# On the Impact of Packet Reordering in MPT-GRE Multipath Networks

Naseer Al-Imareen, Gábor Lencse
Department of Telecommunications
Széchenyi István University
Győr, Hungary
al-imareen.naseer@sze.hu
lencse@sze.hu

***Abstract*—Several factors, such as network delays, congestion, differing path lengths, and varied processing durations at intermediary nodes, can cause out-of-order arrival of the packets, negatively affecting network performance, decreasing throughput, and increasing latency. When packets arrive out of order, the receiver may need to wait for missing packets before assembling a complete message, resulting in additional delays. Furthermore, some applications, like music and video streaming, require continuous and ordered data flow. Any out-of-order arrival of packets can result in jitter and decreased quality. To mitigate the impact of this issue in MPT-GRE multipath networks, MPT contains a packet reordering mechanism. In this paper, we designed a network topology that generated delays and limited the transmission speed on one of the paths. We investigated how the out-of-order arrival of packets influences the throughput aggregation capability of the MPT library with and without enabling the packet reordering feature of MPT. Our network throughput measurements show that by enabling MPT's packet reordering mechanism, MPT can efficiently aggregate the throughput of both symmetric and asymmetric channels.**

**Keywords—*MPT; delay; multipath; reorder; pakets*; *GRE-in-UDP.***

## I. INTRODUCTION

The acceleration of the development of technologies that use multiple interfaces motivated researchers to find solutions and techniques that help take advantage of the multipaths. Every day we rely on intelligent devices (4G/5G smartphones, tablets, laptops), all containing multiple network interfaces. In contrast, TCP/IP protocols are designed to use a single interface per session. In devices with multiple interfaces, data exchange during a communication session cannot take advantage of the multiple interfaces, therefore, only one path is available at a time. Thus, there was a need to develop multipath technologies for the maximum benefit of exploiting multiple interfaces, especially in sessions that require significant amount of data exchange. Moreover, some implementations that need continuous and timely data exchange, such as music and video streaming, necessitate a continuous and ordered flow of data, and any flaw in the order of packets can result in jitter and decreased QoS (Quality of Service) and QoE (Quality of Experience) [1], [2]. As a result, the communication structure and protocols must be constructed to carry out multipaths. A growing number of investigations are focusing on multipath communication as a viable area of investigation. Multipath is crucial to deal with fault tolerance where networks with multipaths are more resistant to failures. If a single path in the network fails, traffic can be switched through different paths, guaranteeing that the network maintains to operate without interruption [3]. Multi-paths solution can distribute traffic more evenly, preventing congestion on any path. This technique can improve network performance and reduce latency. Having multiple paths in a network can provide redundancy, ensuring that data can still be transmitted even if one path is congested or unavailable. Overall, the network that uses a multipath mechanism provides essential benefits that contribute to the network's reliability and performance. MPT [4] and MPTCP [5] are potential alternatives using the multipath approach. Combining congestion control and packet reordering techniques in MPTCP provides improved aggregate throughput performance for varying link delay disparities [6], [7]. Transparent end-to-end connection setup, multipath-allowed congestion control, and the elimination of head-of-line blocking are just a few of the benefits of the MPTCP protocol's adoption [8]. Regarding modern electronic devices, MPT can be a crucial technique for making the most of the multiple network interfaces of our contemporary devices. GRE-in-UDP encapsulation allows MPT to function at the network layer [9].

In our current paper, we designed a network topology to investigate the effect of the out-of-order arrival of the packets on the efficiency of the tunnel throughput aggregation capability of MPT-GRE. We used one path with a fixed transmission speed and no delay, whereas we applied several transmission speeds and delay combinations on the other path. We investigated how the out-of-order arrival of packets influences the throughput aggregation capability of the MPT-GRE software with and without enabling the packet reordering feature of MPT. We also examined how the value of the reordering window parameter influences the effectiveness of packet reordering and thus, the throughput aggregation capability of the MPT-GRE.

The rest of this paper is arranged as follows. In section II, we introduced a summary of MPT. In section III, we explain the packet reordering in MPT. Section IV includes experimental test environments in both hardware and software configurations. Section V displays the various MPT measurements and results. Finally, we present our conclusion.

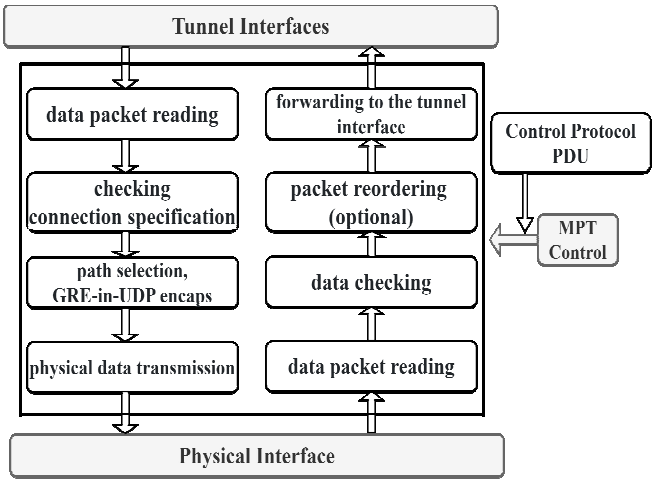


Fig. 1. The MPT-GRE conceptual architecture [13].

## II. MPT Communication Library

MPT is a communication library that enables multipath data transfer over the GRE (Generic Routing Encapsulation) protocol. The MPT-GRE library is an open-source project actively maintained and developed by a research group at the University of Debrecen [4]. It is available for various operating systems and platforms, for example, Linux. It allows multipaths to be utilized simultaneously for increased bandwidth and improved network utilization. MPT software can be used for a variety of applications, including video streaming, and high-performance computing [10]. It encapsulates the data in GRE packets, then the packets are transmitted over one of the multiple paths and the date stream is reassembled at the receiving end. The MPT software generates a logical interface (tunnel) to facilitate communication between hosts. The MPT software offers a standard multipath solution by providing a tunnel interface initialized in the end nodes to define the socket; the MPT software then reads packets arriving at the tunnel interface (IPv4 or IPv6) from the source node. For the UDP portion of this packet, a new GRE will be created before it is delivered via a potential physical route [11], [12]. Fig. 1 shows the conceptual architecture of MPT-GRE. The tunnel interface can be established directly to the physical interfaces using MPT software.

The multi-layered structure of the MPT is depicted here. MPT improves the original GRE-in-UDP concept by allowing multiple physical channels. It's similar to MPTCP, but MPT uses UDP at the bottom, expands on GRE in UDP, and gives us an IP tunnel layer that supports both TCP and UDP [14].

## III. Packet Reordering in MPT

In MPT, the delay of the various channels might vary, and out of order packet arrival can happen during packet sequence transmission. An optional component of the MPT environment provides correctly ordered packet transmission for tunnel communication. When this option is turned on, the receiver stores incoming (unordered) packets in a buffer-array. The packets are then ordered according to the GRE sequence

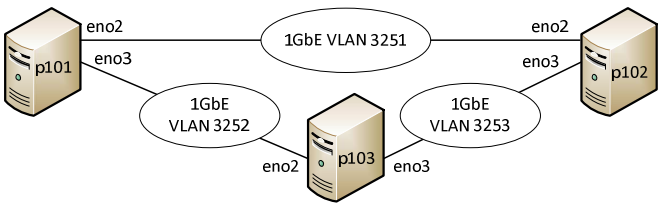


Fig. 2. The Measurement Network Topology.

numbers, ensuring they are transferred to the receiver's tunnel interface correctly. Two parameters control the reordering. The reorder window option specifies the length of the buffer array used for reordering based on generic routing encapsulation sequence numbers. The maximum buffer delay parameter specifies how long a packet can be stored in the buffer array (milliseconds). Even if packets are missing before the packet under consideration, the packet will be sent to the tunnel interface if it is delayed in the buffer array for the given time. MPT will not wait for more time when data isn't received, and it's considered lost. Arriving packets are sent to the logical interface based on their GRE sequence number, ensuring the ordered delivery is maintained even if a packet is lost. If the maximum buffer delay is set too low, and the following number in the sequence doesn't arrive, MPT may incorrectly consider it a lost one; MPT must remove it to ensure order-right delivery [15]. If set too high, the packet loss will be detected too late, lowering communication performance. The reorder window parameter has to be large enough to adjust packets arriving at the maximum line rate from all operational paths of the selected connection within a maximum buffer delay time [13].

## IV. MPT Test Environment

We have constructed a measurement and evaluation test system, as illustrated in Fig. 2.

As a test environment, we used the resources of NICT (National Institute of Information and Communications Technology) StarBED, Japan. We used three DELL PowerEdge R430 servers having two dual core Intel Xeon E5-2683 v4 CPUs and 384GB DDR4-2400 Memory.

Two distinct versions of MPT have been implemented (32-bit and 64-bit). We used mpt-gre-lib64-2019.tar.gz in our paper. We downloaded the MPT file from [16].

In our previous paper [14], we have explained it detail, how the MPT configuration files are to be set. This time we have made the configuration files available on GitHub[17].

## V. Experiments with Packet Reordering

To examine and analyze the impact of the packet reordering mechanism on the MPT, several scenarios were implemented with different cases in terms of enabling and disabling packet reordering and testing the tunnel throughput efficiency in the MPT library using the IPerf3 [18] tool to measure network

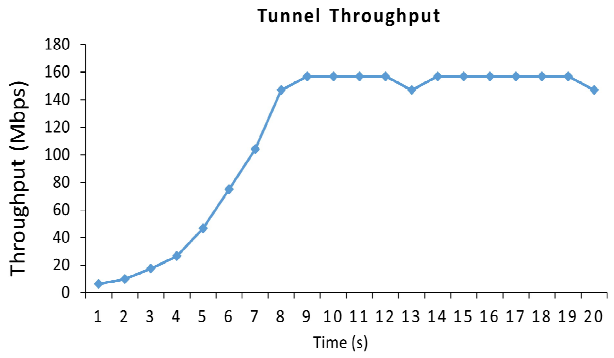


Fig. 3. The transient of the tunnel throughput at the beginning of the measurement (second path speed: 70 Mbps, delay: 50 ms).

throughput in real time. Throughput refers to the amount of data that can be transmitted in each second through the tunnel. On the client end, we used the following style commands:

```
iperf3 -c 192.2.2.2 -t 100 -f M
```

Using this command, we ran an experiment for 100 seconds and got the throughput in MB/s. The MPT library was examined with asymmetric paths at different speeds, 10, 20, 30 ... and 100 Mbps, and with appropriate weight setting for the second path [14]. On the server end, the following command line was used to start IPerf3 in server mode:

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

We used an additional computer to manage the varying delays of bypassing traffic using the Linux **tc** tool. We tested with various delays and speed amounts using the following command:

```
tc qdisc add dev eno2 root netem delay x ms rate y mbit.
```

The values of delay *x* have been set to 10 ms, 20 ms, 30 ms, 40 ms, and 50 ms, whereas the values of transmission speed *y* have been set to 10 Mbps, 20 Mbps, …, and 100 Mbps.

We note that the delay has been added to both network interfaces of the computer, i.e., the packets were delayed in both directions.

### *A. With Packet Reordering Mechanism*

In the first scenario, the packet reordering mechanism in MPT was enabled by setting the values of the reorder window and maximum buffer delay parameters to 900 and 300 ms, respectively.

When we measured the throughput of the MPT tunnel, the results showed an initial transient at the beginning of the tests and then they became stable. A sample for the first 20 seconds results is presented in Fig. 3. It shows the transient state of measurement when the transmission speed is 70 Mbps with a delay of 50 ms.

For calculating the throughput of the MPT tunnel, only the results during the steady state were used. Fig. 4 shows the tunnel throughput as a function of different transmission speeds and delays with packet reordering enabled in MPT.

Our results show that the reordering mechanism of MPT is efficient, and the steady state throughput was decreased only to a small extent due to the delay applied for the second path. For example, when the second path had 40 Mbps speed, the effect of the decrease of the throughput of the tunnel was as follows: in the case of 10 ms, 20 ms, 30 ms, 40 ms, and 50 ms delay the throughput was reduced only by 1 Mbps, 2 Mbps, 5 Mbps, 6 Mbps, and 8 Mbps, respectively.

### *B. Without Packet Reordering Mechanism*

In the second scenario, the MPT has been implemented without the packet reordering mechanism by setting the value

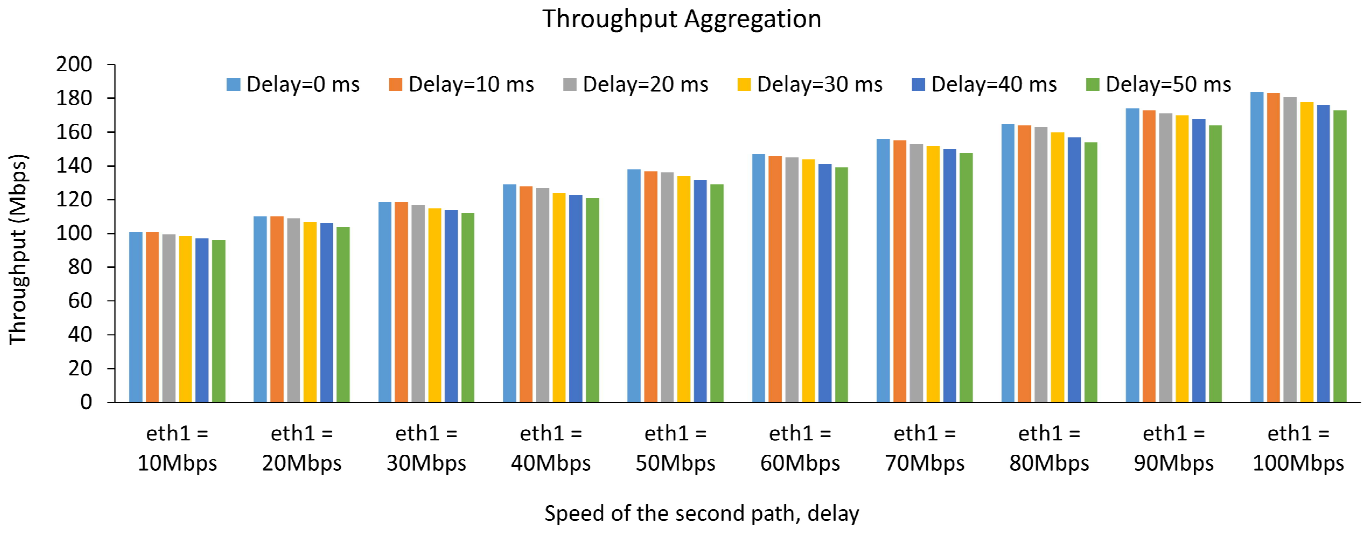


Fig. 4. The tunnel throughput as a function of transmission speed and delay with enabling the packet reordering in MPT.

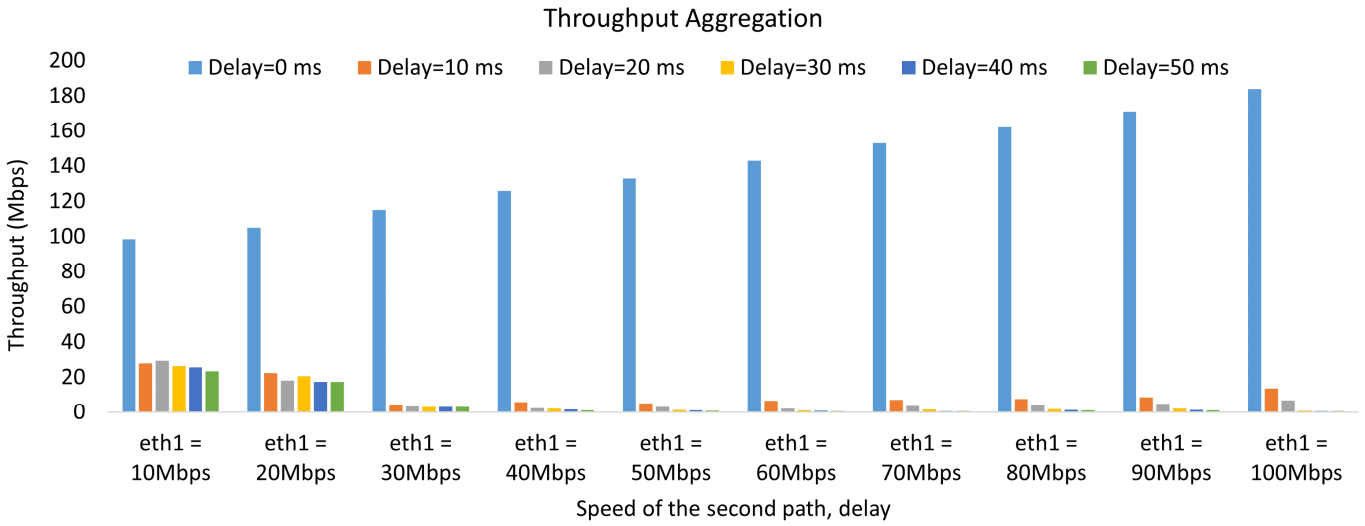


Fig. 5. The tunnel throughput as a function of transmission speed and delay without enabling the packet reordering in MPT.

of reorder window parameter to zero. As it is apparent from the results Fig. 5, introducing the delay negatively affected the throughput of the tunnel. For example, when the second path had 10 Mbps speed, the effect of the delay on the throughput of the tunnel was severe, and the throughput in case of 10 ms, 20 ms, 30 ms, 40 ms, and 50 ms delays was reduced by 70.6 Mbps, 69.1 Mbps, 72 Mbps, 72.7 Mbps, and 75.1 Mbps, respectively.

### *C. Effect of the Reorder Window Parameter*

We have also examined how the value of the reorder window parameter affects the effectiveness of the reordering mechanism of MPT and thus also its throughput aggregation capability. The transmission speeds were 100 Mbps and 50 Mbps for the first path and second path, respectively. As for the value of the delay we kept the previous values (10 ms, 20 ms, 30 ms, 40 ms, and 50 ms). As for the value of the reorder window parameter values, we tested a parameter series that increased exponentially (they were 25, 50, 100, 200, 400, 800, and 1600) to be able to cover a wide range by performing a relatively low number of tests.

Our results in Table I show the effect of the reorder window parameter: if the reorder window size is too small, then the throughput aggregation is inefficient. The throughput

TABLE I. EFFECT OF THE REORDER WINDOW PARAMETER ON MPT THROUGHPUT AGGREGATION.

| Reorder window size | Delay= 10 ms | Delay= 20 ms | Delay= 30 ms | Delay= 40 ms | Delay= 50 ms |
|---|---|---|---|---|---|
| 25 | 15.3 | 7.43 | 4.98 | 3.83 | 2.95 |
| 50 | 28.1 | 13.9 | 9.88 | 8.69 | 5.54 |
| 100 | 55.4 | 26.5 | 17.9 | 14.2 | 11.2 |
| 200 | 137 | 65.6 | 35.7 | 25.6 | 21.1 |
| 400 | 137 | 136 | 134 | 62.1 | 41 |
| 800 | 137 | 136 | 134 | 132 | 130 |
| 1600 | 137 | 136 | 134 | 132 | 129 |

TABLE II. THE LENGTH OF THE TRANSIENT PERIOD.

| Reorder window size | Delay= 10 ms | Delay= 20 ms | Delay= 30 ms | Delay= 40 ms | Delay= 50 ms |
|---|---|---|---|---|---|
| 25 | 0 | 0 | 0 | 0 | 0 |
| 50 | 0 | 0 | 0 | 0 | 1 |
| 100 | 0 | 0 | 0 | 1 | 2 |
| 200 | 1 | 1 | 1 | 2 | 3 |
| 400 | 1 | 2 | 3 | 3 | 4 |
| 800 | 1 | 2 | 3 | 5 | 7 |
| 1600 | 1 | 2 | 3 | 5 | 8 |

aggregation is efficient if the reorder window size is large enough. And further increasing the reorder window size beyond necessity does not have any effect. The appropriate choice of this parameter is one of our future research interests.

Our results in Table II show the length of the transient period, that is, the number of seconds during which the throughput (at the beginning of the measurement) was lower than the throughput in the steady state.

## VI. CONCLUSION

In this paper, the effect of the out-of-order arrival of the packets due to an additional delay at the second path and the effectiveness of the packet reordering mechanism of MPT have been extensively studied. We implemented two main scenarios: the packet reordering mechanism was enabled in the first scenario and disabled in the second one. We measured the efficiency of the throughput aggregation capability of MPT with different combinations of transmission speeds and delays applied to the second path.

In the first scenario, we experienced only a minor decrease of the throughput due to the delay applied to the second path, whereas in the second scenario, the tunnel throughput aggregation was affected significantly: the tunnel throughput deteriorated drastically due to the out-of-order packet arrivals. Thus, we proved the effectiveness of the packet reordering mechanism of MPT.

We have also examined how the value of the reorder window parameter affects the effectiveness of the reordering mechanism of MPT and thus also its throughput aggregation capability. We have found that if the reorder window size is too small, then the throughput aggregation is inefficient. The throughput aggregation is efficient if the reorder window size is large enough, and increasing the reorder window size beyond necessity does not have any effect. The appropriate choice of this parameter is one of our future research interests.

## ACKNOWLEDGEMENT

The measurements were carried out remotely using the resources of NICT StarBED, 2–12 Asahidai, Nomi-City, Ishikawa 923-1211, Japan. The authors would like to thank Shuuhei Takimoto for the possibility of using StarBED.

## REFERENCES

[1] A. Abilov, A. Chunaev, M. A. Lamri, and I. Kaisina, “Real-Time Video Streaming with Application Layer ARQ in UAV Networks: Field Tests,” in 2021 44th International Conference on Telecommunications and Signal Processing (TSP), 2021, pp. 324–328, doi: 10.1109/TSP52935.2021.9522615.

[2] Z. A. Polgar, “Reliable Data Transport Protocol with FEC Mechanism for Erasure Channels,” in 2021 44th International Conference on Telecommunications and Signal Processing (TSP), 2021, pp. 114–119, doi: 10.1109/TSP52935.2021.9522618.

[3] F. Fejes, S. Rácz, and G. Szabó, “Application agnostic QoE triggered multipath switching for Android devices,” in 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–7, doi: 10.1109/ICC.2017.7997450.

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

[5] A. Ford, C. Raiciu, M. Handley, C. Paasch, and O. Bonaventure, “TCP extensions for multipath operation with multiple addresses,” ETF RFC 8684, March 2020, doi: 10.17487/RFC8684.

[6] A. Alheid, D. Kaleshi, and A. Doufexi, “An analysis of the impact of out-of-order recovery algorithms on MPTCP throughput,” in 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, 2014, pp. 156–163, doi: 10.1109/AINA.2014.50.

[7] A. Alheid, A. Doufexi, and D. Kaleshi, “A study on MPTCP for tolerating packet reordering and path heterogeneity in wireless networks,” in 2016 Wireless Days (WD), 2016, pp. 1–7, doi: 10.1109/WD.2016.7461454.

[8] A. Frommgen, T. Erbshäußer, A. Buchmann, T. Zimmermann, and K. Wehrle, “ReMP TCP: Low latency multipath TCP,” in 2016 IEEE international conference on communications (ICC), 2016, pp. 1–7, doi: 10.1109/ICC.2016.7510787.

[9] B. Almási and S. Szilágyi, “Investigating the performance of the MPT multipath communication library in IPv4 and IPv6,” Int. J. Adv. Telecommun. Electrotech. Signals Syst., vol. 5, no. 1, 2016, pp. 53–60, doi: http://dx.doi.org/10.11601/ijates.v5i1.148.

[10] S. Szilágyi, I. Bordán, L. Harangi, and B. Kiss, “Throughput Performance Analysis of the Multipath Communication Technologies for the Cloud,” J. Electr. Electron. Eng., vol. 12, no. 2, 2019, pp. 69–72.

[11] G. Lencse and A. 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, pp. 41–48, doi: 10.11601/ijates.v4i2.112.

[12] S. Szilágyi, F. Fejes, and R. Katona, “Throughput performance comparison of MPT-GRE and MPTCP in the Fast Ethernet IPv4/IPv6 environment,” J. Telecommun. Inf. Technol., no. 2, 2018, pp. 53–59, doi:10.26636/jtit.2018.122817.

[13] G. Lencse, S. Szilagyi, F. Fejes, and M. Georgescu, “MPT Network Layer Multipath Library,” 2021, [Online]. Available: https://datatracker.ietf.org/doc/html/draft-lencse-tsvwg-mpt-09.

[14] N. A. Jabbar and G. Lencse, “Measurement and Analysis of MPT Multipath Throughput in Wire Channels,” in 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2022, pp. 1–5, doi: 10.1109/ICECCME55909.2022.9987956.

[15] A. El-Atawy, Q. Duan, and E. Al-Shaer, “A Novel Class of Robust Covert Channels Using Out-of-Order Packets,” IEEE Trans. Dependable Secur. Comput., vol. 14, no. 2, 2017, pp. 116–129, doi: 10.1109/TDSC.2015.2443779.

[16] F. Fejes, “MPT – multi-path tunnel,” precompiled version can be downloaded from http//github.com/spyff/mpt, 2019.

[17] N. Al-Imareen and G. Lencse, “MPT connections files,” 2023, [Online]. Available: https://github.com/NaseerAJabbar/MPT_Connections_files.

[18] M. Mortimer, “Iperf3 documentation,” 2018, [Online]. Available: https://iperf.fr/iperf-doc.php.