A Measurement Study of Dynamics and Liquid Data Transport in OneWeb LEO Satellite Networks

Owen Perrin, Evan Gossling, Daji Qiao, Hongwei Zhang
Electrical and Computer Engineering, Iowa State University, Ames, Iowa, 50011
Email: {operrin, evang, daji, hongwei}@iastate.edu

***Abstract*—Operators such as SpaceX and Eutelsat have launched mega-constellations of Low Earth Orbit (LEO) satellites to support their LEO satellite networks (LSNs), Starlink and OneWeb, respectively. These networks offer broadband Internet access to previously un-serviceable regions that lack sufficient terrestrial infrastructure. However, these LSNs are not without idiosyncrasies; relatively high latency and jitter accompany throughput variations. Most notably, though, are frequent packet losses that hinder link usage and consistent traffic flows. While consumer-focused LSNs such as Starlink have been extensively studied, other LSNs with distinct properties remain underexplored. In this work, a measurement study of the OneWeb LSN is presented, documenting and detailing an under-studied network which focuses on reliability and enterprise use-cases. Furthermore, we examine the potential benefits of liquid data transport, which allows TCP flows to maintain a large congestion window size and reduce retransmissions, thereby minimizing the impact of packet loss over LSNs. We find that OneWeb generally fulfills its service-level agreements, and liquid data transport may be used to further increase the reliability of TCP flows over LSNs.**

I. Introduction

The rise of mega-constellations such as SpaceX's Starlink, Eutelsat OneWeb, Viasat, and the like has enabled broadband communications in locations previously therewithout. In contrast to Geostationary Earth Orbit (GEO) satellite communications, Low Earth Orbit (LEO) satellites substantially decrease latency while increasing users' capacity and throughput. As such, rural, maritime or battlespace environments with the need for additional reliable connections for redundancy and robustness may utilize these new LSNs.

LEO satellite constellations typically operate at altitudes in the $500 - 1500$ kilometer range, contributing to reduced path loss and improved performance over GEO satellites [1]. However, this comes at a cost: lower altitudes result in much higher mobility as the satellites orbit multiple times per day. As such, handovers between user terminals (UTs), satellites, and ground stations (GSes) must occur frequently. These dynamics can cause applications to see atypical packet loss compared to terrestrial networks. Applications using Starlink, for example, may experience up to 2% packet loss [2].

The remainder of the paper is organized as follows. In Section II, we examine related measurement studies of other LSNs as well as transport-layer modifications that address the potentially negative interactions between congestion control and lossy LSNs. We set the stage for our measurements and experiments in Section III, which details the system under test throughout the rest of the paper. In Section IV, we present the statistical and temporal behaviors of the OneWeb LSN. Next, in Section V, we look at one potential strategy to mitigate TCP congestion control and retransmission issues within LSNs via end-to-end erasure coding. Lastly, Section VI closes and discusses potential avenues for further research.

In summary, this work contributes the following:

- An initial measurement study of the OneWeb LSN is performed to examine the statistical and temporal characteristics that the applications may expect when using the OneWeb LSN;
- We experimentally study the potential benefits of liquid data transport for TCP traffic over lossy LSNs.

II. Related Work

A. Measurements of LEO Satellite Networks

Due to the low access barrier of the constellation and the consumer focus, Starlink remained the subject of most measurement studies at the time of writing. In [2] and [3], preliminary surveys of Starlink's behavior are shown with measurements of latency, throughput, and HTTP performance. In general, Starlink is found to perform sufficiently well as a general Internet service provider, significantly reducing latency and increasing throughput from GEO SatCom. However, packet loss is present and relatively high, especially for the uplink path which could reach a loss rate of up to 2%.

Other works have expanded on this performance and revealed the underlying characteristics of Starlink. The underlying network structure of Starlink is dissected in [4], while the UT-satellite pairing mechanism is inferred in [5]. Granular measurements [6]–[8] point out the presence of a global controller in the Starlink system, which prompts a system reconfiguration every fifteen seconds. This results in noticeably discrete performance characteristics that change as handovers occur. For example, the baseline round-trip time (RTT) may increase due to a new UT-satellite pairing after system reconfiguration.

In contrast, OneWeb and other LEO satellite operators have not been the subject of nearly as many measurement studies. Instead, first-order models, simulations, and emulations have utilized public filings and constellation data to infer and predict system capacity and network performance, such as in [9].

B. Transport Layer Modifications

Several TCP variations have been proposed to alter the protocol to make it more resilient to variability within a LSN.

SaTCP [10] anticipates disruptive events that coincide with Starlink's reconfigurations and handovers. During times when disruptive events are likely to occur, reactions to losses (i.e. the reduction of the send window size) are weakened. In this way, SaTCP differentiates between true congestion events that occur between handover periods and unavoidable loss events that occur *during* handover periods. The emulation results show an increase of 47% in link utilization using a SaTCP-modified TCP CUBIC [11] algorithm. The authors note that link capacity is still suboptimal as the algorithm fails to account for nondeterminism in satellite locations. It is unclear whether *not* reducing the send window size during handover or loss events might negatively impact the network performance.

StarTCP [12] takes the opposite approach of SaTCP; where SaTCP tries to maintain high sending rates throughout handover periods, StarTCP buffers packets during handover periods and waits until the link stabilizes to resume sending. First, StarTCP uses a model based on historical traffic to predict handover events. Next, a separate model determines how far in advance sending should be stopped to prevent losses of in-flight packets. The last component of StarTCP is the probing mechanism, which checks every millisecond for link recovery to resume sending. Full emulation of StarTCP is necessary to better understand how it performs with the entirety of a satellite constellation and network stack accounted for. It may be that even with losses, sending throughout the handover period is more performant than not sending at all. Moreover, the sender and receiver must be wary of retransmission timeouts, which reduce the send window size dramatically.

StarQUIC [13] adopts a similar approach to SaTCP [10], preventing congestion control algorithms from mistakenly overreacting to regular loss events such as handovers. It simply prevents QUIC's congestion control mechanism from decreasing the send window size during regular handover events. However, StarQUIC runs on several assumptions. The core assumption is that the sender can anticipate loss periods, which does not necessarily transfer to other LSNs. Furthermore, the authors mention that devices *not* near a point of presence may need additional mechanisms to estimate the delay between themselves and the LSN in order to accurately determine when congestion window decreases should be prevented.

III. System Design

A. ARA Wireless Living Lab

We conduct our experiments on the ARA wireless living lab [14], located in Ames, Iowa, which allows usage and experimentation with various network equipment and infrastructure. ARA features two connected experimental platforms for advanced wireless research: AraRAN and AraHaul. AraRAN provides access to multiple radio access network (RAN) technologies at seven base station sites and tens of user equipment sites. AraHaul consists of fiber, microwave, millimeter wave, free space optical, and satellite backhaul technologies that interconnect five base stations.

In this study, we utilize AraHaul's OneWeb satellite link, enabled by the OneWeb LSN and user terminal (UT) deployed on the rooftop of Iowa State University's Wilson Residence Hall. Through ARA's online portal[1], we can run experiments from a Dell PowerEdge server directly attached to the OneWeb indoor unit (IDU). Additionally, we utilize a virtual machine (VM) from the Google Cloud Platform, `us-east4-a`, near Ashburn, Virginia. Fig. 1 depicts the system topology.

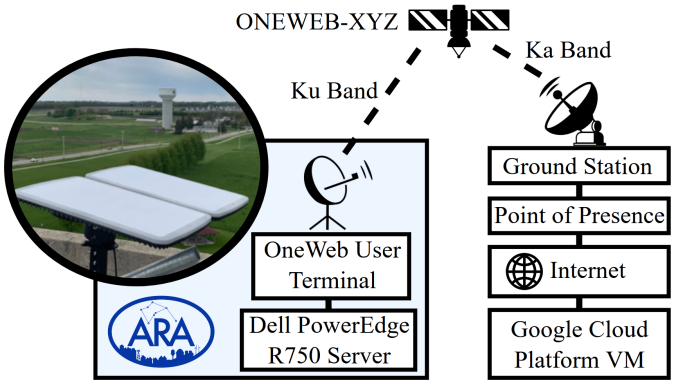

Fig. 1. Network topology of ARA satellite user terminal (UT) deployment and OneWeb infrastructure. The two testing endpoints, the ARA server and the Google Cloud VM, are located in Ames, IA and Ashburn, VA respectively.

The UT connects via the Ku band to the OneWeb satellite constellation. The connected satellite then connects back to OneWeb ground stations via the Ka band in a bent-pipe fashion. These ground stations transmit to and from OneWeb points of presence (PoPs), which are larger data centers where traffic within the OneWeb network is peered with the wider Internet. The service-level agreement (SLA) for ARA's OneWeb connection is $100\,\text{Mbps}$ downlink and $20\,\text{Mbps}$ uplink.

B. OneWeb Satellite Network

The OneWeb satellite network consists of ground infrastructure and a large satellite constellation in LEO. Currently, there are over 630 OneWeb satellites in orbit, with orbital altitudes of about $1200\,\text{km}$ [15]. These satellites, in contrast to Starlink's recent deployments, do not have inter-satellite links. Rather, their near-polar orbits and high altitudes allow a fewer number of satellites to consistently service the globe reliably. Each satellite contains 16 beams for connecting to UTs, and two gateway antennas [9]; the periodic behavior of the UT-satellite inter-beam handovers is shown in our measurements in Section IV.

[1]https://portal.arawireless.org

There are currently 29 OneWeb points of presence (PoPs). PoPs geographically close to the ARA UT are located in Ashburn (Virginia), Miami (Florida), Seattle (Washington), Los Angeles (California), and Honolulu (Hawaii) in the United States, as well as Toronto in Canada, shown in Fig. 2. The ARA UT traffic is generally associated with and sent to the PoP in Ashburn, VA.

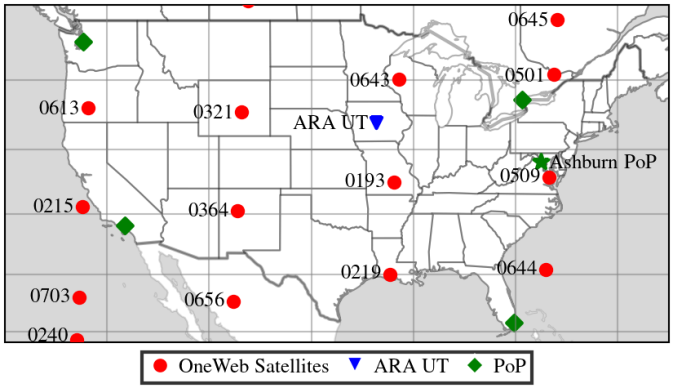

Fig. 2. Snapshot of OneWeb satellite positions, ARA user terminal (UT), and points of presence (PoPs).

IV. OneWeb Measurement

Currently, little work has been done to examine the exact performance of different protocols across the OneWeb LSN, which as we will illustrate, performs and behaves differently from the widely-studied Starlink LSN.

We conduct a systematic measurement study of the OneWeb LSN, focusing on latency and throughput performance between the ARA server and the Google Cloud VM depicted in Fig. 1. The VM is located near a OneWeb PoP where the LSN is peered to the Internet. In this way, minimal time is wasted unnecessarily traversing terrestrial backbone segments, which allows us to primarily observe the behavior of the OneWeb link with no RAN overhead and minimal impact from Internet traversal between the PoP and the VM.

Latency measurements are captured using `ping`, while `iperf3`[2] is used to measure throughput. We include results for UDP, TCP CUBIC [11], and TCP BBR [16]. We utilize CUBIC as it is the most predominant congestion control protocol in conventional operating systems at the time of writing. CUBIC is a loss-based congestion control protocol which regulates its congestion window size—and therefore the volume of traffic being sent—through reductions tied to packet loss events. In comparison, BBR is a delay-based congestion control protocol, as it maintains a model based on the estimated bandwidth and minimum RTT of the network that it traverses.

A. Latency

Fig. 3 visualizes the cumulative distribution function (CDF) of the round-trip time (RTT). More than 6500 RTT samples were collected over a period of multiple days. To establish a baseline, the RTT CDF of a conventional fiber connection between the same endpoints is also shown. Clearly, OneWeb's RTT has much more variability compared to regular, terrestrial networks, a trait which does carry over to other LSNs. Most traffic will see latencies in the $50 - 60$ ms range, although applications should be aware that a significant amount of traffic may exceed 100 ms latency.

The temporal behavior of OneWeb's RTT is shown in Fig. 4. For the most of the time, the UT is able to select a suitable satellite to achieve a base latency of around 50 ms. However, in some instances—as shown at around 18:14—the UT selects a satellite much further away. During the period where SINR is noticeably lower, the UT is connected to ONEWEB-0321, shown in Fig. 2. Prior, the UT connects to 0643, and afterwards to 0193, both are much closer to the UT. These handovers to distant satellites are undesirable but do occasionally occur unpredictably, significantly affecting latency.

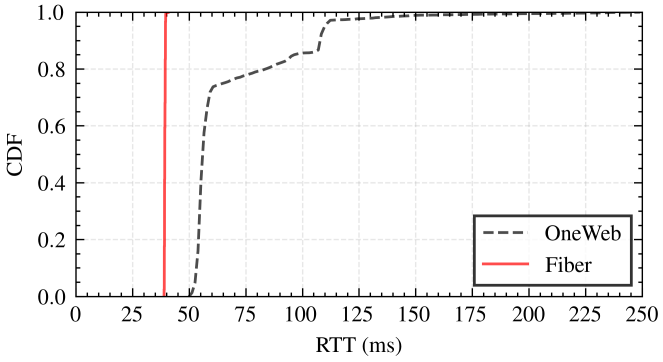

Fig. 3. Cumulative distribution function (CDF) of OneWeb latency between two testing endpoints (in Ames, IA and Ashburn, VA), as compared to the common fiber backhaul.

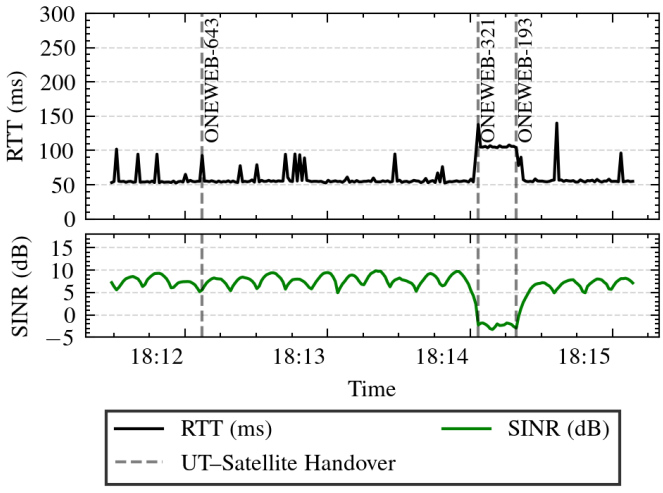

Fig. 4. OneWeb `ping` sample with round-trip time (RTT), signal to interference plus noise ratio (SINR), and inter-satellite handovers.

[2]https://iperf.fr

B. Throughput

The complementary cumulative distribution function (CCDF) of downlink (DL) and uplink (UL) throughput by protocol is shown in Fig. 5. As a baseline, the UDP downlink performs very well, close to the SLA target with minimal tailed behavior where throughput drops below 80 Mbps. Both TCP protocols show significantly more tailing than UDP, but perform similarly to one another in aggregate. We observe uplink performance that is generally the same for all protocols, but has significantly tailed behavior in comparison to downlink; nearly 16% of uplink samples fall below 10 Mbps. Raw UDP throughput is naturally higher than TCP's, which sacrifices throughput via retransmissions and congestion control to maintain reliable and in-order delivery.

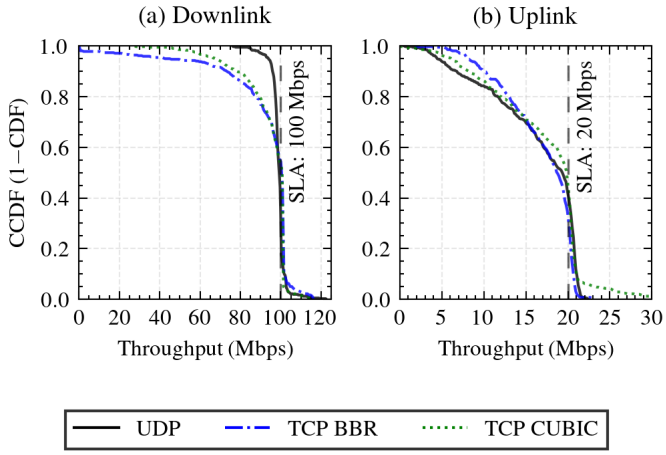

Fig. 5. Complementary CDF (CCDF) of DL and UL throughput. The SLA for the given direction (100 Mbps DL and 20 Mbps UL) is also shown.

For the temporal behavior of throughput by protocol, we examine Figs. 6 and 7, which show samples of raw throughput between the ARA UT and the VM in Ashburn, VA near the PoP. We can see the potential for very bursty packet loss when trying to upload UDP traffic at the SLA bitrate of 20 Mbps. A certain amount of packet loss when transmitting near the SLA is understandable as traffic shaping is necessary to enforce the SLA. However, we note a certain periodicity in the throughput, where dips occur a few times per minute. To better understand the origin of this behavior, we look at Fig. 8, which shows a sample UDP uplink test correlated with the UT's SINR data. We note that the inter-beam handovers, which occur at the valleys of the SINR plot, seem to correlate directly with throughput dips seen by the UDP receiver. Although periodic, losses and performance dips are not clearly predictable.

V. Liquid Data Transport for LSNs

A. Constraints within LSNs & Liquid Data Transport

As seen in the previous sections, packet losses and volatility are inevitable in LSNs. In [2], it is noted that loss rates of up to 2% may be observed for certain Starlink deployments.

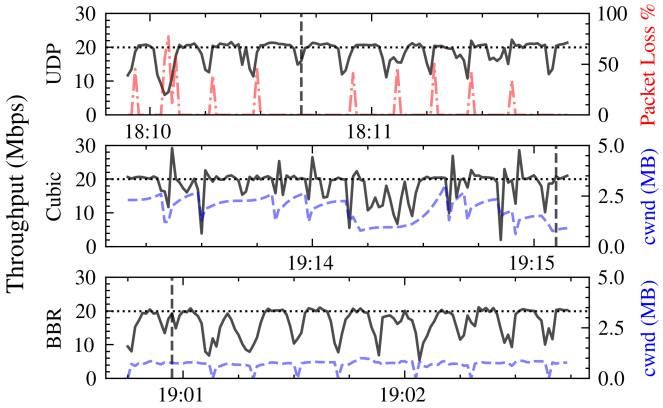

Fig. 6. Temporal UL throughput of UDP and TCP with various congestion control protocols. The UDP packet loss rate and TCP congestion window size (cwnd) are also shown. Inter-satellite handovers are depicted by vertical dashed lines, and the SLA is depicted by horizontal dotted lines.

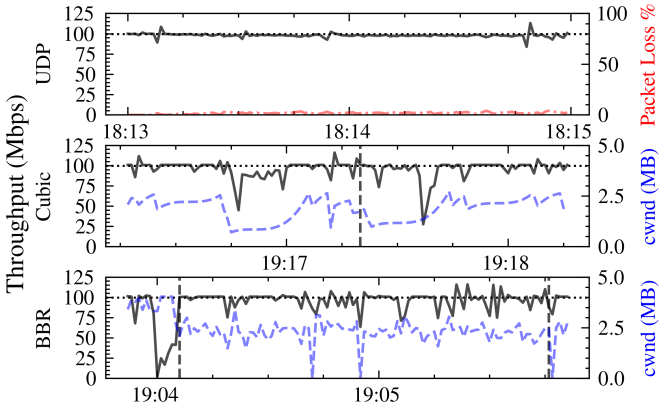

Fig. 7. Temporal DL throughput of UDP and TCP with various congestion control protocols. The UDP packet loss rate and TCP cwnd are also shown. Inter-satellite handovers are depicted by vertical dashed lines, and the SLA is depicted by horizontal dotted lines.

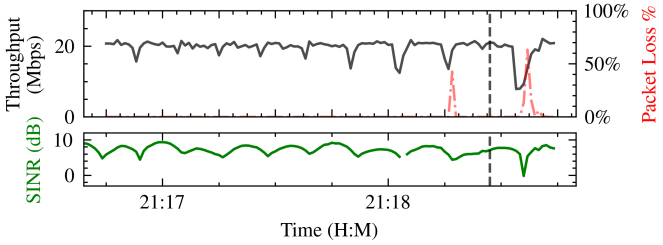

Fig. 8. UDP UL throughput and packet loss percentage, with UT observed SINR. Inter-satellite handovers are depicted by vertical dashed lines.

For many end-users, these are unavoidable occurrences since their UT is a black-box connection to the Internet. TCP flows in particular may react adversely to these losses, especially when the congestion control mechanisms are loss-based, such as CUBIC. There are a few approaches to possibly mitigate this, as we have noted in Section II-B. Modifying the congestion control mechanism or buffering TCP traffic while

using awareness of the UT's condition and satellite connection are approaches that have been proposed and investigated previously. Performance-enhancing proxies employed at the UT may also help by performing local ACKs, and LSN providers may continue to optimize their link-by-link coding and modulation schemes.

Another possible solution to alleviate the challenges incurred from traversing lossy LSNs is liquid data transport [17], [18]. Internally, liquid data transport utilizes RaptorQ forward error correction (FEC) codes [19], [20]. RaptorQ is a type of fountain code and, by extension, an erasure code. This is important for two reasons. First, erasure codes convert source data into a set of encoded symbols such that the original data can be reconstructed once a sufficient number of symbols have been received, which is an amount equal to or slightly greater than the number of original source symbols. This means that the order of arrival of the symbols does not matter, as long as a sufficient amount of symbols have been successfully received. Second, as a fountain code, RaptorQ can generate a potentially unlimited number of encoded symbols from a fixed set of source symbols. This enables the sender to produce as much repair data as needed, offering resilience against packet loss with tunable redundancy. Furthermore, RaptorQ offers high performance due to its linear-time encoding and decoding complexity [19]. These properties make RaptorQ FEC well-suited as the foundation for liquid data transport. Given the unreliable and volatile nature of LSNs, liquid data transport provides a robust solution to support real-time, data-intensive applications over LSNs.

B. Implementation and Results

We implement liquid data transport by using the BitRipple Tunnel [21]. We experiment with liquid data transport over our existing OneWeb LSN and deploy the tunnel in our architecture between the ARA server on ISU campus in Ames, Iowa, and the Google Cloud VM at Ashburn, Virginia, as shown in Fig. 1. The tunnel constructs virtual interfaces to automatically encode and decode application data at each endpoint while encapsulating and transmitting data between endpoints within UDP packets. Between the endpoints, there is a feedback link to provide link state information, which the tunnel uses to dynamically adjust the amount of repair data generated as overhead to recover from packet losses. The overhead itself is correlated with the amount of redundant data transmitted; for example, a redundancy rate of 10% will add 10% extra data as repair data, which will allow the link to overcome up to 10% packet loss.

We examine the statistical performance of liquid data transport in comparison to the baseline transport of TCP with no added coding scheme (aside from any coding which OneWeb may internally utilize). Fig. 9 compares the performance of liquid data with that of un-encoded TCP with BBR and CUBIC congestion control algorithms. We examine a time series of transferring data at a target rate of 10 Mbps over the LSN, which is subject to an artificial loss rate of 1%. This additive 1% loss is used to demonstrate the effectiveness of liquid data in instances where the LSN drops packets, as is possible with LSNs such as Starlink, and to a certain extent, OneWeb. Overcoming this packet loss is important due to the network requirements of real-time applications, of which such applications will incur performance degradation when retransmissions must be made, which is exasperated when using an LSN. This same benefit is possible over other wireless links where loss is frequent, not just LSNs, as shown in [18].

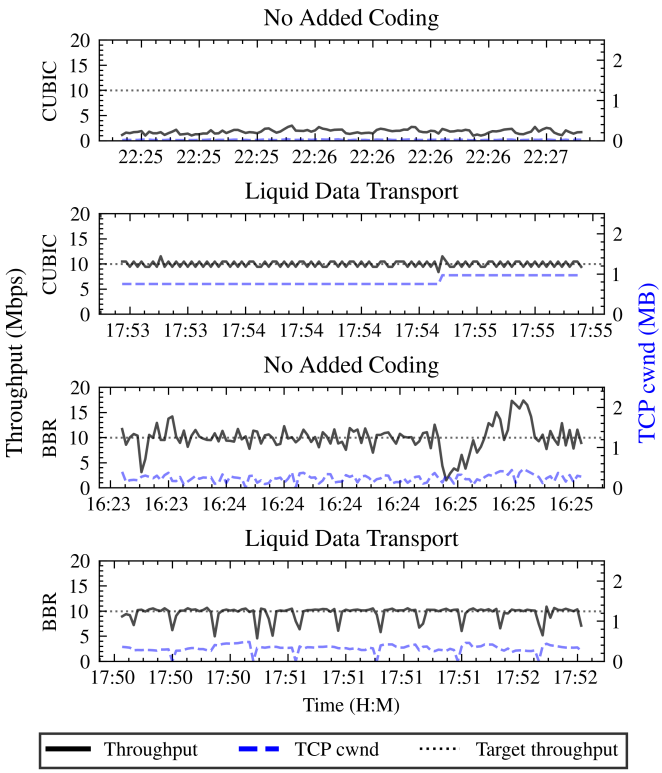

Fig. 9. Throughput uplink samples of TCP CUBIC and BBR, with and without the liquid data transport. All samples were subject to 1% additive loss to simulate lossy LSN conditions. The target bitrate for the `iperf3` test, 10 Mbps, is also shown.

Fig. 9 shows that the performance of CUBIC increases significantly when liquid data transport is utilized compared to without it. To better understand this, we can further analyze the cwnd shown in the figure. As we can see, the cwnd value never increases to a suitable quantity for CUBIC (with no coding) when consistent losses are experienced. Essentially, these loss-based congestion control mechanisms prevent any substantial growth of the cwnd. On the other hand, when a tunnel is established between the TCP sender and receiver by the liquid data transport, packet losses are overcome inside the tunnel using the redundant repair data generated by the erasure code, thus shielding the TCP endpoints. As a result, the cwnd remains around a constant high value that corresponds to the target data rate of 10 Mbps.

BBR's performance is comparatively similar in both cases

as BBR is a delay-based protocol and its model of the link is not drastically affected by these packet losses; not to the degree that CUBIC's loss-based model is. However, as the loss rate increases, the amount of retransmissions BBR (with no coding) must make increases correspondingly [22], whereas liquid data transport is able to overcome the packet loss without incurring additional retransmissions. For the traces shown in Fig. 9, BBR with no coding incurs 1767 retransmissions throughout the trace to transmit the application data, while BBR with liquid data transport only experiences 81 retransmissions.

Using liquid data transport, packet losses are shielded from the TCP endpoints where the congestion control algorithm is executed. It also eliminates or significantly reduces the amount of packet retransmissions. Both factors allow for natural growth and steady-state moderation of the cwnd size, which, in turn, results in increased and stable throughput.

VI. Conclusion

In this paper, we present the statistical and temporal behavior of the OneWeb LSN, which generally succeeds in delivering a reliable and performant Internet service as a black box for applications. We find that, while the effects of OneWeb's handovers are muted when compared to other LSNs studied in the literature, satellite selection can still greatly affect latency and throughput. We also explore the potential benefit of utilizing liquid data transport to improve the TCP performance over lossy LSNs.

As part of the future work, we aim to correlate additional weather data to better understand the performance impact of, for example, precipitation and rain rate on the OneWeb LSN and the ARA UT. Furthermore, we hope that measurement data from other OneWeb UTs in different geographic regions may be made available to form a more complete picture of the OneWeb system. Then, with a baseline understanding of OneWeb's characteristics, a better understanding of application-level performance for various purposes, such as video streaming or latency-sensitive control data, is needed. With a larger dataset, it may also be possible to construct a predictive model of the OneWeb system that can help predict increased latency or throughput dips. Such a model may be utilized to dynamically steer traffic based on both the application and satellite network states in scenarios where the LEO satellite network accompanies other backhaul methods.

Acknowledgment

This work is supported in part by Collins Aerospace, Hughes, BitRipple, and by the NIFA award 2021-67021-33775 and NSF award 2130889.

References

[1] D.-H. Jung, H. Nam, J. Choi, and D. J. Love, "Modeling and analysis of geo satellite networks," *IEEE Transactions on Wireless Communications*, 2024.

[2] F. Michel, M. Trevisan, D. Giordano, and O. Bonaventure, "A first look at starlink performance," in *Proceedings of the 22nd ACM Internet Measurement Conference*, 2022, pp. 130–136.

[3] S. Ma, Y. C. Chou, H. Zhao, L. Chen, X. Ma, and J. Liu, "Network characteristics of leo satellite constellations: A starlink-based measurement from end users," in *INFOCOM 2023-IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[4] J. Pan, J. Zhao, and L. Cai, "Measuring a low-earth-orbit satellite network," in *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2023, pp. 1–6.

[5] H. B. Tanveer, M. Puchol, R. Singh, A. Bianchi, and 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.

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

[7] J. Pan, J. Zhao, and L. Cai, "Measuring the satellite links of a leo network," in *ICC 2024-IEEE International Conference on Communications*, 2024, pp. 4439–4444.

[8] S.-M. Hammer, V. Addanki, M. Franke, and S. Schmid, "Starlink performance through the edge router lens," in *Proceedings of the 2nd International Workshop on LEO Networking and Communication*, 2024, pp. 67–72.

[9] I. Del Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta astronautica*, vol. 159, pp. 123–135, 2019.

[10] X. Cao and X. Zhang, "Satcp: Link-layer informed tcp adaptation for highly dynamic leo satellite networks," in *INFOCOM 2023-IEEE Conference on Computer Communications*, 2023, pp. 1–10.

[11] S. Ha, I. Rhee, and L. Xu, "Cubic: a new tcp-friendly high-speed tcp variant," *ACM SIGOPS operating systems review*, vol. 42, no. 5, pp. 64–74, 2008.

[12] L. Jiang, Y. Zhang, Y. Hu, Y. Cui, and X. Zhang, "Startcp: Handover-aware transport protocol for starlink," in *Proceedings of the 8th Asia-Pacific Workshop on Networking*, 2024, pp. 169–170.

[13] V. Kamel, J. Zhao, D. Li, and J. Pan, "Starquic: Tuning congestion control algorithms for quic over leo satellite networks," in *Proceedings of the 2nd International Workshop on LEO Networking and Communication*, 2024, pp. 43–48.

[14] T. U. Islam, J. O. Boateng, M. Nadim, G. Zu, M. Shahid, X. Li, T. Zhang, S. Reddy, W. Xu, A. Atalar, V. Lee, Y.-F. Chen, E. Gossling, E. Permatasari, C. Somiah, O. Perrin, Z. Meng, R. Afzal, S. Babu, M. Soliman, A. Hussain, D. Qiao, M. Zheng, O. Boyraz, Y. Guan, A. Arora, M. Y. Selim, A. Ahmad, M. B. Cohen, M. Luby, R. Chandra, J. Gross, K. Keahey, and H. Zhang, "Design and implementation of ara wireless living lab for rural broadband and applications," *Computer Networks*, vol. 263, 2025.

[15] OneWeb, https://oneweb.net/our-network, 2024.

[16] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time," *Queue*, vol. 14, no. 5, pp. 20–53, 2016.

[17] J. W. Byers and M. Luby, "Liquid data networking," in *Proceedings of the 7th ACM Conference on Information-Centric Networking*, 2020, pp. 129–135.

[18] E. Permatasari, E. Gossling, M. Nadim, S. Babu, D. Qiao, H. Zhang, M. Luby, J. W. Byers, L. Minder, and P. Aggrawal, "Real-time liquid wireless transport for video streaming in rural and agricultural applications," in *Proceedings of the 3rd Mile-High Video Conference*, 2024, pp. 54–60.

[19] L. Minder, A. Shokrollahi, M. Watson, M. Luby, and T. Stockhammer, "RaptorQ forward error correction scheme for object delivery," RFC 6330, Aug. 2011.

[20] A. Shokrollahi and M. Luby, "Raptor codes," *Foundations and Trends® in Communications and Information Theory*, vol. 6, no. 3–4, pp. 213–322, 2011.

[21] BitRipple, https://www.bitripple.com/solutions, 2024.

[22] Y. Cao, A. Jain, K. Sharma, A. Balasubramanian, and A. Gandhi, "When to use and when not to use bbr: An empirical analysis and evaluation study," in *Proceedings of the Internet Measurement Conference*, 2019, pp. 130–136.