

# MPT-GRE: A Novel Multipath Communication Technology for the Cloud

Szabolcs Szilágyi\*, Imre Bordán\*, Lajos Harangi\*, Benjámin Kiss\*

\*Faculty of Informatics, University of Debrecen  
Kassai út 26. H-4028 Debrecen, Hungary

E-mail: szilagyiszabolcs@inf.unideb.hu, {bordanimre, harangi.lajos.1996, kissbenji14}@gmail.com

**Abstract**—The Cognitive Infocommunications became a hot research area during the past years and it is becoming more and more prevalent. For this reason, telemedicine and the financial services industry are becoming more and more widespread, relying on cloud services. The underlying network infrastructures for the cloud are provided by data centers, which needs to be able to satisfy the new requirements set forth by these systems. The currently deployed data center networks usually have multipath data transfer connections to provide inter-server physical redundancy. Using traditional TCP, they cannot utilize the advantages offered by path aggregation, e.g. improved throughput performance. However, during data center operations both throughput-oriented longer dataflows and delay-sensitive applications can occur. In the case of the latter, even the most representative multipath solution, Multipath TCP (MPTCP), performs poorly, because of its operational architecture. In this paper we present a new L3 multipath communication technology that is capable of efficient data transfer both over TCP and UDP in a dual-path Gigabit Ethernet environment. The MPT network layer multipath communication library – which was proposed to be a possible new basis for the future Cognitive Infocommunications – is capable to use multiple communication channels to create an UDP tunnel which uses GRE tunnel protocol, improving the throughput performance of the cloud-based data centers.

**Keywords**—cloud network; data center; MPT-GRE; multipath communication; tunneling

## I. INTRODUCTION

The notion of Cognitive Infocommunications was proposed in 2012 (see [1]-[2]) and providing a link between infocommunications and cognitive sciences has since proved to be an initiative that sparked a quickly increasing amount of research and new ideas [3]. One of the most popular areas of application is telemedicine, which stores the data needed for its operation in the cloud (see e.g. [4]-[5]).

Cloud computing, which builds its base on datacenters (see Fig.1.), has become one of the most significant infrastructures of the internet [6]-[13]. Thus, multiple CogInfoCom applications including web-based search, file sharing, video streaming, etc., have their storage provided in the cloud. Consequently, as numerous applications communicate through the cloud, the cloud-network has a direct impact on user experience. The data traffic of cloud-based applications can be divided into two groups: throughput-oriented long dataflows,

and delay-sensitive shorter flows. In long dataflows usually, large amounts of data are transferred, often during virtual-machine migration or data synchronization. In short dataflows small amounts of data are transferred, which is typical of interactive applications, e.g. web search. Short dataflows are delay-sensitive and usually require a strict execution deadline, while long dataflows are characterized by a single system requirement: high throughput demand [14].

In this paper we present a multipath communication solution that perfectly meets the aforementioned requirements. The multipath software library named MPT-GRE was implemented at the Faculty of Informatics at the University of Debrecen, and our research group continues to focus on its further development. Our results support the fact that MPT<sup>1</sup> is capable of performing efficient path aggregation in Gigabit Ethernet environments. For this reason, it can be used in cloud-based data centers, improving the performance of CogInfoCom systems.

The second chapter gives a short overview of the base operation of MPT-GRE, while in the third chapter we are going to present the measurement environment. Chapter four discusses the measurement results, giving ideas for further development possibilities in the future.

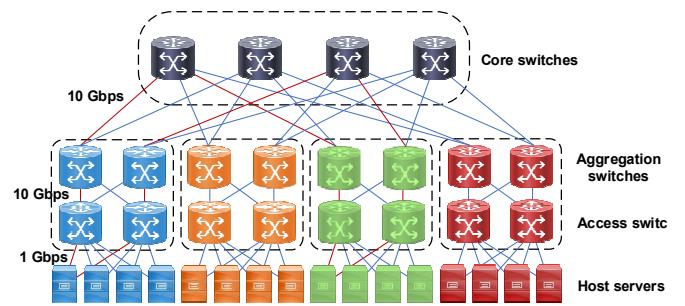


Fig. 1. An example of datacenter topology

## II. MPT-GRE OPERATING PRINCIPLE

As we have discussed in our earlier papers (see e.g. [15]-[16]), the most well-known representative of multipath communication solutions is the multipath extension of

<sup>1</sup> The terms of MPT and MPT-GRE are interchangeable in this paper.

traditional TCP, Multipath TCP - MPTCP. The two main shortcomings of this solution are that MPTCP operates in the Transport-layer, and that it supports data transfer over TCP only (e.g. no UDP support), which can cause problems primarily in case of transferring delay-sensitive multimedia traffic. In fact, identifying these problems was what motivated us to develop a different multipath solution.

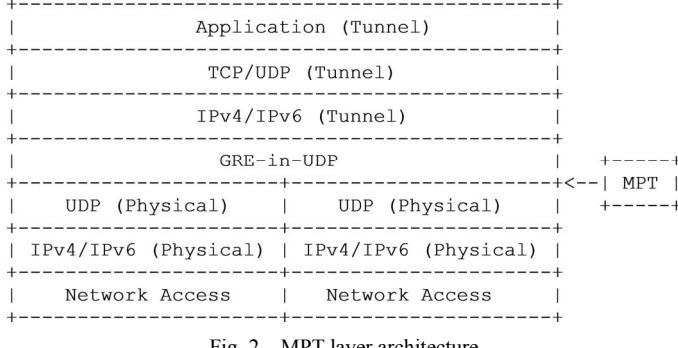


Fig. 2. MPT layer architecture

The current MPT-GRE software-library architecture (see Fig. 2) is based on the description of the IETF RFC 8086, titled „GRE-in-UDP Encapsulation” [17], applying its operation in a multipath environment. In comparison to the traditional TCP/IP technology (see e.g. [18]), a new logical (tunnel) layer was introduced, which is represented via a tunnel interface. The operation above the GRE-in-UDP layer is very similar, with the alteration that the data arriving from the Application-layer is not sent to a physical interface, but it is handed to a tunnel interface. Then, operating below the tunnel layer, the MPT software is responsible for mapping the traffic of the logical interface to the physical interfaces. The operating principle of MPT is discussed in more detail in the following papers and Internet Draft: [16], [19-21].

### III. THE MEASUREMENT ENVIRONMENT

#### A. Hardware and Software Configuration

The efficiency of the MPT was tested on server pairs, as there are no problems with the network bottleneck in the case of several server connections, because uplink connections are faster in size (10 Gbps – see Figure 1). The laboratory measurement setup was composed of two servers having direct physical connections (see Fig. 3). The identical hardware configuration of the two was as follows:

- Gigabyte Z77-D3H motherboard with Intel Z77 chipset
- 8 threads (4 physical cores) on an Intel Core i7-3770K 3.50GHz processor
- 4x4GB 1600MHz DDR3 SDRAM
- Intel PT Quad 1000 Gigabit Ethernet server interface
- Ubuntu 16.04 LTS (Xenial Xerus) 64-bit operating system with 4.4.0-62-generic Linux kernel module.

For remote management we used the integrated NICs that provided internet access to the servers. Naturally, during the given measurements these were shut down to avoid generating

excess traffic. The performance measurements were performed using the quad-port server interfaces, each machine having one such interface card. We only used 2 ports out of the 4 for our measurements; the other two ports were disabled on both servers.

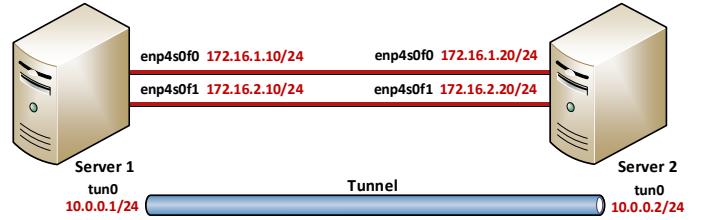


Fig. 3. Two wired paths laboratory measurement environment

The independent physical paths required for the measurement were created between the identically named interface pairs using CAT6 STP cables. The IP address configuration structure can be found in the following table:

Device	Interface	IPv4/IPv6 address prefix	Default gateway
Server 1	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
	tun0	10.0.0.1/24 fec::1:1/112	- -
Server 2	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
	tun0	10.0.0.2/24 fec::1:2/112	- -

Table 1. IPv4 and IPv6 Addressing

#### B. Preparing the MPT-GRE measurements

We carried out our measurements using the MPT version also available on GitHub, uploaded in July 2017 (see [22]). Installing and configuring MPT is very simple and can be performed in a couple of minutes following the instructions in the user guide. After having the required *libc6-dev*, *make*, *gcc*, *gcc-multilib*, *libssl-dev*, *iproute*, *tar* and *gzip* packages installed, what was left was to configure the necessary parameters in the *interface.conf* and *ipv4.conf* configuration files. The *interface.conf* file contained the following parameters:

```

[general]
tunnel_num = 1 ; Accept remote new connection
request
accept_remote = 1
cmdport_local = 60456
cmd_timeout = 25
[tun_0]
name = tun0 ; tunnel mtu: set to 1464 if all the
paths use IPv4, otherwise set to 1440
mtu = 1464
ipv4_addr = 10.0.0.1/24
ipv6_addr = fec::1/64

```

The *ipv4.conf* file configured on Server 1 was setup as follows:

```
;##### Multipath Connection Information: #####
```

```

[connection]
name = MPT_connection_10.0.0.1
permission = 3
ip_ver = 4
ip_local = 10.0.0.1
local_port = 23456
ip_remote = 10.0.0.2
remote_port = 23456
remote_cmd_port = 60456
path_count = 2
network_count = 0
status = 0
reorder_window = 900
max_buffdelay_msec = 44
auth_type = 0

;##### PATHS #####
;#####

[path_0]
interface_name = enp4s0f0
ip_ver = 4
public_ipaddr = 172.16.1.10
gw_ipaddr = 172.16.1.10
remote_ipaddr = 172.16.1.20
keepalive_time = 5
dead_time = 11
weight_out = 1000
cmd_default = 1
status = 0

[path_1]
interface_name = enp4s0f1
ip_ver = 4
public_ipaddr = 172.16.2.10
gw_ipaddr = 172.16.2.10
remote_ipaddr = 172.16.2.20
keepalive_time = 5
dead_time = 11
weight_out = 1000
status = 0

```

Naturally, the IP addresses present in the configuration files were updated for the different measurements to represent the currently used IP version (see Table 1).

To begin our measurements, we wanted to analyze path aggregation performance. We wrote the Python script found below to assist in doing so, adapting it accordingly for the different scenarios (IPv4/IPv6).

```

#!/usr/bin/env python
# coding UTF-8
import sys
import os
import time
import signal
def main():
for i in range(0,10):
os.system("iperf3 -c 10.0.0.2 -t 30 -i 1 -f g | tee
{} &".format("iperf_1interface{}.txt".format(i)))
os.system("sar -u 1 30 | tee {}
&".format("CPU_1interface{}.txt".format(i)))
time.sleep(40)
os.system("./mpt interface enp4s0f1 up")
for i in range(0,10):
os.system("iperf3 -c 10.0.0.2 -t 30 -i 1 -f g | tee
{} &".format("iperf_2interface{}.txt".format(i)))
os.system("sar -u 1 30 | tee {}
&".format("CPU_2interface{}.txt".format(i)))
time.sleep(40)
if __name__ == '__main__':
main()

```

Initially, we shut down one out of the two interfaces of Server 1, and then using the script enabled it again according to the measurement plan. In this manner we ensured the graduated capacity aggregation of the communication paths. Measuring throughput performance was executed via the *iperf3* tool and processor utilization was monitored using the *sar* command, while the results were logged into files with the help of the *tee* program. Every single measurement lasted 30 seconds, and each was repeated 10 times. In each case, less than 1% deviation could be observed in the measurement results. We performed the measurements with regards to the tunnel and the physical interfaces using every IP version combination available (IPv4 over IPv4, IPv4 over IPv6, IPv6 over IPv4 and IPv6 over IPv6).

For measuring file transfer performance, we used the *wget* command as it could be configured so that the destination location was the */dev/null* device instead of the hard drive of the computer. This was important, because the hard drive was not entirely capable of handling the resulting speeds, and as such bottlenecked transfer performance. On the server-side, the file to be transferred was placed in RAM using *tmpfs* type mounting with the help of the *mount* command. All measurements were carried out using a 2GB file. We used the *ifstat* tool to display measurement results, and once again the *tee* program to log them.

```
wget -O /dev/null ftp://10.0.0.2/2GB.zip
```

## IV. MEASUREMENT RESULTS

### A. *iperf3* measurement results

Firstly, we put the path aggregation capacity of MPT-GRE to the test, using the *iperf3* command(s) seen in the script above. As Fig. 4 shows, we applied every possible IP version combination with regards to the tunnel, and the physical interfaces, namely: IPv4 tunnel over IPv4, IPv6 tunnel over IPv4, IPv4 tunnel over IPv6 and IPv6 tunnel over IPv6. The throughput aggregation performance of the MPT-GRE software showed only a minimal decline during the measurements in the order of the previously listed combinations. Overall, we can say that the MPT-GRE solution performed exceptionally during *iperf3* testing in a Gigabit environment.

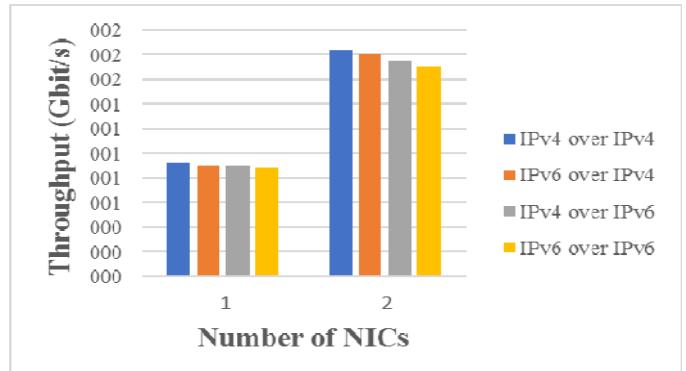


Fig. 4. Testing the MPT iperf3 throughput performance

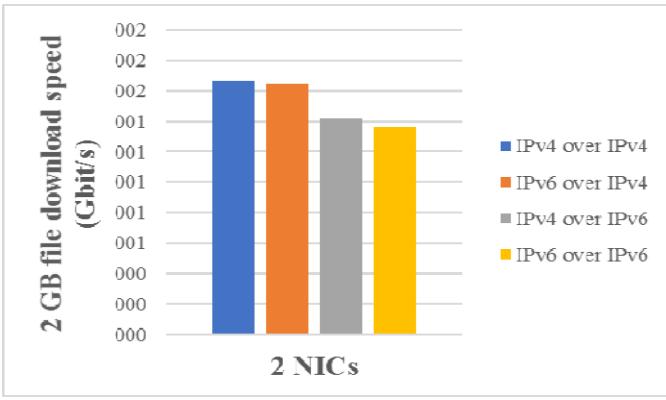


Fig. 5. Wget download speed evaluation of the MPT using 2 interfaces

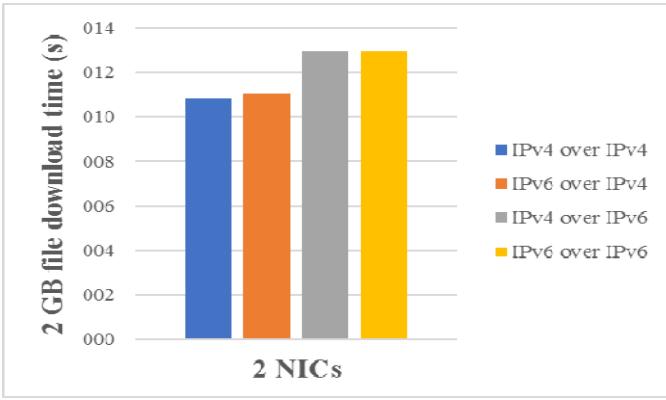


Fig. 6. Wget download time measurement of the MPT using 2 interfaces

We carried out the measurements using 1 interface and using 2 interfaces concurrently. Using one interface, the average throughput was around the 0.92 Gbit/s mark, while in case of using both interfaces the result was 1.84 Gbit/s, which compared to previously published results is unequivocally an improved performance (see [23]-[25]).

### B. FTP measurement results

Next, we measured how efficient MPT-GRE was in case of file transfers, using the `wget` program. Efficiency was examined per interface in this scenario as well. Fig. 5 shows the download speeds measured using two interfaces to download a 2GB file. In the IPv4 tunnel over IPv4 combination the result was 1.66 Gbit/s.

Fig. 6 shows the download time in seconds in the same scenario of downloading a 2GB file, essentially resulting in inversely proportional values to that displayed in the earlier figure. Using MPT-GRE in case of an IPv4 tunnel over IPv4, the download of the 2GB file finished in 10.85 seconds, while the same operation in case of an IPv6 tunnel over IPv6 took around 13 seconds. In case of the other types of measurements, download time results showed minimal deviation.

Fig. 7 shows the throughput performance of MPT in an IPv4 tunnel over IPv4 measurement environment. The other scenarios resulted in similar figures. The throughput displayed via the blue line shows the file download speed measured on a physical interface (without MTP channel aggregation), while

the red line shows the performance measured on the tunnel interface with two under layered physical interfaces. We can observe that in case the 2GB file is downloaded using a physical interface, the download takes 19 seconds with speeds around 0.92 Gbit/s. If we download the same file through the tunnel interface, utilizing 2 physical connections concurrently, the download time decreases to 11 seconds, while the download rate increases to around 1.8 Gbit/s. It can be seen that by using such a dual-path MPT-GRE based system the download time is essentially halved, while the maximum file download speed is doubled in comparison with traditional single-interface solutions.

### C. CPU utilization during iperf3 measurements

Now we are going to present the CPU utilization measurement results of the MPT-GRE multipath software. As previously mentioned among the hardware configuration specifics, each server had an Intel Core i7-3770K 3.50GHz CPU with 8 threads. Figure 8. shows the CPU demand of the MPT software using different IP versions and a varying number (1-2) of physical interfaces. We can observe that the processor utilization increases roughly linearly with the number of physical interfaces. The CPU usage of MPT-GRE in case of an IPv6 tunnel over IPv6 scenario using two physical interfaces increases to 15%. Obviously, this represents the most critical case. Naturally, CPU usage is lower during operation with the other protocol-version combinations.

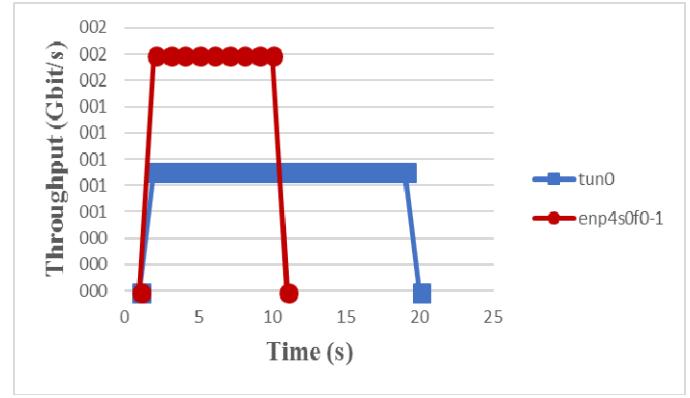


Fig. 7. MPT-GRE IPv4-IPv4 wget throughput performance using 2 interfaces

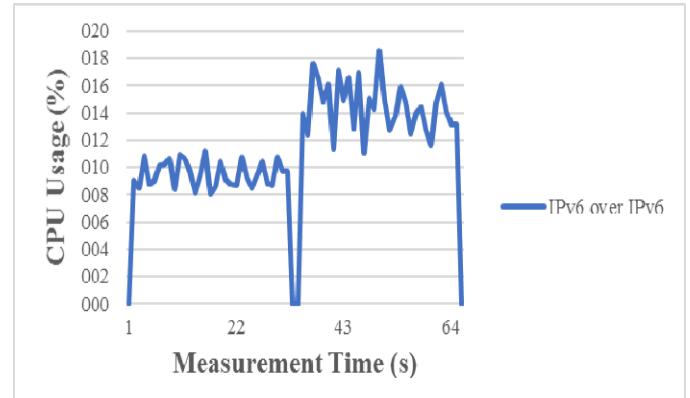


Fig. 8. Evaluation of the MPT CPU utilization

## V. CONCLUSION

This paper introduces a newly developed (in respect of its layered architecture, area of use and measured performance) multipath communication system that is suitable for increasing the efficiency of cloud-based CogInfoCom systems. For performing our measurements, we created a Linux-based, dual server, dual-path Gigabit Ethernet environment, and wrote scripts that (among others) utilized the *iperf3*, *wget*, and *sar* programs. We can say that MPT performed well with regards to both throughput performance and CPU utilization. Therefore, we strongly encourage using this multipath technology in cloud-based CogInfoCom environments, as it is capable to provide efficient path aggregation both over IPv4 and IPv6, and due to its support for operating over UDP, it is suitable for time- and delay-critical applications as well. Our future development plans include extending the performance analysis of MPT-GRE to quad-path Gigabit, and 10Gigabit environments, and implementing a graphical user interface.

## ACKNOWLEDGMENT

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] P. Baranyi, A. Csapó, "Definition and Synergies of Cognitive Infocommunications," *Acta Polytechnica Hungarica*, vol. 9, no. 1, pp. 67-83, 2012.
- [2] A. Csapó, P. Baranyi, "A Unified Terminology for the Structure and Semantics of CogInfoCom Channels," *Acta Polytechnica Hungarica*, vol. 9, no. 1, pp. 85-105, 2012.
- [3] P. Baranyi, A. Csapó, Gy. Sallai "Cognitive Infocommunications (CogInfoCom)", Springer International, 2015.
- [4] I. Péntek, A. Adamkó, Á. Garai: "Cognitive Telemedicine IoT Technology for Dynamically Adaptive eHealth Content Management Reference Framework embedded in Cloud Architecture", 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw 2016.
- [5] Á. Garai, I. Péntek, A. Adamkó: "Interaction-dependent e-Health hub-software adaptation to cloud-based electronic health records", 8th IEEE International Conference on Cognitive Infocommunications, Debrecen, 2017.
- [6] S. Azodolmolky, P. Wieder and R. Yahyapour, "Cloud computing networking: challenges and opportunities for innovations," in IEEE Communications Magazine, vol. 51, no. 7, pp. 54-62, July 2013. doi: 10.1109/MCOM.2013.6553678.
- [7] W. Xia, P. Zhao, Y. Wen and H. Xie, "A Survey on Data Center Networking (DCN): Infrastructure and Operations," in IEEE Communications Surveys & Tutorials, vol. 19, no. 1, pp. 640-656, Firstquarter 2017. doi: 10.1109/COMST.2016.2626784.
- [8] Jose Moura, David Hutchison, "Review and analysis of networking challenges in cloud computing", *Journal of Network and Computer Applications*, Volume 60, 2016, pp. 113-129, ISSN 1084-8045, <https://doi.org/10.1016/j.jnca.2015.11.015>.
- [9] Shamsi, J., Khojaye, M.A. and Qasmi, M.A., "Data-Intensive Cloud Computing: Requirements, Expectations, Challenges, and Solutions", *Journal of Grid Computing*, Springer, 2013 vol. 11, pp. 281-310, <https://doi.org/10.1007/s10723-013-9255-6>.
- [10] M. Chiregi, N. J. Navimipour, "Cloud computing and trust evaluation: A systematic literature review of the state-of-the-art mechanisms", *Journal of Electrical Systems and Information Technology*, 2017, ISSN 2314-7172, <https://doi.org/10.1016/j.jesit.2017.09.001>.
- [11] U. u. Rahman, O. Hakeem, M. Raheem, K. Bilal, S. U. Khan and L. T. Yang, "Nutshell: Cloud Simulation and Current Trends," 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), Chengdu, 2015, pp. 77-86. doi: 10.1109/SmartCity.2015.51.
- [12] Rothenberg C.E. (2010) "Re-architected Cloud Data Center Networks and Their Impact on the Future Internet.", In: Tronco T. (eds) New Network Architectures. Studies in Computational Intelligence, vol 297., pp. 179-184, Springer, Berlin, Heidelberg.
- [13] K. J. Grinnemo and A. Brunstrom, "A first study on using MPTCP to reduce latency for cloud based mobile applications," 2015 IEEE Symposium on Computers and Communication (ISCC), Larnaca, 2015, pp. 64-69. doi: 10.1109/ISCC.2015.7405495.
- [14] W. Wang, L. Zhou and Y. Sun, "Improving Multipath TCP for Latency Sensitive Flows in the Cloud," 2016 5th IEEE International Conference on Cloud Networking (CloudNet), Pisa, 2016, pp. 45-50., doi: 10.1109/CloudNet.2016.53.
- [15] B. Almási and Sz. Szilágyi, "Throughput Performance Analysis of the Multipath Communication Library MPT", TSP 2013 - The 36th International Conference on Telecommunications and Signal Processing, pp. 86-90, ISBN 978-1-4799-0402-0, DOI: 10.1109/TSP.2013.6613897, 2-4 July 2013, Rome, Italy.
- [16] B. Almási, G. Lencse and Sz. Szilágyi, "Investigating the multipath extension of the GRE in UDP technology", Computer Communications, vol. 103, 2017, pp. 29-38, ISSN 0140-3664, <https://doi.org/10.1016/j.comcom.2017.02.002>.
- [17] L. Yong, E. Crabbe, X. Xu and T. Herbert, "GRE-in-UDP Encapsulation", IETF RFC 8086, March 2017.
- [18] A.S. Tanenbaum and D.J. Wetherall, "Computer Networks", Pearson Education, Boston, USA, 5th Ed., 2011.
- [19] B. Almási, "Multipath communication: a new basis for the future Internet cognitive infocommunication", In Proc. CogInfoCom 2013 Conf., Budapest, Hungary, Dec. 2-5, 2013, pp. 201-204, DOI: 10.1109/CogInfoCom.2013.6719241.
- [20] 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", In Proc. CogInfoCom 2015 Conf., Győr, Hungary, Oct. 19-21, 2015, pp. 121-126, DOI: 10.1109/CogInfoCom.2015.7390576.
- [21] G. Lencse, Sz. Szilágyi, F. Fejes, M. Georgescu, "MPT Network Layer Multipath Library", IETF Network Working Group, Internet Draft, June 11, 2018, draft-lencse-tsvwg-mpt-02.txt.
- [22] "MPT - Multi Path Tunnel", source code, <https://github.com/spyff/mpt>
- [23] G. Lencse and Á. Kovács, "Testing the channel aggregation capability of the MPT multipath communication library", World Symposium on Computer Networks and Information Security 2014 (WSCNIS 2014), Hammamet, Tunisia, 13-15 June 2014, ISBN: 978-9938-9511-9-6, Paper ID: 1569946547.
- [24] G. Lencse and Á. Kovács, "Advanced measurements of the aggregation capability of the MPT multipath communication library", International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, Vol. 4. No. 2. (2015.) pp 41-48. DOI: 10.11601/ijates.v4i2.112.
- [25] Á. Kovács, "Comparing the aggregation capability of the MPT communications library and Multipath TCP", In Proc. CogInfoCom 2016 Conf., Wroclaw, Poland, Oct. 16-18, 2016, pp. 157-161, DOI: 10.1109/CogInfoCom.2016.7804542.