Throughput Performance Comparison of MPT-GRE and MPTCP in the Gigabit Ethernet IPv4/IPv6 Environment

SZILÁGYI Szabolcs, BORDÁN Imre, HARANGI Lajos, KISS Benjámin

University of Debrecen, Hungary,
Department of IT Systems and Networks, Faculty of Informatics,
26 Kassai Way, 4028 Debrecen, Hungary, E-Mail: szilagyi.szabolcs@inf.unideb.hu

Abstract *– Multipath communication technologies is one of the current research fields for infocommunications. No better proof for this than having the specification of the MPTCP protocol, considered the field's mothership, integrated into the operating systems of multiple corporations (e.g. Apple, Cisco) shorty after being released. The MPTCP multipath solution is practically the extension of the classic TCP protocol with the application of TCP-subflows. Besides its numerous advantages, this multipath protocol has a big drawback as well, namely that it supports data transfer over TCP only. However, for transmitting multimedia traffic, the UDP protocol is the practical choice. Stemming from this idea, the Networks Research Group at the Faculty of Informatics, University of Debrecen, has started development of a multipath communication technology (MPT-GRE) that aims to fulfil this need, providing an alternate solution besides MPTCP. In this paper we provide a comparison based on performance measurements between the MPTCP as a reference and the MPT solution developed by our research group, illustrated via numerous scenarios implemented over quad-path Gigabit Ethernet and IPv4/IPv6 connections.*

Keywords: *multipath communication; MPT-GRE; MPTCP; performance analysis; system throughput; tunneling.*

I. INTRODUCTION

The networks we use today, wired or wireless, are predominantly based on the classic TCP/IP model. This means a single socket-pair is used to identify the physical interfaces taking part in a communication session using the IP address and port number associations. In case the communication session is interrupted for any reason, it has to be re-established in order to continue the data transfer. On the other hand, the devices used for communication nowadays can have multiple factory-integrated network interfaces (e.g. Ethernet, Wi-Fi, 4G), which could be used for parallel data transfer within a single communication session. This provides the opportunity to increase the aggregated data transfer rates, ensure connection redundancy [1], and last but not least, provide an improved user experience.

The topic at hand has inspired the development of numerous solutions with multipath communication support. Out of all of them, the MPTCP (MultiPath Transmission Control Protocol [2]) became the most well-known representative. Its base specification is detailed in RFC 6824, and further development is also under way. Several renowned manufacturers have considered it good practice to integrate the MPTCP protocol into their own operating systems. For example, MPTCP support is available in numerous Cisco devices since 2013. Apple supports the use of MPTCP by default from the OS X Yosemite version upwards. Then later, in 2017, Apple decided to provide MPTCP functionality for every application with iOS 11. These facts also greatly support the significance and modernity of the multipath communication technologies research field.

The areas where multipath solutions can be utilized are the following:

- Cloud Computing – Link aggregation in data centers.
- Fog Computing – Internet of Things.
- Wi-Fi, 3G, 4G environments – Solution for the roaming problem.
- Cognitive Infocommunications – Telemedicine.
- Time-critical applications – Multimedia.

The contents of this paper are organized as follows: in Chapter 2 we take the MPTCP as reference to briefly compare its operating principles to that of the multipath software architecture developed by us, called MPT [3]. Chapter 3 introduces the environments used for carrying out the different types of measurements, while in Chapter 4 we evaluate the measurement results. Lastly, in Chapter 5 we draw our conclusions and highlight further development opportunities.

II. COMPARISON OF THE MPTCP AND THE MPT-GRE ARCHITECTURES

The MPTCP relies on subflows to establish a communication session with multiple paths and multiple underlying physical interfaces, which is detailed in the IEEE RFC 6824. The figure below gives a comparison

between the classic TCP/IP architecture and the one used by MPTCP (see Fig. 1).

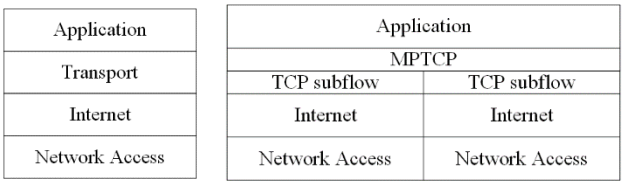

Fig.1. Comparison of the traditional TCP/IP- and the MPTCP layered architecture

Besides the numerous benefits of the MPTCP, a major disadvantage also shows up, namely that it operates in the transport (4.) layer, which means it is unable to support the UDP transfer protocol (it provides communication over TCP only). This drawback can cause problems primarily when dealing with multimedia traffic (audio and/or video transfer), as these applications are time sensitive due to their nature (QoS metrics set the maximum allowed rate of e.g. latency, jitter, and packet loss). When communicating over TCP, these values cannot always be guaranteed towards the multimedia applications.

The realization of this fact gave the starting idea and motivation for developing a new multipath technology that can overcome the stated problem. Fig. 2 shows the software-architecture of the MPT[1] (MultiPath Tunneling) solution developed by our research group, which is built on a completely new underlying concept. Its operation is based on the multipath extension of the GRE-in-UDP standardized tunneling technique specified in RFC 8086, which allows the mapping of a logical (tunnel) interface to physical interfaces in the network layer (3. layer) already, and as such it enables support for both TCP and UDP protocols over either IPv4 or IPv6. A new logical (tunnel) layer has been introduced in the MPT implementation, which hides the operating logic and provides a traditional interface towards the upper layers.

The data units arriving from the application layer get forwarded to the tunnel interface, which has the MPT functionality working below it. This logic is responsible for mapping the traffic to the physical interfaces and re-packing the data units.

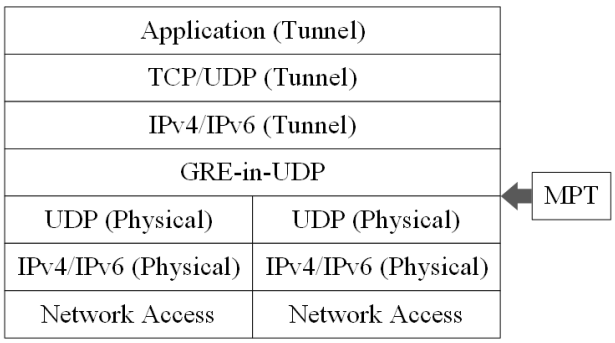

Fig. 2. The MPT-GRE architecture

[1] MPT and MPT-GRE are interchangeable terms

The second figure clearly shows that MPT allows data transfer over either TCP or UDP, and that the IP version used on the logical (tunnel) interface, and the IP version used on the physical interfaces are completely independent (every possible combination is supported: IPv4 over IPv4, IPv4 over IPv6, IPv6 over IPv4, and IPv6 over IPv6). A more detailed description of the operating logic of MPT can be found in [4] and [5].

III. THE MEASUREMENT ENVIRONMENT

We have used two server machines to set up the different measurement scenarios. Fig. 3 shows how the connection between the two endpoints was installed. The system specifications for both computers were as follows:

- Gigabyte Z77-D3H motherboard with Intel Z77 chipset.
- Intel Core i7-3770K 3.50 GHz processor with 4 cores and 8 threads.
- 4 X 4 GB 1600 MHz DDR3 SDRAM.
- Intel PT Quad 1000 Gigabit Ethernet server adapter.
- Ubuntu 16.04 LTS (XenialXerus) 64-bit operating system with 4.4.0-62-generic Linux kernel module.

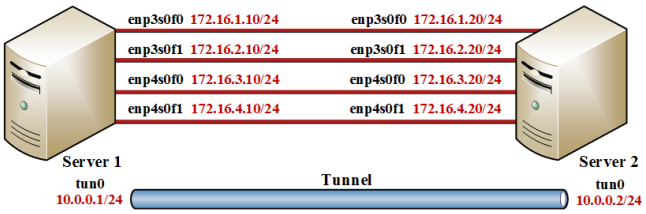

Fig. 3. Four wired paths Gigabit Ethernet laboratory measurement environment

We have connected the servers using four CAT6 STP patch cables, thus providing four independent physical paths. The integrated network interface cards were used for remote management purposes. These integrated NICs were disabled for the duration of the MPTCP-based measurements in order to achieve the most representative measurement results possible.

For the MPTCP-based measurements that served as a reference, we have downloaded the 0.94 stable version from the official website of the MPTCP (see [2]). The kernel module was installed as per the provided official instructions. In order to increase the performance of the MPTCP, we have issued the following configuration setting:

```
sysctl net.mptcp.mptcp_checksum='0'
```

For the MPT measurements, we have used the latest version of the software available from GitHub [6]. We have compiled the code and performed the setup of the configuration files according to the instructions in the user manual. In the beginning we have set out to use the latest version of Ubuntu Server (18.04 LTS), however, for

some reason the MPT was not operating effectively. Unfortunately, we could not uncover the true reason behind this anomaly, so we were forced the use the previous version (16.04 LTS).

We have carried out the following comparison measurements using each multipath solution: *iperf3*-based throughput performance analysis, examining the download speed and time of a 10GB file using FTP, CPU usage measurement while using *iperf3*.

We have used bash and Python scripts to automate the measurements. Each and every measurement was repeated ten times. The measurement results showed a minimal deviation of less than 1% in every case. For both the MPTCP and the MPT-GRE, we have designed the measurement scenarios so that all the possible IP version combinations were included. In practice, this resulted in four scenarios for the MPT (all IPv4, IPv4 over IPv6, IPv6 over IPv4, all IPv6), and two scenarios for the MPTCP (all IPv4, all IPv6) measurements.

IV. MEASUREMENT RESULTS

A. *iperf3-based Measurements*

Firstly, we have limited our test series to have reading/writing of data performed only from memory to memory to avoid the speed limitations imposed by writing to, and reading from a disk. To accomplish this, we have used the *iperf3* networking software with the following parameters:

```
iperf3 -O 1 -c 10.0.0.2 -t 30 -i 1 -f g
```

The results were logged using the *tee* program. As Fig. 4 shows, we have experienced a nearly linear increase in throughput while gradually enabling the network interfaces. Using one path with the MPT-GRE the throughput was 0.9 Gbps, using two paths it was 1.8 Gbps, with three paths we measured 2.7 Gbps, while using all four paths resulted in 3.6 Gbps, which compared to previously published results is unequivocally an improved performance (see e.g. [7]).

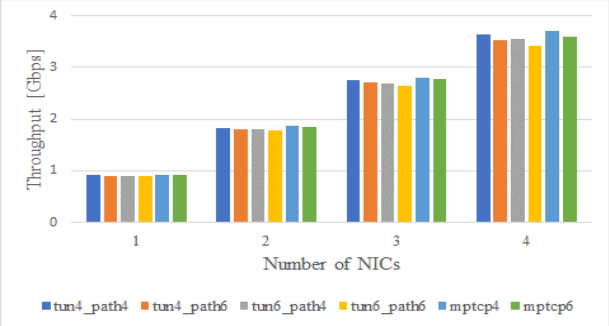

Fig. 4. MPT-MPTCP *iperf3* test comparison

The MPTCP achieved only marginally better results: 0.93 Gbps, 1.86 Gbps, 2.79 Gbps, and 3.71 Gbps respectively. All in all, we can say that the MPTCP managed to perform some 3% better compared to the MPT in both IPv4 and IPv6 environments.

B. *FTP Measurements*

The next round of scenarios involved FTP-based measurements. The machine that can be seen on the left side of Fig. 3 was set up as an FTP server that was used to download a 10GB file to the computer on the right. We have performed the *ramdisk* configuration using the following script:

```
   sudo moun -t tmpfs -o size=11G tmpfs
/var/ftp/pub/
   cp /var/ftp/10GB.zip /var/ftp/pub/
```

The FTP download process itself was automated as follows:

```
   !/bin/bash
   #HOST="[fec0::2]"
   #HOST="10.0.0.2"
   #HOST="172.16.1.2"
   HOST="[fec1:300::2]"
   wget ftp://$HOST/pub/10GB.zip -O /dev/null -
-report-speed=bits 2>&1
```

Fig. 5 shows the performance achieved with four aggregated paths. In the case where we have used IPv4 with MPT on both the tunnel and the physical interfaces, the 10GB file was downloaded with a speed of 3.36 Gbps.

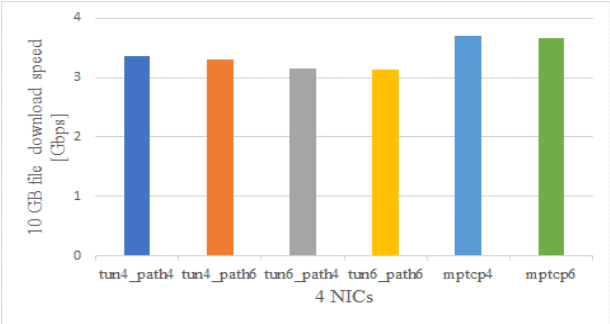

Fig. 5. MPT-MPTCP FTP download speed comparison aggregating 4 interfaces

In the other possible combinations performance has deteriorated slightly with 3.31, 3.15, and 3.13 Gbps respectively. The MPTCP has performed slightly better in this round of tests as well: we have measured a download speed of 3.7 Gbps over IPv4, and 3.66 Gbps over IPv6.

We continued by performing FTP download speed comparisons between the MPT and the MPTCP (Fig. 6-7). First, we have carried out a baseline measurement between the two servers while ensuring that both the MPT and the MPTCP were disabled. This resulted in a download speed of 0.98 Gbps, taking 84 s to complete the download. We can see on Fig. 6 that if we use four paths concurrently instead of just one, the throughput can be quadrupled (1), resulting in the download taking just a quarter of the baseline measurement's duration to complete (2):

$$T(n) = n \cdot T(1)\ [\mathrm{Gb/s}] \qquad (1)$$

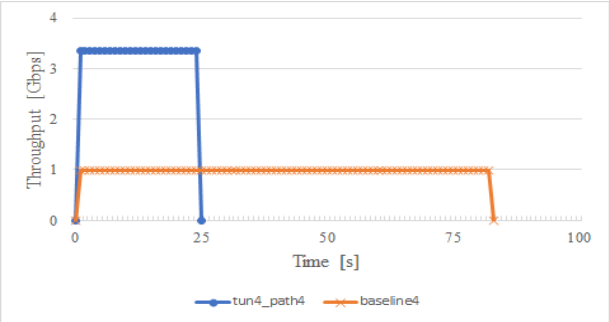

Fig. 6. The MPT-GRE IPv4-IPv4 FTP throughput performance using 4 Gigabit Ethernet interfaces

$$t(n) = \frac{t(0)}{n} \text{ [s]} \quad (2)$$

The same can be said in case of the MPTCP as well (Fig. 7) with the difference being that in this case there is no tunnel interface, but only the four physical interfaces that the MPTCP can perform effective throughput aggregation with. Both of the examined protocols achieved similarly good results in this test series as well.

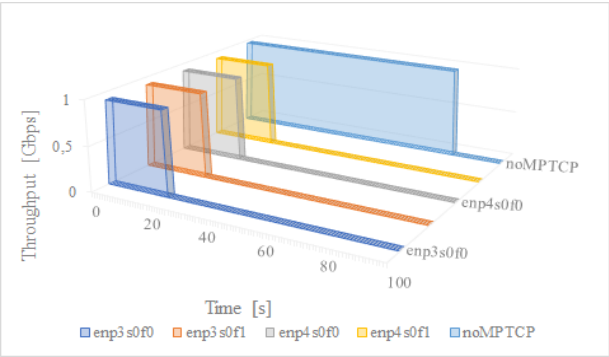

Fig. 7. MPTCP IPv4 FTP throughput performance using 4 interfaces

C. *CPU Utilization*

Finally, we have looked at the CPU resource requirement of the MPT and the MPTCP solutions. Fig. 8 shows the most critical corner case, namely the CPU performance figures while using IPv6. Naturally, in the other less taxing cases results were a bit better, however to keep the paper tidy those are not shown here. Somewhat of a linear progression is also discernible in this context, depending on the number of enabled paths.

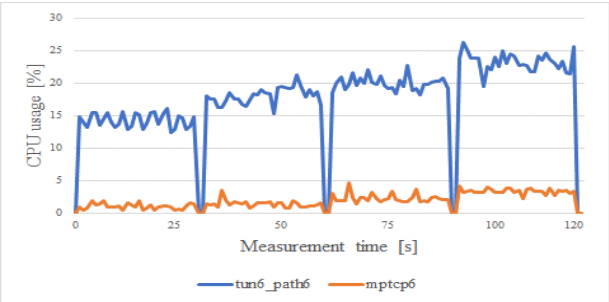

Fig. 8. MPT-MPTCP CPU utilization comparison

However, the high resource usage of the MPT is also conspicuous, hovering around 15% while using one path, and reaching even 25% with four paths enabled. In contrast, the MPTCP seems much more resource efficient with CPU usage hovering around 1-2% using one path, and reaching 3.8% with four paths enabled, but never going over 4%.

V. CONCLUSIONS

In our paper we have used an IPv4-IPv6 environment with four Gigabit Ethernet paths to provide a comparison between the reference values achieved with the MPTCP and the results achieved with the MPT-GRE multipath communication software library developed by our research group. Both solutions have proved to be an effective approach for multipath communication. A significant difference could be seen in CPU utilization, in favor of the MPTCP. As our main conclusion, we can say that the MPTCP can be the preferable choice primarily for multipath data transfer over TCP when low resource usage is key. The main advantage and preferable use case for the MPT is transferring multimedia traffic over UDP. Nevertheless, throughput performance is more than adequate using either solution. The edge that the MPTCP has in efficiency can primarily be attributed to its kernel-level implementation. With that being said, a kernel-level implementation of the MPT-GRE is also among our plans for the future. Either one of these software solutions can be effectively applied for the purposes of cloud-, or fog-computing, telemedicine, wireless environments (roaming problem), and in datacenters as well.

ACKNOWLEDGMENTS

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.

REFERENCES

[1] I. Fitigau, N. Tomai, G. Toderean, “Network Communication Reliability Analysis in Terms of Fault Tree”, Journal of Electrical and Electronics Engineering, vol. 8, no. 1, pp. 13 – 16, May 2015.

[2] "The MPTCP Project official website", - http://multipath-tcp.org/ [April 2019]

[3] "The MPT-GRE Project official website", [March 2019] - https://erlang2.inf.unideb.hu/~szilagyi/index.php/en/mpt/

[4] B. Almási, G. Lencse, S. Szilágyi, "Investigating the Multipath Extension of the GRE in UDP Technology", Computer Communications, vol. 103, issue. C, pp. 29–38, 2017. https://doi.org/10.1016/j.comcom.2017.02.002

[5] G. Lencse, S. Szilágyi, F. Fejes, M. Georgescu, "Internet Draft: MPT Network Layer Multipath Library" - https://tools.ietf.org/html/draft-lencse-tsvwg-mpt-03 [March 2019]

[6] F. Fejes, "MPT source code on GitHub" – [March 2019] https://github.com/spyff/mpt/

[7] Á. Kovács, "Comparing the aggregation capability of the MPT communications library and Multipath TCP", in Proc. 7th IEEE Int. Conference on Cognitive Infocommunications (CogInfoCom), 2016, pp. 157–161. https://doi.org/10.1109/CogInfoCom.2016.7804542