QPEP in the Real World: A Testbed for Secure Satellite Communication Performance

Julian Huwyler
ETH Zurich
jhuwyler@student.ethz.ch

James Pavur
University of Oxford
james.pavur@gmail.com

Giorgio Tresoldi
and Martin Strohmeier
Cyber-Defence Campus
first.last@armasuisse.ch

***Abstract*—Although new technologies are on the rise, traditional Geostationary Earth Orbit (GEO)-based satellite internet is a crucial piece of critical communications infrastructure used by many, for example in the maritime sector. Previous work found that much GEO traffic is unencrypted, as there is a lack of secure, yet performant ways to communicate for end users. QPEP, a hybrid between a traditional Performance Enhancing Proxy and a VPN, aims to solve this issue but has only been tested in simulations. This work presents a newly developed testbed, which is used to collect real-world results for QPEP. Two different satellite links, one using Ka-band, the other Ku-band, were analyzed. In the Ka band, we find that QPEP offers on average 80% more goodput compared to OpenVPN. The page load time is reduced on average by 17% and the 95th percentile is reduced by 25% compared to OpenVPN. Although the average page load time of QPEP is higher compared to the unencrypted, proprietary PEP of the provider, the 95 percentile is equivalent. While satellite environments are often a black box that is difficult to evaluate scientifically, we show that in typical settings QPEP can prove its benefits in the real world.**

I. Introduction

By 2028, an expected average of 990 satellites are going to be launched yearly, resulting in an increase of 332% for the years 2019-2028 compared to the previous decade [18]. In the year 2020, about 1'600 satellites were used for communication [18]. A large user of these communication services is the maritime industry, as many ships rely on satellites for ship-to-shore communication. As 90 % of global trade relies on ships, with an annual expected growth of 3.4%, it is an important sector, which still lags behind in digitalization [1]. Therefore, there is growing demand for high-speed and ultra-reliable maritime communications [16].

Ships often use GEO-based satellite communication to communicate to the shore [12] as it only needs roughly six satellites to enable worldwide communication and is already well established within the industry.

Workshop on Security of Space and Satellite Systems (SpaceSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-89-4
https://dx.doi.org/10.14722/spacesec.2023.239792
www.ndss-symposium.org

As the GEO is at 35'786km, signals suffer a long delay, posing challenges to the performance and security of the satellite link. Traditionally, internet service providers use Performance Enhancing Proxies (PEPs) to improve the performance of TCP-based traffic. However, PEPs are often unencrypted, leaving the customer with an insecure connection. One of the popular options to enable encryption over otherwise insecure links is to use a VPN. Unfortunately, VPNs cannot profit from PEP due to the need of deep packet inspection for the functionality of PEPs. Therefore we, as a customer, are facing a trade-off between performance and security. Recently discovered vulnerabilities of unencrypted satellite traffic and the performance versus security trade-off motivated the development of a new protocol for secure, yet performant satellite internet: QPEP. It combines the techniques of PEP and VPN, with the objective to improve performance of secure traffic. QPEP makes use of the newly developed QUIC protocol, which has only been standardized recently. First results on a custom built simulated testbed look promising, as QPEP can even outperform traditional PEPs. Knowing its potential in this simulated testbed, it would be interesting to have measurements over real satellite links to further prove QPEP's benefits. This is where this work will continue the evaluation of QPEP by providing new results and insights on the performance in the real world.

Contributions

The goal of this work is to collect real-world measurements of QPEP. Thus, we contribute the following:

- We create a testbed to measure performance of different protocols over a satellite link. An automated measurement testbed using Docker was developed from the existing simulated one.
- We conduct first measurements of QPEP, which were continuously monitored over several months. Our measurements corroborate QPEP's simulations but also identify further areas for potential improvement for QPEP.

II. Background

In this section, we briefly explain the specifics of satellite communication, PEPs and QUIC.

A. Satellite Communication

Satellites typically communicate in a microwave frequency range between 3 GHz and 30 GHz, containing different bands for a variety of purposes. For satellite broadband communication the Ka, Ku and C-band are used, while the other bands are used, amongst other things, for GPS and satellite phones (L-band), weather/ship radar (S-band) and military purposes (X-band) [6]. The K-band portion is not suitable for long range transmission due to the resonance frequency of water vapor at 22.24 GHz [17]. The Ka-band offers a higher bandwidth but is susceptible to so-called rain-fade. To lower the effect of rain-fade, one can use the Ku-band or for even less susceptibility the C-band.

To connect to the internet via a satellite, we need a satellite dish, to send and receive signals, a satellite and a ground station/gateway which relays our signals. For a satellite in the GEO we face a few challenges and limitations. First of all is the latency: Because of the large distance, the signal takes about 130ms to reach the satellite, which adds up to 520ms (4x 130 ms) of Round Trip Time (RTT) at best, due to the signal having to travel from the customer terminal to the satellite , back to a ground station and the same way back, resulting in four times the delay caused by the large distance. Secondly, the satellite dish and the satellite must have line-of-sight (LOS) as the high frequency signals are susceptible to multi-path fading.

B. Performance Enhancement Proxies

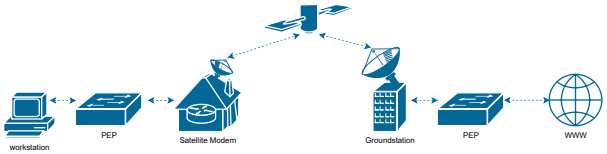


Fig. 1: A setup with a distributed Performance Enhancement Proxy (PEP), with PEP appliances on each side. Often they are directly integrated into the satellite modem and the ground station of the provider.

As summarized in RFC3135 [7], PEPs are used to improve performance on degraded links, where link layer optimizations do not work, e.g. for links with high delay bandwidth product. They are commonly used in satellite and other wireless links with high delay or under adverse conditions such as a high error rates. PEPs can be classified by design choices, such as distribution, symmetry, splitting and transparency.

Typically, PEP work on the transport and/or the application layer. Improvements are implemented using different means such as acknowledgment mechanisms, tunneling mechanisms, data compression, disconnection handling, priority based multiplexing or protocol boosters (additional error correction or jitter control).

Using a PEP has certain implications, most notably it breaks end-to-end semantics, which makes the usage of encryption at the IP layer, like IPSec, impossible, as most PEP improvements rely on the inspection of TCP packets. Other implications are the additional processing power and memory requirements for the inspection.

C. QUIC Protocol

QUIC was originally developed by Google in 2013 [10] to tackle challenges presented by the growing demand for reduced latency of web sites and applications. The broader usage of HTTPS using the TCP/TLS stack adds latency due to the way it initializes a connection using many RTTs. Most connections on the internet are short transfers, hence this generates a substantial overhead. Another problem known from HTTP traffic is the Head of Line blocking, which occurs when a TCP packet re-transmission blocks the stream from the HTTP application.

QUIC has been standardized in May 2021 by the IETF in RFCs 9000, 9001, 9002 [9], [15], [8]. It is the default transport protocol to use with HTTP version 3 and is different to the one from Google.

QUIC uses UDP as its underlying transport protocol. Because UDP is unreliable on its own, QUIC implements its own congestion control and loss recovery mechanisms. It sends data using a stream, which can be multiplexed for parallel data transfer. QUIC establishes a connection in one RTT for a new connection and 0-RTT for an already established connection, allowing for a significantly lower latency compared to the TCP/TLS stack. QUIC also houses a mechanism for network path migration in case the path changes. To ensure privacy and confidentiality QUIC uses TLS1.3, but it only uses TLS for key negotiation and uses its own transport streams to send traffic instead of the TLS record layer.

D. QPEP

As a consequence of the lack of encryption in PEPs and the performance overhead of traditional VPNs, QPEP [13] aims at providing a secure yet performant way to improve satellite communication, overcoming the trade-off between security and performance.

As seen in Figure 2, QPEP is implemented as a distributed PEP, establishing a QUIC tunnel between two nodes. Such an implementation ensures compatibility with other web services that do not run over QUIC. Another benefit of the distributed architecture is, that it can operate transparently. Up on starting up the client launches a long-lived QUIC tunnel with the server. Each incoming TCP connection to the client is assigned to an unique QUIC stream in this tunnel. The client does not tunnel TCP packets directly, but selectively terminates TCP connections, drops spurious acknowledgments and converts only relevant data to a QUIC packet to be sent in the tunnel.

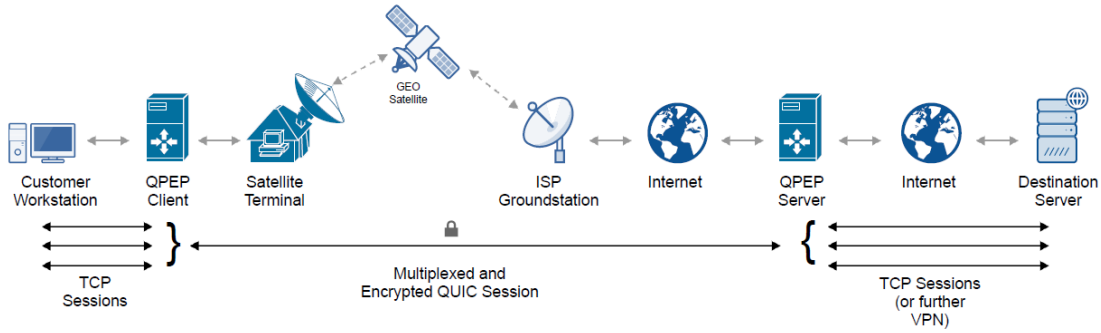


Fig. 2: The distributed architecture of QPEP simplified. Graphic from the original QPEP paper. [13]

The QPEP code written in Go is publicly available and open-source[1]. At the time of writing only the TCP/IP stack has been implemented to be tunneled. The 0-RTT option of QUIC has not been implemented.

III. Related Work

A. GEO Satellite Security

Very Small Aperture Terminal (VSAT) networks are widely used in the maritime industry and have revolutionized the communication from vessels to shore, aiding digitization. Pavur et al. demonstrate attacks against VSAT networks using equipment less than $400 [12], ultimately motivating the development of QPEP. Using a custom tool, the authors extracted 1.3TB of real-world satellite data, which originate from boats in an area of about 26 million square kilometers. Not only did they find sensitive data from some of the world's largest maritime companies, but also personal data like passport numbers and credit card details, showing the impact of insecure satellite communication. Lastly, they also present ways to intercept and modify TCP sessions under certain conditions, allowing for man-in-the-middle and DoS attacks.

B. Real-World Satellite Testbeds

In [5] an analysis of the performance of splitting PEP over a commercial Ku satellite link is executed. The performance gain is examined in different scenarios. It was discovered that splitting is highly sensitive to random losses as well as the number of simultaneous connections. Furthermore, the performance gain of using a PEP increases with larger file sizes.

[14] analyzes the performance of QUIC over a Ka satellite link using two websites between a local time of 2-4PM. It found that PLT doubled under QUIC compared to an optimized TCP connection. It was difficult to conduct controlled experiments because of specific optimizations by the satellite operator.

[1]https://github.com/ssloxford/qpep

In [2], the authors compare Google's QUIC to the accelerated transport layer over a GEO satellite link. To evaluate the performance, they downloaded a 1GB file from Google Drive over real and emulated satellite links. They found that QUIC achieved only 20% of the throughput of the accelerated transport layer.

In [4], Deutschmann et al. measured the performance of GEO internet of three different providers. For the performance evaluation measurements such as one-way delay, bulk-data transfer and website download times were considered. They found, that QUIC outperforms TCP tunneled in VPN, but not TCP optimized with a PEP. QUIC, for some providers, had a larger jitter for PLTs, which is, according to the authors, caused by the larger jitter in UDP traffic. They state that QUIC suffers from the non-applicability of the PEPs.

IV. Testbed Design

To measure the performance of QPEP and compare it to other protocols in a consistent way, a simulated testbed is provided by [13]. We briefly cover the simulated testbed, before explaining the design of the real-world testbeds we created to measure the performance of QPEP over commercially available satellite links.

A. Existing OpenSAND Testbed

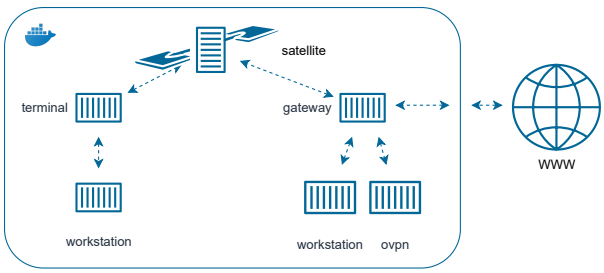


Fig. 3: The simulation testbed using OpenSAND.

The original QPEP testbed is implemented using Docker containers. These containers are connected as

depicted in Figure 3, with separate subnets for the terminal, satellite and the gateway. It represents a typical satellite internet setup with a customer workstation connected to the satellite terminal, which sends and receives all traffic. Adjacent to the gateway on the other side, a workstation with various purposes and an OpenVPN server is placed, dedicated to the measurements. The gateway container is connected to the world wide web, to be able to browse the web from the customer workstation over the simulated satellite link.

OpenSAND allows for an advanced simulation of satellite channels. Rather than only artificially adding a higher delay, it has more advanced features like attenuation modeling and SNR emulation.

B. Real-world Testbed

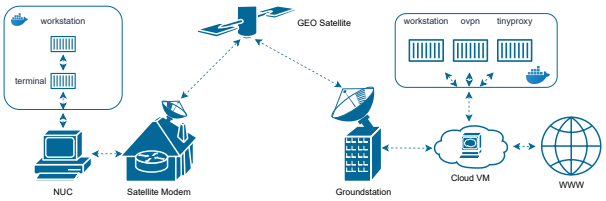


Fig. 4: The Docker-based testbed, which runs over a real satellite link.

To ensure that the results are comparable, our real-world testbed was built on the simulated one. It is a split version with containers on each side of the satellite link as depicted in Figure 4. The containers are, whenever possible, identical to the one from the simulated testbed. One of the main differences is the absence of the gateway container, as it is unnecessary in the cloud for the connection to the satellite. The PEPsal on the gateway side was moved to the workstation. An additional container, which houses a https proxy was added. This ensures that all the traffic from for the PLT passes the cloud, such that the results are comparable and the protocols that pass the cloud anyway (QPEP and OpenVPN) are not disadvantaged as there might be a more direct way from the satellite ground station to the webserver of a specific website. An overview of the functions of each container can be found in Table I.

Name	**Function**
workstation (terminal)	hosts all client side tests (iperf and PLT)
terminal	hosts client sideof QPEP and PEPsal appliance and connects to the host (and further to satellite)
workstation (gateway)	hosts the QPEP server as well as the IPerf server and a PEPsal appliance
ovpn	hosts the OpenVPN server
tinyproxy	hosts a http/https proxy to route traffic for PLT

TABLE I: Containers of the testbed and their purpose.

Since the PLT measurements and PEPsal are designed to run on x86 architecture instead of ARM, we used Intel NUCs instead of Pis for the terminal side. These computers are connected to the satellite modem. Additionally, they have an out-of-band access via a terrestrial network to be able to remotely control measurements over the satellite link and upload data without using data volume from the satellite link or interfering with ongoing measurements.

Fig. 5: Picture of one of the real-world setups.

For the gateway, we decided to deploy the server-side on cloud virtual machines. To minimize the influence of the Internet on our measurements, we placed the cloud virtual machines as close as possible to the physical location of the ground stations. The setup and a guide for the deployment of the testbed can be found in the repository at https://github.com/jhuwyler/qpep.

Our outdoor setup in a weather-proof box is depicted in Figure 5 and has withstood rain and heat waves.

C. Automated Measurement Design

Collection	Purpose	Docs	Total Size
iperf_CH	Channel characterizations	340	275.1 KB
iperf_TCP	Goodput measurements using TCP	764	3.9 MB
iperf_UDP	Goodput measurements using UDP	66	466.9 KB
sitespeed	PLT measurements	663	649 KB

TABLE II: The total number of documents in our different collections over three months.

In order to enable periodic measurements and to take advantage of the available unlimited data volume during nighttime, all measurements were automated. Crontab scheduled our Python scripts, orchestrated the testbeds and measurements. These Python scripts are mostly the same those used in the simulated testbeds, but adapted to our distributed setup and with improvements on robustness. The scripts are run on the client and orchestrate the Docker containers and setups on the cloud as well, allowing for measurements to be started automatically from one point. Another addition is the automatic upload to a MongoDB database (as most results come in JSON format). Table II lists

```
{
"_id":{"$oid":"60de6489561f6f57c58bf538"},
"date":{"$date":"2021-07-02T02:57:44.391Z"},
"testbed":"real-world A",
"scenario":"dist_pepsal",
"ping":1085,
"measurements":{...}
}
```


Fig. 6: Sample document from the *sitespeed* collection.

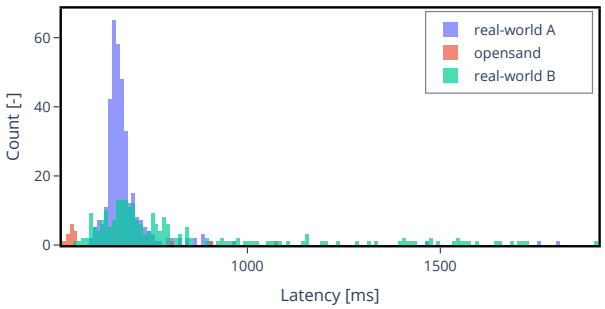


Fig. 7: Histogram of pings of the three testbeds.

the total number of documents in our database. Each document contains data from metrics collected by a specific testbed and scenario as well as the date and a latency measurement. The latency measurement ensures that the measurement was performed over the satellite link, as well as to collect a latency of the satellite link over time. Figure 6 shows a sample document from the *sitespeed* collection, to show how the data is stored.

V. Experimental Design

A. Metrics

To run a first benchmark of the different protocols, we measure two distinct metrics: Goodput and PLT.

1) Goodput Measurements: We use IPerf measure the good put of different scenarios. The IPerf server was running on the gateway workstation container and the IPerf client initialized the measurements from the client workstation container. We measured mainly using the TCP protocol and set the reverse option of IPerf to measure the download speed at the client side instead of upload. For each measurement, we used payloads with sizes ranging from 0.5 up to 10 Mbits in 250 kbit intervals, similar to previous simulations [13].

2) Page Load Time Measurements: The loading time of a web page is crucial for the user experience and in general can be a good indicator for performance. Loading a page requires the opening of different connections, sometimes even to multiple servers, and loading different smaller files.

B. Test Scenarios

We evaluate QPEP in our testbed against similar test scenarios to the simulations in the original paper.

1) Plain Connection: The plain connection has no performance improvements nor security features enabled. We measure it to compare the bare connection of the satellite link, without traffic enhancements or a VPN. We use a proxy server for the PLT measurements to route the connections to a webserver through our server-side testbed to improve the comparability of the scenarios.

2) PEPsal: PEPsal, an open-source implementation of an unencrypted PEP [3] is the only openly available PEP. We used this integrated PEP in both an integrated (only working on the client side) and a distributed (working on the client as well as the gateway side of the satellite link) manner.

3) OpenVPN: OpenVPN [11] is a popular VPN. It has no specific satellite optimization. We used it to compare QPEP to another encrypted protocol. OpenVPN was configured to work over UDP on the default port 1194. We configured an OpenVPN server on the gateway side of the satellite link and use the standard client software on our workstation on the terminal side of the satellite link to create a tunnel.

4) QPEP: We used the standard QPEP configuration [13], making sure we can compare it to previous results. One thing that was added to the QPEP implementation in comparison to previous work, is the ability to set the port of QPEP via a command line flag.

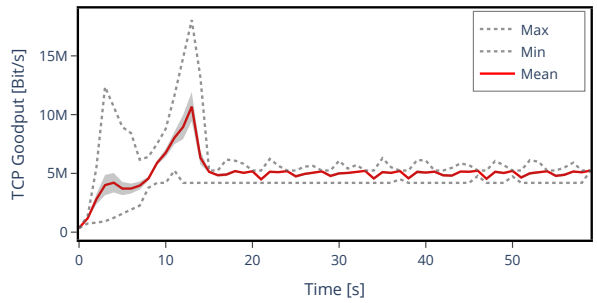
VI. Preliminary Evaluation

We present some early results from our testbeds.

A. Channel Characterization

We first characterize the raw satellite links. Figure 7 shows 2.5 months of pings within the two real-world and the OpenSAND setups. For the pings the latency at the end of each TCP goodput measurement was used. The ping is independent of enhancements and protocols, so we used measurements with and without provider enhancements. The results indicate that the real-world channels had a higher mean than the simulated OpenSAND one, as Setup A had 685ms, Setup B 831 ms and OpenSAND 610 ms. It has to be noted that the latency in the OpenSAND testbed can be adjusted. We further see that the channel of Setup B had a much wider distribution than Setup A.

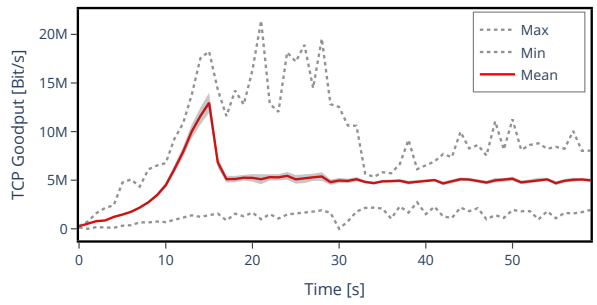
Figure 8 shows the channel throughput measurements. On the left we see the measurements with the enhancements disabled, on the right with the enhancements enabled. The satellite link on Setup A clearly resembles the simulated OpenSAND. The effect of the



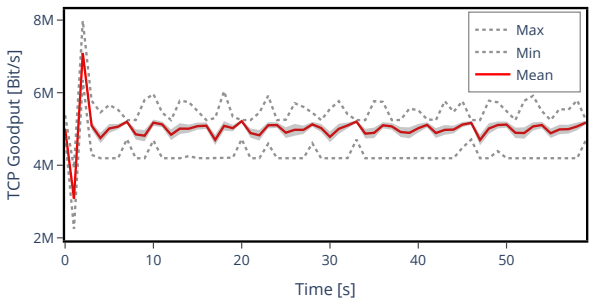
(a) OpenSAND

Testbed	Days	Count
OpenSAND	1	60
Setup A	2	91
Setup A PEP	4	40
Setup B	1	79
Setup B web-caching	2	40

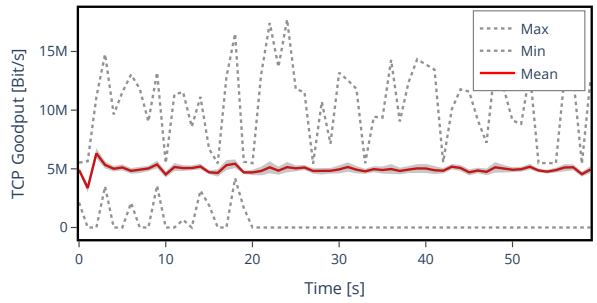
(b) The number of days and measurements.



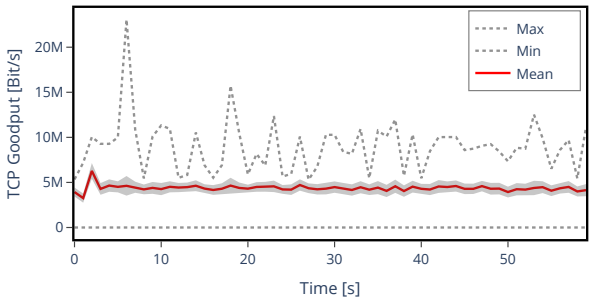
(c) Real-world Setup A no PEP



(d) Real-world Setup A with PEP



(e) Real-world Setup B no web-caching



(f) Real-world Setup B with web-caching

Fig. 8: Goodput measurements over 60s averaged over multiple measurements, 95 % confidence interval in gray.

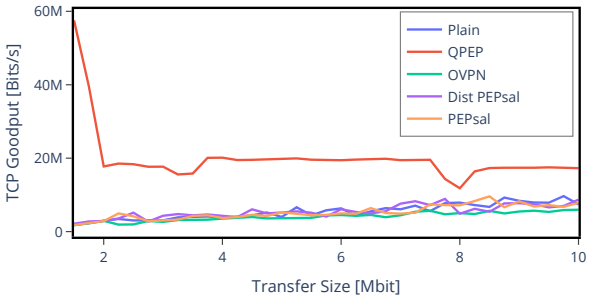
PEP on Setup A can also be seen as the slow start of TCP was eliminated. The similarity of the two plots of the link in Setup B can be explained by the fact that only website caching was disabled by the provider. As measuring with IPerf does not involve queries to websites, there is no impact. As there is no TCP slow start, we can infer that Setup B still has an active PEP. It was not possible to have this PEP disabled, hence we exclude Setup B from further analysis. Furthermore, we observed that the OpenSAND testbed had a significantly better UDP performance than the real-world testbeds.

B. Goodput Measurements

From the results in Figure 9, we can see that Setup A performs similar to the simulations by the OpenSAND testbed. In both plots we can see, that QPEP outperformed the other scenarios, although QPEP had a higher throughput in the OpenSAND testbed with an average of 19.86 Mbit/s compared to an average of 6.07 Mbit/s in Setup A. We can also see, that if the PEP of the provider was turned on, it improved the throughput of all the TCP traffic from an average of 3.73 Mbit/s to 24.43 Mbit/s for the plain scenario on Setup A. We see, that QPEP and OpenVPN could not benefit from the PEPs, as they are both UDP based. QPEP, compared to the OpenSAND testbed, could not outperform the PEPs of the providers. Compared to OpenVPN, QPEP had a better performance with a throughput of 6.62 Mbit/s compared to 3.67 Mbit/s for OpenVPN. Especially for smaller file sizes QPEP could prove its benefits.

C. Page Load Time Measurements

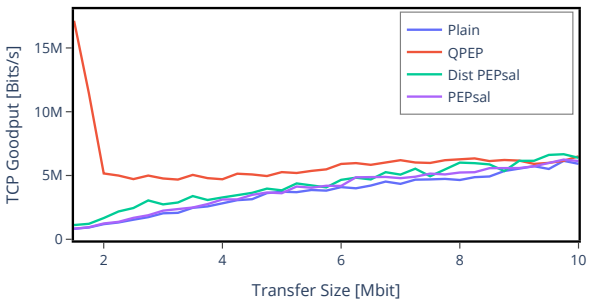
Figure 10 provides the plots of the PLT measurements of the Alexa Top 20 list. We can see that the



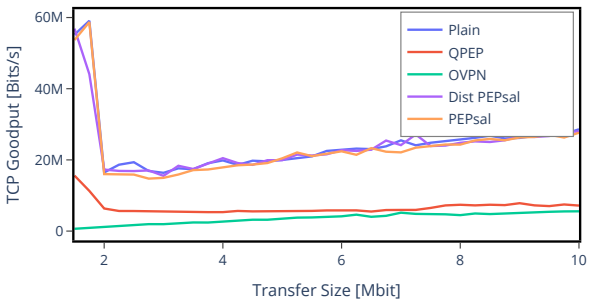
(a) OpenSAND

Testbed	Days	Count
OpenSAND	1-3	4-8
Setup A	18-25	18-25
Setup A PEP	14-15	14-15

(b) The number of measurements.



(c) Real-world Setup A no PEP



(d) Real-world Setup A with PEP

Fig. 9: Goodput measurements for the different testbeds. Note that OpenVPN is only present in the PEP-enabled measurements, because of errors in the measurement of the non-PEP. The count is per transfer size.

different testbeds showed diverse behaviours. While the plot of the OpenSAND testbed shows significant differences between the different scenarios, these differences were smaller on the other testbeds. On Setup A with the provider PEP turned on QPEP outperformed OpenVPN, as we can see that it improved the PLT at lower loading times and improves the 95th percentile by 9%, eventhough QPEP had a 3% higher average PLT. If the provider's PEP was turned off, QPEP could outperform the plain connection and every other protocol, as it had an average PLT of 31.37 seconds compared to 34.96 seconds of the plain connection.

VII. Discussion

We briefly discuss the obtained results, including some of the underlying reasons and potential consequences for future real-world tests.

A. Setup A without PEP matches OpenSAND

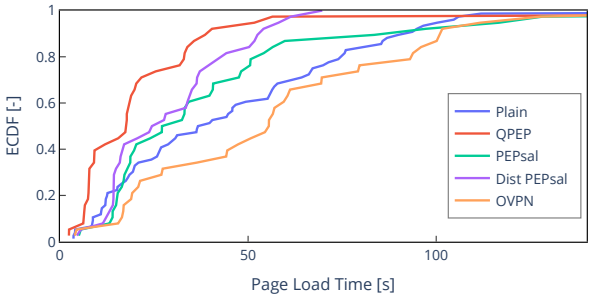
Considering channel characteristics, goodput and PLT, the Setup A testbed without activated PEP closely resembled the simulated OpenSAND testbed. Setup B seems to be a less reliable link, with channel measurements deviating from the average of the other testbeds. It also did not allow us to completely disable the PEP. We regard the B link as a special case, as the link and network reached "end-of-service" according to the provider.

The OpenSAND testbed is a simulated network with no congestion as the performed measurements are the only traffic on the satellite link itself but also the only traffic on the gateway side. Therefore UDP traffic was nearly lossless, consequently more reliable than in the other testbeds. Judging from the first measurements, QPEP might benefit even more from this nearly lossless channel compared to OpenVPN.

The provider networks remain "black-boxes", which makes it difficult to gain detailed insights into the performance of certain scenarios. It remains unclear, for example, if certain performance degradation of UDP traffic was caused by the satellite channel itself or by the link from the ground station to the virtual machine in the cloud, however it is assumed that the main limitation is the satellite link, since satellite links are less reliable than general networks.

B. QPEP performs better than OpenVPN

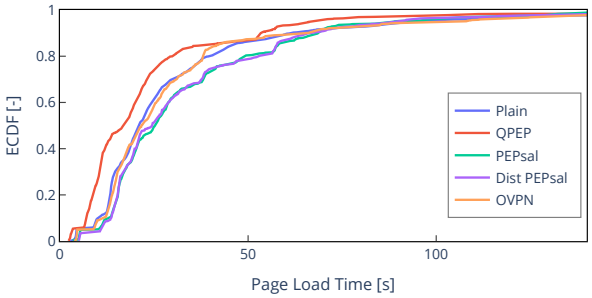
In general, QPEP could prove its benefits, considering PLT on the A testbed had the performance of unencrypted traffic accelerated by the provider PEP. Considering goodput, QPEP showed less performance compared to the simulated results, which is likely caused by the increased loss of UDP traffic in the real-world setups. The PEP of the provider however has no effect on QPEP itself, which agrees with the findings from [4].



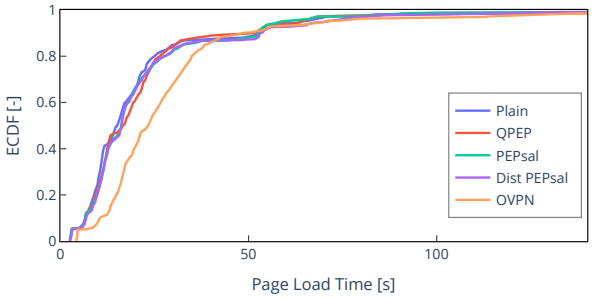
(a) OpenSAND

Testbed	Days	Count
OpenSAND	1-2	38-76
Setup A	19-37	759-1592
Setup A PEP	16	484-640

(b) The number of measurements for the PLT measurements.



(c) Real-world Setup A no PEP



(d) Real-world Setup A with PEP

Fig. 10: The ECDF function of the PLT of the different testbeds. The count is cumulative for all websites.

C. Smilar PLT for QPEP with lower Goodput

On Setup A, we observed that QPEP had less goodput than PEP-enhanced traffic, but similar PLT. Therefore, for practical websurfing use cases, QPEP still has an advantage over traditional PEP and OpenVPN, even in setups where it has goodput.

D. A QPEP parameter search may be beneficial

An investigation of the configuration parameters might lead to better results with more suitable parameters for the satellite links (as also suggested in [14]). Anecdotally, QPEP had rather long page load times on Setup B for some less visited websites. This might indicate some unwanted behavior of QPEP, possibly the re-transmission window might be too large, such that QPEP is stalled by lost packets.

Overall, we can conclude that QPEP's use of QUIC can improve performance as previously indicated by [4]. For some specific instances, however, it has a poorer performance compared to OpenVPN, which might potentially be solved by tuning the parameters properly such that the use of QUIC can be beneficial. In general, we can say that QPEP dominates OpenVPN, which is its main goal, as QPEP was developed to provide better encrypted traffic performance over satellite links.

VIII. Future Work

Future work will see more experiments on all testbeds, including tests of Wireguard and other potential VPN implementations. If possible, other GEO providers should be tested to understand the issues in the satellite ISP ecosystem to the fullest extent possible. Ideally, cooperation with a provider should be established, to understand the performance-enhancing black box and possible throttling of services. Beyond this, the testing of LEO setups such as Starlink and Iridium and whether they can benefit from PEPs and user-space encryption should also be pursued.

IX. Conclusion

QPEP offers a way for anybody to secure their traffic, while improving performance in high-delay bandwidth products such as satellite networks. This work provided two real-world testbeds for QPEP, which was previously only assessed in simulations, and conducted some first real-world measurements.

A total of 1'833 measurements were collected over 103 days. The PLT was reduced on average by 17% and the 95th percentile is reduced by 25% compared to OpenVPN. While the average PLT of QPEP was higher compared to the unencrypted, proprietary PEP of the provider, the 95 percentile was the same. While real-world measurements are sometimes difficult to compare, since the underlying network is a black box, we conclude that QPEP can principally keep its promises.

REFERENCES

[1] V. Babica, D. Sceulovs, and E. Rustenova, "Digitalization in maritime industry: Prospects and pitfalls," in *ICTE in Transportation and Logistics 2019*, E. Ginters, M. A. Ruiz Estrada, and M. A. Piera Eroles, Eds. Cham: Springer International Publishing, 2020, pp. 20–27.

[2] J. Border, B. Shah, C.-J. Su, and R. Torres, "Evaluating quic's performance against performance enhancing proxy over satellite link," in *2020 IFIP Networking Conference (Networking)*, 2020, pp. 755–760.

[3] S. A. D. Lacamera, "Pepsal: A tcp performance enhancing proxy for satellite links." [Online]. Available: https://github.com/danielinux/pepsal

[4] J. Deutschmann, K.-S. Hielscher, and R. German, "Satellite internet performance measurements," in *2019 International Conference on Networked Systems (NetSys)*, 2019, pp. 1–4.

[5] N. Ehsan, M. Liu, and R. J. Ragland, "Evaluation of performance enhancing proxies in internet over satellite," *International Journal of Communication Systems*, vol. 16, no. 6, pp. 513–534, 2003. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.593

[6] ESA, "Satellite frequency bands." [Online]. Available: https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Satellite_frequency_bands

[7] J. Griner, J. Border, M. Kojo, Z. D. Shelby, and G. Montenegro, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," RFC 3135, Jun. 2001. [Online]. Available: https://rfc-editor.org/rfc/rfc3135.txt

[8] J. Iyengar and I. Swett, "QUIC Loss Detection and Congestion Control," RFC 9002, May 2021. [Online]. Available: https://rfc-editor.org/rfc/rfc9002.txt

[9] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," RFC 9000, May 2021. [Online]. Available: https://rfc-editor.org/rfc/rfc9000.txt

[10] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, ser. SIGCOMM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 183–196. [Online]. Available: https://doi.org/10.1145/3098822.3098842

[11] OpenVPN Inc., "Openvpn." [Online]. Available: https://openvpn.net/

[12] J. Pavur, D. Moser, M. Strohmeier, V. Lenders, and I. Martinovic, "A tale of sea and sky on the security of maritime vsat communications," in *2020 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, may 2020, pp. 1384–1400. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SP40000.2020.00056

[13] J. Pavur, M. Strohmeier, V. Lenders, and I. Martinovic, "Qpep: An actionable approach to secure and performant broadband from geostationary orbit." Internet Society, 2021.

[14] L. Thomas, E. Dubois, N. Kuhn, and E. Lochin, "Google quic performance over a public satcom access," *International Journal of Satellite Communications and Networking*, vol. 37, no. 6, pp. 601–611, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sat.1301

[15] M. Thomson and S. Turner, "Using TLS to Secure QUIC," RFC 9001, May 2021. [Online]. Available: https://rfc-editor.org/rfc/rfc9001.txt

[16] T. Wei, W. Feng, Y. Chen, C.-X. Wang, N. Ge, and J. Lu, "Hybrid satellite-terrestrial communication networks for the maritime internet of things: Key technologies, opportunities, and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 8910–8934, 2021.

[17] Wikipedia, "Ka band." [Online]. Available: https://en.wikipedia.org/wiki/Ka_band

[18] T. Wood, "Who owns our orbit: Just how many satellites are there in space?" [Online]. Available: https://www.weforum.org/agenda/2020/10/visualizing-easrth-satellites-sapce-spacex