Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)
16-18 November 2022, Maldives

Measurement and Analysis of MPT Multipath Throughput in Wire Channels

Naseer A. Jabbar
Department of Telecommunications
Széchenyi István University
Győr, Hungary
al-imareen.naseer@sze.hu

Gábor Lencse
Department of Telecommunications
Széchenyi István University
Győr, Hungary
lencse@sze.hu

***Abstract*— There is a significant growth in the use of the current intelligent and portable devices such as smartphones, laptops, and tablets that result in several network interfaces and various potential paths for communication. However, the mechanism used today via the Internet uses only one path for a communication session. This paper presents multipath throughput tests for the MPT network layer multipath communication library, a novel and promising solution for real-time transmission based on GRE-in-UDP encapsulation that provides an IPv4 or IPv6 tunneling (logical interface). We investigate the efficiency of channel capacity aggregation of MPT in dealing with physical paths via wired channels and test the cases when the physical paths have the same speed or radically different speeds.**

***Keywords*— GRE-in-UDP, MPT, MPTCP, Tunneling, Throughput.**

I. INTRODUCTION

The current environment of the Internet allows only a single path to transfer packets in a communication session. The single path connection technologies cannot use the advantages that are available with multiple interfaces. There are many motives behind resorting to multipath technologies, and as a result of the rapid development and significant growth in the field of communications requires building and designing technologies that support multipath in the communication session as many of the current intelligent information devices (such as laptops, smartphones, and tablets) have multiple network interfaces (Wi-Fi, 5G, and 4G), today the attention and focus is on the topic of multipath communication a prominent study issue. Throughput can be significantly and broadly improved in environments that support multiple interfaces in a communication session. There are various promising solutions that support the multipath technique, e.g. (MPT [1] and MPTCP [2]. MPT is considered one of the essential technologies in taking advantage of the multiple interfaces provided by modern communication devices. MPT uses the GRE-in-UDP encapsulation and it works in the network layer [1]. In contrast, MPTCP uses the TCP protocol within the transport layer [3]. The focus of our paper is on multipath communication using the network layer. MPT can be downloaded for free for most Linux distributions (32-bit, 64-bit) from [4].

Our paper is organized as follows: Section II gives a brief overview of MPT. Section III describes our measurement environment including the MPT configuration. In section IV, we disclose our measurements and results. In Section V, we discuss our results, and give the direction of our future research. Finally, section VI concludes our paper.

II. BRIEF OVERVIEW OF MPT

MPT is a supportive method for a multipath solution that operates within the network layer. It was developed and designed by a research group at Debrecen University, Hungary [1]. MPT takes advantage of GRE-in-UDP capabilities to provide multipath tunneling. MPT can act as a router through which packets can be routed between different networks through tunnel endpoints. This feature means establishing a multipath connection from one site to another. The IP version of the tunnel is independent of the IP version of the path so that the MPT can be used for IPv6 transition purposes [5], [6].

The MPT library creates a tunnel interface between hosts. The logical interface is mapped to the physical interfaces directly by MPT software [7]. Determining the connection session requires applications to custom the private IP address of the tunnel interface. When an application sends an IP packet, it uses the address of the tunnel path, i.e., the address of the logical interface. Thus, there is no need to modify the communication programs for applications.

The possibility that the MPT library provides through the tunnel interface helps in mapping among the tunnel interface and the physical interfaces. For example, when changing the physical interface address or the physical interface itself, the application does not know about this change. It continues to work through the tunnel interface, where the MPT library reorganizes the mapping from the logical interface to the physical interface based on the latest update. Transmission of video in high quality in real-time is one of the challenges that can be solved using multipath technology. The solutions developers and researchers propose to produce and develop technologies for the multipath problem are essential in facing this challenge.

The MPT library provides a typical solution for multipath by giving a logical path (tunnel) that is initialed in the terminal hosts to define the socket, where the MPT library reads the packets coming at the logical interface (IPv4 or IPv6) originating from the sender host; This packet will be wrapped in a new GRE on the UDP part before being sent over a potential physical route [8]. The MPT-GRE conceptual architecture can be seen in Fig. 1.

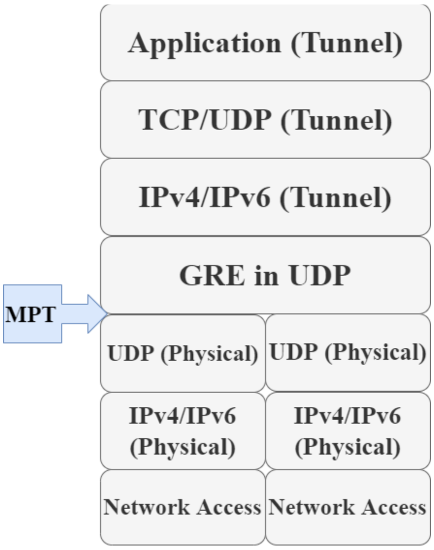

Fig. 1. The Conceptual Architecture of MPT-GRE. [1]

III. TEST ENVIRONMENT

A. *Hardware and Fundamental Configurations*

In our experiments, we used two computers (DELL Precision Workstation 490) according to the following specifications:

- Motherboard with Intel 5000X chipset.
- 8 GB 533MHz DDR2 SDRAM.
- Intel Xeon 5140 2.33 GHz dual core processors.
- Broadcom NetXtreme BCM5752 Gigabit Ethernet controller.

An Intel PT Quad 1000 type four-port Gigabit Ethernet controller was added for each computer. Two Gigabit Ethernet ports were used in each computer. A Cisco Catalyst 2950 switch linked the PCs, limiting the transmission speed to 100Mbps or 10Mbps isolating the two physical connections using VLANs. In the topology and IP address configuration of the test network utilized in the IPv4 tunnel over IPv4 connections testing are depicted in Fig. 2. Debian 8.11 (Jessie) GNU / Linux operating system was installed on both computers. The intent of the measurement network is to examine the throughput efficiency of the MPT software, and to monitor and verify network performance in a network topology that has been designed.

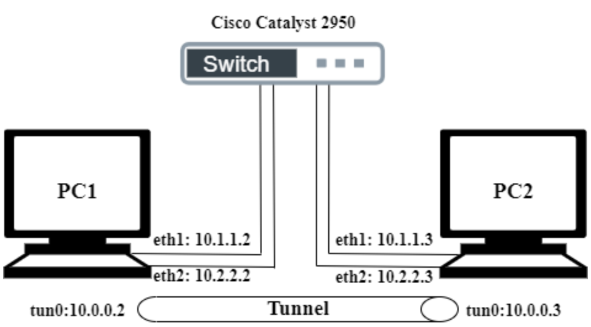

Fig. 2. The Measurement Network Topology.

B. *MPT Software Configuration*

The MPT implementation contains two versions (32-bit and 64-bit). In our paper, version mpt-gre-lib64-2016-06-16.tar.gz was used. We have downloaded the MPT from [4]. The following are the details of the two configuration files: (the paths are determined depending on the installation directory of MPT.) The first file `conf/interface.conf` was:

```
; ####### General Interface Information #######
  60456         # The local cmd UDP port number
  2             # the interfaces number
  1             # Accept remote request
; MPT software make mptsrv as a server
  25            # cmd_timeout
; ############ Information tun0 ##############
  tun0          # Name of tunnel interface
  1440          # Maximum Transfer Unit
  10.0.0.2/24   # IPv4 address and prefix length
```

The same configuration method in the above file was used for the other interface on the second computer. The different types of tunnels files were placed in independent connection files. Also, all the information about IP addresses and a prefix length for each tunnel created by MPT software have been prepared.

On the other hand, the session connection file has been prepared and configured in which the connection configurations IPv4 over IPv4 as defined in the `conf/connections /IPv4.conf` file:

```
; ####### Connections Info. ########
  1             # Connections Num.
; ####### Connection Details #######
MPT_Connection  # Connection Name
3               # Permission Send (1)/Receive (2)
4               # The Version of IP
10.0.0.2        # Local_IP
50230           # Local_Port
10.0.0.3        # Remote_IP
50230           # RDP (Remote_Data_Port)
60456           # UDP Port Number_Remote
2               # Paths_Num.
0               # Networks_Num
5               # Time of Keepalive Message
5               # Dead_Timer (sec)
0               # Status of Connection
0               # Auth_Type
0               # Auth_Key
; ########### Path0 info. ##########
eth1            # Interface Name
4               # The Version of IP
10.1.1.2/24     # Public IP
10.1.1.2        # Gateway_IP
10.1.1.3/24     # Remote IP
5               # Keepalive Time
25              # Dead Time
100             # Weight_out
100             # Weight_in
1               # CMD_default
; ####### Path1 information ########
eth2            # Interface Name
4               # IP_Version
10.2.2.2/24     # Public IP
10.2.2.2        # Gateway_IP
10.2.2.3/24     # Remote IP
5               # Keepalive Time
25              # Dead Time
100             # Weight_out
100             # Weight_in
1               # CMD_default
0               # Path status
```

The remaining paths for this connection were prepared and configured in the same manner that was adopted for the above two paths. It is worth noting that the configuration files have to adhere to a tight structure, including comment-only lines. In [5], the researchers suggested that this be adjusted for the most widely used freestyle configuration files using keyword parsing.

IV. MPT Measurements and Results

To examine and analyze the network throughput running according to MPT software, more scenarios were implemented using the *IPerf3* command, a real-time tool for measuring network performance. It is a cross-platform, open-source client-server program that may be used to test the throughput between two hosts.

We will include the basic used commands in this paper. The following listed commands are IPv4 tunnels over IPv4 measurements. The iperf3 command was:

```
iperf3 -c 10.2.2.2 -t 30 -f M
```

This command ran a 30-sec experiment and returned the throughput in MB/sec units. In iperf terminology, this is referred to as the client-side. The server, on the other hand, was launched with the following command line:

```
iperf3 -s -f M
```

For the throughput measurement, we used the topology shown in Fig. 2, which consists of two computers and four Ethernet links, two for each host, which are connected in a peer-to-peer manner. Two independent paths were established between the two hosts (eth1, eth2). In addition to the logical interface provided by the MPT software (tun0). Table I shows the complete IP address specification for the working environment.

The Cisco Catalyst 2950 switch was used to limit the transmission speed. In our paper, three different scenarios are adopted depending on the transmission speed. The measurements were repeated ten times for each case. The average and standard deviation of the ten throughput values were calculated.

In the first case, the transmission speed was 100Mbps for both paths (eth1, eth2), that is, symmetric paths were used. The case was tested using different time factors (30s, 70s, and 100s). The results are shown in Table II. The results are very stable, the throughput value does not depend on the time factor, and the standard deviation is very low compared to the average. The aggregation is highly efficient, the throughput of the tunnel (184Mbps) is nearly double the single path throughput (94Mbps).

In the second case, the transmission speed was 10Mbps for the first path (eth1) and 100Mbps for the second one (eth2), that is, asymmetric paths were used. The results are shown in Table III. As expected, the throughput of the 10Mbps path is about one tenth of the throughput of the 100Mbps path. The single path results are still very stable as shown by the low standard deviation values. However, the throughput of the tunnel is relatively low: it is much less than the throughput of the 100Mbps path. Therefore, the "aggregation" is very much inefficient. The relatively high standard deviation value shows a kind of instability of the results. We plan to investigate the root causes of this phenomenon in future work.

In the third case, the transmission speed was 1000Mbps for the first path (eth1) and 100Mbps for the second one (eth2). We note that this time the eth1 interfaces of the two computers were interconnected directly, bypassing the switch in order to be able to achieve 1000Mbps speed.

TABLE I. IP Address Details

Computer	Interface	IP address and prefix length	gateway
PC1	eth1	10.1.1.2/24	10.1.1.2
	eth2	10.2.2.2/24	10.2.2.2
	tun0	10.0.0.2/24	-
PC2	eth1	10.1.1.3/24	10.1.1.3
	eth2	10.2.2.3/24	10.2.2.3
	tun0	10.0.0.3/24	-

TABLE II. Measurement Results (Transmission Speed: eth1=100Mbps, eth2=100Mbps)

Cases Time	Interface	Transfer (Mbytes)	Throughput (Mbits/sec)	Standard Deviation
Case 1: Time=30sec	eth1	337.5	94.37	0.048
	eth2	337.5	94.38	0.063
	tun0	658.8	184	0
Case 2: Time=70sec	eth1	786.1	94.21	0.032
	eth2	786.6	94.27	0.048
	tun0	1536	184	0
Case 3: Time=100sec	eth1	1126.4	94.2	0
	eth2	1126.4	94.2	0
	tun0	2191.36	184	0

TABLE III. Measurement Results (Transmission Speed: eth1=10Mbps, eth2=100Mbps)

Cases Time	Interface	Transfer (Mbytes)	Throughput (Mbits/sec)	Standard Deviation
Case 1: Time=30sec	eth1	35.57	9.947	0.187
	eth2	337.6	94.39	0.032
	tun0	80.04	24.689	5.571
Case 2: Time=70sec	eth1	77.57	9.294	0.121
	eth2	786.1	94.22	0.042
	tun0	140.18	16.79	5.165
Case 3: Time=100sec	eth1	111.2	9.322	0.062
	eth2	1124.352	94.2	0
	tun0	194.3	16.29	4.34

TABLE IV. Measurement Results (Transmission Speed: eth1=1000Mbps, eth2=100Mbps)

Cases Time	Interface	Transfer (Mbytes)	Throughput (Mbits/sec)	Standard Deviation
Case 1: Time=30sec	eth1	3368.96	942	0
	eth2	337.5	94.37	0.067
	tun0	658.778	184	0
Case 2: Time=70sec	eth1	7856.128	942	0
	eth2	786	94.2	0
	tun0	1536	184	0
Case 3: Time=100sec	eth1	11264	942	0
	eth2	1126.4	94.2	0
	tun0	2191.36	184	0

As expected, the results for the third case are shown in Table IV. The measured throughput of the 1000Mbsps link was approximately ten times higher than that of the 100Mps link. However, it could contribute to the channel aggregation approximately as much as the 100Mbps link. This phenomenon is even differing from the previous case. We suspect that one factor could be the lack of computing power to handle the aggregation. This phenomenon surely deserves more investigation, and we plan to do so.

The results of the measurements are displayed in Figs. 3–11. These graphs depict the values of interface throughput for the various test scenarios.

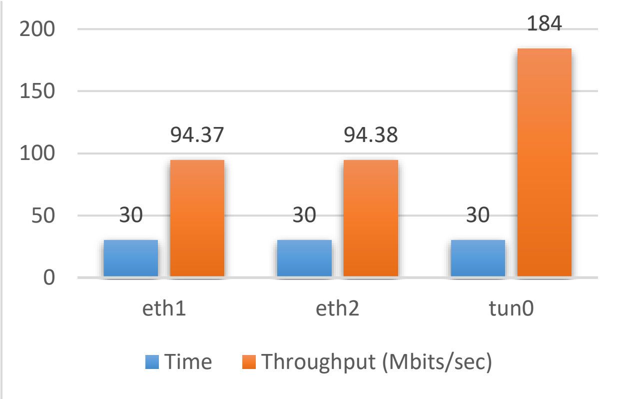

Fig. 3. Throughput Test (eth1=100, eth2=100, Time: 30 Sec).

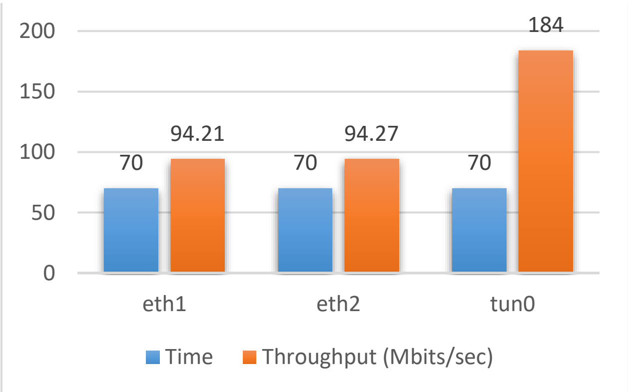

Fig. 4. Throughput Test (eth1=100, eth2=100, Time: 70 Sec).

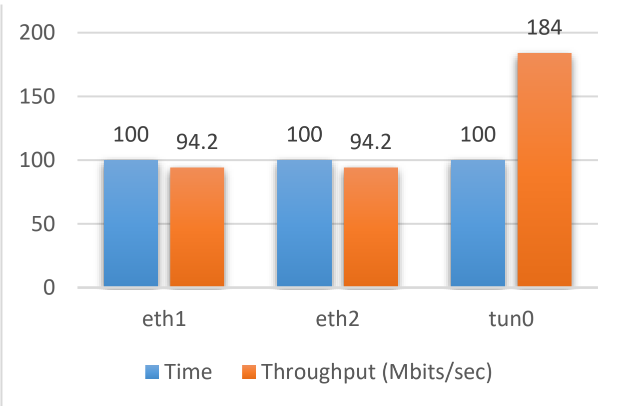

Fig. 5. Throughput Test (eth1=100, eth2=100, Time: 100 Sec).

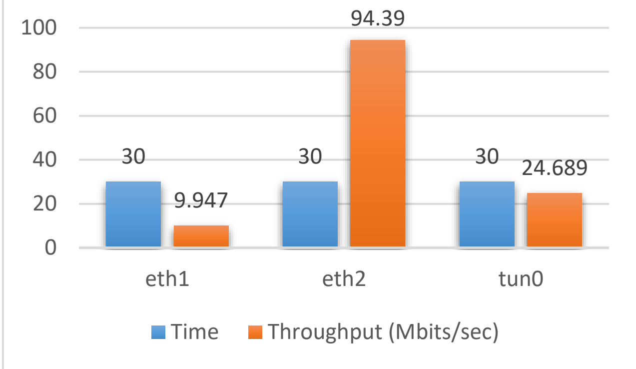

Fig. 6. Throughput Test (eth1=10, eth2=100, Time: 30 Sec).

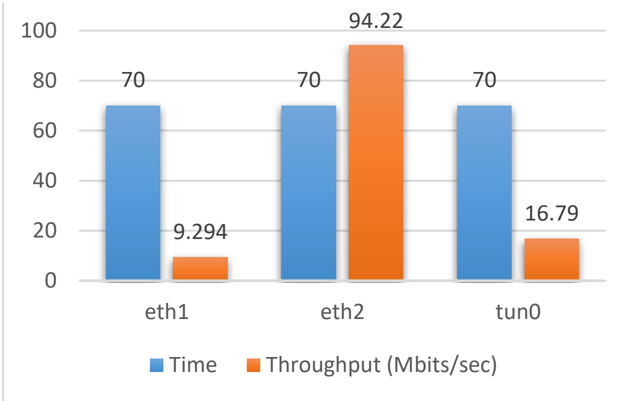

Fig. 7. Throughput Test (eth1=10, eth2=100, Time: 70 Sec).

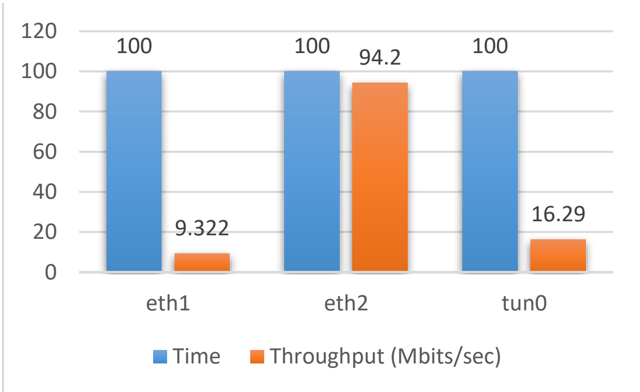

Fig. 8. Throughput Test (eth1=10, eth2=100, Time: 100 Sec).

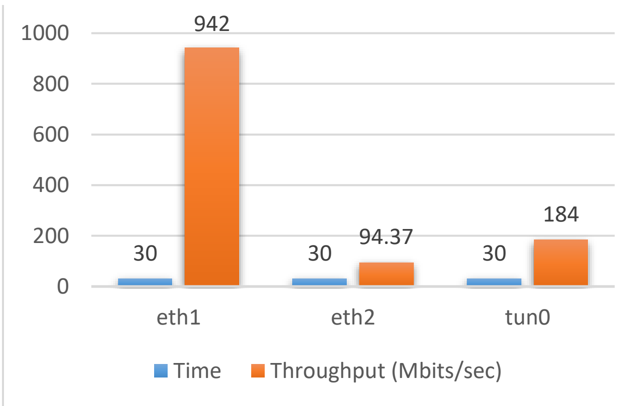

Fig. 9. Throughput Test (eth1=1000, eth2=100, Time: 30 Sec).

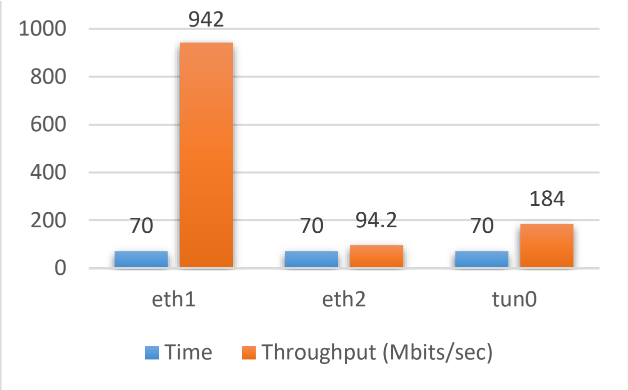

Fig. 10. Throughput Test (eth1=1000, eth2=100, Time: 70 Sec).

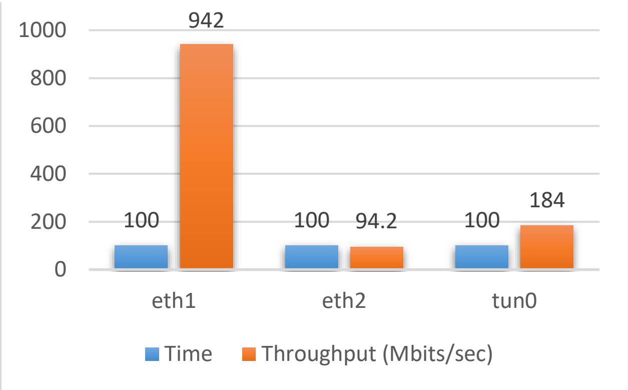

Fig. 11. Throughput Test (eth1=1000, eth2=100, Time: 100 Sec).

V. DISCUSSION AND FUTURE DIRECTIONS OF RESEARCH

The resource constraint of the shared nodes in a mobile environment, though multipath techniques, increases the efficiency of communication sessions. Therefore, several factors must be considered while working on developing a multipath library, such as energy efficiency. On the other hand, one of the challenges and problems open to the MPT network layer multipath communication library is focusing on the metrics that affect QoE or QoS in the mobile environment as a result of mobility and congestion control.

When testing the aggregation capability of the MPT network layer communication library, we have encountered and interesting phenomenon. Whereas its aggregation efficiency was very high using symmetrical paths, the aggregation was rather inefficient when asymmetrical paths were used. We consider this topic important, because as far as we are aware other papers examined only the symmetric case, e.g., in [5], [6], [7], [8], and [9]. Therefore, we plan to examine the aggregation capability of MPT using asymmetric paths with different speeds. We plan to analyze how the ratio of the speeds of the paths influences the efficiency of the aggregation.

VI. CONCLUSION

We have tested the channel aggregation capability of the MPT library that provides a network layer multipath solution based on GRE-in-UDP between two hosts with the possibility of using any transport layer protocol (TCP or UDP). The performance of the throughput aggregation of the MPT library was tested for two physical paths having the same or one order of magnitude different speeds. The measurements were taken using the industry standard iperf3 real-time benchmarking tool. Our results confirmed the results of other researchers that MPT can efficiently aggregate the transmission capacity of paths with the same speed. On the other hand, we pointed out that aggregation is rather inefficient, if the paths have significantly different speeds. As our direction of future research, we plan to investigate the root cause of this phenomenon.

REFERENCES

[1] B. Almási, G. Lencse, and S. Szilágyi, "Investigating the multipath extension of the GRE in UDP technology," Comput. Commun., vol. 103, pp. 29–38, 2017.
https://doi.org/10.1016/j.comcom.2017.02.002

[2] A. Ford, C. Raiciu, M. Handley and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses", IETF, Jan. 2013, RFC 6824.

[3] N. A. Jabbar, and G. Lencse, "An Overview of the Multipath Technologies, their Importance and Types," Proceedings of The Technical University of Sofia, 72(1), pp. 21–28, 2022.
https://doi.org/10.47978/TUS.2022.72.01.004

[4] B. Almási, "MPT library", precompiled version can be downloaded from: http://irh.inf.unideb.hu/user/almasi/mpt/, 2016.

[5] G. Lencse and Á. Kovács, "Advanced Measurements of the Aggregation Capability of the MPT Network Layer Multipath Communication Library," Int. J. Adv. Telecommun. Electrotech. Signals Syst., vol. 4, no. 2, 2015.
https://doi.org/10.11601/ijates.v4i2.112

[6] S. Szilágyi, I. Bordán, L. Harangi, and B. Kiss, "Throughput performance comparison of MPT-GRE and MPTCP in the gigabit Ethernet ipv4/ipv6 environment," J. Electra. Electron. Eng., vol. 12, no. 1, pp. 57–60, 2019.
https://doi.org/10.26636/jtit.2018.122817

[7] G. Lencse, Á. Kovács, "Testing the channel aggregation capability of the MPT multipath communication library", In Proc. World Symposium on Computer Networks and Information Security 2014 (WSCNIS 2014), Hammamet, Tunisia, Jun. 13-15, 2014.

[8] S. Szilágyi and I. Bordán, "Throughput performance measurement of the MPT-GRE multipath technology in emulated WAN Environment," CEUR Workshop Proc., vol. 2874, pp. 187–195, 2021.
http://ceur-ws.org/Vol-2874/short17.pdf

[9] S. Szilágyi and I. Bordán, "Investigating the Throughput Performance of the MPT-GRE Network Layer Multipath Library in Emulated WAN Environment," Acta Technica Jaurinensis., vol. 15, no. 1, 2022.
https://doi.org/10.14513/actatechjaur.00639