2019 International Conference on Networked Systems (NetSys)

Satellite Internet Performance Measurements

Jörg Deutschmann, Kai-Steffen Hielscher, and Reinhard German
Computer Networks and Communication Systems
Friedrich-Alexander University Erlangen-Nürnberg (FAU)
Erlangen, Germany
{joerg.deutschmann, kai-steffen.hielscher, reinhard.german}@fau.de

***Abstract*—Satellite Internet with geostationary satellites is one way to provide Internet access all over the world. In Europe, there are three major satellite operators. In this paper, their performance is compared with respect to one-way delays, bulk data transfers, and website download times. When one-way delays are measured with UDP packets, the forward link generally shows lower delays than the return link. The bulk data transfers reveal how close actual data rates get to the advertised link rates. Regarding website download times, the web protocols HTTP/1.1, HTTP/2, and QUIC are considered. The results of our TCP measurements show that satellite Internet heavily relies on Performance Enhancement Proxies (PEPs). TCP connections over VPN tunnels cannot benefit from PEPs and therefore result in poor performance. QUIC can perform better than TCP tunneled in VPNs, but performs worse than TCP connections optimized by PEPs. One operator generally shows more stable performance regarding delays, data rates and page load times.**

***Index Terms*—Satellite Internet, Performance Measurements, UDP, TCP, HTTP, QUIC**

I. Introduction

Internet access via satellite covers large areas with high data rates and is especially useful for users that have no access to terrestrial broadband Internet (e.g., fiber, DSL, cable or LTE). Given that Internet browsing is today's dominant traffic, the performance of satellite Internet as perceived by end users is crucial. Internet access using geostationary satellites suffers from large propagation delays, which results in poor performance when TCP is used. Performance Enhancement Proxies (PEPs) [1] are middleboxes used to overcome the problem of large link delays and are deployed for commercial satellite Internet access. PEPs basically split TCP flows transparently in two separate flows, one between user and local PEP (satellite modem), and another one between remote PEP (satellite operator network) and the Internet. The protocol between local PEP and remote PEP is chosen by the operator and optimized for large delays (e.g., TCP Hybla [2] or proprietary protocols).

In this paper, simple yet meaningful performance measurements for three different satellite operators are described. The operators are Avanti, Astra, and Tooway. The systems are black box tested without having access to internal system information of the operators. The obtained data can also be used for input modeling for a future simulation framework, e.g., for multipath communication combining satellite and terrestrial Internet access as described in [3]. We consider three different performance measurements:

- One-way delays: These are important for delay-sensitive applications and commonly use UDP as a transport protocol.
- Bulk-data transfers: Transferring large chunks of data reflects the download of large files over TCP connections.
- Website download times: Web protocols further influence the performance. Most common are HTTP/1.1 (with or without TLS1.2) and HTTP/2 (with TLS1.2), both based on TCP. Recently Quick UDP Internet Connections (QUIC), a reliable transport protocol with encrypted transport layer headers, is deployed on a large scale by Google and currently being standardized in the IETF [4].

Disclaimer: The presented measurements and evaluations are snapshots of the performance of current satellite networks. As network operations are subject to continuous change, investments in hardware or software may alter the outcome of the experiments described in this work. Therefore we do not refer to the operators by their name but simply by Operator A, Operator B and Operator C.

This paper is structured as follows: Section II reviews related work, and Section III compares the functional features of satellite Internet access provided by the three operators. The measurement setups are described in Section IV and evaluated in Section V. Finally, Section VI concludes this paper.

II. Related Work

There are several studies of web performance over satellite links comparing different HTTP variants [5]–[7], but none of these compares the performance of different satellite operators. In [5] the authors even noticed that the used satellite system has shown great variations in RTT, which caused some evaluation troubles. There is awareness that PEPs cannot optimize encrypted connections [8]–[11]. In [11] the performance of QUIC and HTTP/2 is compared, but only for one operator and one QUIC implementation.

III. Functional Comparison

In order to provide a fair comparison, we have chosen similar tariffs for all providers. The advertised upload throughput ranges from 2 Mbit/s to 6 Mbit/s, the advertised download throughput ranges from 15 Mbit/s to 30 Mbit/s. More details will be given in the evaluation section. All tariffs have Fair

This research has been funded by the Federal Ministry of Economics and Technology of Germany in the project Transparent Multichannel IPv6.

978-1-7281-0568-0/19/$31.00 ©2019 IEEE

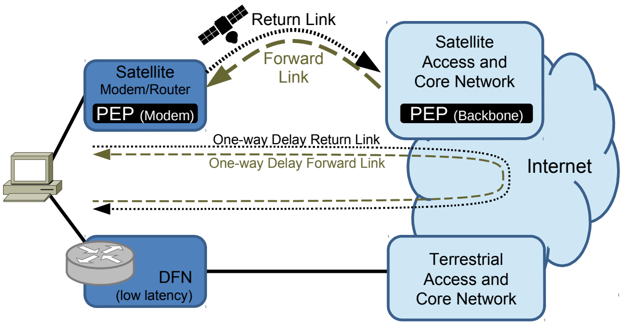

Fig. 1. Setup

Use Policies (FUPs) which throttle the data rates after a certain amount of data has been transferred. We only took measurements without throttled links. The local PEP of Operator C immediately sends ACKs to the initial SYN packet without checking if the destination is reachable. This saves one RTT, but is a rather harsh violation of the end-to-end principle. In case the destination in the Internet is not reachable, the connection is aborted by setting the RST flag. Operator C was also observed to use an HTTP accelerator, which looks into unencrypted HTTP/1.1 GET requests, pre-fetches the content from the server and provides it at the user modem. Thereby the time from HTTP request to response is significantly reduced. However, in our experiments only sometimes and only a few of the objects requested via HTTP/1.1 were pre-fetched. Two of the operators provide public IPv4 addresses for the user, but only one of them allows access from the Internet (the other one blocks ports with a firewall). One operator uses carrier-grade NAT. However, with more expensive professional tariffs, all operators have publicly accessible IPv4 addresses in their portfolio. None of the operators supports IPv6 at the time of writing.

IV. Measurement Setup

Modems and antennas were provided by the satellite operators. Tooway uses ViaSat hardware, Astra and Avanti both use Gilat hardware. The satellite dishes are mounted on the top of the Wolfgang-Händler building in Erlangen, Germany (coordinates 49.57374N, 11.02706E). Please note that satellite operators cover different areas with spot beams or cluster codes, similarly to cellular networks. The performance may depend on the equipment location.

All setups were run on recent hardware (Intel Core i5-3470S 2.9 GHz and Intel PRO/1000 Gigabit Ethernet controllers). The operating system was Ubuntu 18.04 LTS (Kernel 4.15.0-34-generic; TCP settings: CUBIC congestion control, SACK and window scaling enabled, IW 10, no ECN).

One test cycle tests one after another: forward/return link delays, forward/return link bulk data transfers and every web protocol. This test cycle was run 100 times for every operator, 24 hours a day over the period of one week. All measurements were analyzed regarding the time and weather conditions, but no significant dependencies were observable over that timeframe.

A. One-way delays

For measuring one-way delays, having the same physical host for sending and receiving packets simplifies the setup because clock synchronization among different hosts is not necessary. We have a host which has one dedicated ethernet interface for each satellite operator and another ethernet interface connected to the DFN gigabit backbone network [12] as illustrated in Fig. 1. The sending and receiving applications are separated via Linux network namespaces. Timestamps are extracted from `tcpdump` traces running at the relevant interfaces. In order to avoid delays by ARP request or other link setup times, we sent ten UDP packets to the destination host and afterwards another ten UDP packets for measuring the delays.

B. Bulk data transfers

A simple form of throughput and goodput measurement can be achieved by bulk data transfers. We only consider the transport protocol TCP and focus on the goodput results. Due to the limited accessibility from the Internet, connections must be initiated from the local side. Therefore, an object with the size of 10 Mbyte (10^9 Byte) is transferred via a socket based client/server program transmitting random data. The duration of a flow is then measured from the first SYN to the last FIN/ACK at the local side. We note that the flow duration on the backbone side is different due to the PEP's Split TCP, but the differences are neglible for large data transfers.

During the presented measurement timeframe, Operator B limited the throughput of a single flow in the forward link to 10 Mbit/s; the motivation for this is unknown. As we still want to examine the maximum throughput capabilities of the Internet access link, we start three TCP flows simultaneously. We observed a very good TCP fairness (the three flows finished almost simultaneously), therefore the overall goodput can be calculated as the sum of the goodput of all flows. For Operator A and C, where the flow limitation does not apply, it does not matter if the goodput is measured over a single flow or multiple parallel flows.

The physical setup is identical to Fig. 1. We also tested the performance when the server is located at a different, well-connected location (dedicated root server provided by Hetzner Online GmbH, Nuremberg) but did not obtain different results.

C. Website transfers

For web access evaluations, we designed two typical static websites as used in nowadays Internet. Both websites consist of 34 objects. The first website contains objects from 4 kB to 300 kB resulting in a total size of 1.4 MB, the second website hosts objects of sizes between 4 kB and 4 MB resulting in a total size of 10 MB. Test results with a third website with a total size of 2.9 MB are not included in this paper because the results do not provide additional insights. We note that the outcomes of tests on that layer depend on a lot of parameters on lower layers, but as mentioned in the introduction, the purpose of these tests is the performance perceived by users with a configuration setup with default settings.

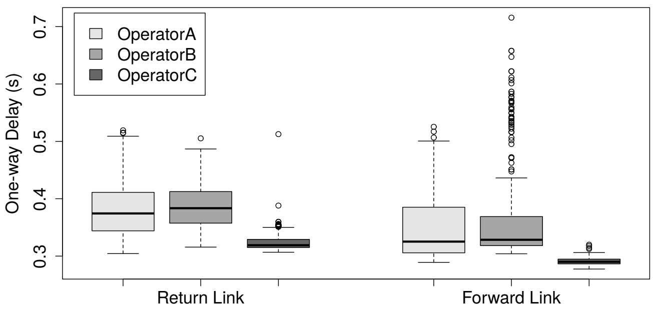

Fig. 2. One-way delays (Setup in Sec. IV-A)

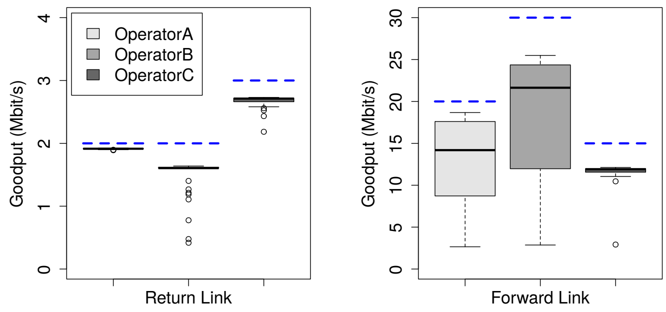

Fig. 3. Bulk data transfers with TCP (Setup in Sec. IV-B)

The pysical setup is again similar to Fig. 1, where the client connects via the satellite modem and the server is connected via the DFN backbone network. Similar to the bulk data transfers, running the webserver on a dedicated root server yields the same results.

The HTTP server was Apache 2.4.29 with `mod_http2` module running on the above-mentioned computer and operating system. The websites were accessed in two ways: First, using direct TCP connections which allows the operation of PEPs. Second, via an OpenVPN (version 2.4.4) UDP tunnel which prevents PEPs from doing split TCP and flow manipulation. The tunnel endpoints were set up in the corresponding network namespace on the client and server interface, respectively. Regarding QUIC, two different server implementations were tested:

- Chromium QUIC [13], git commit `19eaae6`, Sept. 2018
- quic-go [14], git commit `ffdfa1f`, August 2018

Both use Google QUIC version 43 (Q043) and CUBIC congestion control (quic-go's CUBIC implementation is actually based on the Chromium code). Finally, for both HTTP and QUIC, we deleted cache contents before every test run, so that every website access is a first-time access without session resumption advantages. All objects are served from a single server (no Domain Sharding). HTTP/2 Server Push or Stream Prioritization were not enabled.

The website download times are measured at the client using the Performance Timing API [15]. The time from `connectStart` to `domComplete` represents the page load time (PLT). We used the Google Chrome browser 69.0.3497.100, automated with Selenium Chromedriver 2.41.578700.

V. Performance Evaluation

This section describes the results obtained with the setups described before. All boxplots show the median, the first/third quartile, and outliers for values less/more than 1.5 times the interquartile range.

A. One-way delays with UDP packets

The results for one-way delays with UDP packets are plotted in Fig. 2. There are two major findings. First, the delays in the forward link are lower than the delays in the return link for all three operators, which is probably caused by the channel access mechanisms. Second, Operator A und B suffer from larger variations than Operator C. This suggests that Operator C has implemented better Quality of Service (QoS) strategies.

B. Bulk data transfers

The results of the bulk data transfers are shown in Fig. 3. In the return link, Operator A and B specify a maximum throughput of 2 Mbit/s and Operator C specifies a maximum throughput of 3 Mbit/s. Considering the protocol overhead by IPv4 and TCP, all operators achieve their advertised return link throughputs. In the forward link, the maximum throughputs are as follows: Operator A 20 Mbit/s, Operator B 30 Mbit/s, and Operator C 15 Mbit/s. Operator C reaches the advertised throughput most reliably.

C. Website transfers

The results of the website transfers are shown in Fig. 4. The findings related to HTTP are:

- HTTPS with TLS1.2 requires two more RTTs for initial connection setup compared to unencrypted HTTP/1.1, which has a significant impact for small websites.
- HTTP over VPN always performs worse than plain TCP because the PEP cannot optimize TCP flows. Yet with HTTP over VPN, the browser opening multiple parallel TCP connections (as it is done with HTTP(S)/1.1) seems to be better than HTTP/2.
- Operator B limited a single flow to 10 MBit/s, which has a negative impact for HTTP/2 connections, especially for large websites.

Regarding QUIC, the findings are:

- QUIC uses encryted transport layer headers which are not alterable by PEPs. Still, QUIC is able to perform better than HTTP over UDP tunnels, especially when Operator C is used. Larger send and receive buffers or more SACK ranges might be the reason.
- Because QUIC runs atop UDP, the larger PLT variations experienced with Operator A and B might be related to the higher UDP jitter described in Section V-A.
- Performance tests with QUIC webservers should be taken with a grain of salt, because QUIC is currently work in progress. Chromium QUIC [13] explicitely states that

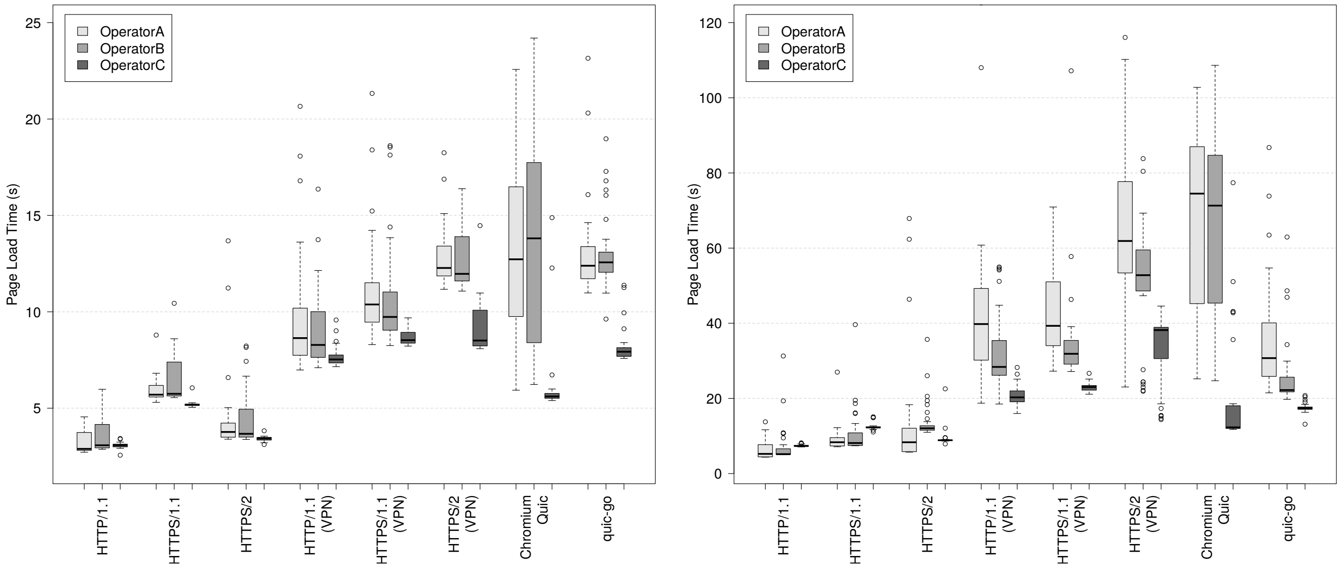

Fig. 4. Download of a small-sized website (left) and a large-sized website (right)

it "is [not] performant at scale". Although the system utilization (CPU, Memory, ...) was never critical for both implementations, Quic-go often performed better than Chromium Quic.

In summary, the HTTP variants perform as expected. HTTP/TCP sent over encrypted tunnels as well as QUIC suffers from the non-applicability of PEPs. However, our QUIC measurements can only give a first impression. Further investigations are needed to fully evaluate QUIC performance in satellite networks. Operator C provides most stable page load times which fits the previous results.

VI. Conclusion

In this work, we measured and evaluated the performance of satellite Internet as perceived by end users. Three operators were compared considering different setups and parameters. Generally, one-way delays were lower in the forward link than in the return link. Operator A und B suffered from large variations, which might be caused by buffer bloat and should be examined more thoroughly in the future. For bulk data transfers, the goodput results were in the expected range. Operator C showed more stable delays and data rates than Operator A and B. For web protocols, it was identified that limiting the data rate of a TCP flow (Operator B) is detrimental for HTTP/2. It was shown that PEPs are an important component of satellite Internet access. QUIC with default configuration settings as well as TCP flows tunneled in VPNs cannot benefit from PEPs, which leads to higher page load times.

References

[1] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations," RFC 3135, June 2001.

[2] C. Caini and R. Firrincieli, "TCP Hybla: a TCP enhancement for heterogeneous networks," *International Journal of Satellite Communications and Networking*, vol. 22, no. 5, pp. 547–566.

[3] J. Deutschmann, K. Hielscher, T. Keil, and R. German, "Multipath communication over terrestrial and satellite links," in *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pp. 119–121, June 2018.

[4] "IETF QUIC Working Group." https://datatracker.ietf.org/wg/quic/about.

[5] R. Secchi, A. C. Mohideen, and G. Fairhurst, "Performance Analysis of Next Generation Web Access via Satellite," *International Journal of Satellite Communications and Networking*, vol. 36, pp. 29–43, December 2016.

[6] A. Cardaci, L. Caviglione, E. Ferro, and A. Gotta, "Using SPDY to improve Web 2.0 over satellite links," *International Journal of Satellite Communications and Networking*, vol. 35, pp. 307–321, June 2016.

[7] L. Caviglione, N. Celandroni, M. Collina, H. Cruickshank, G. Fairhurst, E. Ferro, A. Gotta, M. Luglio, C. Roseti, A. Abdel Salam, R. Secchi, Z. Sun, and A. Vanelli Coralli, "A deep analysis on future web technologies and protocols over broadband GEO satellite networks," *International Journal of Satellite Communications and Networking*, vol. 33, pp. 451–472, July 2015.

[8] R. McKinney and R. Lambert, "VPN over Satellite, A comparison of Approaches." http://www.vsat-systems.com/PDFs/VPN-over-satellite-large.pdf.

[9] "QUIC Behaviour over satellite networks." https://groups.google.com/a/chromium.org/d/msg/proto-quic/k_tPUf6JkkA/Lny7OgnPxrwJ, January 2015.

[10] J. Border and C.-J. Su, "QUIC over Satellite." https://trac.tools.ietf.org/bof/trac/wiki/BofIETF101, March 2018. HotRFC Lightning Talks, IETF101.

[11] L. Thomas, E. Dubois, N. Kuhn, and E. Lochin, "QUIC and SATCOM," *CoRR*, vol. abs/1810.04970, 2018.

[12] H.-M. Adler, P. Eitner, K. Ullmann, H. Waibel, and M. Wilhelm, "X-WiN: The Network Infrastructure of DFN." https://www.dfn.de/fileadmin/1Dienstleistungen/XWIN/X-WiN-Broschuere_englisch.pdf.

[13] "The Chromium Projects: QUIC, a multiplexed stream transport over UDP." https://www.chromium.org/quic.

[14] "A QUIC implementation in pure go." https://github.com/lucas-clemente/quic-go.

[15] World Wide Web Consortium, "Navigation Timing." www.w3.org/TR/navigation-timing/. W3C Recommendation 17 December 2012.

All Internet links were last accessed on 2018-10-01.