behaviors across the 2195 measurement runs. We apply straightforward Euclidean k-means clustering, as it proved to be most visually informative in comparison to Dynamic Time Warping (DTW) and soft DTW, which we also examined. The number of clusters is set to 8 in order to capture a reasonable amount of variation in cluster structure. The results from clustering with 100 ms time slots are shown in Figure 3 for the 4 clusters with largest number of instances out of the 8. For every cluster the centroid of the cluster is shown with a yellow line, along with 50 randomly selected example runs from the cluster. The measurement run shown in Figure 2b was allocated to cluster 4, and is shown as the red line. Some notable aspects are visually apparent, such as the duration of the runs being different as a consequence of the speedtest software not using a single fixed duration for tests.

Noticeable in all cluster centroids is a kink around the 7 to 8 seconds time span, where a considerable reduction in throughput is evident. We note that the clustered centroid does not fully capture the extent of this drop since the location of the drop can vary a little bit in time, and thus becomes averaged out so that the yellow line does not go as deep down as each individual experimental run does. The remaining 4 clusters showed similar trends as the 4 displayed clusters, mainly varying in run duration, which overall varied between 9 to 15 seconds. To confirm the underlying cause for the observed kink, we performed additional measurement runs using iperf3 with 10 second test duration and 10 connections (had kink), using 5G as access (no kink), and then also



**Figure 3: Time series clustering over the 2195 measurement runs. Four out of eight clusters are shown. Notable is that practically all runs show a marked throughput decrease located at a point varying between 6.5 and 8 seconds.**



**Figure 4: Throughput of 3 min iperf3 runs, 100 ms slots.**

long running (3-min) iperf3 tests which provided crucial information in regards to the observed consistent throughput drop as detailed below.

### 3.4 The 15 seconds throughput anomaly

We performed 25 3-minute iperf3 measurement runs with 10 connections, illustrated in the left graph in Figure 4 with the run which had the median average throughput out of the 25 runs. With this longer timescale, it is apparent that kinks systematically occur every 15 seconds. The reason for the kink position being relatively stable in time in Figure 3 is the implicit synchronization between the cron-job executing the test runs every 5 minutes, and the underlying Starlink process generating these 15 second drops, which consequently occur in even multiples over the 5 minute inter-test time. We confirmed this with additional cron runs using a different start offset.

We cannot, from our data, determine the process within Starlink responsible for this anomaly. However, a recent work [3] uses high-performance radio hardware in a passive radar context and performs a Starlink signal analysis. Based on observations in spectrograms, the authors state that "frequency allocation and beam switching appear to change in intervals of approximately 15.5 s". Such a frequency reallocation and beam switching process is consistent with the connectivity disruptions we observe, and thus we find it likely that it is the underlying cause, and that the actual interval is 15 seconds as given by our high-precision measurements.

All measurement runs considered so far have used multiple parallel TCP connections so as to fully load the network for link characterization. For the 3-minute iperf3 test runs we additionally interleaved runs with a single TCP connection, in-between the 10-connection runs. These 1-connection runs are representative of the throughput observed by a single long running Linux Cubic [18] TCP connection, such as a large file transfer. The results for the 1-connection run with median throughput are shown in the right graph of Figure 4. The kinks are no longer as clearly visible, but that is simply because the single TCP connection struggles to fully utilize the available link capacity to the same extent as for the 10-connection runs. As these results show, influenced by the periodic Starlink throughput kinks, the 1-connection effective throughput is only 47 percent of the 10-connection throughput for these two median runs, and it is 46 percent for the overall average across the 25 runs.

### 3.5 Micro-scale throughput variation

We now continue the analysis focusing on shorter time scales, and in particular on the burst behavior observed in Figure 2d. There, for both high and low throughput regimes, the arrival of traffic appears to be bursty with millisecond breaks between traffic bursts. We can recall that one 1500 byte packet arriving every millisecond correspond to a rate of 12 Mbps. Consequently, if packets were somewhat evenly spaced there would be no milliseconds without traffic given the typical 100+ Mbps throughput observed in our measurements.

During the experiments the throughput observed at the receiver can reasonably be assumed to mostly be the result of constraints in the bottleneck link, which here would be the Starlink access. Thus, the queue in front of the bottleneck link will be non-empty so that there are almost always packets to be sent when the constrained link can accept them.

To allow further exploration of the burst behavior, we process our per-packet data to locate the bursts. To delineate the bursts, we use a minimum interburst delay of 1.5 ms and require a burst to have at least two packets to be considered as a burst. Since we are interested in the burst behavior



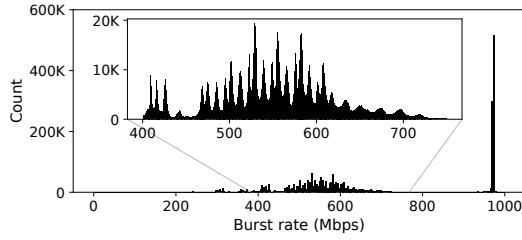


Figure 5: Burst rate distribution.

during full traffic load, we consider only the traffic that are present within 1 second margin away from the start and end of the speedtest measurement run, i.e., one second in from the start and end of the measurements runs illustrated in Figure 2b. With this definition of the bursts, we process the 422 million packets that are within the margin, and locate 2375690 bursts.

As a first step, we consider the distribution of burst rates, i.e., the average throughput within each burst which is simply the number of bytes in the burst multiplied by 8 and divided by the burst duration. The duration is measured with the help of the high-precision hardware timestamping functionality in the NIC. The distribution of the burst rates is shown in Figure 5. We can observe a tall peak with burst rates in the region above 970 Mbps, which is close to back to back packet transmission when one considers the overhead of Ethernet headers and inter-frame distances. These bursts with very high burst rates form a group of 863,998 bursts out of the almost 2.4 million. Here we note that the Starlink equipment is connected with Gigabit Ethernet, and it is known that Ethernet networks can show a bursty behavior where part of the explanation is the hardware offloading functionality in the sending NIC [23]. The NIC hardware accepts larger data chunks into its buffers and then empties these buffers at line rate after it has performed offloading operations such as TCP segmentation and checksumming [5].

The second set of bursts are those where the burst rate varies between 150 and 750 Mbps, where the majority of bursts lie within the range of 400 to 750 Mbps. The distribution of this second group is shown in the inset in Figure 5. There are a number of apparent peaks for the burst rates, which fits well with the behavior observed for the one millisecond graph shown in Figure 2d where some particular rate values occurred several times.

Figure 6 shows the distribution of burst lengths in packets, and burst duration in milliseconds, for the two burst groups. It is clear that there is a qualitative difference between these two groups. The high burst rate group has a short burst length and burst duration, and the distribution in terms of length and duration is similar. This can be contrasted to the group with lower burst rates, where there is a marked

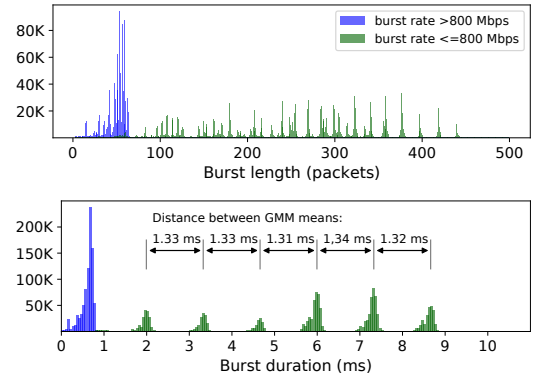


Figure 6: Burst length and duration distributions.

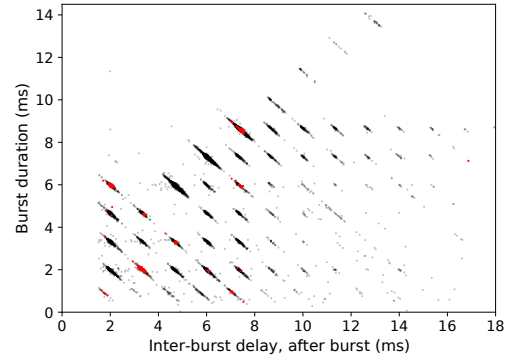


Figure 7: Burst duration vs. interburst delay scatterplot.

difference in distribution between burst length and burst duration. While the burst length shows an assortment of distinct peaks, the burst duration shows six consistent peaks with equal distance between them. Recent work [7] using specialized signal capture equipment deduced that the frame period,  $T_f$ , is  $1/750 \approx 1.33$  ms. Applying Gaussian Mixture Modeling (GMM) [12] to our measurements corroborates that the mean distance between peaks becomes 1.33 ms. The larger numbers of peaks for the burst lengths in packets in comparison to the related burst duration in milliseconds is due to the use of different modulations, such as 4-QAM and 16-QAM [7], which results in different transmission rates, and thus different amount of packets for the same burst duration.

We next consider only observations with burst rate lower than 800 Mbps. Burst duration and interburst delay, i.e., the duration of the break after a burst, are shown in Figure 7. We restrict the x and y ranges to highlight an appropriate region based on the marginal distributions, and subsample by a factor 100 from the approximately 1.5 million bursts falling within the displayed region. In the figure, in which we also overlay in red the 545 bursts present in the example run

shown in Figure 2, there are several notable aspects. There are horizontal bands in the figure which correspond to the burst durations observed in Figure 6, as would be expected. In addition, it can be observed that also the interburst delay shows similar, but vertical, banding where there are distinct regions of burst delays that are spaced with similar time distance as the burst durations. Further, diagonal banding can also be noted within the many small clusters of time-constrained observation collections. This marked diagonal banding visible in the figure is again consistent with time slotting behavior, where the sum of the burst duration and interburst delay has a few fixed values that is some multiple of a fixed sub-slot size.

## 4 CONCLUSIONS

Using a hardware timestamping measurement setup we have collected and analyzed a large number of Starlink measurements. We quantified the Time-Of-Day effects, and enhanced the understanding of Starlink throughput variation by visualizing over several timescales. We observed and analyzed the persistent presence of periodic throughput drops consistent with 15 second interval frequency reallocation and beam switching. Further, our measurements corroborate recent physical layer determination of the frame time as 1.33 ms. These results highlight the utility of network layer assisted inference of Starlink system aspects and point towards future opportunities of utilizing these techniques for research on scheduling, modulation switching behavior, and other system details.

## ACKNOWLEDGMENTS

The authors wish to thank Jonas Karlsson and Tobias Vehkärvi for assistance with the data collection. This research was partly funded by the Swedish KK-foundation.

## REFERENCES

- [1] 3GPP. 2022. 3GPP TS 23-700-27. Study on 5G system with satellite backhaul (Release 18). *3GPP TS 23-700-27 v18.0.0 (2022-12)* (2022).
- [2] M Mahdi Azari, Sourabh Solanki, Symeon Chatzinotas, Oltjon Kodheli, Hazem Sallouha, Achiel Colpaert, Jesus Fabian Mendoza Montoya, Sofie Pollin, Alireza Haqiqatnejad, Arsham Mostaani, et al. 2022. Evolution of non-terrestrial networks from 5G to 6G: A survey. *IEEE communications surveys & tutorials* (2022).
- [3] Rodrigo Blázquez-García, Diego Cristallini, Martin Ummenhofer, Viktor Seidel, Jörg Heckenbach, and Daniel O'Hagan. 2023. Experimental comparison of Starlink and OneWeb signals for passive radar. In *2023 IEEE Radar Conference (RadarConf23)*. IEEE, 1–6.
- [4] Aizaz U. Chaudhry and Halim Yanikomeroglu. 2021. Laser Intersatellite Links in a Starlink Constellation: A Classification and Analysis. *IEEE Vehicular Technology Magazine* 16, 2 (2021), 48–56. <https://doi.org/10.1109/MVT.2021.3063706>
- [5] Yuchung Cheng and Neal Cardwell. 2016. Making Linux TCP Fast. In *Netdev 1.2, The Technical Conference on Linux Networking*. Tokyo, Japan. <https://netdevconf.info/1.2/session.html?yuchung-cheng>
- [6] Johan Garcia and Per Hurtig. 2016. KauNetEm: Deterministic network emulation in Linux. In *Netdev 1.1, The Technical Conference on Linux Networking*. Seville, Spain.
- [7] Todd E Humphreys, Peter A Iannucci, Zacharias M Komodromos, and Andrew M Graff. 2023. Signal structure of the starlink ku-band downlink. *IEEE Trans. Aerospace Electron. Systems* (2023).
- [8] Mohamed M Kassem, Aravindh Raman, Diego Perino, and Nishanth Sastry. 2022. A browser-side view of starlink connectivity. In *22nd ACM Internet Measurement Conference (IMC '22)*. 151–158.
- [9] Zeqi Lai, Hewu Li, Yangtao Deng, Qian Wu, Jun Liu, Yuanjie Li, Jihao Li, Lixin Liu, Weisen Liu, and Jianping Wu. 2023. StarryNet: Empowering Researchers to Evaluate Futuristic Integrated Space and Terrestrial Networks. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. 1309–1324.
- [10] Sami Ma, Yi Ching Chou, Haoyuan Zhao, Long Chen, Xiaoqiang Ma, and Jiangchuan Liu. 2022. Network Characteristics of LEO Satellite Constellations: A Starlink-Based Measurement from End Users. *arXiv preprint arXiv:2212.13697* (2022).
- [11] Kyle MacMillan, Tarun Mangla, James Saxon, Nicole P Marwell, and Nick Feamster. 2023. A Comparative Analysis of Ookla Speedtest and Measurement Labs Network Diagnostic Test (NDT7). *Proc. ACM on Measurement and Analysis of Computing Systems* 7, 1 (2023), 1–26.
- [12] Geoffrey McLachlan and David Peel. 2000. *Finite Mixture Models*. John Wiley & Sons Hoboken, NJ.
- [13] François Michel, Martino Trevisan, Danilo Giordano, and Olivier Bonaventure. 2022. A First Look at Starlink Performance. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC '22)*. 130–136. <https://doi.org/10.1145/3517745.3561416>
- [14] Mohammad Neinavaie and Zaher M Kassas. 2023. Unveiling Starlink LEO satellite OFDM-like signal structure enabling precise positioning. *IEEE Trans. Aerospace Electron. Systems* (2023).
- [15] Ravi Netravali, Anirudh Sivaraman, Somak Das, Ameesh Goyal, Keith Winstein, James Mickens, and Hari Balakrishnan. 2015. Mahimahi: Accurate Record-and-Replay for HTTP. In *USENIX Conference on Usenix Annual Technical Conference (USENIX ATC '15)*. 417–429.
- [16] Ookla. 2023. *Speedtest CLI: Internet Speed Test for the Command Line*. Retrieved June 16, 2023 from <https://www.speedtest.net/apps/cli>
- [17] Mohammad Rajiullah, Giuseppe Caso, Anna Brunstrom, Jonas Karlsson, Stefan Alfredsson, and Ozgu Alay. 2023. CARL-W: a Testbed for Empirical Analyses of 5G and Starlink Performance. In *In 3rd ACM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases (5G-MeMU '23)*. <https://doi.org/10.1145/3609382.3610510>
- [18] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger. 2018. *CUBIC for Fast Long-Distance Networks*. RFC 8312. RFC Editor.
- [19] Starlink. 2023. *Order Starlink*. Retrieved July 6, 2023 from <https://www.starlink.com/>
- [20] Winfried Stock, Christian A Hofmann, and Andreas Knopp. 2023. LEO-PNT with Starlink: Development of a burst detection algorithm based on signal measurements. In *WSA & SCC 2023; 26th International ITG Workshop on Smart Antennas and 13th Conference on Systems, Communications, and Coding*. VDE, 1–6.
- [21] Marcos AM Vieira, Matheus S Castanho, Racyus DG Pacifico, Eleron RS Santos, Eduardo PM Câmara Júnior, and Luiz FM Vieira. 2020. Fast packet processing with ebpf and xdp: Concepts, code, challenges, and applications. *ACM Computing Surveys (CSUR)* 53, 1 (2020), 1–36.
- [22] Haoyuan Zhao, Hao Fang, Feng Wang, and Jiangchuan Liu. 2023. Real-time Multimedia Services over Starlink: A Reality Check. In *Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video*. 43–49.
- [23] Yongxiang Zhao, Baoxian Zhang, Cheng Li, and Changjia Chen. 2017. ON/OFF Traffic Shaping in the Internet: Motivation, Challenges, and Solutions. *IEEE Network* 31, 2 (March 2017), 48–57.