

Throughput Performance Measurement of the MPT-GRE Multipath Technology in Emulated WAN Environment*

Szabolcs Szilágyi, Imre Bordán

University of Debrecen, Faculty of Informatics
szilagyi.szabolcs@inf.unideb.hu
bordanimre@gmail.com

*Proceedings of the 1st Conference on Information Technology and Data Science
Debrecen, Hungary, November 6–8, 2020
published at <http://ceur-ws.org>*

Abstract

Internet architecture enables only a single data path between two communication endpoints within a communication session. On the other hand, decent communication equipments (laptops, tablets, phones) are equipped at the factory with several network interfaces (Ethernet, Wi-Fi, 3G, 4G). It does not worth not using these hardware-given possibilities, which could increase the performance of the communication between two devices, using two or more communication paths. In this paper we presented a possible solution by implementing the MPT-GRE software library. This software was developed under Linux and is based on a totally new architecture, in comparison with the classical TCP/IP model, providing an easy-to-use extension of the current TCP protocol stack (see Figure 1). In our previous papers we investigated its performance in various laboratory measurement environments. In this paper we tried to do it in a much more realistic environment, using the Dummynet WAN emulation software. The measurement results confirmed that the MPT multipath solution could efficiently aggregate the performance of physical connections in the emulated WAN environment as well.

Copyright © 2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

*This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

Keywords: MPT-GRE, multipath communication, DummyNet, throughput, WAN Emulator

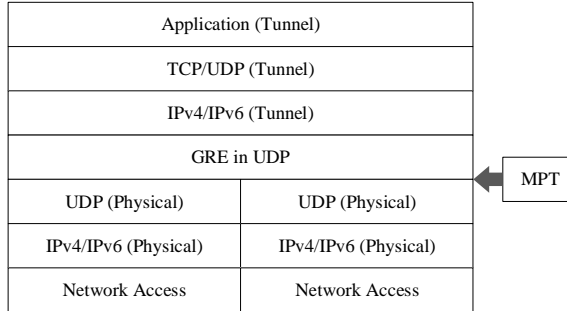


Figure 1. Layered architecture of the MPT-GRE.

1. Introduction

Multipath communication technologies are one of the hot research topics nowadays. What better proof of this than Apple and Cisco integrating MPTCP¹, considered as the flagship of multipath technologies, into their operating systems [8]. With the help of multipath communication, we can increase throughput while also employing redundant data paths.

In our earlier publications (see e.g. [1–3, 5–9]) we have presented a multipath communication technology (MPT-GRE² [4]) developed by our research group, which we have built on the standardized GRE-in-UDP tunneling technology³. We have examined its effectiveness with the help of numerous scenarios in our test environment, comparing results with MPTCP as a reference. All of the scenarios have showed that our MPT-GRE solution is capable of efficient path-aggregation both in Fast Ethernet and Gigabit Ethernet IPv4/IPv6 environments.

The testing environments we have used previously (see e.g. Figure 3) can be considered ideal in the sense that they did not contain any network environment parameters that could negatively affect network performance (e.g. delay, jitter, packet-loss). For this reason, we find it important to further examine the effectiveness of MPT-GRE in a more realistic environment (see e.g. Figure 2).

When wants to test a newly developed networking software in a realistic environment, we practically have three possibilities:

- internet

¹The MPTCP Project official website: <https://www.multipath-tcp.org/>

²The MPT-GRE Project official website: <https://irh.inf.unideb.hu/~szilagyi/index.php/en/mpt/>

³GRE-in-UDP Encapsulation standard: <https://tools.ietf.org/html/rfc8086>

- network simulation
- network emulation

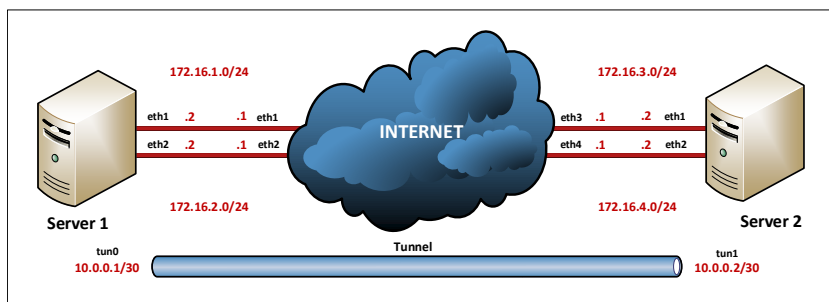


Figure 2. A real network testbed example.

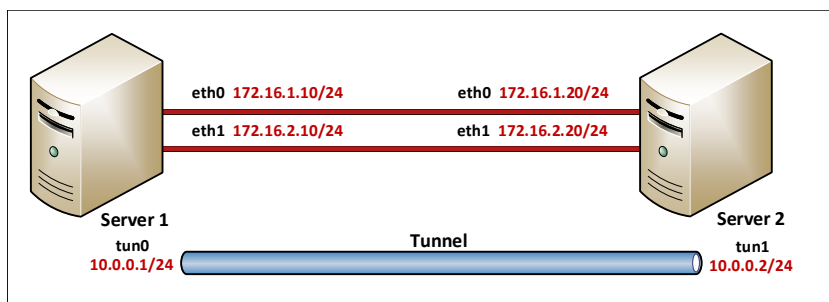


Figure 3. Our previous “ideal” testbed.

Using the internet can be given in the case of single-path communication. However, being able to test multipath systems, the presence of dual-home technology is essential, i.e. we need to have multiple ISP connections available.

Network simulation aims to replicate the key parameters of desired network environments with the help of mathematical models, with greater or lesser success.

The essence of network emulation is to replicate real network behaviors. Two main types exist:

- hardware realization (see e.g. Figure 4)
- software implementation

The first contains advanced technological solutions, but it is, in turn, quite a costly method. The latter is not always capable of providing a reliable and precise

test environment, but it is cost-effective. Some examples of network emulator software: Dummy Cloud⁴, Dummynet⁵, NETEM⁶, NIST Net⁷, SoftPerfect Connection Emulator⁸, WANem⁹.

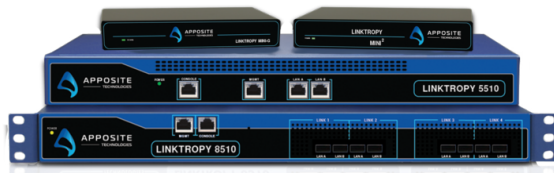


Figure 4. Hardware-based network emulators.
(Source: <https://www.apposite-tech.com>)

Given that a hardware implementation of a WAN emulator sufficient for our goals would be around 6000 EUR + VAT¹⁰, after having reviewed the software solutions, our choice was Dummynet.

The Dummynet WAN emulator was developed in 2010 at the University of Pisa, and later got integrated into the FreeBSD operating system¹¹. It provides a suitable framework for testing multipath solutions, enabling the setup of packet-delay, jitter and packet-loss network parameters [10]. It also has good documentation, including numerous code examples¹².

2. Measurement Environment

We created a dual-path Fast Ethernet IPv4 WAN emulated measurement environment (see Figure 5). We downloaded a 1 GB file from the fileserver on the left onto the server on the right. Network parameters were controlled on the intermediate server that had Dummynet installed on the kernel level.

⁴Dummy Cloud official website: <http://www.dummycloud.com/>

⁵Dummynet Project official website: <http://info.iet.unipi.it/~luigi/dummynet/>

⁶NetEm's manual page: <https://man7.org/linux/man-pages/man8/tc-netem.8.html>

⁷NIST Net home page: <https://www-x.antd.nist.gov/nistnet/>

⁸SoftPerfect Connection Emulator:

<https://www.softperfect.com/products/connectionemulator/>

⁹WANem official web page: <http://wanem.sourceforge.net/>

¹⁰Linktropy Mini-G's price at November 2020: <https://www.digital-hands.eu/products/apposite/linktropy-mini-g/>

¹¹FreeBSD Manual Pages: <https://www.freebsd.org/cgi/man.cgi?dummynet>

¹²Using Dummynet in FreeBSD: http://noahdavids.org/self_published/using_dummynet.html

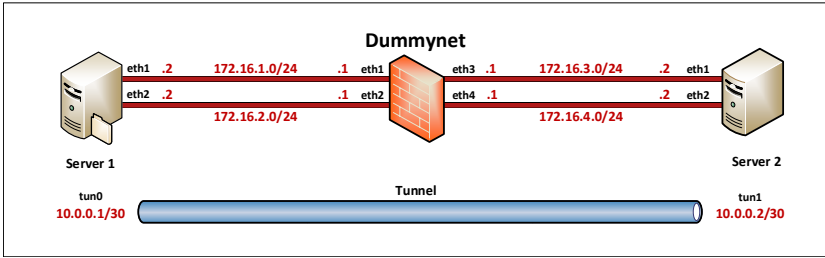


Figure 5. Our new measurement testbed with Dummynet.

All three machines were running Linux Ubuntu operating system. We examined the effect of packet-delay, jitter and packet-loss on file download speed, download time, and CPU performance. Bash and Python scripts – available on our website¹³ – were used to automate the measurement process. We repeated each series of measurements ten times.

3. Measurement Results

First, we checked how packet-delay affected download speed (see Figure 6).

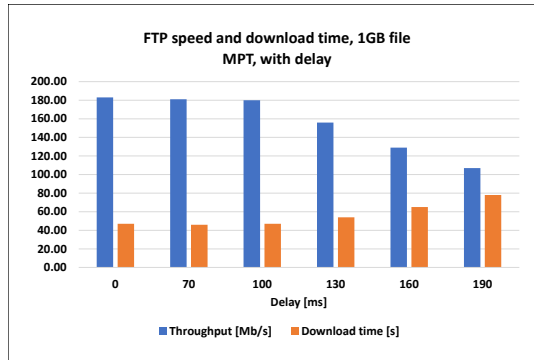


Figure 6. The effect of the delay on the FTP throughput and download time.

We gradually increased the delay values on a scale of 0-190 ms using Dummynet. Everything proved to be stable until 100 ms. Above 100 ms, we experienced a continuous decrease in file download speeds. Using the 190 ms delay value, the download speed decreased to 107 Mb/s, while download time increased from 47 seconds to 78 seconds.

¹³Our test scripts can be downloaded from: <https://nas01.inf.unideb.hu/share.cgi?ssid=03Csnis>

A similar effect could be experienced in the case of increasing jitter values (see Figure 7). With a 160-190 ms delay fluctuation, the download speed practically decreased by half, while the download time doubled.

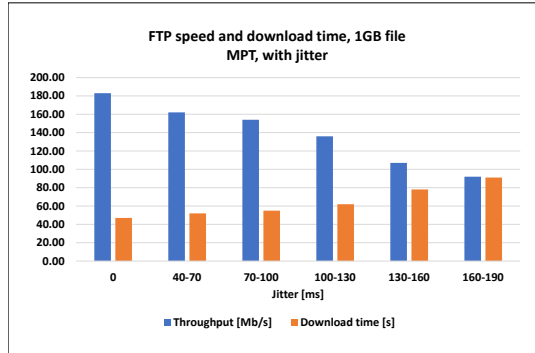


Figure 7. The effect of the jitter on the FTP throughput and download time.

Applying even a minimal data-loss rate (1 ‰), we witnessed a drastic performance decline (see Figure 8). The download speed fall to a quarter, while the download time quadrupled. Therefore, we did not experiment with further data-loss rate values.

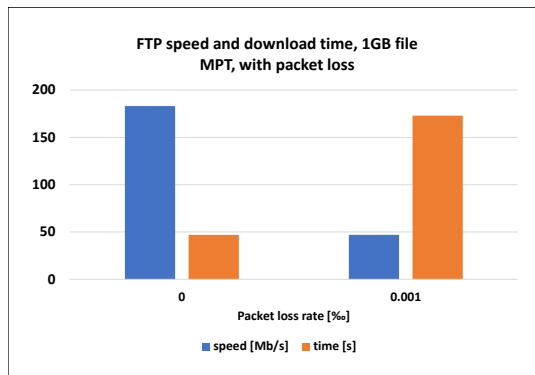


Figure 8. The effect of the packet loss on the FTP throughput and download time.

Regarding the effect of packet-delay on CPU performance, we did not experience significant fluctuation (see Figure 9). CPU utilization hovered between 15-22% in every case.

Introducing jitter however, had noticeable effects on CPU utilization (see Figure 10). With higher jitter values, we experienced a drop in CPU utilization.

While examining CPU loads, the effect of packet-loss also proved to be drastic (see Figure 11). Using a data-loss rate of 1 ‰, utilization dropped from 17 to 5.7 percent.

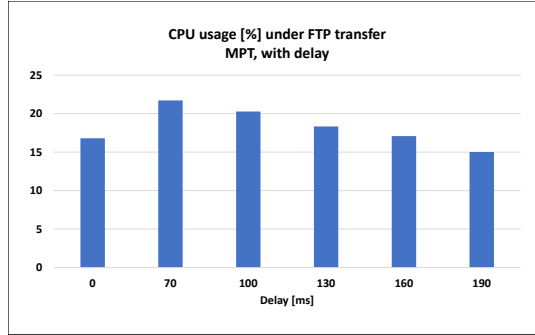


Figure 9. The effect of the delay on the CPU usage.

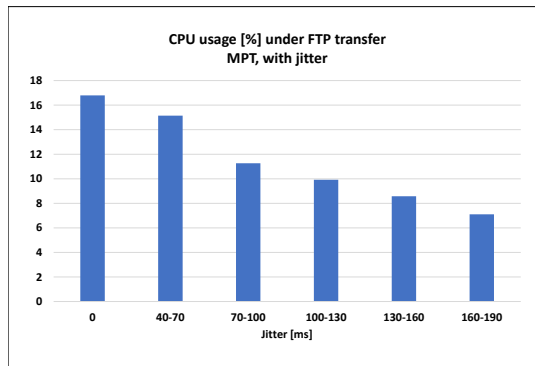


Figure 10. The effect of the jitter on the CPU usage.

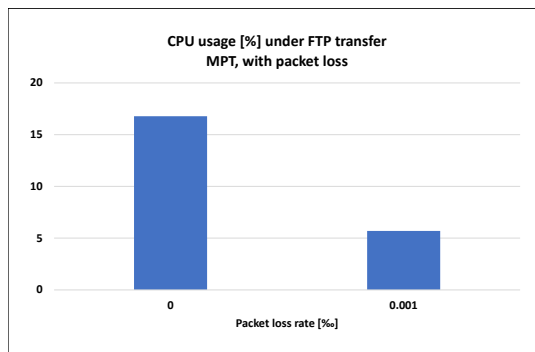


Figure 11. The effect of the packet loss on the CPU usage.

We also carried out further measurements, mixing the parameters of the different paths. E.g. using only delay on one path, while using only jitter on the other. These scenarios brought similar results as well.

4. Conclusions

In our current paper, we extended the performance-analysis of our own multipath solution, MPT-GRE, using an emulated WAN environment. We examined the effect of different network parameters, like e.g. packet-delay, jitter and data-loss rate on file download speed, download time, and CPU utilization. The worst performance we experienced was with the application of the 1 % packet-loss rate. Among our future goals we are planning to extend our measurements to Gigabit Ethernet IPv4/IPv6 environments, and we would also like to get hands-on experience with the capabilities offered by hardware WAN emulators.

References

- [1] B. ALMÁSI, G. LENCSE, S. SZILÁGYI: *Investigating the Multipath Extension of the GRE in UDP Technology*, Computer Communications 103 (2017), pp. 29–38, DOI: <https://doi.org/10.1016/j.comcom.2017.02.002>.
- [2] B. ALMÁSI, S. SZILÁGYI: *Investigating the Throughput Performance of the MPT Multipath Communication Library in IPv4 and IPv6*, International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems 5.1 (2016), pp. 53–60, DOI: <https://doi.org/10.11601/ijates.v5i1.148>.
- [3] B. ALMÁSI, S. SZILÁGYI: *Throughput Performance Analysis of the Multipath Communication Library MPT*, in: TSP 2013 – The 36th International Conference on Telecommunications and Signal Processing, Rome, Italy, 2013, pp. 86–90, DOI: <https://doi.org/10.1109/TSP.2013.6613897>.
- [4] G. LENCSE, S. SZILÁGYI, F. FEJES, M. GEORGESCU: *MPT Network Layer Multipath Library - Internet Draft v6*, 2020, URL: <https://tools.ietf.org/html/draft-lencse-tsvwg-mpt-06> (visited on 10/10/2020).
- [5] S. SZILÁGYI, I. BORDÁN: *The Effects of Different Congestion Control Algorithms over Multipath Fast Ethernet IPv4/IPv6 Environments*, in: Proceedings of the 11th International Conference on Applied Informatics (ICAI 2020), vol. 2650, Eger, Hungary, 2020, pp. 341–349, URL: <http://ceur-ws.org/Vol-2650/paper35.pdf>.
- [6] S. SZILÁGYI, I. BORDÁN, L. HARANGI, B. KISS: *MPT-GRE: A Novel Multipath Communication Technology for the Cloud*, in: 9th IEEE International Conference on Cognitive Informatics : CogInfoCom 2018 Proceedings, Piscataway (NJ), USA, 2018, pp. 81–86, DOI: <https://doi.org/10.1109/CogInfoCom.2018.8639941>.
- [7] S. SZILÁGYI, I. BORDÁN, L. HARANGI, B. KISS: *Throughput Performance Analysis of the Multipath Communication Technologies for the Cloud*, Journal of Electrical and Electronics Engineering 12.2 (2019), pp. 69–72.
- [8] S. SZILÁGYI, I. BORDÁN, L. HARANGI, B. KISS: *Throughput Performance Comparison of MPT-GRE and MPTCP in the Gigabit Ethernet IPv4/IPv6 Environment*, Journal of Electrical and Electronics Engineering 12.1 (2019), pp. 57–60.

- [9] S. SZILÁGYI, F. FEJES, R. KATONA: *Throughput Performance Comparison of MPT-GRE and MPTCP in the Fast Ethernet IPv4/IPv6 Environment*, Journal of Telecommunications and Information Technology 3.2 (2018), pp. 53–59,
DOI: <https://doi.org/10.26636/jtit.2018.122817>.
- [10] D. T. TIEU, B. C. NGUYEN, L. M. LE, Q. M. PHAM, Q. T. NGUYEN, D. T. VO: *Automatic test framework for video streaming quality assessment*, in: 2015 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), IEEE, 2015, pp. 214–218,
DOI: <https://doi.org/10.1109/NICS.2015.7302193>.