Strategically Positioning On-Board PEPs in LEO-based NTN for TCP Throughput Improvement

Kyeongnam Park, Kyungha Kim, Hyungjoon Shin, Hojeong Lee and Hyogon Kim
Department of Computer Science and Engineering
Korea University

2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall) | 979-8-3503-2928-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/VTC2023-Fall60731.2023.10333683

***Abstract*—Non-terrestrial network (NTN) is a technology that enables mobile devices such as smartphones to directly communicate with Low Earth Orbit (LEO) satellites and access Internet services from anywhere on the globe. Although initial commercial releases offer only limited bandwidth, the technology is expected to rapidly expand its capacity with "mega-constellations." In this paper, we explore novel configuration options of the NTN architecture that will help boost the TCP throughput performance of an NTN user. We first demonstrate that traditional performance-enhancing proxies (PEPs) deployed on only terrestrial gateways do not have an overwhelming merit over end-to-end TCP Hybla. However, we show that an on-board PEP strategically deployed on satellites can achieve superior performance than both TCP Hybla and the PEP on the terrestrial gateway.**

***Index Terms*—Non-terrestrial network (NTN), TCP, throughput, performance-enhancing proxy (PEP), on-board proxy**

I. INTRODUCTION

THE 3rd Generation Partnership Project (3GPP) Release 17 is the latest standard for cellular communication, which includes specifications for non-terrestrial networks (NTN). NTN complements the role of terrestrial mobile networks (TN), enhancing reliability and service continuity. Commercial products and services have already begun providing satellite communication on smartphones since the end of 2022 [1], [2]. Innovations in satellite launch technology is accelerating this technological disruption by rapidly realizing "mega-constellations" of low-earth orbit (LEO) satellites [3]. It calls for an overall reassessment of existing techniques in terms of their capability to cope with the peculiarities of satellite connections that incorporate individual NTN user equipment, *i.e.*, smartphones. In particular, the impact on the performance Transmission Control Protocol (TCP) needs investigation, as majority of Internet applications will continue to use it in the foreseeable future.

TCP performance can be impacted by satellite links in two significant ways. Firstly, satellite links can experience a higher rate of link errors due to factors such as weather and satellite elevation. This can lead to packet loss even in the absence of network congestion, resulting in a decreased congestion window size and lower throughput. Secondly, satellite links can have a large round-trip time (RTT), which slows down the growth of the congestion window. This is because the congestion window growth is directly proportional to the RTT. Combined, these idiosyncrasies make the recovery of the congestion window after packet losses on satellite links difficult, further deteriorating the throughput.

There are currently two approaches to addressing TCP performance degradation in satellite communication. The first approach is to increase the TCP congestion window size at a rate proportional to the round-trip time (RTT), as seen in the TCP Hybla algorithm [4]. The second approach is to split the TCP connection into two or more legs along its end-to-end path and use a performance-enhancing proxy (PEP) [5] to isolate the negative impact of the satellite link from the adjacent non-satellite links. TCP Hybla focuses on addressing the delay-induced throughput drop, while PEPs deal with both delay and link loss. TCP Hybla is preferable when TCP cannot be split, for instance due to security reasons [6], [7]. It is also less expensive than deploying PEPs. However, TCP Hybla may not be able to achieve as high of a throughput as the PEP-based approach as it does not have a design feature to deal with link losses. Another advantage of PEPs is that the satellite link is transparent to end systems.

In this paper, we investigate the PEP-based approach to enhancing TCP performance in the novel context of NTN. Specifically, we explore the potential of an on-board PEP operating on LEO satellites themselves. We anticipate that the increasing processing power of today's technology may soon enable more sophisticated on-board computing on LEO satellites [8]. This idea aligns with 3GPP's standardization efforts to introduce regenerative payloads (including edge computing) on some satellites [10]. While PEP has proven effective in high RTT communication using geosynchronous earth orbit (GEO) satellites, its application has been less explored in the LEO setting, where the RTT is relatively shorter but has the potential for larger error rates. Moreover, in NTN, there is no special node that can accommodate the PEP functionality on the user side since the user terminal connects directly to the LEO satellite. In this context, this paper conducts a fresh investigation into TCP performance tested with various configurational alternatives.

II. RELATED WORK

TCP Hybla [4] addresses slow congestion window growth caused by large round-trip time (RTT) by inflating the congestion window using a factor $\rho \geq 1$ compared to TCP NewReno. This factor is the ratio of the real RTT to a "normalized" RTT (RTT_0), with $\rho = \max[RTT/RTT_0,\ 1]$, where the normalized RTT is applied equally to all TCP connections. TCP Hybla multiplies ρ to both the elapsed time t and congestion window W in the NewReno algorithm. This counteracts the lower congestion window growth rate and quickly fills the bandwidth-delay product proportional to the RTT. In essence, TCP Hybla adjusts all TCP connections with different RTTs to behave like the one with the normalized RTT. The congestion window evolution of TCP Hybla as a function of time t is as follows:

$$W(t) = \begin{cases} \rho \cdot 2^{\frac{\rho t}{RTT}}, & 0 \leqslant t < t_\gamma; \\ \rho[\rho \cdot \frac{t-t_\gamma}{RTT} + \gamma], & t \geqslant t_\gamma \end{cases}$$

where γ is the Slow Start (SS) threshold and t_γ is the time that takes TCP to grow the congestion window to the threshold. However, TCP Hybla is not designed to address non-congestion link losses. In this paper, we propose to use a PEP to handle such losses while still leveraging the large RTT-handling capabilities of TCP Hybla.

Several PEP implementations are available to address different network problems. For instance, split TCP is a commonly used technique to mitigate the impact of large round-trip times (RTTs) on satellite links by adjusting TCP parameters on different legs. When dealing with asymmetric links with significantly different upstream and downstream bandwidths, ACK filtering is typically employed. For handling wireless packet losses caused by collisions or interference, Snoop [11] is an integrated PEP solution that retransmits locally detected lost data segments on the wireless link using duplicate TCP acknowledgements. Conversely, a distributed PEP like D-Proxy [12] detects lost segments by examining sequence numbers and then retransmitting them. In this paper, we focus on split TCP as a PEP solution to address packet loss issues.

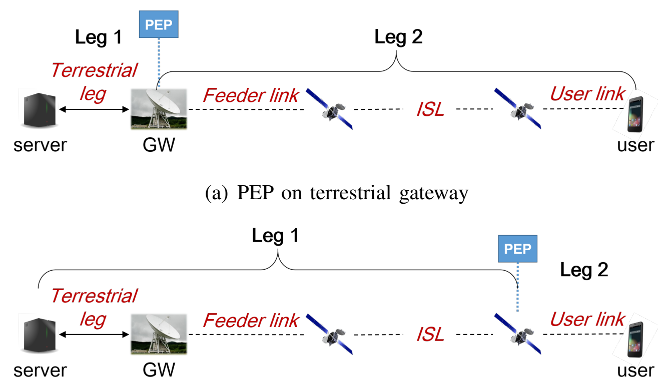
A work most relevant to ours is Luglio et al. [9] that proposes on-board proxy on geo-synchronous (GEO) satellites. It demonstrates that the throughput improvement by using such proxy is dependent on the on-board cache size. Although the work shares the same idea of positioning PEP on satellites, our work parts from it in that we investigate the issue of the most strategic positioning of PEP because LEO mega-constellations can have more numerous candidate locations than GEO. Moreover, the strategic position shifts over time as LEO satellites constantly move across the sky, whereas GEO satellites do not.

The compatibility issues between PEP and network-layer security protocols like IPsec are well documented [5]. However, one potential solution is to use Transport Layer Security (TLS) instead of IPsec. Similarly, PEPs cannot handle QUIC connections [6]. Compatibility between PEP and QUIC [13] will be explored in future works. For the purposes of this paper, we assume that PEPs can be deployed within the cellular network, and security concerns are addressed separately by the cellular infrastructure.

III. Sketch of Solution Approach

We propose deploying a PEP on board a LEO satellite. Specifically, we propose to strategically place a PEP on the satellite that serves the user link that connects directly to the user's smartphone. This idea of having an intelligent data plane function aligns with 3GPP's standardization efforts to introduce edge computing on some satellites [10] and the assumption that some works already make about future satellite computing capacity [8]. To the best of our knowledge, no prior research has been conducted on deploying an onboard PEP.

The idea of deploying a PEP on a LEO satellite parts from previous research on distributed PEPs deployed on two terrestrial gateways [5], [14], which showed significant throughput improvement. In NTN, however, a user's smartphone is mobile hence difficult to maintain a stable connection to another PEP on its side. Therefore, the distributed PEP architecture does not fit well with Non-Terrestrial Network (NTN) as envisaged by 3GPP. NTN can still use a terrestrial gateway on the feeder link side, though. Under this constraint, this paper further explores several PEP configurations including a single PEP on terrestrial gateway, a PEP on different satellites, and combinations of a terrestrial PEP and satellite PEPs. For instance, some single PEP placements investigated in this paper are shown in Fig. 1. The gNB as well as the core network also comes between the GW and server [15] are not shown for brevity.



(a) PEP on terrestrial gateway

(b) PEP on satellite

Fig. 1. Example of splitting TCP with a PEP in NTN

In NTN, because user's smartphone should directly connect to a satellite without a terrestrial gateway, the small form factor (hence small antenna size) and low transmit power of the smartphone can only support low data rate. It characterizes the satellite communication that is considered in this paper.

In determining the number and the location(s) of the PEP(s), Fig. 1 provides an insight. If a PEP is placed on a terrestrial gateway, the two most error-prone links (*i.e.*, user link and feeder link) get to be both on one side of the split TCP connection ("Leg 2"), as depicted in Figure 1(a). In contrast, if the PEP is placed on a satellite near the user (Figure 1(b)), errors are more evenly distributed between the two legs of the connection. Since the end-to-end throughput will be the minimum of the two split connections, placing the PEP on the satellite providing the user link is expected to produce better throughput. In the next section, we will investigate strategic locations for PEPs in NTN, through extensive simulation experiments.

IV. Performance evaluation

We perform simulation experiments in a NTN setting as envisioned by 3GPP for pedestrians in areas of extreme coverage [17] holding a smartphone. We use the ns-3 simulator with the ns-3-leo [18] extension to model the movement of LEO satellites in the simulation. Each simulation is run for 30 seconds, and all results are averaged over five simulation instances. The simulated topology is illustrated in Fig. 2, and detailed simulation configuration is summarized in Table I.

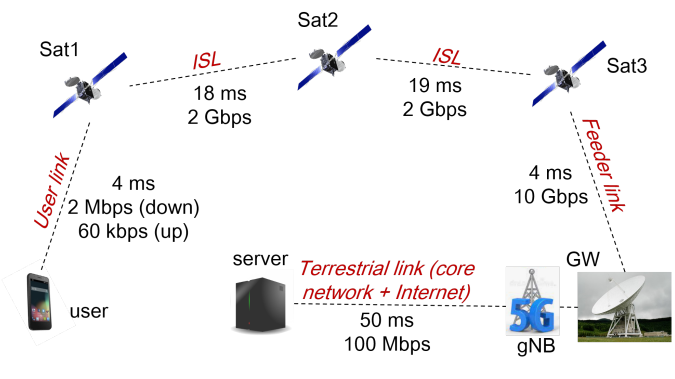


Fig. 2. Simulated NTN topology

TABLE I
SIMULATION PARAMETERS AND DEFAULT VALUES

	Parameter	Value
End hosts	Source location	(51.399°, 10.536°)
	Destination location	(40.76°, -73.96°)
	Rec. socket buf. size	max(128 KB, 0.4 BDP)
	Transfer size	70 MB
Gateway	Rec. socket buf. size	4.2 MB (S1G) 2.5 MB (S12G)
Satellite groups 1;2	Altitude	1,200 km ; 1,180 km
	Inclination	20° ; 30°
	Planes	32 ; 12
	Satellites per plane	16 ; 10
User/feeder links	Elevation angle	20°
	Tx power	105.9 dBm
	Free space path loss	189.3 dB
	Atmospheric loss	4.8 dB
	Link margin	0.36 dB
	Rx gain	31.8 dBi
	Rx loss	0.0 dB
	Bandwidth	10 Gbps (feeder) 60 kbps (user; upload) 2 Mbps (user; download)
	Delay	*variable*
	Congestion control	Hybla
ISL	Bandwidth	2 Gbps
	Delay	*variable*
	Loss	0
	TCP variant	Hybla
Terrestrial leg	Bandwidth	100 Mbps
	Delay	100 ms
	Congestion control	Hybla, NewReno

A. Bandwidth

The user link can be used in either direct or indirect connectivity mode (via a router, as in the case of Starlink), with the former being the focus of our investigation in this paper. The direct connectivity mode is intended for non-professional consumers who are mobile and connect directly to the satellite using the sub-7 GHz band [10]. The user link is assumed to have a capacity of 60 kbps for uplink and 2 Mbps for downlink [17]. This relatively small capacity is mainly due to the limitations of the smartphone's form factor and transmit power. For the feeder link, we use the default Telesat constellation configuration in ns-3-leo, which has a large bandwidth and will not bottleneck the TCP connection. The inter-satellite link (ISL) is also assumed to have a large bandwidth. The path from the gateway (GW) to the server consists of terrestrial links, which goes through the 5G core network and the Internet connected by a User Plane Function.

B. Error

We based our experiment on a recent report on the Starlink packet error rate [19], which ranges from 0.1% to 0.6%. To investigate the effect of this error rate, we simulate error rates of 0.1% and 1% in this paper. We assume that inter-satellite links (ISL) are highly reliable in LEO constellations [16]. Therefore, the error rate due to the satellite connection is implemented by the user and feeder links, and it is assumed to be evenly distributed between them. We also assume packet losses on the terrestrial segment, albeit much smaller than the satellite segments. In the simulation, we consider five scenarios with different error rates, which are presented in Table II.

TABLE II
ERROR SCENARIOS – PACKET ERROR RATES (PER)

Error Scenario		Satellite link	Terrestrial
1	No error	0	0
2	Low sat. + low terrestrial	10^{-3}	10^{-6}
3	Low sat. + high terrestrial	10^{-3}	10^{-3}
4	High sat. + low terrestrial	10^{-2}	10^{-6}
5	High sat. + high terrestrial	10^{-2}	10^{-3}

C. Delay

We assume fixed delays for the user, feeder, and ISL links in our simulations, as shown in Fig. 2, even though the delays of LEO satellites steadily change due to their movement. This assumption is reasonable since we are analyzing the TCP behavior over a relatively short duration (*i.e.*, 30 seconds).

D. TCP configuration

Unless stated otherwise, the TCP variant used in our simulations is Hybla, both for end-to-end and split connections. We set the normalized RTT for Hybla (*i.e.*, RTT_0) to 50 ms. To avoid receiver bottleneck due to socket buffer shortage, we set the buffer size to $\max$[128 KB, 0.4 BDP], where BDP is the bandwidth-delay product of the TCP connection. To achieve better performance with TCP Hybla, we enable the Selective ACK (SACK) option, which is turned on in all our simulations.

E. PEP placement

To investigate the effects of PEP locations and numbers, we conducted experiments using various configurations, as shown in Table III. We considered two different congestion control algorithms for our end-to-end TCP connections: NewReno (NR) and Hybla (HY). We also experimented with a single PEP in three different locations: two satellites and the gateway (GW). In addition, we investigated the impact of having two PEPs located on Sat1 and GW, and three PEPs located on Sat1, Sat3, and GW. All TCP connections, except for NR, utilized Hybla, whether in split or end-to-end mode.

TABLE III
TCP AND PEP CONFIGURATIONS

Configuration	# of PEPs	PEP location(s)	TCP variant
NR	0	-	NewReno
HY			Hybla
S1	1	S1	
S3	1	S3	
GW	1	GW	
S1G	2	Sat1, GW	
S13G	3	Sat1, Sat3, GW	

F. Simulation results

Fig. 3 compares the throughput of data transmission from the server to the user's smartphone. It shows that Configuration NR, which employs end-to-end NewReno, experiences significant degradation as the error condition increasingly worsens from Scenario1 to Scenario5. NewReno struggles to recover from the reduced congestion window size after packet losses, particularly under the given large RTT condition that is typical for the NTN topology.

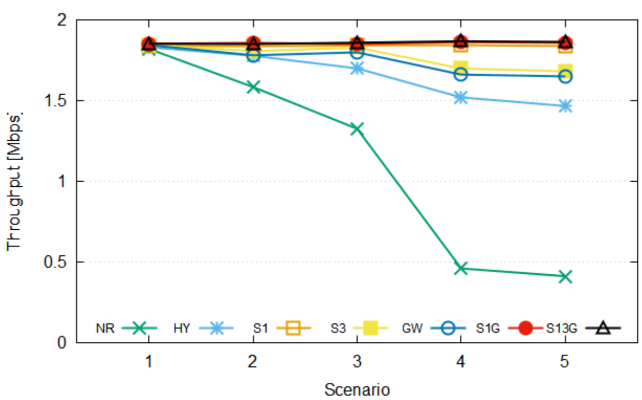


Fig. 3. Summary of throughput in download.

In contrast, Configuration HY, *i.e.*, end-to-end Hybla, demonstrates better performance than NewReno. It fares acceptably in low-error scenarios (2 and 3), with only a slight decrease in throughput compared to Scenario 1. However, in Scenarios 4 and 5, where the satellite link error rate increases up to 1%, it faces challenges achieving high throughput. This outcome is expected, given that Hybla is designed for handling large RTT, but not specifically for managing high link errors on satellite links. Therefore, to handle high errors on satellite links, TCP Hybla alone is insufficient, and additional mechanisms such as PEP should be considered.

A single PEP at GW (Configuration GW) produces slightly higher throughput than end-to-end Hybla, by up to 12.5%. Given the cost of deploying and operating a PEP, however, it may be more cost-effective to use end-to-end Hybla instead of placing a PEP on the gateway. Note that this result agrees with an earlier investigation with a similar testbed configuration [21]. On the other hand, a PEP at Sat1 produces more clear throughput gap of up to 25% compared to Hybla.

Careful consideration should be given if satellites are considered as candidate locations. Obviously, using more PEPs (*e.g.* Configurations S13G and S1G) achieves the best result. However, even with a single PEP, comparable performance can be attained. In particular, Configuration S1 performs close to the multi-PEP configurations, *i.e.*, within 1.4% of the best performance. The comparison between Configurations GW and S1 bears out the insight set forth in Section III that the most error-prone links should be separated across the PEP. Configuration S1 produces up to 11% more throughput than Configuration GW. On the other hand, Configuration S3 does not produce as good performance. Although Configuration S3 also splits the error-prone links, longer Leg 2 seems to be subject to stronger oscillations under Hybla. A detailed analysis of the simulation traces reveals that it makes Leg 2 spend longer in Fast Recovery (FR) and Slow Start (SS) than Congestion Avoidance (CA), as showin in Fig. 4. It means that there are more packet losses, i.e., three duplicate ACKs that lead to FR and more bursty losses that cause the retransmission timer expiry and eventually, Slow Start. The additional losses are caused by the longer feedback loop in Configuration S3. The more frequent packet losses reduce throughput for the configuration. All these results strongly suggest that a strategic selection of PEP location is necessary to minimize the cost of PEP deployment while maximizing its benefit.

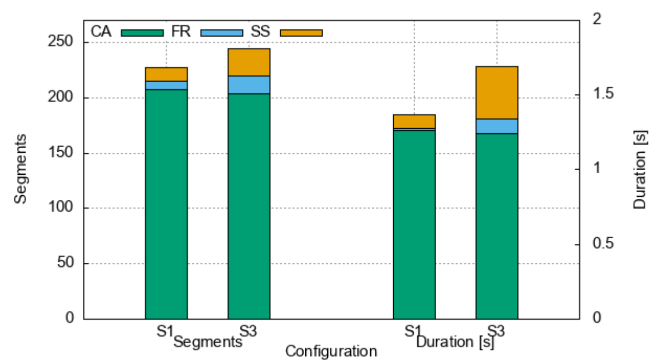


Fig. 4. Effect of longer control loop in Configuration S3.

In the NTN architecture, the user link typically has the least bandwidth and the largest error rate, making Sat1 the best location for error reduction and maximizing data flow through the bottleneck link from Sat1 to the user. The rule of thumb, therefore, is to place the PEP in the satellite that connects to the user terminal. In terms of the length of the feedback loop, it is the shortest for the error-prone link. As we saw in Fig. 4, it helps reduce the congestion losses and achieve better throughput.

Finally, frequent handovers in NTN using LEO constellations mean that many, if not every, satellite should be equipped with PEP functionality. To minimize on-board resource usage, however, the functionality should only be turned on for user links. For instance, if a satellite is an intermediate hop for a certain connection (*e.g.* used only as ISL), the PEP functionality should not be activated on it. It will decrease the energy footprint and the memory requirement on LEO satellites equipped with the PEP functionality.

V. Conclusion

This paper explores the possibility of a performance enhancing proxy (PEP) on a LEO satellite in the 3GPP NTN architecture. Different performance dynamics is obtained with direct satellite connection from the user's smartphone to a LEO satellite. The PEP on the ground gateway does not provide the best throughput performance due to the long RTT to the user that slows congestion window recovery and higher probability of having two error-prone satellite links on one side. Activating PEP functionality on the satellite that provides the user link is the most effective. Future work is needed in the areas of data uploads from users' smartphones to the server, various TCP congestion control algorithms, disparate distributions of link loss rates and round-trip times (RTTs), different types of cross traffic, and selective PEP activation.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience program (IITP-2023-2020-0-01819) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation.

References

[1] Apple, Use Emergency SOS via satellite on your iPhone 14, [Online]. Available at: https://support.apple.com/en-us/HT213426.

[2] T-Mobile, T-Mobile Takes Coverage Above and Beyond With SpaceX, [Online]. Available at: https://www.t-mobile.com/news/un-carrier/t-mobile-takes-coverage-above-and-beyond-with-spacex, Aug. 25, 2022.

[3] A. C. Boley and M. Byers, "Satellite mega-constellations create risks in Low Earth Orbit, the atmosphere and on Earth," *Nature Scientific Reports*, 11:10642, May 2021.

[4] C. Caini and R. Firrincieli, "TCP Hybla: a TCP enhancement for heterogeneous networks," *International Journal of Satellite Communications and Networking*, 22:547-566, 2004.

[5] J. Border *et al.*, *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*, RFC 3135, June 2001.

[6] J. Border et al., "Evaluating QUIC's Performance Against Performance Enhancing Proxy over Satellite Link," in Proc. IFIP Networking Conf., pp. 755-760, 2020.

[7] C. Mogildea, J. Deutschmann, K.-S. Hielscher, and R. German, "QUIC over Satellite: Introduction and Performance Measurements," in Proc. Ka Conference, Sep. 2019. [Online] Available at: https://www.researchgate.net/publication/351282748_QUIC_OVER_SATELLITE_INTRODUCTION_AND_PERFORMANCE_MEASUREMENTS.

[8] J. Kim, *et al.*, "Space Mobile Networks: Satellite as Core and Access Networks for B5G," *IEEE Comm. Magazine*, 60(4):58-64, April 2022.

[9] M. Luglio, M. Y. Sanadidi, M. Gerla, and J. Stepanek, "On-Board Satellite Split TCP Proxy," *IEEE JSAC*, 22(2):362-370.

[10] N. Chuberre, "3GPP NTN Standardization: Past, Current and Future," a presentation at Mobile Korea 2022, [Online]. Available at: https://www.youtube.com/watch?v=WgccPhJYpR0.

[11] H. Balakrishnan, S. Seshan, and R. H. Katz, "Improving TCP/IP Performance over Wireless Networks," in Proc. of the 1st annual international conference on Mobile computing and networking, pp. 2-11, 1995.

[12] D. Murray, T. Koziniec and M. Dixon, "D-Proxy: Reliability in Wireless Networks," in Proceedings of 16th Asia-Pacific Conference on Communications (APCC), pp. 129-134, 2010.

[13] H. Cruickshank, R. Mort, and M. Berioli, "Broadband Satellite Multimedia (BSM) Security Architecture and Interworking with Performance Enhancing Proxies," in Proceedings of International Conference on Personal Satellite Services (PSATS), pp 132–142, 2009.

[14] J. Deutschmann, K. Hielscher, and R. German, "Satellite Internet Performance Measurements," in Proc. NetSys, pp. 1-4, 2019.

[15] 3GPP, *Solutions for NR to support non-terrestrial networks (NTN) (Release 16)* ; TR 38.821 V16.2.0, Mar. 2023.

[16] M. I. Shehata *et al.*, "Optical Inter-Satellite Link over Low Earth Orbit: Enhanced Performance," *Journal of Physics: Conference Series*, 2128, 2021.

[17] 3GPP, *Study on Scenarios and Requirements for Next Generation Access Technologies*; TR 38.913 V17.0.0, Mar. 2022.

[18] T. Schubert, L. Wolf, U.Kulau, "ns-3-leo: Evaluation Tool for Satellite Swarm Communication Protocols," *IEEE Access*, 10:11527-11537, Jan. 2022.

[19] D. Paessler, "Analyzing Starlink speed and availability over 10 months with PRTG," Apr. 2022. [Online]. Available at: https://blog.paessler.com/analyzing-starlink-speed-and-availability-over-10-months-with-prtg.

[20] G. Montenegro *et al.*, *Long Thin Networks*, RFC 2757, Jan. 2000.

[21] C. Caini, R. Firrincieli, and D. Lacamera, "PEPsal: a Performance Enhancing Proxiy designed for TCP satellite connections," in Proc. IEEE VTC-spring, pp. 2607-2611, 2006.

[22] R. Sallantin *et al.*, "An end-to-end alternative to TCP PEPs: Initial Spreading, a TCP fast start-up mechanism," *International Journal of Satellite Communications and Networking* , 34:75-91, 2016.

[23] N. Parvez, A. Mahanti and C. Williamson, "An Analytic Throughput Model for TCP NewReno," *IEEE/ACM Trans. on Networking*, 18(2):448-461, Apr. 2010.

[24] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP Throughput: A Simple Model and Its Empirical Validation," *ACM SIGCOMM CCR*, 28(4):303-314, Oct. 1998.