Paper

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

Szabolcs Szilágyi, Ferenc Fejes, and Róbert Katona

Department of Informatics Systems and Networks, Faculty of Informatics, University of Debrecen, Hungary

https://doi.org/10.26636/jtit.2018.122817

Abstract—This paper presents the MPT-GRE software, a novel multipath communication technology founded on the Generic Routing Encapsulation (GRE) protocol in UDP tunneling RFC specification. It enables the creation of a GRE protocol-based UDP tunnel built on several communication channels. On the other hand, MPTCP is one of the most typical representatives among multipath communication technologies, basing its operation on the utilization of TCP subflows. The authors compare the path-aggregation capabilities of these two technologies using four Fast Ethernet channels. The tests were carried out with the iperf3 network bandwidth measurement tool, and while transferring data using the FTP protocol over both IPv4 and IPv6.

Keywords— channel capacity aggregation, MPTCP, MPT-GRE, multipath communication, performance analysis, tunneling.

1. Introduction

The Internet communication environment based on the TCP/IP protocol stack provides only one path per communication session for data transfer. The one-path approach is acceptable in the case of systems that only operate with one network interface or a single point of exit towards the Internet. However, most of the devices in use today are factory equipped with multiple network interfaces which, for example, include Ethernet ports for wired networks and multiple radio interfaces for Wi-Fi and mobile data connections (e.g. 3G, LTE).

The classic, single-path based communication technologies are not capable of harnessing the multi-interface capabilities of devices. The communication-related performance (e.g. throughput) can be improved, if the network environments support the use of multiple data paths in a given communication session. Furthermore, in the case of interface malfunction or overload, the capability of automatic handover between the interfaces of a device in a communication session can improve user quality of experience. Several multipath solutions have already been devised, operating in the data link and transport layers [1]–[4]. The most well-known representative among multipath communication technologies is Multipath TCP (MPTCP)[1] situated in the transport layer [5].

A new architecture is introduced in this paper that provides a very easy-to-use extension to the current TCP/IP protocol stack, enabling the use of multiple paths between communication endpoints. The new multipath environment was implemented in the MPT-GRE software tool[2]. The architecture of MPT-GRE is entirely different from that of MPTCP (see Figs. 1 and 2). MPT-GRE provides both TCP and UDP transport protocol support to applications and it operates in the third layer (network layer), while MPTCP operates in the fourth layer (transport layer).

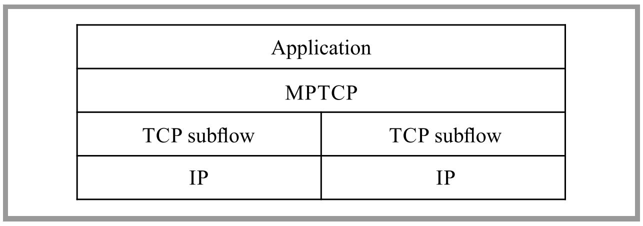


Fig. 1. Architecture of the MPTCP protocol stack.

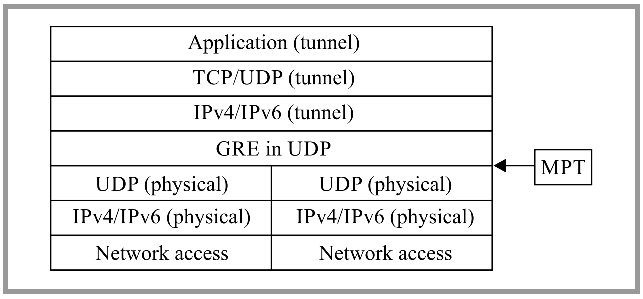


Fig. 2. MPT-GRE conceptual architecture.

The MPT-GRE software was designed on Linux systems with the aim of creating a laboratory setup for measurements, which helped us carry out a performance analysis of its multipath capabilities. Currently it is also available

[1] MPTCP – http://multipath-tcp.org/

[2] MPT-GRE – MultiPath Technology based on Generic Routing Encapsulation in UDP RFC specification: http://irh.inf.unideb.hu/user/szilagyi/mpt/

for Raspbian based systems, and the development of an Android version is in progress [6]–[8]. Our results support the fact that the MPT-GRE multipath solution is capable of efficiently aggregating the throughput of several interfaces used for communication.

The rest of the paper contains a brief look at the operating principles of MPT-GRE and MPTCP in Section 2, while in Section 3 the environment used for measurements is described. Section 4 gives a summary of the results, followed by plans and final conclusions.

2. Multipath Solutions

In this section, we briefly present MPTCP, the most typical representative of these technologies, and then describe the operating principle of the MPT-GRE software library developed.

2.1. MPTCP Operating Principle

The traditional TCP/IP communication infrastructure e.g. [9], is limited to a single IP address per communication endpoint. The current IP technology uses IP address pairs and TCP or UDP port numbers for identifying the communication endpoints (sockets).

MultiPath TCP (MPTCP) operates with an entirely new layered architecture [10]. The new MPTCP sub-layer provides a communication interface to applications, while the TCP sub-flow sub-layers, situated directly below the MPTCP sub-layer, are responsible for creating the multipath communication environment. The TCP protocol operating in the sub-flow sub-layer is responsible for providing flow control and for ensuring reliability (see Fig. 1).

The available literature presents quite a few examples that examine the throughput capabilities of MPTCP. For example, Paasch *et al.* in [11] presented a measurement environment that utilized six 10 Gb links between two endpoints, with an aggregated throughput of 50 Gbps, which is regarded world-wide as the leading performance in the field. Naturally, besides its numerous advantages, MPTCP has a couple of drawbacks:

- tuning of the application layer or the operating system might be necessary to achieve optimal operation,
- operates in the transport layer,
- provides TCP-only support, which can cause problems, e.g. during the usage of multimedia applications.

These problems motivated us to develop a software solution supporting multipath communication that eliminates the aforementioned disadvantages.

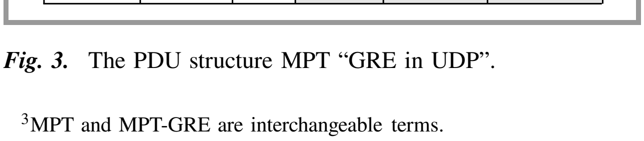
2.2. MPT-GRE Operating Principle

The multipath communication architecture is based on the IETF RFC 8086 "GRE in UDP encapsulation" [12]–[14], extending its operating principle to a multipath environment (Fig. 2).

In addition to the classic layered architecture (see e.g. [9]), we introduced a new logical (tunnel) interface (Fig. 2). The layers above the GRE in UDP (tunnel) operate identically to the traditional environment, with the exception that the data arriving from the application layer is sent to a logical (tunnel) interface instead of a physical one. Below the logical interface it is possible to map the communication session to multiple physical interfaces. In practice, the MPT[3] software is situated between these two sections, designed to control the whole operation.

The basic operating principle is as follows. First a logical (tunnel) interface is created on the endpoints, which the applications can use for socket identification. The MPT-GRE software reads the packet arriving to the tunnel interface (IPv4 or IPv6) on the sending host. This packet is encapsulated in a new GRE in UDP segment before being forwarded on a possible physical path. On the receiving host, the header of the GRE in UDP segment is removed before forwarding the embedded data (which is the original packet forwarded from the sender's tunnel interface) to the receiving host's tunnel interface. The (logical) connection between the communicating hosts is of the direct, point-to-point type. With this architecture, no modifications whatsoever are required in the application, as it uses a single logical interface (the tunnel interface) during the entirety of the communication session. In addition, the application can use the UDP transport protocol above the tunnel interface, as the solution is not limited to the TCP protocol like in the case of MPTCP. If the MPT-GRE library uses the UDP protocol during the encapsulation process, retransmission and flow-control services will not be provided below the tunnel interface.

In Fig. 3, the PDU structure of the MPT-based communication is presented. The grey blocks denote the packet arriving from the application layer, which gets forwarded to the logical (tunnel) interface. The MPT-GRE software will read the packet arriving to the tunnel interface and will assign one of the physical interfaces' IP address to it. It has to be noted that the MPT-GRE library is a dual-stack implementation. The separation of the physical and logical IP layer allows for the use of any IP version (IPv4 or IPv6) as well as all combinations, e.g. IPv4 below the tunnel, IPv6 above the tunnel, or *vice versa.* The receiving host listens on the designated UDP port number and after stripping the headers of the incoming packets it forwards the embedded PDUs to the logical interface. Above the tunnel interface, the application layer can normally process the incoming



IP packet v4/v6	UDP fixed port	GRE	Tunnel IP v4/v6	Tunnel TCP/UDP	Application data

Fig. 3. The PDU structure MPT "GRE in UDP".

[3] MPT and MPT-GRE are interchangeable terms.

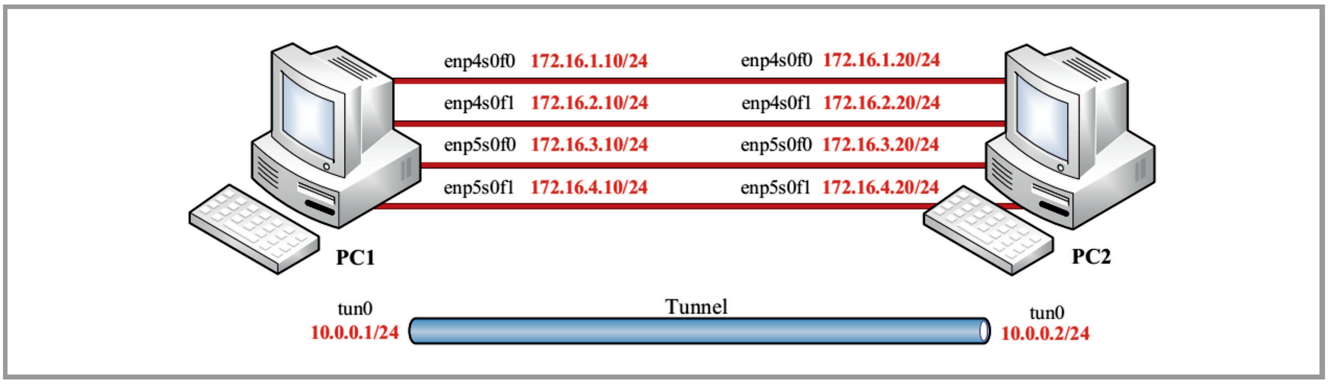


Fig. 4. Four wired paths laboratory measurement environment.

Table 1
IPv4 and IPv6 addressing

Device	Interface	IPv4/IPv6 address prefix	Default gateway
PC1	enp4s0f0	172.16.1.10/24 2001:db8:acad:1::10/64	172.16.1.20/24 2001:db8:acad:1::20/64
	enp4s0f1	172.16.2.10/24 2001:db8:acad:2::10/64	172.16.2.20/24 2001:db8:acad:2::20/64
	enp5s0f0	172.16.3.10/24 2001:db8:acad:3::10/64	172.16.3.20/24 2001:db8:acad:3::20/64
	enp5s0f1	172.16.4.10/24 2001:db8:acad:4::10/64	172.16.4.20/24 2001:db8:acad:4::20/64
	tun0 (only for MPT-GRE)	10.0.0.1/24 fec::1:1/112	–
PC2	enp4s0f0	172.16.1.20/24 2001:db8:acad:1::20/64	172.16.1.10/24 2001:db8:acad:1::10/64
	enp4s0f1	172.16.2.20/24 2001:db8:acad:2::20/64	172.16.2.10/24 2001:db8:acad:2::10/64
	enp5s0f0	172.16.3.20/24 2001:db8:acad:3::20/64	172.16.3.10/24 2001:db8:acad:3::10/64
	enp5s0f1	172.16.4.20/24 2001:db8:acad:4::20/64	172.16.4.10/24 2001:db8:acad:4::10/64
	tun0 (only for MPT-GRE)	10.0.0.2/24 fec::1:2/112	–

packets. The detailed operating principle of MPT-GRE is described in [15] and [16].

3. The Measurement Network Environment

The laboratory environment for the measurements comprised two PCs that were directly connected by wire (Fig. 4). The computers had the following configuration:

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

Integrated network interface cards were used for remote management purposes and during measurements, the integrated NICs were shut down to avoid excess traffic. The performance measurements were carried out between the 4-port server adapters. Both PCs had one of these installed. As these were Gigabit interfaces, we limited their speed to create the Fast Ethernet measurement environment.

The identically named interface pairs were connected via CAT6 STP cables to create the independent physical paths needed for performing the measurements. The applied addressing scheme is summarized in Table 1.

3.1. MPT-GRE Measurements Setup

Even though we used the same physical laboratory environment for MPT-GRE and MPTCP measurements, the implementations of the two multipath technologies required different configuration steps.

First, the MPT-GRE was downloaded and installed. After the packages were installed, the necessary network parameters in the two basic configuration files were modified.

At the start, we disabled three out of the four interfaces on PC1, and then enabled them one-by-one using the Python script according to the measurement schedule, ensuring the gradual aggregation of the communication channel. We used *iperf3*[4] to measure the throughput, and the *sar*[5] command to examine processor usage. The *tee*[6] program was

[4] https://github.com/esnet/iperf
[5] https://linux.die.net/man/1/sar
[6] http://pubs.opengroup.org/onlinepubs/9699919799/utilities/tee.html

used to log the results into files. Each measurement had a duration of 30 s and was repeated 10 times. Less than 1% deviation could be observed in the results for each set of the repeated measurements. We performed the measurements using all possible IP version combinations in regards to both the tunnel and the physical interfaces (IPv4 over IPv4, IPv4 over IPv6, IPv6 over IPv4, IPv6 over IPv6).

Following the approach of papers [17]–[22], we also carried out for real file transfer performance measurements using scripts.

For FTP performance-analysis we run the scripts on PC1, while PC2 was configured as the FTP server. A 1 GB file was used for each measurement, which was deleted after download completion with the help of the script. We used the *ifstat*[7] tool to display the measurement results. Logging was performed once again with the help of *tee*.

3.2. MPTCP Measurements Setup

We installed MPTCP v0.91 for measurements in *GRUB*[8], which we selected directly after restarting the computer, in order to load the MPTCP kernel module. Next, we configured the routing between the interfaces, and the *mptcp_enabled* and *mptcp_path_manager* parameters.

Finally, with the help of the scripts used for the MPT measurements, we performed the required measurement sequence. At the start, only the first interface was in an active status, the other three were shut down. The appropriately timed reactivation of the interfaces was handled by the script. The MPTCP kernel module was installed on both hosts. The measurement scripts were executed on PC1. Just like in the case of the MPT measurements, we used PC2 as the *iperf3* and FTP server. It has to be noted that for performing the MPTCP measurements, we had to disable the integrated interfaces used for remote access, as MPTCP automatically assigned these to the list of interfaces participating in the MPTCP process. During the MPTCP measurements MPT was not running on the hosts.

4. Measurement Results

4.1. iperf3 Measurements

First, we performed a comparison of the path aggregation capabilities of MPT-GRE and MPTCP. The previously presented *iperf3* command was used for this purpose. As Fig. 5 shows, in the case of MPT-GRE we applied all possible IP version combinations with regards to the physical interfaces and the tunnel interface: IPv4 tunnel over IPv4, IPv6 tunnel over IPv4, IPv4 tunnel over IPv6 and IPv6 tunnel over IPv6. The throughput aggregation capability of the MPT-GRE software minimally decreased compared to the previous list. The same could be observed in the case of MPTCP, where, due to a different implementation procedure, we only examined two cases: when the paths used either IPv4 or IPv6 homogenously. Overall, MPTCP performed slightly better in *iperf3* tests. However, the difference in performance was around 0.9% in all cases. We performed the tests using 1, 2, 3, and 4 interfaces simultaneously. On average, both for MPTCP and MPT-GRE for IPv4, we measured a throughput of 92 Mbps in the case of a single interface, 184 Mbps in the case of two interfaces, 276 Mbps in the case of three interfaces, and 368 Mbps in the case of four interfaces used.

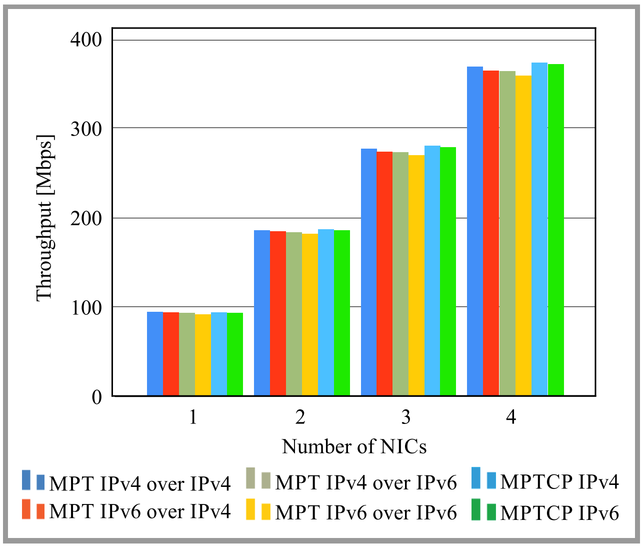


Fig. 5. MPT-MPTCP *iperf3* test comparison. (See color pictures online at www.nit.eu/publications/journal-jtit)

4.2. FTP Measurements

In the next round, the efficiency of MPTCP and MPT-GRE using the FTP protocol was examined. Figure 6 shows the measured download speed of a 1 GB file using four interfaces. The results are close in this case as well, with MPTCP achieving download speed of 368.47 Mbps over IPv4, while in the case of MPT-GRE with IPv4 tunnel over IPv4, 366.89 Mbps was measured.

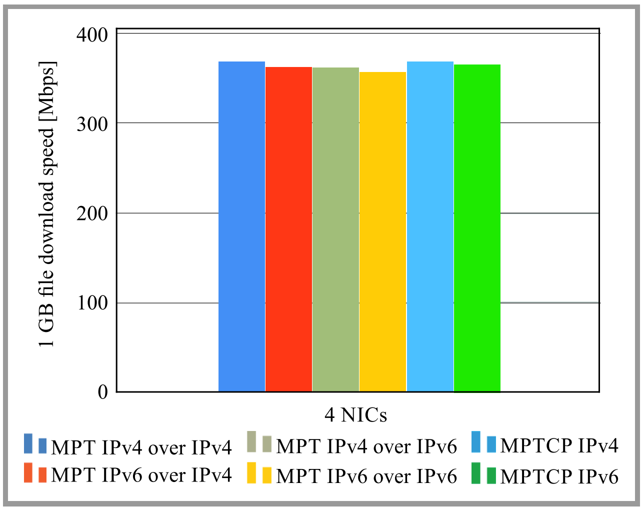


Fig. 6. MPT-MPTCP FTP download speed comparison using 6 interfaces.

[7] https://sourceforge.net/projects/ifstat/
[8] http://www.gnu.org/software/grub/

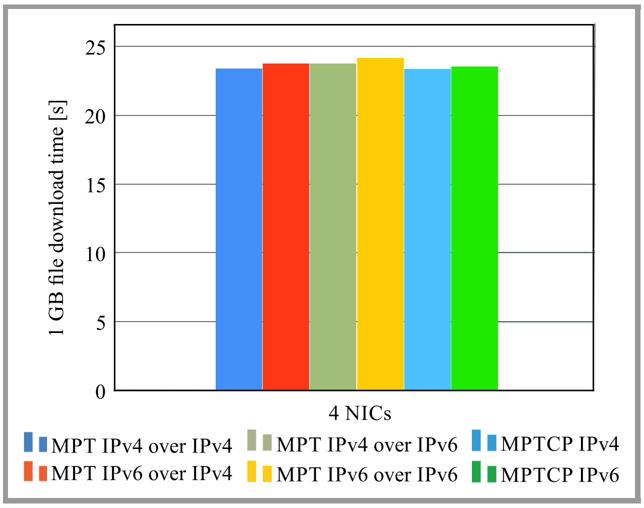


Fig. 7. MPT-MPTCP FTP download time comparison in case of 4 interfaces.

Figure 7 shows the measured download times of a 1 GB file, where values are inversely proportional to Fig. 6. Using MPT-GRE with IPv4 tunnel over IPv4, the 1 GB file was successfully downloaded in 23.41 s, while using MPTCP (IPv4) the duration was around 23.31 s. In case of other sets of measurement results, a minimal deviation could be observed in the file download time.

Figure 8 shows the MPT throughput in an IPv4 tunnel over IPv4 test. Other IP version combinations resulted in similar figures. The result represented with the orange line shows the file download speed measured on the physical interfaces, while the red line shows the throughput measured using the tunnel interface. It is clearly visible that in the case the 1 GB file downloaded using a physical interface, the resulting download duration is 94 s with a download speed of around 96.6 Mbps. If the same file is downloaded through the tunnel interface, with 4 parallel physical connections, the download time is reduced to 25 s, while the download speed increases to around 372.76 Mbps. It can be established that using an MPT-GRE based system the time needed to download the file is practically reduced to a quarter, while the maximum download speed is four times that of a traditional single-path solution.

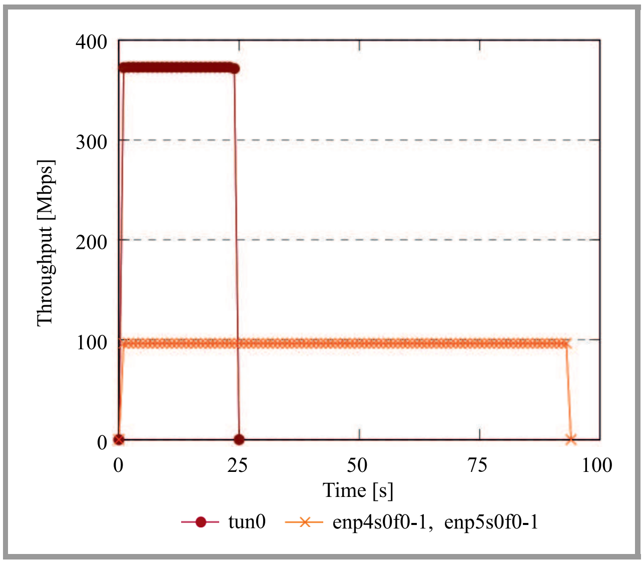


Fig. 8. MPT-GRE IPv4-IPv4 FTP throughput performance using 4 interfaces.

In Fig. 9 we can see the path aggregation performance of MPTCP, similarly during a 1 GB file download. The same can be observed as in the case of MPT-GRE: the download time is reduced to a quarter when utilizing the four physical interfaces at the same time.

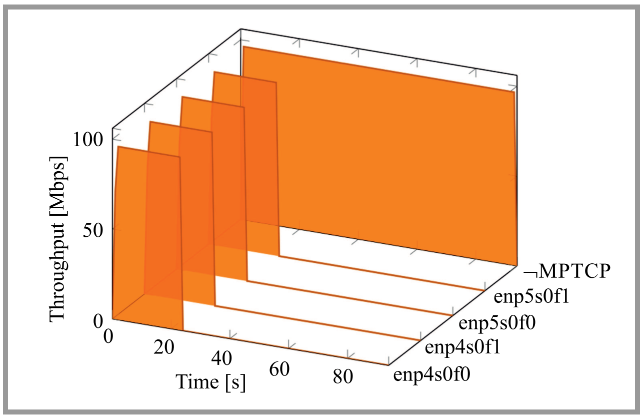


Fig. 9. MPTCP over IPv4 FTP throughput performance using 4 interfaces.

4.3. CPU Utilization in Case of iperf3

In the following section, the CPU utilization results of the MPT-GRE and MPTCP multipath solutions are presented. Figure 10 shows CPU utilization of the MPT and MPTCP solutions while operating with a different number of physical interfaces (1–4), operating over IPv6. This is the most resource-intensive mode of operation. It can be observed that processor utilization is in a linear relationship with the number of physical interfaces, and that the CPU utilization

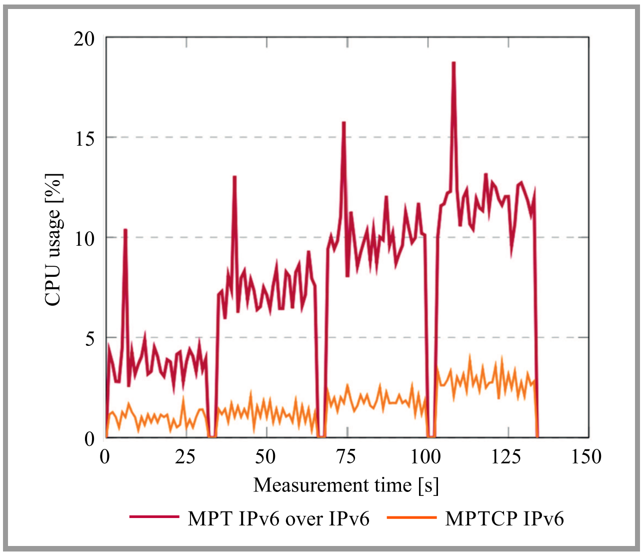


Fig. 10. MPT-MPTCP CPU utilization comparison.

of MPTCP is lower compared to that of MPT-GRE. While the CPU utilization of MPTCP with 4 physical interfaces over IPv6 hovers around the 3% mark, in the case of MPT-GRE with IPv6 tunnel over IPv6 this increases to 12%. Evidently, these are the most critical cases, and when using the other protocol-version combinations the CPU utilization of both solutions is lower.

5. Conclusions

This paper presents two multipath communication systems based on different technologies, comparing their path aggregation capacity, efficiency, and processor utilization. MPTCP is a transport layer, while MPT-GRE is a network layer implementation. In the majority of the measurements MPTCP slightly outperformed MPT-GRE. However, the performance differences were quite minimal. As a result, we highly recommend the use of both technologies. MPTCP for those who primarily concentrate on minimal processor utilization, and MPT for those who would like to communicate over TCP/UDP flexibly with multipath capabilities, e.g. in the case of multimedia traffic (interactive voice, video streaming).

Our future plans include the efficiency review of these two technologies in Gigabit Ethernet and 10 Gigabit Ethernet environments, and the publication of an MPT-GRE RFC draft.

Acknowledgements

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] B. Almási and A. Harman, "An overview of the multipath communication technologies", in *Proc. Adv. on Wireless Sensor Networks 2013 Conf. AWSN 2013*, Debrecen, Hungary, 2013, pp. 7–11.

[2] P. Dong, J. Wang, J. Huang, H. Wang, and G. Min, "Performance enhancement of multipath TCP for wireless communications with multiple radio interfaces", *IEEE Transact. on Commun.*, vol. 64, no. 8, pp. 3456–3466, 2016 (doi: 10.1109/TCOMM.2016.2584615).

[3] M. Li, A. Lukyanenko, Z. Ou, A. Ylä-Jääski, S. Tarkoma, M. Coudron, and S. Secci, "Multipath transmission for the internet: A survey", *IEEE Commun. Surveys Tutor.*, vol. 18, no. 4, pp. 2887–2925, 2016 (doi: 10.1109/COMST.2016.2586112).

[4] Y. Yu, S. Fang, K. M. M. Aung, C. H. Foh, H. Li, and Y. Zhu, "A layer 2 multipath solution and its performance evaluation for Data Center Ethernets", *Int. J. of Commun. Sys.*, vol. 27, no. 11, pp. 2555–2576, 2014 (doi: 10.1002/dac.2488).

[5] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses", *RFC 6824, RFC Editor*, Jan 2013 [Online]. Available: http://tools.ietf.org/html/rfc/6824

[6] B. Almási, "A solution for changing the communication interfaces between Wi-Fi and 3G without packet loss", in *2015 38th Int. Conf. on Telecommun. and Signal Proces. TSP*, IEEE, Prague, Czech Republic, 2015, pp. 1–5 (doi: 10.1109/TSP.2015.7296429).

[7] F. Fejes, R. Katona, and L. Püsök, "Multipath strategies and solutions in multihomed mobile environments", in *2016 7th IEEE Int. Conf. on Cognitive Infocommun. CogInfoCom*, IEEE, Wrocław, Poland, 2016, pp. 79–84 (doi: 10.1109/CogInfoCom.2016.7804529).

[8] F. Fejes, S. Rácz, and G. Szabó, "Application agnostic QoE triggered multipath switching for Android devices", in *Proc. 2017 IEEE Int. Conf. on Commun.: Bridging People, Communities, and Cultures, ICC*, Paris, France 2017, pp. 1585–1591 (doi: 10.1109/ICC.2017.7997450).

[9] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, Boston: Pearson Education, 2011.

[10] C. Paasch *et al.*, "The MPTCP project official website", [Online]. Available: http://multipath-tcp.org/

[11] C. Paasch, G. Detal, S. Barré, F. Duchêne, and O. Bonaventure, "The fastest TCP connection with multipath TCP", *ICTEAM*, UCLouvain, Louvain-la-Neuve, Belgium, 2013.

[12] B. Almási, M. Kósa, F. Fejes, R. Katona, and L. Püsök, "MPT: A solution for eliminating the effect of network breakdowns in case of HD video stream transmission", *2015 6th IEEE Int. Conf. on Cognitive Infocommun. CogInfoCom*, IEEE, Győr, Hungary, 2015, pp. 121–126 (doi: 10.1109/CogInfoCom.2015/7390576).

[13] S. Szilágyi, "The MPT-GRE project official website", [Online]. Available: http://irh.inf.unideb.hu/user/szilagyi/mpt/

[14] L. Yong, E. Crabbe, X. Xu, and T. Herbert, "GRE-in-UDP encapsulation", *RFC 8086, RFC Editor*, 2017.

[15] B. Almási, G. Lencse and S. Szilágyi, "Investigating the multipath extension of the GRE in UDP technology", *Computer Commun.*, vol. 103, no. C, pp. 29–38, 2017 (doi: 10.1016/j.comcom.2017.02.002).

[16] G. Lencse, S. Szilágyi, F. Fejes, and M. Georgescu, "MPT Network Layer Multipath Library", *IETF Network Working Group, Internet Draft*, Jun 2017 [Online]. Available: https://www.ietf.org/iol/draft-lencse-tsvwg-mpt-01.txt

[17] B. Almási, "Multipath communication – a new basis for the future internet cognitive infocommunication", in *Proc. 2013 IEEE 4th Int. Conf. on Cognitive Infocommun. CogInfoCom*, Budapest, Hungary, 2013, pp. 201–204 (doi: 10.1109/CogInfoCom.2013.6719241).

[18] B. Almási and S. Szilágyi, "Multipath FTP and stream transmission analysis using the MPT software environment", *Int. J. of Adv. Research in Computer and Commun. Engineer.*, vol. 2, no. 11, pp. 4267–4272, 2013.

[19] B. Almási and S. Szilágyi, "Throughput performance analysis of the multipath communication library MPT", in *Proc. 2013 36th Int. Conf. on Telecommun. and Signal Proces. TSP*, Rome, Italy, 2013, pp. 86–90 (doi: 10.1109/TSP.2013.6613897).

[20] B. Almási and S. Szilágyi, "Investigating the throughput performance of the MPT multipath communication library in IPv4 and IPv6", in *Proc. Int. J. of Adv. in Telecommun., Electrotec., Signals and Sys.*, vol. 5, no. 1, pp. 53–60, 2016 (doi: 10.11601/ijates.v5i1.148).

[21] Á. Kovács, "Comparing the aggregation capability of the MPT communications library and multipath TCP", in *Proc. 2016 7th IEEE Int. Conf. on Cognitive Infocommun. CogInfoCom*, Wrocław, Poland, 2016, pp. 157–162 (doi: 10.1109/CogInfoCom.2016.7804542).

[22] G. Lencse and Á. Kovács, "Advanced measurements of the aggregation capability of the MPT multipath communication library", *Int. J. of Adv. in Telecommun., Electrotech., Signals and Sys.*, vol. 4, no. 2, pp. 41–48, 2015 (doi: 10.11601/ijates.v4i2.112).

Szabolcs Szilágyi received his M.Sc. in Electrical Engineering and Computer Science from the University of Oradea, Romania in 2009, and his Ph.D. from the University of Debrecen, Hungary in 2015. He has been working for the Department of Informatics Systems and Networks, Faculty of Informatics, University of Debrecen since 2013. At

present, he is a senior lecturer. The area of his research includes performance analysis of communication networks and investigating multipath communication technologies. He is the new head of the Multipath Communication Research Group at the Faculty of Informatics, University of Debrecen.

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

Ferenc Fejes received his B.Sc. in Computer Engineering from the University of Debrecen in 2015. He is currently working on his master's degree in the same field at the University of Debrecen. He expanded his industrial experience at the Ericsson Traffic Laboratory, Budapest. His research interests include multipath networking, QoS/QoE analysis of wireless networks, open source and experimental technologies.

E-mail: fejes@openmailbox.org
Department of Informatics Systems and Networks
Faculty of Informatics
University of Debrecen
26 Kassai Way
4028 Debrecen, Hungary

Róbert Katona received the B.Sc. degree in Computer Science Engineering from the University of Debrecen, Debrecen, Hungary, in 2016 with a specialization in infocommunication networks. Since 2016, he has been pursuing M.Sc. in the same field at the University of Debrecen. Being a member of the MPT research team at the local Faculty of Informatics, his main research interests include multipath communication networks.

E-mail: robert.k@opmbx.org
Department of Informatics Systems and Networks
Faculty of Informatics
University of Debrecen
26 Kassai Way
4028 Debrecen, Hungary