Computer Communications 234 (2025) 108090

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Modeling and predicting starlink throughput with fine-grained burst characterization

Johan Garcia [a,b,*], Matthias Beckerle [a], Simon Sundberg [a], Anna Brunstrom [a]

[a] Department of Mathematics and Computer Science, Karlstad University, Karlstad, Sweden
[b] Icomera AB, Gothenburg, Sweden

ARTICLE INFO

Keywords:
Starlink
Low-earth orbit satellite networks
Throughput modeling
N-gram prediction

ABSTRACT

Leveraging a dataset of almost half a billion packets with high-precision packet times and sizes, we extract characteristics of the bursts emitted over Starlink's Ethernet interface. The structure of these bursts directly reflects the physical layer reception of OFDMA frames on the satellite link. We study these bursts by analyzing their rates, and thus indirectly also the transition between different physical layer rates. The results highlight that there is definitive structure in the transition behavior, and we note specific behaviors such as particular transition steps associated with rate switching, and that rate switching occurs mainly to neighboring rates. We also study the joint burst rate and burst duration transitions, noting that transitions occur mainly within the same rate, and that changes in burst duration are often performed with an intermediate short burst in-between. Furthermore, we examine the configurations of the three factors burst rate, burst duration, and inter-burst silent time, which together determine the effective throughput of a Starlink connection. We perform pattern mining on these three factors, and we use the patterns to construct a dynamic N-gram model predicting the characteristics of the next upcoming burst, and by extension, the short-term future throughput. We further train a Deep Learning time-series model which shows improved prediction performance.

1. Introduction

Low-Earth orbit (LEO) systems have made it possible to provide Internet connectivity across wide areas that were previously underserved in terms of wireless communication access. Such access has generally been provided by cellular networks with limited coverage in sparsely populated areas. Providing passengers onboard vehicles such as trains and buses with Internet connectivity when traversing areas of no, or poor, cellular connectivity is also a use case for LEO systems, as is the use of Starlink for 5G backbone connectivity [1,2]. In contrast to cellular networks, the details of LEO systems, and SpaceX's Starlink [3] in particular, is opaque. Thus, many system design decisions can only be inferred through measurements as the amount of publicly provided information on system details is limited. In this work, which is an extension of our previous conference paper [4], we delve into the technical details of Starlink.

Several measurement studies have been performed to evaluate different aspects of the Starlink network [5–8]. In this work, we in particular focus on the satellite downlink mechanisms that yield a particular observed throughput at the network layer. By utilizing a large dataset of high-precision per-packet measurements, we study the behavior of the transmission bursts that are associated with how Starlink allocates system resources between concurrent users.

Previous studies [6,9] have noted the existence of a 15 s reconfiguration interval, as well as an OFDMA frame time of $1.\overline{33}$ ms. Our recent study [10] on physical layer details highlight that the data in these frames is transmitted in bursts observable at the network layer. By analyzing the timings of these bursts, it is possible to infer further detail regarding the physical layer characteristics. The number of frames per burst in our measurements typically vary between 1 and 7. Between these bursts, there are inter-burst pauses, where no data is received by the user.

On longer time scales (i.e. multi-burst time scales) we note that the throughput obtained by a user is a function of (a) the physical transmission rate of the frames, (b) the number of frames in a burst, and (c) the inter-burst silent time. Throughput is thus a combination of the prevailing radio channel conditions as managed by (a) and the scheduling across competing users as managed by (b, c). We here focus on an investigation of these factors with the aim to obtain additional knowledge regarding Starlink's rate transition behavior. In comparison to the previous conference paper [4], we here provide more details on

* Corresponding author at: Department of Mathematics and Computer Science, Karlstad University, Karlstad, Sweden.
E-mail addresses: johan.garcia@kau.se (J. Garcia), matthias.beckerle@kau.se (M. Beckerle), simon.sundberg@kau.se (S. Sundberg), anna.brunstrom@kau.se (A. Brunstrom).

https://doi.org/10.1016/j.comcom.2025.108090
Received 30 May 2024; Received in revised form 24 December 2024; Accepted 1 February 2025
Available online 8 February 2025

the patterns of changes for these three factors, and how we can predict the likely burst characteristics of the next future burst.

Building on our previous studies of Starlink burst rates [9,10], we here study how the transition between rates are performed within the Starlink network. Based on the analysis of 2.4 million data bursts observed in our dataset of around half a billion packet sizes and arrival times we arrive at four main insights:

- rate changes in the Starlink physical transmission rates predominantly occur between adjacent rates,
- changes to burst rate or length are usually done in conjunction with a short transitory burst,
- a clear relationship can be empirically observed between the effective throughput and the underlying system mechanisms of burst rates and burst timings,
- and the patterns of changes in burst rates and timings can be mined, and used to construct N-gram and Deep Learning based predictors of upcoming burst characteristics.

We believe that the findings reported in this paper can be applicable to analytical, simulation, and emulation based modeling of the Starlink link throughput variations, and provide additional knowledge regarding the Starlink physical layer behavior. Given the proprietary nature of the Starlink network, documentation on system design details is scarce, and we believe that measurement and modeling studies like this one can contribute to an enhanced system understanding.

The structure of this article is as follows. In the next section we review related work, followed by a section on data collection. In sections 4 and 5 we leverage our earlier work [9,10] to provide a background on Starlink throughput characteristics and burst rate distribution and the associated clusters. Section 6 then models the transitions between burst rates, which is extended in Section 7 to jointly model rate and duration transitions. The burst rate, burst duration, and inter-burst delay maps to an effective throughput, which is modeled and empirically examined in Section 8. Next, in Section 9 we elaborate on how sequence pattern mining can be used to find the typical patterns of changes in burst characteristics. Section 10 considers how these patterns can be used to construct prediction models. Finally, Section 11 provides the conclusions.

2. Related work

Research on inferring Starlink system level details based on network layer measurements is relatively limited. Our previous investigations have explored Starlink's throughput across various time scales [9] going from days to milliseconds. That work was the first to confirm both the reconfiguration interval and the frame timing based exclusively on network layer measurements. We made an initial observation of the bursty behavior of received Starlink traffic from the CPE Ethernet interface, and made an initial characterization of burst durations and inter-burst pauses. A later deepened analysis [10] showed that our first initial burst characterization however had incorrect burst time measurements due to the sub-bursts occurring within the bursts. We devised a correction mechanism, and by using these updated burst characteristics we were able to construct a radio resource allocation model that provides specific details to the general frame structure described in [11]. Our analysis showed that a model where each Starlink OFDMA frame uses 287 symbols over 1000 subcarriers for user data transport of full size packets provides an excellent match for the base rate found in our empirical measurements.

More broadly, the existing literature that relates to our study can be categorized into two principal groups. The first group comprises measurement-focused studies, which examine network performance in terms of throughput and latency as perceived by users [5,8,12]. In [13] several aspects of Starlink are explored through comprehensive measurements. In particular, latency and throughput are characterized along with environmental impact factors such as precipitation. The study also reports on data collected in sparsely populated remote areas. Similarly focusing on the effect of precipitation, [14] collect measurements at two central Europe locations, and report an impact from rain on downlink throughput. The potential impact on Starlink performance by VPN usage along with the use of CUBIC or BBRv2 congestion control is examined in [15]. Focusing on latency, [16] uses the HitchHiking distributed measurement approach to collect data from more than 2400 Starlink users, and highlight the interplay between Starlink's use of Inter-satellite links and user latency. Also focusing mainly on latency aspects, [17] measure the one way delay both outside-in and inside-out from the Starlink access network, also reporting increased variance caused by inter-satellite links.

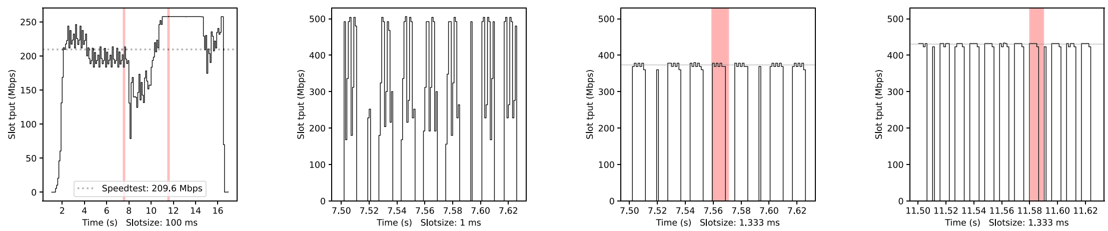
Starlink also has the potential to be used in conjunction with cellular infrastructure to provide better connectivity than what is possible with a single technique in isolation. In [18] three cellular operators and two Starlink setups are measured under mobility, with the conclusion that Starlink outperforms cellular in open areas, and that the complementary nature of the networks offer potentially improved performance. Also examining joint cellular and Starlink measurements, [19] examines the coverage and reliability of Starlink and one cellular network under mobility in sparsely populated regions around the arctic circle, arriving at similar conclusions. Mobility is also considered in [20], which examines Starlink performance from the vantage point of a service van driving in central Europe, highlighting the reduced performance observed in urban areas.

Typically, these measurement studies do not gather data with the level of timing detail and accuracy necessary for in-depth inference about system characteristics. Their primary aim is to describe the performance as experienced by users and the influencing elements, like satellite visibility and meteorological conditions. In the study by Tanveer et al. [7], the investigation measures round-trip times and packet loss rates using probing packets sent every 20 ms. Another study by Mohan et al. [6] assesses latency at intervals of 3 ms and throughput with 100 ms granularity. In contrast, our analysis enables us to deduce throughput with a level of detail extending to instantaneous per-packet throughput, thanks to timing measurements accurate to tens of nanoseconds.

The second category of relevant research originates from the fields of electrical and/or aeronautical engineering. These studies are primarily concerned with the physical layer details of the Starlink system, often to aid secondary applications like positioning [21–23] or passive radar systems [24]. They employ specialized radio technology to scrutinize the physical layer properties. In these studies, measurements of user data throughput are not a focus, as the interest lies in these secondary applications. An important contribution to our understanding of the physical layer comes from the blind signal identification work conducted in [11], where details of Starlink's Ku-band transmissions were uncovered, including the overall frame structure.

3. Data collection

To collect data we perform a measurement campaign utilizing a Starlink deployment on the roof of the main building of the Department of Computer Science at Karlstad University. Our Starlink deployment includes a Gen-2 Starlink kit, which includes: (a) a satellite dish with an electronic phased antenna; (b) a motorized base for self-orientation of the dish; and (c) a WiFi router with an Ethernet adaptor. It is connected to our measurement machine via an Ethernet network, see [25] for further details. To generate measurement traffic over the Starlink access, we utilize the Ookla Speedtest command line tool [26] using a specified Speedtest server located in Stockholm. An IP-geolocation on the Starlink exit node indicates that it is also located in Stockholm. Each Speedtest run employs multiple parallel TCP connections and each measurement run has a duration of 9 to 15 seconds [27]. We set up a cron-job to execute a run every 5 min over approximately a week, for a total of 2195 runs. The traffic generated by the measurement



(a) DL throughput, 100 ms slotsize (b) Left band (1a), 1 ms slotsize (c) Left band (1a), $1.\overline{33}$ ms slotsize (d) Right band (1a), $1.\overline{33}$ ms slotsize

Fig. 1. Starlink downlink throughput over two time scales. The red regions in Fig. 1a are shown in Fig. 1b-d.

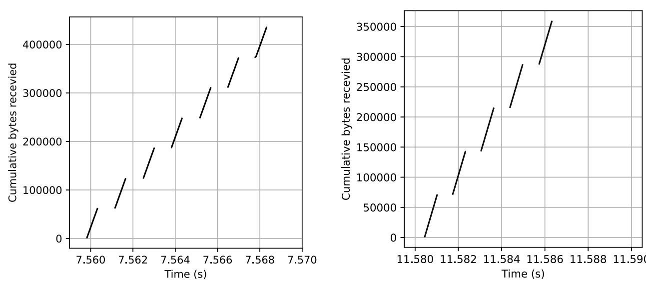


Fig. 2. The seven sub-bursts (each with data from one Starlink OFDMA frame) within the red burst in Fig. 1c (left), and the five sub-bursts from the red burst in Fig. 1d (right).

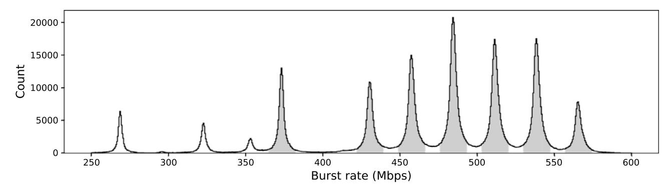


Fig. 3. Burst rate distribution for all burst durations, with cluster extent marked in gray.

runs is captured using tcpdump. Additionally, we use eBPF [28] 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. The IPDs are then stored along with packet sizes in a compact format using a minimum of just 4 bytes per packet, as described in [29]. The IPD data is well suited for studying link throughput variations at multiple timescales, and has a much smaller storage footprint than pcap files.

The measurements are collected 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 precision 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.

4. Starlink throughput characteristics

To provide some background on Starlink throughput variation, and illustrate the ability of our measurement approach to examine Starlink throughput at varying time scales, we now provide an initial overview of Starlink downlink throughput. We illustrate using one of the 2195 measurement runs, choosing the displayed timescales to be relevant for our subsequent discussion on Starlink's physical rate and its temporal variation.

Fig. 1 illustrates the throughput evolution over time for the example run at two timescales. The complete measurement run, shown in Fig. 1(a), shows the throughput as it appears when calculated over slots of 100 ms. Notably, there is significant variation in Starlink throughput during this measurement run, with a distinct dip around the 8-second mark. This dip corresponds to the Starlink network's periodic reconfiguration which takes place every 15 s. We have in a previous work [9] identified this periodic dip based on network measurements, and assessed its potential impact on TCP throughput performance. The issue of reconfiguration interval is also discussed in [6].

In Fig. 1(a), two small regions are marked in red. The leftmost region is further detailed in Fig. 1(b), using a finer slotsize of 1 ms for throughput computation. Here, the y-axis scale extends up to 500 Mbps, in contrast to the 250 Mbps used in Fig. 1(a). It is clear that there is considerable variation in the throughput where there are periods without any traffic followed by short bursts of traffic. However, we should note that 1 millisecond is not a suitable choice for the slot size, given that it has been shown that Starlink has a frame time of $1.\overline{33}$ milliseconds. Thus, aliasing effects are likely, and in fact, this is what is present in Fig. 1(b). If we take exactly the same data and instead use the slot size equal to the Starlink frame time, then we get the appearance that is shown in Fig. 1(c) which shows a stable rate during the data bursts. Included in the figure is a horizontal reference line at 373.3 Mbps, one of the identified Starlink physical rates, as will be discussed later.

In Fig. 1(d) the left red band of Fig. 1(a) is shown, which covers a stable region of higher throughput than the region of Fig. 1(c). Here we can note that the general structure of having short data bursts followed by silent regions is the same. The throughput obtained in the bursts is however different, as is the duration of the bursts, and as is the inter-burst silent time. Here, the horizontal reference line is placed at 430.5 Mbps.

By increasing the time resolution further, we can consider the behavior within one of these bursts. The bursts marked in red in Figs. 1(c) and 1(d) are shown in Fig. 2, but now using the cumulative bytes received on the y-axis instead of throughput. Within each burst, there is a clear periodic pattern visible in the form of sub-bursts, with the left subfigure having seven sub-bursts and the right five sub-bursts. We have in related work [10] examined these sub-bursts and their relation to the physical layer OFDMA frame structure of Starlink, and noted that the sub-bursts as visible in Fig. 2 correspond to data received in a Starlink physical layer frame, when that data is emitted out on the Starlink Ethernet interface. The slope of each sub-burst in Fig. 2 thus corresponds to the Ethernet line rate. The time between the start of consecutive sub-bursts is exactly $1.\overline{33}$ ms, matching the Starlink physical frame time [11]. In our related work [10] we also derived a set of commonly occurring transmission rates based on an aggregate analysis of burst characteristics. These transmission rates are listed in Table 1, and will be used in the further analysis. We note the match between Fig. 1(c) and rate #6, and Fig. 1(d) and rate #7.

5. Burst rate distribution and clusters

We now consider the specific burst rates identified in our previous work [10] as listed in Table 1, with the aim to provide a more

Table 1
Empirical and hypothesized Starlink physical rates, model rate difference to baseline in steps of 27 Mbps, and error between empirical rate and model rate in Mbps. From [10], where details on possible model parameters listed in the comment column are also provided.

#	Emp.rate	Theo.rate	Rate diff	Error	Comment
0	134	133.5	-11×27	1	
1	215	214.5	-8×27	1	
2	242	241.5	-7×27	1	
3	269	268.5	-6×27	0	
4	323	322.5	-4×27	0	
5	353	349.5	-3×27	4	No model match
6	373	376.5	-2×27	−3	No model match
7	**430.57**	**430.5**	**0**	**0.07**	**287sym 4QAM**
8	458	457.5	$+1 \times 27$	0	+18sym 16QAM
9	485	484.5	$+2 \times 27$	0	+36sym 16QAM
10	512	511.5	$+3 \times 27$	0	+54sym 16QAM
11	539	538.5	$+4 \times 27$	0	+72sym 16QAM
12	565	565.5	$+5 \times 27$	0	+90sym 16QAM
13	645	646.5	$+8 \times 27$	−1	

granular examination of the distribution of rates, and the rate transition behavior. To enable the examination of burst characteristics we, similarly to our previous work, process the packets captured during the measurement runs to identity 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 during full traffic load, we consider only the traffic that is present within a 1 second margin away from the start and end of the Speedtest measurement run, e.g., one second in from the start and end of the example measurement run in Fig. 1(a). With this definition of the bursts, we process the 422 million packets that are within the margins, and locate 2 375 690 bursts.

Fig. 3 provides a histogram of the resulting burst rates. As our interest here lies in examining the transitions between different burst rates, we need a mechanism to assign the observed bursts into clusters where each cluster corresponds to one of the rate peaks in the figure, a rate which also has an equivalent line in Table 1 where it is numbered. In the following we use the number from Table 1 to identify the clusters. We use a rate width of 9 Mbps on either side of the empirical rate to assign observed bursts into a cluster. Thus, the gray areas in Fig. 3 correspond to the regions associated with clusters 6, 7, 8, 9, 10, 11, and 12, which are the ones we consider in this study.

We now consider a more detailed analysis of the burst rate distribution. As described in [10] and illustrated in Fig. 2, bursts are composed of a number of sub-bursts, each corresponding to the data in a physical layer frame. In Fig. 4 we provide the distribution of burst rates subdivided by the number of sub-bursts/frames present in each burst. Notably, bursts consisting of a single frame – a burst length of 1 – display a distinctive spiky distribution pattern. For bursts encompassing only a single frame, the obtained rate is directly proportional to the count of packets within that frame. Beyond single-frame bursts, the distributions have some commonality in that they all exhibit similar peak locations. However, the relative prominence of these peaks varies with the burst duration. How large fraction of the observations that fall outside any cluster also varies, as does the degree to which the peaks are well defined. For the 2 frame case, the peaks are less defined and more observations become non-clustered. One can also note that there are other variations when comparing bursts of differing lengths. For instance, in the case of bursts lasting 8 frames, lower rates constitute a larger fraction of the distribution compared to bursts that span 4 frames.

6. Burst rate transitions

Having assigned all burst observations to either one of the clusters, or noted them as not clustered, we can perform an analysis of how

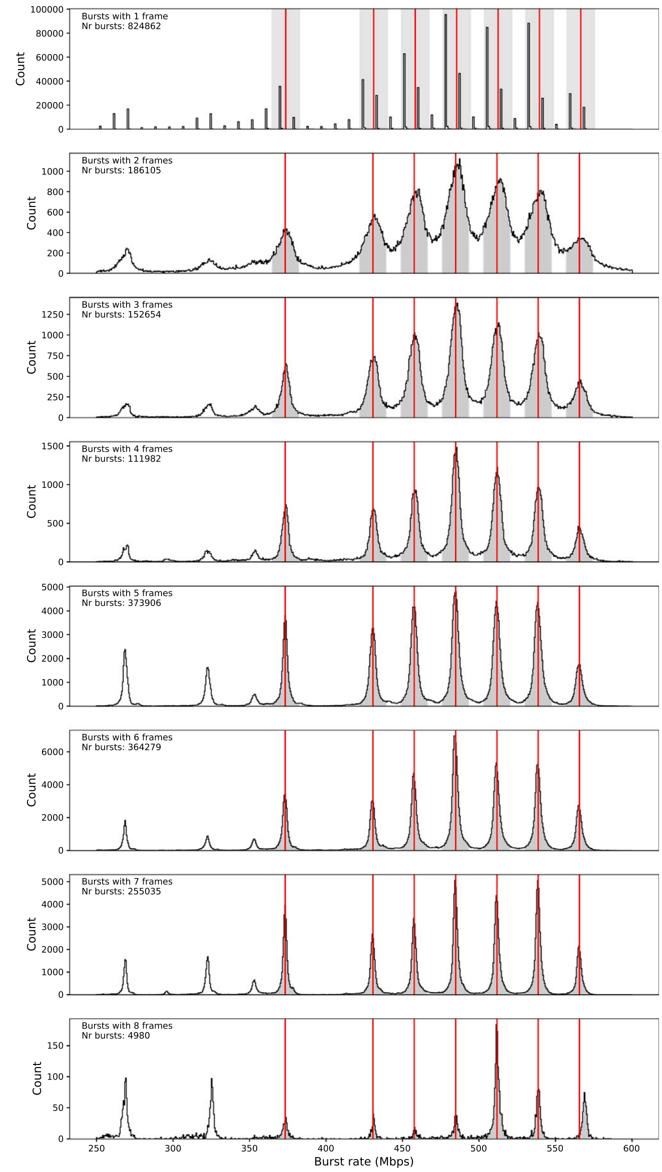


Fig. 4. Burst rate distribution per burst duration, considered in number of $1.\overline{33}$ ms frames in burst. Cluster extent marked in gray. The red lines mark the cluster centers of the 7 considered clusters. Note difference in y-axis scaling.

the changes from one burst rate to another burst rate is performed. We can note from Figs. 1(c) and 1(d) that the burst rates appear fairly stable. To examine this further, we process all burst rate observations to obtain a transition matrix. A transition occur from one burst to the next burst subsequent in time, and each of these bursts have an associated cluster or are unclustered. We show the resulting transition probabilities between the burst rate clusters in Fig. 5. Here, there is a clear structure that can be observed. For clusters 7 to 12, the transitions follow a chain behavior where for a given cluster rate, transitions are occurring either to the rate above or the rate below. Or alternatively, and in fact more commonly, to a rate outside of any cluster. The most common transition is, for all clusters, back to the current cluster rate. In terms of the chain behavior, one can observe that there is a push towards the middle. In the sense that the probability of going from 7 to 8 is higher than going from 8 to 7, and correspondingly higher to go from 12 to 11 than from 11 to 12. This holds also for the other clusters. The observed chain-like behavior appears consistent with a radio resource allocation mechanisms that tries to optimize the physical

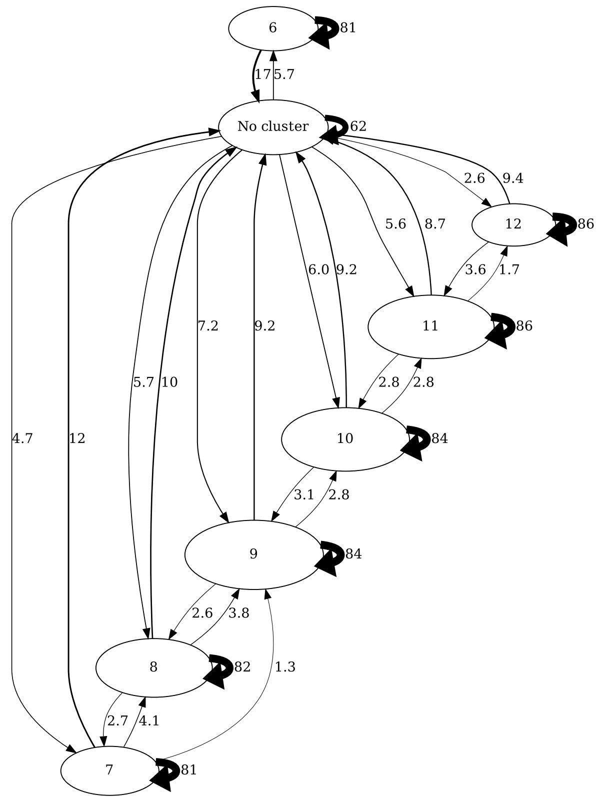


Fig. 5. Transition probabilities between states representing clusters of burst rates. The state numbers are the # in Table 1. The area of the state's ellipse is proportional to the number of observed bursts in the cluster. Line thickness is proportional to transition probability. Probabilities < 1% are not shown.

rate to prevailing channel conditions. As far as we are aware, an observation of this behavior has not been previously reported. A notable exception is cluster 6, which does not have significant transition to the other clusters, but rather moves to and from unclustered rates. We note that from a rate modeling perspective as seen in Table 1, cluster 6 is an outlier.

7. Joint rate and duration transitions

We now focus on both the burst rate, and the burst duration, as a joint characteristic of each network burst. This perspective enables us to construct a more comprehensive transition matrix that encapsulates the interactions between these two properties. To get a manageable graph size we limit our analysis to bursts that are clustered, and which fall within a burst duration range of 1 to 7 frames. Consequently, this approach results in 49 distinct states, where each C_D state represents a unique combination of burst rate and duration. The transition probabilities are presented in Fig. 6. Despite the complexity and potential visual overload of the figure, several observations can be discerned:

- Transition Patterns Between Clusters: An interesting pattern emerges in transitions between states that refer to different rate clusters, i.e. where the C of C_D changes. These transitions are predominantly, if not exclusively, to states representing short one-frame bursts at the new burst rate. These states are identifiable as those ending with a '_1', indicating a 1-frame burst.
- Transition Patterns Within Clusters: It can be observed that the change of burst duration within a cluster, i.e. where the D of C_D changes, also predominantly occurs through a transitory single frame state '_1'. This implies that transitions between bursts of varying lengths are often mediated by a short, intervening single-frame burst.
- Sticky State Transition Pattern: In the majority of cases, but not universally, we observe that the most common transition for a state is a return to itself.
- Exceptions to Sticky State Transition Pattern: States that end with 2, 3, or 4 and thus have that number of frames display a different transition behavior. These states are typically transitory, meaning that the most frequent transition observed is not a return to the same state, but rather a move to another distinct state.
- Isolation of Cluster 6: In consistence with the results shown in Fig. 5, Cluster 6 continues to demonstrate a unique pattern of isolation. Also for this expanded transition matrix, it is evident that there are also no transitions from any of the 6_D states to any state belonging to another C_ cluster.

While we cannot ascertain the underlying mechanism for these observations, they nevertheless contribute to additional knowledge regarding the design and functioning of the Starlink physical layer and radio resource allocation. As our measurement approach only requires hardware in the form of a hardware timestamp capable NIC, our measurements can easily be extended by others to obtain additional insights into this behavior. As our results are based on a single measurement campaign at a single location, the results are not expected to cover the full range of variability that are likely present in the physical layer. We however expect overall observations such as the tendency towards rate changes to neighboring rates, and performing cluster transitions through single-frame states, to hold also for the general case. We provide the complete 49×49 transition probabilities for this 49-state model[1] to allow further exploration.

8. Evaluating effective throughput

We now also consider the interburst silent time, which, together with burst rate and burst duration, forms the components for constructing the effective throughput observed over longer timescales. As depicted in Fig. 1, there are interburst silent periods, which can vary in time duration. The relative proportion of time spent within a burst and between bursts is a major factor in determining the effective throughput on the scale of tens of milliseconds. We can define a complete burst to be consisting of the burst duration, and the associated interburst silent time that follows after that particular burst. With the duration and silent times, and with consideration of the burst rate, we can now easily compute the effective throughput during such a complete burst. Fig. 7 shows the results of such computations for the cluster burst rates and burst lengths examined in the preceding transition analysis, along with the length of the interburst silent times over a range of values that has been empirically observed. For these considered configurations the effective throughput varies from slightly above 25 Mbps up to almost 500 Mbps.

We now examine the empirical distribution of the three components of effective throughput in our collected measurements. To highlight the differences in physical burst rate and scheduling that lead to the effective throughputs that are observed, we consider three different subsets of measurement runs. We select the 10 percent of measurement runs with the lowest, middle and highest throughput over the complete run as reported by the Ookla Speedtest. The mean throughput of the measurement runs within these three deciles is 142, 222, and 268 Mbps, respectively. The bursts within these respective runs are examined, with the configuration of the three components determined and used to create a 2d histogram. These histograms are shown in Fig. 8, which highlights the apparent differences between the deciles.

The runs with the lowest throughput have a considerable amount of single frame bursts, i.e. C_1, with relatively short interburst times. Longer bursts are mainly present in the cluster with lowest rate out of the ones considered for analysis and display. For these low throughput

[1] https://git.cs.kau.se/pub/slinkmarkov

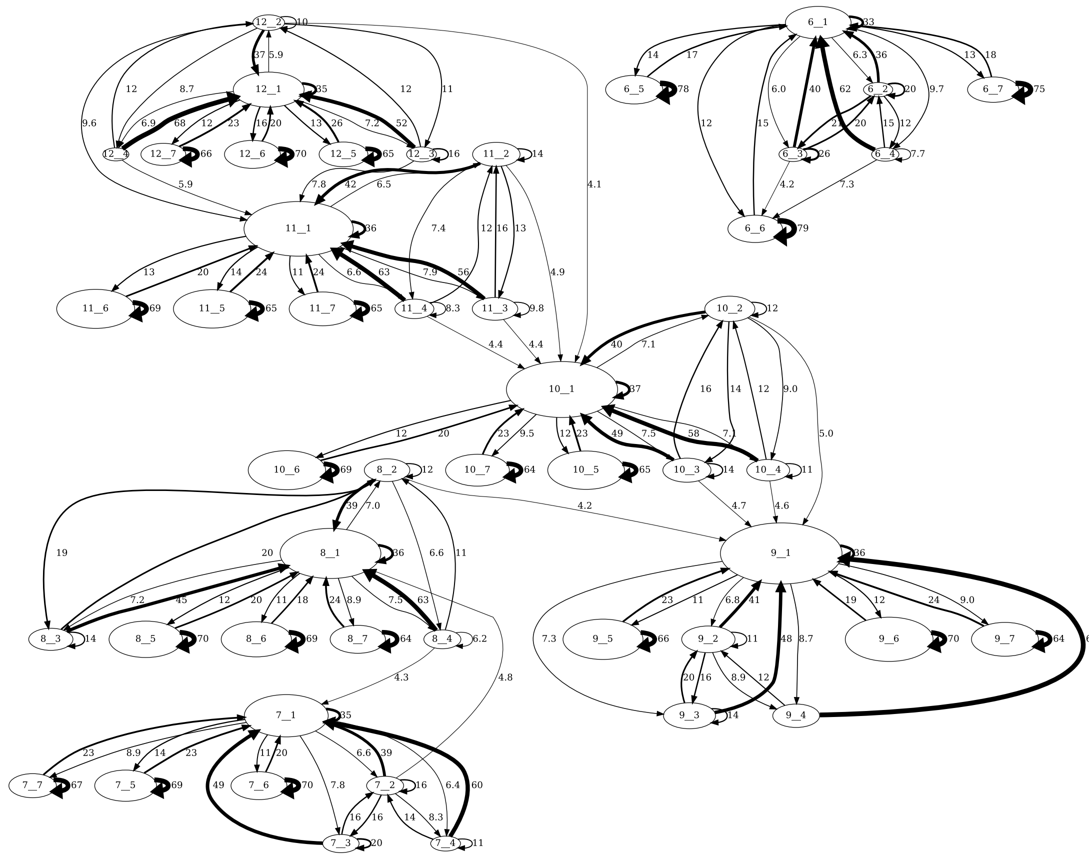


Fig. 6. Transition probabilities between the 49 combined C_D states where C is the burst rate cluster, and D the burst duration in number of $1.\overline{33}$ ms frames. The area of the state's ellipse is proportional to the number of observations in the state. Line thickness is proportional to transition probability. Probabilities < 4% are not shown.

runs, there are also a number of bursts that fall into the lower burst rate clusters which are not shown here. Due to space and legibility considerations the number of clusters considered in analysis and visualization needs to be limited.

In comparison, the runs with throughput around the median of all runs show a distribution in the middle of the examined configurations. For these runs, the most common configuration is 8 ms bursts with a rate of 484 Mbps, and an interburst silent time of $5.\overline{33}$ ms, i.e. 4 on x-axis, and 9_6 on y-axis. Notable is the somewhat peculiar pattern of the combinations of burst duration and inter-burst time. Across the burst rates, the common combinations are 5+3, 6+4 and 7+5 which sum to 8,10, and 12 time slots. So rather than summing to a constant burst duration plus interburst time for the scheduling, this seems to be done in multiples of two time slots.

Finally, the runs within the highest throughput decile share a similar behavior with the median throughput runs with regards to the distribution among burst durations and inter-burst times. However, the burst rates are now higher, with the majority of the bursts falling within the two highest rate clusters. We can note that these results by necessity reflect the conditions at the time of data collection and for the specific site of data collection. Data collected at other sites, and with different amount of competing users is very likely to have different characteristics from the ones illustrated here. Nevertheless, the fundamental approach of using these three constituent components as factors when examining the effective observed throughput over longer time scales is generally applicable across sites and time periods, and more measurements and accompanying analyses may lead to further improved understanding of Starlink system aspects.

9. Burst sequence mining

We now continue the examination by considering how the complete burst characteristics, i.e. the triplet of burst rate cluster, burst duration, and inter-burst delay (C_D_I), vary over time. It is clear from the transition probabilities of the C_D states in Fig. 6, and the related discussion in Section 7, that patterns can be expected in how the burst triplet varies over time. To further examine the patterns of variation of the burst triplet, we perform frequent substring mining to detect frequent patterns of variation. We process the burst sequence of each run and then aggregate the counts of the identified patterns.

More formally, one can consider S as a sequence of length n, where $S = s_1, s_2, \dots, s_n$ and s_i is the i-th element of the sequence. In our case, the sequence S consists of C_D_I burst characteristic triplets with a total sequence length, n, of 2 375 690. A substring $S_{i..j}$ of S is defined as the triplet elements from index i to j where $1 \le i \le j \le n$. Thus, $S_{i..j} = s_i, s_{i+1}, \dots, s_j$. The frequency $f(S_{i..j})$ of a substring $S_{i..j}$ is defined as the number of times that the substring $S_{i..j}$ appears anywhere in S. We define $\mathrm{M}(S, k, w)$ as the set of all substrings of length k whose frequency in S is greater than or equal to w. Formally,

$$\mathrm{M}(S, k, w) = \{S_{i..j} \mid j - i + 1 = k \text{ and } f(S_{i..j}) \ge w\}$$

To efficiently locate the substrings and count the number of occurrences, we utilize suffix arrays and longest common prefix arrays as implemented in `pydivsufsort` [30]. To understand the impact of substring length on the number of patterns that are present in the triplet sequence, we vary k over the range $2 \dots 20$ and search for all substrings in S with $w >= 2$. In total, this leads to the detection of 1 935 420 distinct substrings over this range of k. The frequency of occurrence for the most frequent distinct substrings are shown in Fig. 9 for a range of substring lengths. As expected, shorter substrings are more frequent than longer, with the most frequent substring for $k = 2$ occurring 41367 times. As the substring length increases, it can be noted that even the most frequent substrings are not overly frequent.

Table 2
The 8 most common substrings for $k = 2$, 4, 7 and 10. If triplet C_D_I occurs N times in a row it is abbreviated as C_D_I **xN**.

k = 2		k = 4		k = 7		k = 10	
Freq	Substring	Freq	Substring	Freq	Substring	Freq	Substring
41367	9_6_4 **x2**	13 866	9_5_3 **x4**	2490	9_6_4 **x4**, 9_1_4, 9_6_4 **x2**	1640	9_6_4 **x5**, 9_1_4, 9_6_4 **x4**
33869	9_5_3 **x2**	13 823	8_5_3 **x4**	2438	9_6_4 **x3**, 9_1_4, 9_6_4 **x3**	1619	9_6_4 **x4**, 9_1_4, 9_6_4 **x5**
31487	11_6_4 **x2**	12 428	11_5_3 **x4**	2425	9_6_4 **x2**, 9_1_4, 9_6_4 **x4**	1599	9_1_4, 9_6_4 **x5**, 9_1_4, 9_6_4 **x3**
31080	10_6_4 **x2**	12 369	9_6_4 **x4**	2351	10_1_4 **x7**	1544	9_6_4 **x2**, 9_1_4, 9_6_4 **x5**, 9_1_4, 9_6_4
31006	11_5_3 **x2**	12 280	10_5_3 **x4**	2029	9_6_4 **x5**, 9_1_4, 9_6_4	1537	10_1_4 **x10**
30814	8_5_3 **x2**	10 340	6_5_3 **x4**	1971	9_1_4, 9_6_4 **x5**, 9_1_4	1532	9_6_4 **x3**, 9_1_4, 9_6_4 **x5**, 9_1_4
30538	10_5_3 **x2**	9402	11_6_4 **x4**	1928	9_6_4, 9_1_4, 9_6_4 **x5**	1517	9_6_4, 9_1_4, 9_6_4 **x5**, 9_1_4, 9_6_4 **x2**
26233	8_6_4 **x2**	9301	7_5_3 **x4**	1928	8_5_3 **x2**, 8_1_3, 8_5_3 **x4**	1443	8_5_3 **x7**, 8_1_3, 8_5_3 **x2**

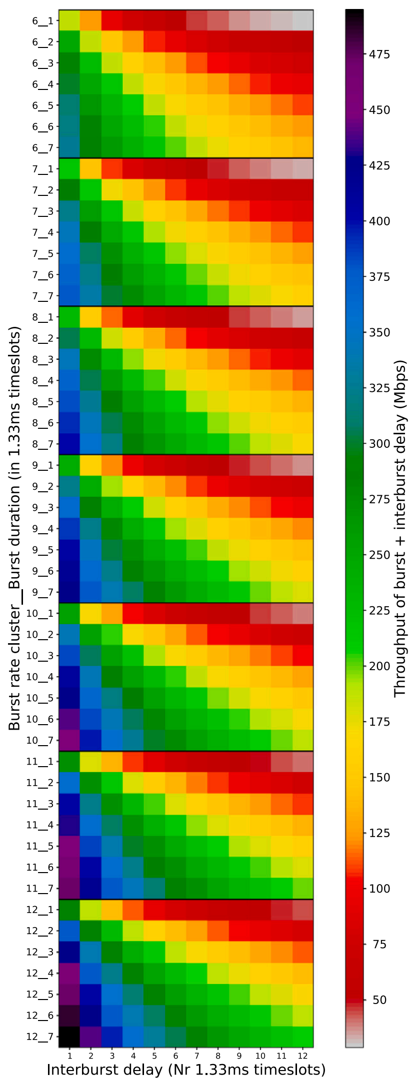


Fig. 7. Throughput when computed over sum of burst duration and inter-burst delay, for some burst rate cluster C and burst duration D, noted as C_D on y-axis.

The data comes from 2195 runs, so substrings with a frequency of occurrence less than 2195 have no possibility of occurring at least once in every one of the examined runs. The eight most frequent substrings for four different k are provided in Table 2. As can be seen the longer substrings are typically composed of the smaller ones. Also, it can be noted that the patterns expected by the C_D transition probabilities of Fig. 6 are also seen for the C_D_I burst characteristic triplets. The transitions occurring via states with a duration of one, i.e. transitions to C_1 and back, here gives rise to multiple similar substrings varying only in the offset location of the C_1_I triplet as exemplified for $k =$ 7. Furthermore, Table 2 again illustrates that transitions back to the same state are the most common, as for example all the most common substrings for $k \leq 4$ are only repetitions of a specific triplet.

10. Burst sequence prediction

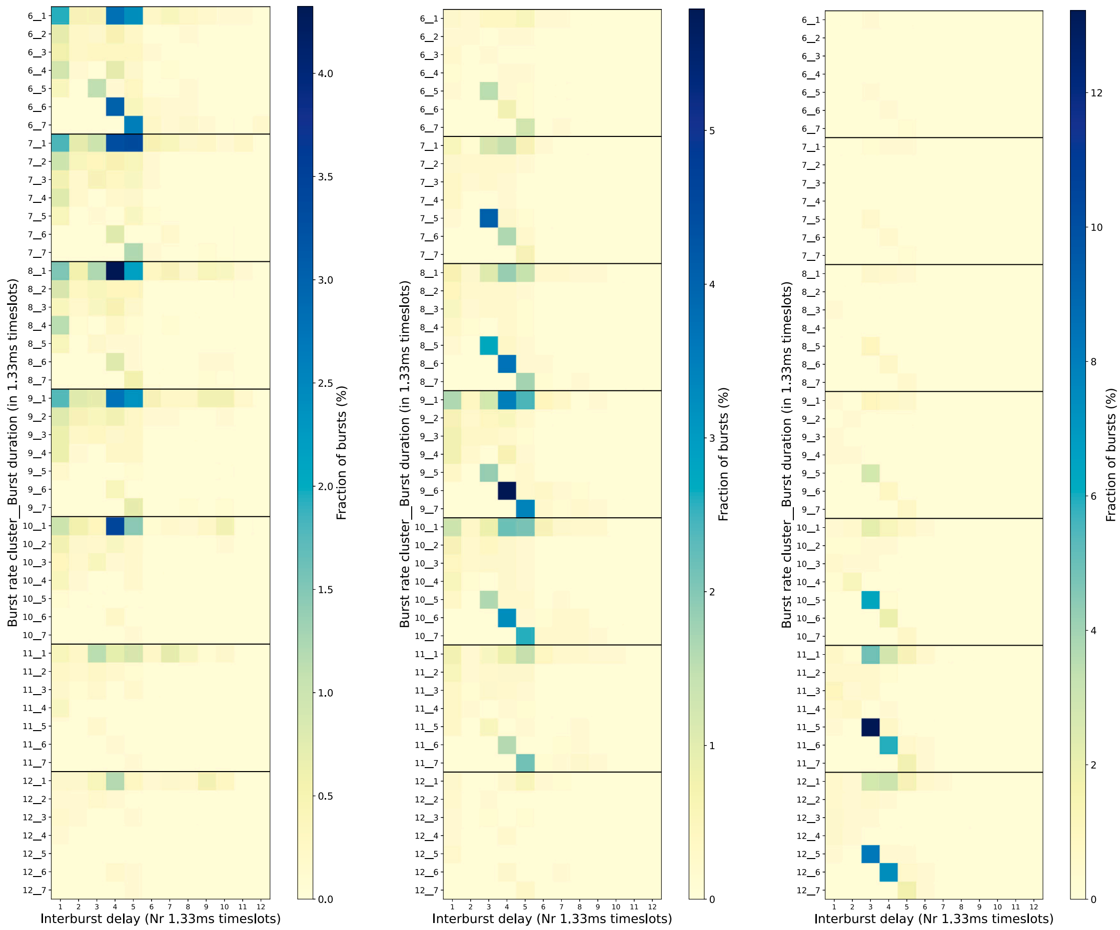
The structure evident from the pattern mining can also be exploited for prediction. We now consider the prediction of the burst characteristics for the next burst, based on the characteristics of the preceding bursts. We wish to predict the triplet of burst rate cluster, burst duration, and inter-burst delay as that allows us to later calculate the effective future bandwidth for the predicted burst (if the prediction is correct). Such information of future available throughput can be beneficial for congestion control mechanism, throughput-adaptive applications, and multi-connectivity schedulers.

In contrast to the preceding transition modeling, for the prediction evaluation we consider the full range of all 14 clusters from Table 1 as well as the unclustered observations, giving 15 categories for burst rate cluster. For the burst duration the full range of observed durations, i.e between 1 and 13 slots, is considered. The range over which the inter-burst delay varies is considerably larger, with 96 different values observed in our data. It can be noted that the combinations of these three characteristics in our data gives rise to 2360 unique C_D_I triplets, which is a considerably higher number than the 49 C_D states used to compute the transition probabilities shown in Fig. 6. We examine two approaches for burst prediction. First we perform prediction with N-gram modeling to iteratively smaller substrings based on match occurrence frequency. Then, we employ Deep Learning to examine how much improvement in prediction performance can be achieved with such more elaborate modeling.

10.1. N-gram burst prediction

From the perspective of constructing a predictor, the commonly used one-hot encoding technique would lead to 2360 features just to represent one previous triplet, possibly leading to dimensionality issues as we also want to consider a history of several previously observed triplets. Instead of using one-hot encoding, we instead take inspiration from N-gram modeling to create an iterative Maximum Likelihood Estimator for predicting the next triplet. More formally, we want to predict the next element L_{k_L+1} of a sequence where the preceding elements of observed triplets form the string $L_{h..k_L}$, where h is the starting index beginning at 1, and k_L represents the length of L. Now recall that we have $\mathrm{M}(S, k, w)$ as the set of all substrings of length k in S whose occurrence frequency is at least w. These substrings are utilized to predict the next element of $L_{h..k_L}$, based on patterns identified in S as follows:

- Search for prefix matches for $L_{h..k_L}$ in $\mathrm{M}(S, k_L + 1, w)$, and select the match with highest frequency of occurrence.
- If matched, use the last element in the matched substring, S_{k_L+1} as the prediction.
- If no match is found, search for a match using $L_{h+1..k_L}$.



(a) Throughput within 0-10 percentile range. (b) Throughput within 45-55 percentile range. (c) Throughput within 90-100 percentile range.

Fig. 8. Frequency of occurrence of different burst rate clusters and burst durations (C_D) on *y*-axis, and inter-burst delays on *x*-axis, for all bursts within three different throughput deciles of the 2195 measurement runs.

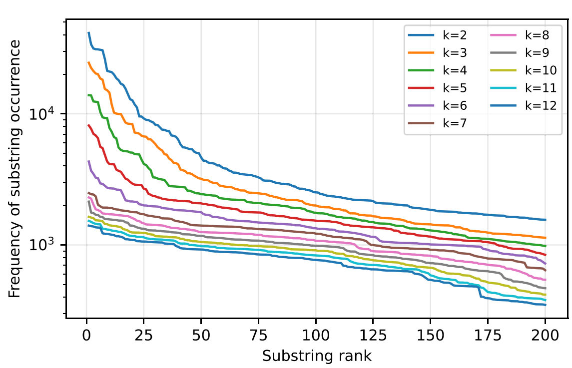


Fig. 9. Number of occurrences for frequent substrings of varying length.

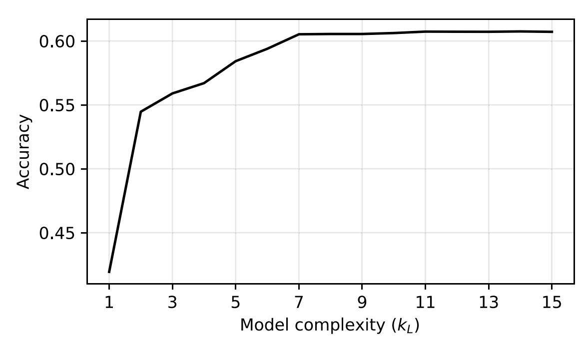


Fig. 10. Prediction accuracy for next burst triple prediction as a function of model complexity.

- Continue adjusting h until a match is found, or until $k_L - h$ is reduced to a minimum specified length.
- If no matches are found through all iterations, use L_{k_L} as the next predicted element.

Based on the described approach, we implement a custom predictor/classifier in the `scikit-learn` framework [31]. We can now examine the prediction performance of our predictor for varying model orders. We perform a 80/20 train test split so that 80% of the collected runs are put into the training data, and 20% of the runs are used for testing. For the testing runs, 50 randomly selected locations are picked to be predicted, with their preceding burst triplets as input to the predictor. The training time for the model is around 11 min on an i9-10920X CPU. As baseline, a zero-order model that always predicts the single triplet which is most frequent in the data achieves an accuracy of 2.56%. Another straightforward model that simply predicts the next triplet to be the same as the immediately preceding triplet has an accuracy of 30.2%.

A first order N-gram model (i.e. $k = 2, k_L = 1$) has a model complexity corresponding to the Markovian transition models shown in Figs. 5 and 6. The prediction accuracy for first order, as well as higher order models, is shown in Fig. 10. It can be seen that the first order model improves considerably over the simple previous triplet

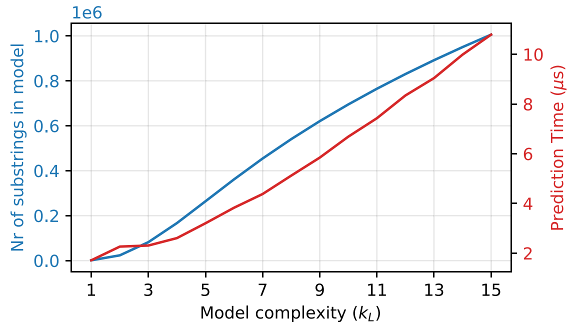


Fig. 11. Number of substrings and prediction time as a function of model complexity.

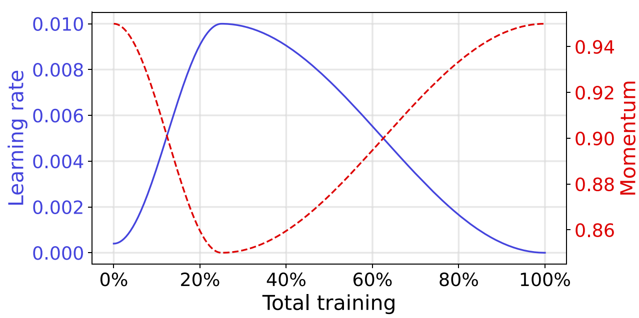


Fig. 12. Deep learning one-cycle training: learning rate (solid, blue line) and momentum (dashed, red line) over time for 15 epochs default training.

model, and that the accuracy improves with model complexity. After $k_L = 7$ the performance levels out at around 0.6. We can thus note that our predictor in the majority of cases can correctly predict the burst characteristics of the upcoming burst. As illustrated in Fig. 7 the throughput follows directly from such burst characteristics, and thus the precise future throughput can also be correctly predicted in the majority of cases, on the short-term burst time-scale of tens of milliseconds. By applying auto-regressive prediction to our N-gram model these times can be further extended. Such extensions, as well as exploring transformer-based models for prediction, are left for future work.

We now consider the computational and memory complexity of this prediction approach. Fig. 11 shows the cumulative number of substrings held in the model as model complexity increase. Also shown is the average prediction time for a single prediction when performed on a single core of the 10920X CPU. As the lookup into $\mathrm{M}(S, k, w)$ is performed with hashing, the prediction times are very modest. Furthermore, there is potential for further runtime optimization of the implementation by creating a direct lookup table, which could lead to prediction times less than 100 nanoseconds [32]. From a memory perspective, the demands of the model is also modest in that 20 megabytes of RAM should suffice for any of the model complexities under consideration.

10.2. Deep learning based burst prediction

Applying Deep Learning for time series prediction is resource-intensive but offers high predictive power, making it a valuable method. We selected InceptionTimePlus [33], a Deep Learning model optimized for multivariate time series analysis, to address our problem. InceptionTimePlus is a convolutional neural network designed for multichannel 1-dimensional inputs, comprising 3 949 905 trainable parameters. We employ the ReLU activation function and cross-entropy as the loss function.

Table 3
Deep Learning performance metrics for different epochs (E), sliding window sizes (WS) and maximum learning rate (LR) for predicting burst triplets. Only the differences to default are noted. Default configuration: 15 E, WS 100, LR 1e-2.

Configuration	Percentage correctly predicted				
	Triplets	Elements	C	D	I
Default F0	72.40	86.65	83.82	89.33	86.79
Default F1	70.53	85.62	82.80	88.40	85.65
Default F2	71.32	85.82	83.29	88.28	85.89
Default F3	71.44	86.09	83.55	88.76	85.96
Default F4	72.57	86.79	83.80	89.49	87.08
F0 to F4 Average	71.65	86.19	83.45	88.85	86.27
Default run 1	72.37	86.71	83.79	89.37	86.99
Default run 2	72.57	86.79	83.92	89.40	87.05
WS 10	68.88	84.61	82.52	87.74	83.58
WS 25	71.96	86.44	83.27	89.31	86.74
WS 50	72.44	86.71	83.67	89.44	87.02
WS 200	72.46	86.71	83.74	89.40	86.98
LR 5e−3	72.04	86.55	83.63	88.96	87.06
LR 2e−2	71.52	86.22	83.69	88.85	86.12
LR 3e−2	71.01	85.97	83.47	88.68	85.75
5 E	71.43	86.16	83.45	88.84	86.19
10 E	72.19	86.57	83.73	89.24	86.75
30 E	71.94	86.27	83.62	88.72	86.48
50 E	72.62	86.81	83.87	89.31	87.26
WS 050, 3 E	70.51	85.73	82.98	88.56	85.64
WS 050, 10 E	72.19	86.54	83.49	89.33	86.81

10.2.1. Dataset

To keep a straightforward setup and maximize the amount of training examples, we concatenate all triplet data into a single training set. While this creates boundary effects as the model is exposed to disjoint triplet sequences from different runs, the model is expected to be sufficiently sophisticated to handle such noise in the training data. Data preprocessing employs a sliding window approach with a window size ranging from 50 to 200 and a stride of 1. This ensures that, for a window size of 100, each instance, spanning the 101st to the second-to-last 100th sample, is reused 100 times in the training dataset. To prepare the data for deep learning, the C_D_I triplets used with N-gram modeling are here split into their three constituent elements, with each elements assigned to one model channel. Thus, for a sliding window size of 100, we now have 2 375 590 instances with the dimensions (3, 100). Correspondingly, the target data Y comprises of 2 375 590 instances with dimension (3, 1).

To ensure a robust evaluation of the model, we divide the dataset into training and validation sets. The first 80% of the data is used for training, and the remaining 20% is reserved for validation. This ensures that no validation triplet is part of the sliding windows of the training set. For the default case (sliding window size of 100), the validation set contains 475 118 triplets.

10.2.2. Training strategy

We conducted preliminary testing on a small subset of the training data to identify a reasonable starting point for hyperparameter optimization. Using 15 epochs, a sliding window size of 100, and a maximum learning rate of 0.01, we achieve the best results in this pretest, which then informs our default evaluation strategy on the full dataset.

To enhance training efficiency, we adopt one-cycle training, a technique that reduces training times and can improve model accuracy [34]. Unlike using static values for learning rates and momentum, one-cycle training dynamically adjusts these parameters during the training process. Specifically, we set the maximum learning rate of 0.01 to be reached at 25% of the training cycle, as recommended. Momentum is simultaneously adjusted to reach its minimum value at the same 25% mark. Fig. 12 illustrates the dynamic adjustments of

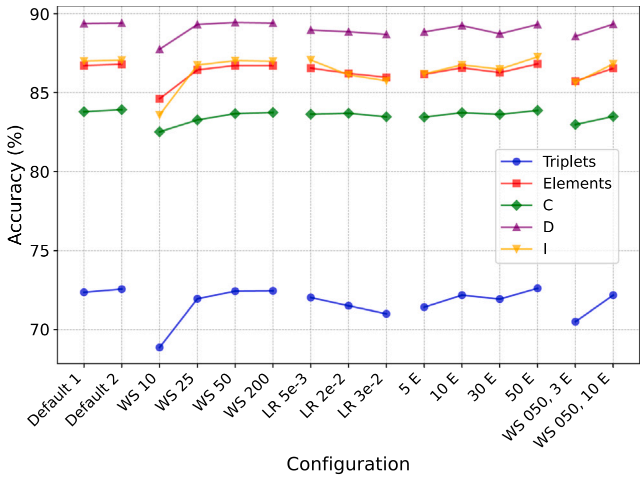


Fig. 13. Deep learning performance metrics for different epochs (E), sliding window sizes (WS) and maximum learning rates (LR) for predicting burst triplets. Only the differences to default are noted. Default configuration: 15 E, WS 100, LR 1e-2.

learning rate and momentum throughout the training cycle.

Instead of predicting the entire triplet as a single unit, we train the model to predict each individual value separately. This approach reduces the number of possible classes compared to triplet prediction and allows for a more granular evaluation. The training is conducted using a batch size of 64, which is the InceptionTimePlus default, dividing the training set into 29 695 batches. The InceptionTimePlus model is trained on an Nvidia A40 GPU, and the training time per epoch is between 6.5 to 7 min. The variation in training time per epoch is mostly influenced by the sliding window size where a larger window size increases training times slightly. We explore the hyperparameter space by varying the sliding windows size, the learning rate and the amount of epochs trained.

10.2.3. Results

The performance of the InceptionTimePlus model on the dataset is summarized in Table 3 and visualized in Fig. 13. The model demonstrates robust performance across various hyperparameters, highlighting its effectiveness in predicting the individual values of the triplets. While we achieve the best results when training for 50 epochs with a sliding window size of 100, all results are rather close to each other.

We also evaluate our results using 5-fold cross-validation with the default hyperparameters (Default F0–F4 in Table 3). Slight variation can be observed across the folds, this is likely due to data variability. Despite this, the results remain consistent across different hyperparameter settings, indicating that overfitting to the validation set is unlikely.

On the same Nvidia A40 GPU hardware, our average inference time per prediction is approximately 66 μs, with a GPU memory usage around 670 MiB. These values could potentially be further optimized.

Overall, the results indicate that this model performs better than N-gram, achieving around 86% element-wise matches, and approximately 72% for matching triplets compared to 61% for N-gram. Variations in hyperparameter selection yielded no significant improvements, suggesting that a time- and resource-efficient configuration might be preferable. An efficient setup found consists of a sliding window size of 50, combined with 10 to 15 epochs of one-cycle training at a maximum learning rate of 0.01

11. Conclusions

This work uses receiver side network measurements collected with the aid of a hardware-timestamp NIC to perform studies on Starlink physical layer rates, and in particular rate transitions. We construct rate transition matrices and study the resulting transition structure for hints on Starlink physical layer structure and behavior. Our results show that there is structure in the transition behavior, such as particular transition steps associated with rate switching, and that rate switching occurs mainly to neighboring rates. We also perform pattern mining to identify the most frequent patterns of burst characteristics in terms of burst rate, burst duration, and inter-burst silence time. We show how these patterns can be utilized to construct both an N-gram and a Deep Learning model that can predict upcoming burst characteristics, and by extension short term throughput. Although limited to single-location measurements, this work advances the understanding of Starlink's complex dynamics and highlights promising directions for further research.

CRediT authorship contribution statement

Johan Garcia: Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Matthias Beckerle:** Methodology, Software, Writing – original draft, Writing – review & editing. **Simon Sundberg:** Writing – review & editing, Software, Resources, Data curation. **Anna Brunstrom:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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

Data availability

Link to repo with model has been provided.

References

[1] M.M. Azari, et al., Evolution of non-terrestrial networks from 5G to 6G: A survey, IEEE Commun. Surv. & Tutorials (2022) 2633–2672.

[2] 3GPP, 3GPP TS 23-700-27: Study on 5G system with satellite backhaul (Release 18), 2022, 3GPP TS 23- 700- 27 V18. 0. 0 (2022- 12).

[3] Starlink, Order Starlink, 2023, [Online]. Available: https://www.starlink.com/.

[4] J. Garcia, S. Sundberg, A. Brunstrom, Fine-grained starlink throughput variation examined with state-transition modeling, in: 2024 19th Wireless on-Demand Network Systems and Services Conference, WONS, 2024, pp. 69–76.

[5] F. Michel, M. Trevisan, D. Giordano, O. Bonaventure, A first look at Starlink performance, in: Proceedings of the 22nd ACM Internet Measurement Conference, IMC '22, 2022, pp. 130–136.

[6] N. Mohan, A.E. Ferguson, H. Cech, R. Bose, P.R. Renatin, M.K. Marina, J. Ott, A multifaceted look at Starlink performance, in: Proceedings of the ACM on Web Conference 2024, 2024, pp. 2723–2734.

[7] H.B. Tanveer, M. Puchol, R. Singh, A. Bianchi, R. Nithyanand, Making sense of constellations: Methodologies for understanding Starlink's scheduling algorithms, in: Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies, 2023, pp. 37–43.

[8] H. Zhao, H. Fang, F. Wang, J. Liu, Realtime multimedia services over Starlink: A reality check, in: 33rd Workshop on Network and Operating System Support for Digital Audio and Video, 2023, pp. 43–49.

[9] J. Garcia, S. Sundberg, G. Caso, A. Brunstrom, Multi-timescale evaluation of Starlink throughput, in: Proceedings of the 1st ACM Workshop on LEO Networking and Communication, 2023, pp. 31–36.

[10] J. Garcia, S. Sundberg, A. Brunstrom, Inferring Starlink physical layer transmission rates through receiver packet timestamps, in: Proceedings of the 25th IEEE Wireless Communications and Networking Conference, WCNC, 2024, pp. 1–6.
[11] T.E. Humphreys, P.A. Iannucci, Z.M. Komodromos, A.M. Graff, Signal structure of the Starlink Ku-band downlink, IEEE Trans. Aerosp. Electron. Syst. (2023).
[12] M.M. Kassem, A. Raman, D. Perino, N. Sastry, A browser-side view of Starlink connectivity, in: 22nd ACM Internet Measurement Conference, IMC '22, 2022, pp. 151–158.
[13] S. Ma, Y.C. Chou, H. Zhao, L. Chen, X. Ma, J. Liu, Network characteristics of LEO satellite constellations: A Starlink-based measurement from end users, in: IEEE INFOCOM 2023-IEEE Conference on Computer Communications, IEEE, 2023, pp. 1–10.
[14] D. Laniewski, E. Lanfer, B. Meijerink, R. van Rijswijk-Deij, N. Aschenbruck, WetLinks: a large-scale longitudinal starlink dataset with contiguous weather data, 2024, arXiv preprint arXiv:2402.16448.
[15] J. Deutschmann, S. Jahandar, K.-S. Hielscher, R. German, Internet via satellite: GEO vs. LEO, OpenVPN vs. Wireguard, and CUBIC vs. BBR, in: Proceedings of the 1st ACM MobiCom Workshop on Satellite Networking and Computing, 2023, pp. 19–24.
[16] L. Izhikevich, M. Tran, K. Izhikevich, G. Akiwate, Z. Durumeric, Democratizing LEO satellite network measurement, Proc. the ACM Meas. Anal. Comput. Syst. 8 (1) (2024) 1–26.
[17] J. Pan, J. Zhao, L. Cai, Measuring the satellite links of a LEO network, in: IEEE International Conference on Communications, 2024, pp. 1–6.
[18] B. Hu, X. Zhang, Q. Zhang, N. Varyani, Z.M. Mao, F. Qian, Z.-L. Zhang, LEO satellite vs. Cellular networks: Exploring the potential for synergistic integration, in: Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies, CoNEXT 2023, 2023, pp. 45–51.
[19] C. Beckman, J. Garcia, H. Mikkelsen, P. Persson, Starlink and cellular connectivity under mobility: Drive testing across the arctic circle, in: 2024 Wireless Telecommunications Symposium, WTS, IEEE, 2024, pp. 1–9.
[20] D. Laniewski, E. Lanfer, S. Beginn, J. Dunker, M. Dückers, N. Aschenbruck, Starlink on the road: A first look at mobile starlink performance in central Europe, 2024, arXiv preprint arXiv:2403.13497.
[21] Z.M. Kassas, et al., Navigation with multi-constellation LEO satellite signals of opportunity: Starlink, OneWeb, Orbcomm, and Iridium, in: Position, Location and Navigation Symposium, PLANS, 2023, pp. 338–343.
[22] S. Kozhaya, J. Saroufim, Z.M. Kassas, Starlink for PNT: a trick or a treat? in: Proceedings of the 37th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2024, 2024, pp. 3779–3788.
[23] W. Stock, C.A. Hofmann, A. Knopp, LEO-PNT with Starlink: Development of a burst detection algorithm based on signal measurements, in: WSA & SCC 2023; 26th ITG Wshop on Smart Antennas and 13th Conf on Systems, Communications, and Coding, VDE, 2023, pp. 1–6.
[24] R. Blázquez-García, D. Cristallini, M. Ummenhofer, V. Seidel, J. Heckenbach, D. O'Hagan, Experimental comparison of Starlink and OneWeb signals for passive radar, in: Radar Conference, 2023, pp. 1–6.
[25] M. Rajiullah, G. Caso, A. Brunstrom, J. Karlsson, S. Alfredsson, O. Alay, 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, 2023, pp. 1–7.
[26] Ookla, Speedtest CLI: Internet speed test for the command line, 2023, [Online]. Available: https://www.speedtest.net/apps/cli.
[27] K. MacMillan, T. Mangla, J. Saxon, N.P. Marwell, N. Feamster, A comparative analysis of Ookla speedtest and measurement labs network diagnostic test (NDT7), Proc. ACM Meas. Anal. Comput. Syst. 7 (1) (2023) 1–26.
[28] M.A. Vieira, M.S. Castanho, R.D. Pacífico, E.R. Santos, E.P.C. Júnior, L.F. Vieira, Fast packet processing with eBPF and XDP: Concepts, code, challenges, and applications, ACM Comput. Surv. 53 (1) (2020) 1–36.
[29] S. Sundberg, J. Garcia, A. Brunstrom, Characterizing wireless link throughput with eBPF and hardware timestamps, in: 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2023, 2023, pp. 302–308.
[30] L. Abraham, pydivsufsort, Zenodo, 2023, [Online]. Available: https://github.com/louisabraham/pydivsufsort.
[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[32] J. Garcia, T. Korhonen, DIOPT: Extremely fast classification using lookups and optimal feature discretization, in: 2020 International Joint Conference on Neural Networks, IJCNN, IEEE, 2020, pp. 1–8.
[33] H.I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D.F. Schmidt, J. Weber, F. Petitjean, InceptionTime: Finding AlexNet for time series classification, Data Min. Knowl. Discov. 34 (6) (2020) 1936–1962.
[34] L.N. Smith, A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay, 2018, arXiv preprint arXiv:1803.09820.