

Article



# Performance Analysis of MPT-GRE Multipath Networks Under **Out-of-Order Packet Arrival**

Naseer Al-Imareen \* D and Gábor Lencse D

Department of Telecommunications, Széchenyi István University, H-9026 Győr, Hungary; lencse@sze.hu \* Correspondence: al-imareen.naseer@sze.hu

Abstract: Network packets may arrive out of their original order due to network delays, transmission speed variations, congestion, or uneven resource distribution. These factors cause significant challenges to network performance. These challenges result in jitter, packet loss, and reduced throughput, negatively affecting the efficient arrangement of packets. The Multipath tunnel-Generic Routing Encapsulation (MPT-GRE) architecture addresses this issue through a packet reordering mechanism designed for multipath GRE with User Datagram Protocol (UDP) encapsulation networks. This study investigates and analyses the path-specific delays, jitter, and transmission speed constraints to evaluate the influence of out-of-order packets on the MPT-GRE tunnel throughput aggregation capability. By comparing scenarios with and without the re-ordering mechanism, the results demonstrate that the reordering mechanism substantially improves the traffic throughput in symmetric and asymmetric channel configurations. Additionally, the study emphasizes the critical role of optimizing the reordering window parameter for effective performance. These findings confirm that packet reordering mechanisms significantly enhance MPT-GRE network performance by reducing the negative effects of delays and out-of-order arrivals.

**Keywords:** multipath; MPT; GRE-in-UDP; throughput; jitter; delay

# 1. Introduction

The rapid evolution of technologies leveraging multiple network interfaces has motivated extensive research into optimizing multipath utilization. Devices like smartphones with 4G/5G and Wi-Fi connectivity, tablets, and laptops demonstrate the prevalence of various network interfaces in modern computing systems [1]. However, traditional Transmission Control Protocol (TCP)/Internet Protocol (IP) protocols are designed to operate over a single network interface, creating inefficiencies and limiting the potential of multipath communication [2]. This restriction prevents the simultaneous use of multiple pathways for data exchange, reducing efficiency and reliability and underscores the need for innovative multipath technologies. Quality of Service (QoS), and throughput aggregation are two key factors influencing the performance of IP network applications [3].

Multipath networking has revolutionized data transmission by enabling data streams to utilize multiple paths, enhancing bandwidth utilization. Multipath solutions allow for data division across various paths but also introduce challenges, notably out-of-order packet delivery. In multipath environments, differences in latency, bandwidth, and packet loss between paths often lead to packets arriving out of sequence. This can severely degrade application performance, especially in latency-sensitive scenarios like real-time video streaming, online gaming, and financial trading, where timely and ordered packet delivery is critical. Packet reordering can cause choppy video playback, gameplay lag,



Academic Editors: Christos I. Bouras and Dimitris Kanellopoulos

Received: 23 April 2025 Revised: 19 May 2025 Accepted: 22 May 2025 Published: 24 May 2025

Citation: Al-Imareen, N.; Lencse, G. Performance Analysis of MPT-GRE Multipath Networks Under Out-of-Order Packet Arrival. Electronics 2025, 14, 2138. https://doi.org/10.3390/ electronics14112138

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/ licenses/by/4.0/).

or transactional errors, significantly impacting user experience [4]. Efficient solutions to reduce the effects of out-of-order packet arrival are crucial for maximizing the benefits of multipath networking. Addressing these challenges enhances performance and ensures the reliability and QoS required for modern real-time applications.

Multipath networks are significant because they appear more resilient in the face of interruptions and ensure data arrival through multiple paths, thus ensuring fault tolerance. Multipath solutions ensure data transfer without interruption, even in the event of congestion or the unavailability of a path. The network can redirect traffic to alternative channels in case of a single-channel failure, ensuring uninterrupted operation. Studies have shown that networks that operate according to multipath solutions, such as Multipath GRE-in-UDP Technology (MPT) [5] and Multipath TCP Technology (MPTCP) [6], are two potential alternatives for conducting multipath communications. Traffic allocation and network resources are distributed more equitably, reducing congestion on individual paths, improving overall network performance, and lowering latency. Implementing multipath technology significantly enhances network reliability and efficiency.

One of the main challenges in MPT-GRE multipath networks is handling the out-oforder arrival of packets, which occurs due to the heterogeneous nature of path characteristics such as delay, jitter, and bandwidth asymmetry. This disorder severely impacts the performance of real-time applications by causing buffer overflows, playback issues, and inefficient retransmissions. While MPT aims to aggregate throughput across multiple paths, the reordering problem introduces inefficiencies that can negate these benefits. Existing solutions often fail to adapt to varying network conditions dynamically or require significant buffering, which is unsuitable for latency-sensitive applications like live video streaming. Therefore, there is a need for more efficient and adaptive packet reordering mechanisms that can maintain throughput while minimizing delay and loss.

This paper investigates the issue of out-of-order packet arrival, which can significantly degrade the performance of MPT-GRE multipath networks. This challenge is especially critical for real-time communication applications where maintaining consistent throughput and minimizing packet loss are essential. While MPT-GRE networks offer improved throughput aggregation, packet reordering often compromises their performance, particularly in asymmetric or congested network paths.

To address this challenge, the research focuses on a central question: How does the packet reordering mechanism in the MPT-GRE network affect tunnel throughput aggregation under varying network conditions such as delay, jitter, and transmission rate differences? Specifically, this study examines the behavior of MPT-GRE with and without the reordering mechanism across symmetric and asymmetric path configurations. Additionally, the impact of reordering window tuning on overall throughput performance is analyzed, offering insights into how this mechanism can mitigate the adverse effects of out-of-order packet arrival and improve network performance.

The primary objectives of this work are to (1) Improve the performance of MPT by enabling a packet reordering mechanism for MPT-GRE multipath networks and (2) Evaluate its impact on key performance metrics such as throughput, latency, and packet loss under various network conditions. We supplied a network setup to evaluate the performance of MPT-GRE tunnel throughput aggregation when packets are subject to out-of-order arrival. Specifically, one route maintained a steady transmission speed and did not experience any delays, while the other route had unique bandwidths and delays. Our investigation's main goal was to determine how the packet reordering mechanism in the MPT-GRE library while packets arriving out of sequence affects the tunnel throughput of the MPT-GRE library. Additionally, a study was carried out to evaluate the tunnel throughput and determine the efficacy of the out-of-order mechanism. The main goal was

to establish how tuning the reorder window size impacts throughput improvement. The findings demonstrated that the optimal packet window size was determined according to the specified delay conditions.

The main contributions of this paper are as follows:

- Comprehensive Performance Evaluation: This study offers a detailed analysis of the performance of the MPT-GRE multipath network by systematically measuring tunnel throughput under different network conditions. A comparative assessment was conducted both with and without the activation of the packet reordering mechanism. The findings indicate a significant improvement in network performance when the reordering mechanism is enabled, underscoring its essential role in enhancing throughput efficiency.
- Reorder Window Parameter Evaluation: The research included experiments with various sizes of reorder window parameters. It was found that an appropriately sized reorder window parameter, in combination with the packet reordering mechanism, further improves network performance. This contribution provides valuable insights into optimal parameter settings.
- Impact of Jitter and Delay on Measurements: This analysis examines the effects of jitter and delay while calculating the round-trip time (RTT) statistics to validate the results.

The rest of this paper is structured as follows: Section 2 presents an overview of MPT, Section 3 examines related literature, and Section 4 provides detailed insights into packet reordering within the MPT-GRE framework. The Section 5 describes the experimental settings, including the hardware environment and MPT-GRE configurations. Section 6 presents several tests and experimental findings associated with MPT-GRE. The final Section discusses the conclusions drawn from this study and future research directions.

# 2. MPT- GRE Network Layer Multipath

#### 2.1. MPT-GRE Library

The primary purpose of the MPT library is to create a flexible and robust multipath communication framework that can aggregate the capacities of multiple paths to ensure reliable data transmission, allowing the use of different IP versions (IPv4 or IPv6) independently on the tunnel and physical layers.

The MPT-GRE software enables data transmission across multiple routes using the Generic Routing Encapsulation (GRE) protocol. The project was developed and upheld by a group of researchers from Debrecen University [7]. It is free and open-source and available on [8]. MPT works at the network layer and, taking advantage of the characteristics of the GRE-in-UDP protocol, creates a tunnel between the client and the server. Additionally, MPT can act as a router, efficiently routing the packets across multiple networks and tunnel endpoints, thus providing reliable communication between sites. Unlike MPTCP and Huawei GRE Tunnel Bonding Protocol, MPT uses a different approach for multipath networks. MPT-GRE technology helps improve network speed by almost doubling symmetric network throughput. It addresses the benefits of creating logical connections, often called tunnels, to facilitate communication between clients and servers and to enable multiple channels to be used simultaneously. It encapsulates data into MPT-GRE packets, which are then sent over one of several available paths for the data stream to its destination, using multiple interfaces to ensure faster media flow. Subsequently, the MPT-GRE library examines incoming packets (IPv4 or IPv6) received at the tunnel originating from the first node. Within the User Datagram Protocol (UDP) component, transmission involves creating a novel Generic Routing Encapsulation (GRE) through an alternate physical route [9,10]. MPT-GRE system has packet reordering mechanisms that rearrange packets before delivery to ensure the accuracy and consistency of transmitted data, ultimately enhancing the

network's reliability and efficiency. Figure 1 illustrates the MPT-GRE layer architecture. The MPT-GRE software enables the direct implementation of the tunnel interface onto the physical interfaces.

The multiple-layer structure illustrates MPT-GRE's intricate architecture. MPT-GRE refines the initial concept of GRE-in-UDP by accommodating multiple physical interfaces. While sharing similarities with Multipath TCP (MPTCP), MPT distinguishes itself by employing UDP as the foundational layer, extending the functionality of GRE in UDP. This enhancement establishes an IP tunnel layer that facilitates support for Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). The congestion control mechanism can utilize the control interface of the MPT 64-bit version software library to influence and modify the attributes and parameters of the pathways. For instance, a path experiencing congestion can be deactivated or activated after it returns to an uncongested state [10–12]. Furthermore, the allocation of traffic across several paths can be adjusted based on the network infrastructure's current condition using the control protocol.



Figure 1. The MPT-GRE layer architecture [13].

Figure 2 depicts the MPT software component in action, allowing for packets' transmission and reception through a defined mechanism. Initially, the MPT application software dispatches an IP packet to the tunnel interface, which is subsequently intercepted by the MPT software. The MPT software examines the packet to identify the proper connection specifications, which are the foundation for MPT's multipath communication. When two MPTSRV nodes connect from the sender and receiver ends, a UDP tunnel is created to transfer packets between them. For instance, a host with Ethernet and Wi-Fi interfaces can create numerous MPT connections with a remote host. Each connection can use different paths, which facilitates traffic distribution. These connections circumvent physical interfaces and correspond to different communication sessions established based on host limitations, such as File Transfer Protocol (FTP) downloads.



Figure 2. The theoretical process of the MPT mechanism [13].

Essential connection requirements for efficient real-time packet routing may include peer IP addresses or port numbers, enabling allocation on a per-packet or flow basis. Communication parameters can be changed using control protocol instructions while the system runs. When a packet is received from the tunnel interface, the MPT software coordinates its transmission to the remote physical interface according to the connection's specified state and distribution parameters. Real-time evaluation of connection metrics determines path selection, followed by encapsulating the packet into a GRE in a UDP segment that adheres to GRE header standards.

Given that out-of-order packet delivery is a significant challenge in multipath networks, the MPT-GRE library allows us to precisely test the impact of various reorder window parameter sizes and maximum buffer delay. These evaluations align with our research objectives, which aim to optimize these parameters to enhance network efficiency and ensure that data are delivered sequentially, thereby improving overall network performance.

In contrast, the packet reception process is relatively straightforward: when packets reach the physical interface, they are examined to ensure their compliance with the specified GRE-in-UDP port number [5]. If relevant, the MPT software conducts the following validation operations: verifying the connection's legitimacy, checking the GRE sequence numbers, and confirming the GRE Key values. Successfully validated packets are then forwarded to the Transport and Application Layers via the tunnel interface for further processing. Figure 3 shows the encapsulation of the various protocol data segments.

|--|

Figure 3. Protocol Data Unit (PDU) architecture for MPT communication [13].

#### 2.2. Packet Reordering of MPT-GRE

The packet reordering mechanism in MPT-GRE is designed to maintain the correct sequence of packets in multipath networks by utilizing GRE sequence numbers and bufferbased reordering. Selecting parameters such as maximum buffer delay and reorder window size is critical in balancing performance metrics such as throughput and latency. Variations in channel delays within MPT-GRE environments may cause packets to arrive out of sequence due to changes in channel latency during transmission. Packet reordering becomes apparent in multipath tunneling environments, where diverse paths introduce variable delays. The MPT-GRE multipath network incorporates an optional mechanism to ensure proper packet sequencing, addressing this challenge. This elective element within the MPT-GRE environment facilitates the accurate sequencing of packet transmission for effective tunnel communication. Upon activation, the recipient accumulates incoming packets (initially unordered) within a buffer array, subsequently arranging them according to GRE sequence numbers to guarantee precise transmission to the receiver's tunnel interface.

The buffer delay parameter sets the maximum time a packet can be held in the buffer. In contrast, the reorder window parameter specifies the size of the reorder buffer array depending on the GRE sequence numbers. The packets are sorted according to the GRE sequence numbers, ensuring the ordered transmission to the receiver's tunnel interface. The reorder window parameter defines the buffer size that temporarily stores out-of-order packets from different paths until they are reassembled in the correct sequence for delivery. The reorder window size must be carefully adjusted to accommodate all packets arriving within the maximum buffer delay interval, accounting for the highest possible data rate across all active paths. If the reorder window is too small, it may overflow, leading to packet drops and degraded network performance. The size of the reorder window is measured in a number of packets, as it defines the length of the buffer array used by the MPT receiver to temporarily store and sort incoming out-of-order packets based on their GRE sequence numbers. A value of 0 disables reordering, while the maximum configurable size is 10,000 packets.

However, expanding the reorder window beyond the necessary limit provides no additional benefit. A properly sized reorder window enables the system to handle out-of-order packets efficiently, ensuring seamless and in-order delivery. This efficient operation is essential for maintaining high throughput and reliability in multipath network environments.

These two parameters govern the reordering procedure. If packet loss occurs before evaluation, particularly when packets are delayed in the buffer for the designated time, the packet is transmitted to the tunnel interface without waiting for additional data. MPT-GRE does not extend its waiting time when data are deemed lost or not received.

In adherence to the preservation of ordered delivery, incoming packets are directed to the logical interface according to their GRE sequence numbers, ensuring continuity even when packets are lost. However, if the buffer's maximum delay is set too low and the next sequence number fails to arrive, MPT-GRE may mistakenly classify it as lost. In that case, MPT-GRE may erroneously classify it as lost and remove it to maintain correct order delivery. Packet reordering is prevalent within the Internet infrastructure and is managed inconspicuously by user and application-level processes. Conversely, if the maximum buffer delay is set too high, delayed packet loss is detected too late, compromising communication performance. To address this problem, the reorder window size must be sufficiently large to accommodate packets arriving at the maximum line rate across all operational paths within the selected connection while adhering to the pre-scribed longest possible buffer delay [13].

#### 3. Related Work

Investigating packet reordering is necessary for multipath network research due to its impact on network performance and user experience. Previous studies have identified the challenges and proposed various techniques to manage reordering, such as adaptive algorithms and buffer management. Gao, Ling, and Yuan examined the impact of packet reordering through simulation-based multipath experiments using various aggregating paths [14]. Additionally, an analytical model was introduced to elucidate the fundamental behavior of packet reordering in Concurrent Multipath Transfer (CMT). Through comparative analysis experiments, the model acknowledged the inevitability of packet reordering.

Multipath technologies optimize the way packets are transmitted on different network paths. These technologies help increase network capacity and improve data transfer efficiency. However, the utilization of these technologies introduces additional delays in various ways.

Congestion control and packet reordering algorithms can be combined in multipath networks to improve network performance and throughput [15,16]. The advantages of MPTCP and MPT-GRE systems include providing continuous communication between the two ends of the network, improving user experience, and increasing communication efficiency. This innovation helps use multiple channels to transfer data in a balanced distribution load and improves the network's overall performance. These benefits improve performance and congestion management in networks.

MPT-GRE is an important step to ensure high-performance multipath networks in telecommunications networks. Increased performance in load-balancing GRE traffic enables MPT-GRE to perform more efficiently at the network layer [17].

Zinner et al. conducted a performance evaluation of packet reordering in concurrent multipath transmissions for network virtualization [18]. Their study introduced an innovative approach to packet reordering to enhance overall network performance. The findings demonstrated a substantial improvement in the performance of transport virtualization for computer resources, particularly in reducing packet loss rates and enhancing throughput.

Some research has increasingly focused on addressing the challenges of out-of-order (OFO) packet delivery in multipath transmission. For instance, a notable contribution is the DRL-based OFO-Friendly Multipath Scheduling (DOFMS) framework, which employs a Double Deep Q Network (DDQN) to regulate path scheduling dynamically and introduces a novel OFO evaluation index to capture packet reordering phenomena better [19]. The DOFMS framework demonstrates a 25% reduction in OFO rate and a 4% improvement in bandwidth over conventional approaches, highlighting the benefits of combining reinforcement learning with dynamic multipath scheduling. While such learning-based methods offer promising improvements in delay-sensitive environments, our work differs in its focus on real-world implementations of MPT-GRE tunnels. Specifically, we investigate the impact of OFO arrivals under varied network conditions and evaluate a practical, parameterized packet reordering mechanism designed to restore transmission order and optimize throughput in MPT-GRE networks.

Almási and Szilágyi presented a study on the effectiveness of reaggregation in an MPT environment. They explicitly explore how the bandwidth aggregation feature of MPT can be applied to transport streaming media through data upload [20]. However, their findings revealed instances of packet reordering when discrepancies in the paths' bandwidths were present. Furthermore, the study observed an increased variance in one-second throughput compared to the single-path scenario, suggesting a potential correlation between elevated variance and packet reordering within the multipath environment.

Recent studies have explored multipath transmission to enhance performance in Remote Direct Memory Access (RDMA) networks, particularly within data center networks (DCNs). Multipath transmission, including packet spraying, has been shown to alleviate network congestion, enable load balancing, and improve overall efficiency. One such advancement is ORNIC, a high-performance RDMA Network Interface Card (NIC) designed for multipath transmission, which supports both in-order and out-of-order packet reception. ORNIC directly writes out-of-order packets to user memory without reordering and uses a dynamically allocated bitmap structure to ensure data integrity and detect packet loss. Researchers concluded that ORNIC handles out-of-order packets with minimal performance degradation, offering significant advantages over conventional NICs, which experience substantial throughput drops [21].

Also, in [1], a shared storage-based packet reordering and deduplication system (SSPRD) was proposed to improve multipath transmission over a Wide Area Network (WAN). SSPRD uses a dynamically utilized shared storage model to achieve optimal buffer usage among multiple sessions to counter the impact of out-of-order and duplicate packets in Software-Defined WAN (SD-WAN) networks. By packet classification based on their descriptors and a sub-buffer-based algorithm for handling timeouts, SSPRD manages lost or delayed packets efficiently. With the capability to support four concurrent sessions, it achieves a reorder buffer depth of 15,360 packets per session while conserving storage space by 62.5%. The system achieves a high-performance packet deduplication rate of 10 Gbps for 1500-byte packets and a packet reordering rate with significantly improved buffer efficiency compared to any traditional method.

Some studies leverage network intermediate devices to address out-of-order packet issues in multipath transmission. AFR-MPRDMA [22] implements a fast retransmission mechanism that detects congestion on slower paths and retransmits blocked packets via faster paths to enhance transmission efficiency. Additionally, the receiver dynamically adjusts buffer allocation to minimize latency and prevent unnecessary retransmissions. While AFR-MPRDMA reduces packet blocking and improves flow completion time by up to 61% compared to existing RDMA-based schemes, its requirement for the sender RNIC to track all in-flight packets results in high memory consumption, posing scalability challenges in large-scale multipath environments.

Ineffective packet scheduling across multiple paths can lead to degraded performance, sometimes making MPTCP perform worse than single-path TCP, necessitating advanced scheduling strategies to mitigate reordering [23–25]. Various reordering recovery mechanisms, such as TCP-DOOR and D-SACK, have been analyzed, demonstrating that their effectiveness depends on path delay differences, emphasizing the importance of selecting the proper mechanism to sustain high throughput [26]. Adaptive solutions, such as OLAPS and AEPS, dynamically adjust packet scheduling based on real-time network conditions, improving throughput while mitigating reordering in dynamic environments like mobile networks [24,25]. Additionally, queueing delay-aware methods, such as QDAPS, optimize packet spraying in data center networks by selecting paths based on queueing delays, ensuring in-order arrival and minimizing reordering effects [27].

S. Szilagy and I. Bordan [28] introduced MPT-GRE to utilize multiple network interfaces in parallel, validating the solution in controlled laboratory environments. The research extended the performance analysis to more realistic conditions using Dummynet WAN emulation software integrated into FreeBSD 13.1-RELEASE. The study focused on the impact of various network parameters—including packet delay, jitter, and packet loss—on file download speed, time, and CPU utilization in dual-path Fast Ethernet and Gigabit Ethernet IPv4 WAN emulated environments. Results showed that while MPT-GRE efficiently aggregated performance in Fast Ethernet scenarios, its performance in Gigabit Ethernet environments was significantly lower, especially under minimal network impairments. Comparative measurements against MPTCP demonstrated that although MPTCP outperformed MPT-GRE, both solutions struggled under the emulated Gigabit conditions. The authors concluded that Dummynet may not be ideal for evaluating multipath Gigabit Ethernet environments and highlighted the need for further optimization of MPT-GRE to improve its efficiency and robustness under higher-capacity network conditions.

Despite differing application domains, the MPT architecture and the integrated predictor observer feedback control approach share a fundamental objective: mitigating the adverse effects of delay to maintain system stability and performance. While the control strategy addresses unbounded input time delay in spacecraft systems through predictive and observer-based compensation [29,30], MPT deals with delay-induced packet reordering in multipath IP networks using buffering and reordering mechanisms. In both cases, the systems adopt adaptive techniques to handle dynamic environments and ensure reliable output, reflecting a shared emphasis on delay-aware design for real-time responsiveness.

Also, some previous papers primarily examined the impact of delay, jitter, and packet loss on throughput performance, download time, and CPU utilization using MPT-GRE, highlighting general performance degradation. However, they did not specifically address how packet reordering mechanisms could mitigate these issues in multipath scenarios. Additionally, some studies have explored packet reordering using MPTCP. This study builds on that work by focusing on the relationship between path-specific delays, jitter, and throughput in MPT-GRE networks, with particular emphasis on optimizing packet reordering mechanisms and reorder window parameters. It demonstrates that enabling reordering strategies can significantly improve tunnel throughput, even under asymmetric and degraded network conditions. This research is unique in its in-depth study of packet reordering mechanisms, an area not extensively explored in previous studies.

Several existing studies have explored packet reordering and its effects, mainly in the context of general multipath protocols or traditional TCP. While these studies provide a broad understanding of the effects of reordering, they often need more detailed analysis specific to newer or less popular protocols such as MPT-GRE. This study addresses the abovementioned gap by focusing on the MPT-GRE library, which combines GRE and UDP encapsulation to provide a new approach to multipath communications. By evaluating MPT-GRE's handling of packet reordering, this research offers a new understanding of the performance and improvement of this specific protocol performance.

MPT stands out by integrating multipath transmission at the network layer. It uses GRE tunneling to efficiently aggregate throughput across multiple paths depending on its packet reordering mechanism without the need for complex scheduling or extensive packet tracking. Unlike solutions like ORNIC and AFR-MPRDMA, which focus on NIC or sender-side optimizations, MPT reduces memory overhead and scalability issues. While MPTCP and other methods also struggle with path delay differences and packet reordering, MPT provides an efficient mechanism for arranging packets, especially with real-time delays and jitters.

# 4. Experiment Setting

We used the StarBED resources provided by Japan's National Institute of Information and Communications Technology (NICT) called StarBED. StarBED allowed us to establish a thorough assessment and measurement system, as shown in Figure 4. Our setup included three DELL servers, the PowerEdge R650 series was sourced from Dell Technologies, Round Rock, TX, USA, featuring each two Intel Xeon Gold 6330N 2.2G CPUs, 512GB 3200MT/s, dual rank DDR4 SDRAM, 480GB SSD, and four 25GbE Network Interfaces.

The experimental design was carefully prepared to evaluate the performance of the MPT-GRE multipath network, with a particular focus on packet reordering mechanisms and their impact on throughput and network efficiency. Our experiments were conducted using two operating systems: Debian 11.7 and Ubuntu 22.04 GNU/Linux with the default congestion control algorithm in both, which is Cubic. The choice to use Ubuntu 22.04 and Debian 11.7 was primarily driven by technical compatibility and experimental consistency with the MPT-GRE library, which is developed and maintained in a Linux environment. Ubuntu and Debian are widely used, stable Linux distributions that offer robust networking support, kernel-level customization, and access to low-level network stack features and capabilities essential for accurately measuring and evaluating tunnel

throughput, reordering behavior, and TCP performance metrics. The 64-bit version of MPT was implemented. The experiments varied the size of the reorder window parameter to evaluate its effect on network performance. The reorder window size is critical as it determines the buffer capacity for out-of-order packets and directly influences the efficiency of the packet reordering mechanism. Experimenting with different reorder window sizes helps identify the optimal balance between performance and resource utilization. The MPT source file used in this study was mpt-gre-lib64-2019.tar.gz, which can be obtained from the developer's GitHub page [8].



Figure 4. The Measurement MPT-GRE Network Topology.

A previous publication comprehensively detailed the MPT configuration file settings. For this current work, the configuration files have also been made accessible on GitHub [31].

#### 5. Experiments with Packet Reordering

We conducted several scenarios to investigate the influence of packet reordering on MPT-GRE and determine the optimal value for the reorder window size. An effective evaluation of tunnel throughput is crucial to better understanding MPT-GRE operation. Multiple scenarios were implemented to compare the tunnel throughput when the packet reordering mechanism was enabled (and the extent to which it could restore out-of-order packets) versus when it was disabled. The iperf3 tool was employed for real-time network throughput measurement [32] with its default configuration, which corresponds to a single TCP flow and packet size of 1460 bytes, SACK is enabled by default, and the timeout value = 3 s. In this context, throughput signifies the volume of packets transmitted each second through the tunnel. The source node operations involved the utilization of the following command:

iperf3 -c 10.0.0.3 -t 100 -f M

A Python 3.10 script was employed to organize and execute the experiments over 100 s. This script not only facilitated the systematic execution of experiments but also automated the storage of tunnel throughput results. The use of scripting streamlined the experimental process and improved the reliability and repeatability of our measurements. This meticulous approach, involving automated data collection, ensures a consistent examination of the impact of packet reordering on MPT-GRE and helps determine the optimal reorder window size. The MPT-GRE assessment involved examining it under varying conditions of asymmetric paths at speeds ranging from 100 Mbps to 1000 Mbps. Appropriate weight settings for the second path were configured according to established parameters. On the server end, the initiation of IPerf3 in server mode was carried out using the following command line:

iperf3 -s -f M

An additional server was utilized to oversee the fluctuating delays associated with bypassing traffic. The experiment involved testing with diverse delay configurations, and the designated server functioned as a bridge by using the following content of the/etc/network/interfaces file to set up its network interfaces:

# The bridge interfaces iface enp152s0f0 inet manual iface enp152s0f1 inet manual auto br0 iface br0 inet static bridge\_ports enp152s0f0 enp152s0f1 address 10.2.2.6 broadcast 10.1.1.255 netmask 255.255.255.0 The following commands were used: brct1 addbr br0 brct1 addif br0 enp152s0f0 enp152s0f1

The traffic control command (tc) was used to add the delay on the middle server (x047): *tc qdisc add dev enp*152*s*0*f*0 *root netem delay <x> ms* 

The set delay values <x> were configured as 1 ms, 2 ms, 3 ms, 4 ms, and 5 ms, respectively. The selected delay values (1–5 ms and 10–50 ms) were chosen to simulate a representative range of real-world network conditions, from low-latency environments such as data centers or local networks (1–5 ms) to higher-latency wide area networks or cross-continental links (10–50 ms). This range allows for a comprehensive evaluation of the MPT-GRE tunnel's performance under varying delay scenarios and reflects both typical and challenging conditions faced by multipath communication systems. The inclusion of gradual increments enables the observation of throughput trends and the impact of delay asymmetry on the effectiveness of the packet reordering mechanism. This delay was applied to both network interfaces of the intermediary server within the second path, resulting in the intentional delay of packets in both transmission directions. Transmission speeds on both ends of the communication channel were regulated using the Linux traffic control command (tc) tool. The transmission speed values, denoted as variable y, range from 100 Mbps to 1000 Mbps and have been configured in increments of 100 Mbps.

tc qdisc add dev eno152s0f0 root tbf rate <y> mbit burst 32kbit latency 100ms.

There is one more important condition that needs to be mentioned. To enhance network performance and resource utilization by distributing incoming network packets across multiple CPU cores, Receive-Side Scaling (RSS) was enabled on both interfaces in Ubuntu 22.04 using the following commands:

ethtool -N enp152s0f0 rx-flow-hash udp4 sdfn

ethtool -N enp152s0f1 rx-flow-hash udp4 sdfn

However, one of the established nodes, the Debian 11.7, Linux distribution, does not support RSS settings with the Intel E810-XXV network interfaces on the servers.

In the subsequent experiments, the tunnel throughput was measured to substantiate the efficacy of the packet reorder mechanism of MPT-GRE. The optimal size of the reorder window parameter was determined in two scenarios. The Debian 11.7 Linux system was utilized in the first scenario without activating RSS. Transmission speeds were set to 100 Mbps for the first path and ranged from 10 Mbps to 100 Mbps for the second path, with delays varying from 10 ms to 50 ms in 10 ms increments. In contrast, the Ubuntu 22.04 Linux system was used with RSS activated and with transmission speeds ten times higher than the first case: 1000 Mbps for the first path and ranging from 100 Mbps to 1000 Mbps for the second path, with delays of {1, 2, 3, 4, 5} ms. The experimental parameters used

for evaluating the MPT tunnel throughput under various conditions are summarized in Table 1.

	Parameters							
System	Transmission Speeds	Packet Reordering Mechanism	RSS	Reorder Window Sizes	One-Way Delay Values	Jitter Values	Combined Delay/Jitter Conditions	
Ubuntu 22.04	100 Mbps to 1000 Mbps (with 100 Mbps increments)	Enabled and Disabled	Enabled	25, 50, 100, 200, 400, 800, 1600, 3200, 6400, 10000	{10, 20, 30, 40, 50} ms	{2, 4, 6, 8, 10} ms	{10/2, 20/4, 30/6, 40/8, 50/10} ms	
Debian 11.7	10 Mbps to 100 Mbps (with 10 Mbps increments)	Enabled and Disabled	Disabled	25, 50, 100, 200, 400, 800, 1600, 3200, 6400, 10000	{1, 2, 3, 4, 5} ms	-	-	

Table 1. Summary of Experimental Parameters for MPT Tunnel Evaluation.

#### 5.1. Efficiency Analysis with Packet Reordering Mechanism

In this experiment, we enabled the packet reordering mechanism by setting the reorder window parameter to 10,000 and the maximum buffer delay value to 300 ms. Upon evaluating the performance of the MPT-GRE on a Debian Linux system by measuring tunnel throughput, we observed an initial transient period followed by stabilization.

Figure 5a illustrates a sample of results for the initial 20 s, showcasing the transient period at a transmission speed of 70 Mbps and a delay of 50 ms using the Debian system.





Figure 5b illustrates a transient state from the initial 20 s interval, elucidating the transient period of measurement at a bandwidth of 700 Mbps with a delay of 5 ms using the Ubuntu system. It should be noted that when tunnel throughput was measured on the Ubuntu system, the transmission speed was ten times higher than on the Debian system.

The results presented in Figure 5b indicate that the initial second was deliberately adjusted, making the transient period shorter compared to Debian, as in Figure 5a. Latency issues and an increased likelihood of data loss cause transient periods in the MPT network. As a result, these issues lead to reduced tunnel throughput and a clear decrease in the overall performance of the MPT-GRE network. Solving these challenges requires implementing congestion control algorithms, which are carefully applied to evaluate their efficiency.

The differences in transient behavior between Debian and Ubuntu, as shown in Figure 5, warrant closer examination. The variation in both duration and severity of these transient periods likely arises from the one-order-of-magnitude difference in their trans-

mission speeds. Other factors may also contribute, including disparities in kernel versions, default network settings, driver performance, and scheduling mechanisms between the two operating systems. Ubuntu, with its more recent kernel and enhanced support for high-speed interfaces, may achieve stable throughput more quickly. In contrast, Debian's potentially more conservative configurations tend to result in extended transient phases.

The practical impact of packet reordering is particularly significant in latency-sensitive applications such as multimedia streaming and online gaming. Reordering can lead to playback interruptions, audio-video desynchronization, or lag, all of which negatively affect the user experience. In gaming scenarios, even minimal misordering may introduce jitter, delayed feedback, or inconsistencies that compromise performance. Implementing an effective packet reordering strategy, as explored in this study, not only supports sustained tunnel throughput but also contributes to smoother, more dependable application behavior in real-world conditions.

Figure 6 illustrates the effect on the tunnel throughput and the relationship between different transmission speeds and delays in the multipath MPT-GRE environment on Ubuntu 22.04, specifically when the packet reordering mechanism is enabled; the results indicate the efficacy of reordering mechanisms in Multipath MPT-GRE, demonstrating that the stable throughput experienced only a marginal reduction attributable to the delay introduced in the other path. For instance, with the second path operating at 500 Mbps, no noticeable impact on tunnel throughput was observed with 1 ms and 2 ms delays. However, with delays of 3 ms, 4 ms, and 5 ms, the throughput experienced only a nominal reduction of 1 Mbps. This result confirms that the chosen reorder window size was appropriate, as it effectively mitigated the impact of delays on throughput while maintaining stability.



**Figure 6.** Tunnel throughput: various bandwidths and delay with MPT-GRE packet reordering enabled under Ubuntu 22.04.

The experimental data highlight the effectiveness of the packet reordering mechanism in the multipath MPT-GRE environment. Delays only result in small throughput decreases, validating the chosen reorder window size as optimal and ensuring throughput stability.

The packet reordering mechanism shows high efficiency by maintaining throughput despite delay variations. The multipath MPT-GRE environment is scalable, accommodating various transmission speeds without significant performance degradation. The setup's stability under different delay conditions confirms its reliability. Therefore, the multipath MPT-GRE environment with an enabled packet reordering mechanism provides stable, efficient, and reliable throughput despite transmission delays.

Figure 7 depicts the tunnel throughput to diverse transmission speeds, 100 Mbps for the first path and 10 Mbps to 100 Mbps for the second path, with delays of 10 ms to 50 ms, showcasing the efficiency of the packet reordering mechanism when activated in multipath MPT-GRE within the Debian 11.7 environment. Additionally, the results show that the reordering process in MPT-GRE is effective. The reorder window size chosen for the experiment was appropriate, ensuring system stability and mitigating the negative effects of delay on tunnel throughput.



**Figure 7.** Tunnel throughput: various bandwidth and delay with MPT-GRE packet reordering enabled under Debian 11.07.

Both figures demonstrate that the packet reordering mechanism effectively maintains throughput stability. There are only slight decreases in throughput as delays increase. However, Figure 7 shows a more significant impact of delays due to the broader range of delays tested. Additionally, while Figure 6 focuses on higher transmission speeds, Figure 7 provides insights into the mechanism's performance at lower speeds. In summary, the figures demonstrate the robustness of the packet reordering mechanism across different speeds and delay conditions, ensuring stable and efficient throughput in multipath MPT-GRE environments.

As expected, an adequately configured reorder window size positively impacts throughput. Proper window sizes enable the system to handle more out-of-order packets, which reduces retransmissions and maximizes throughput. This outcome validates our initial hypothesis that optimizing the reorder window can effectively increase the amount of data successfully transmitted across multiple paths.

#### 5.2. Efficiency Analysis Without Packet Reordering Mechanism

We disabled the packet reordering mechanism during the implementation of the MPT-GRE by setting the reorder window size to 0.

The results in Figure 8 show the adverse impact of introducing delays on tunnel throughput in Ubuntu 22.04. For instance, when the first path operated at 1000 Mbps and the second path at 100 Mbps, the measured tunnel throughput under nominal conditions (without delay) was 1020 Mbps. In contrast, introducing delays had a pronounced negative effect on tunnel throughput. Specifically, with delays of {1, 2, 3, 4, 5} ms, the throughput experienced reductions to 156 Mbps, 95.3 Mbps, 66.4 Mbps, 57.9 Mbps, and 54.3 Mbps, respectively. This finding underscores the substantial impact of delays on tunnel effectiveness and highlights the importance of the packet reordering mechanism in mitigating these



adverse effects. The noticeable drop in performance shows how even minor delays can significantly reduce tunnel throughput.

**Figure 8.** Tunnel throughput: transmission speed and delay without MPT-GRE packet reordering enabled under Ubuntu 22.04.

The results also showed decreased combined tunnel throughput in the Debian 11.7 environment. Figure 9 demonstrates that, like the observations in the Ubuntu 22.04 environment, introducing delays in the Debian setup led to a noticeable reduction in tunnel throughput. This decline highlights the importance of packet reordering mechanisms in multipath MPT-GRE configurations. The consistent performance degradation across different operating systems highlights the universal challenge posed by delays in network throughput aggregation and emphasizes the need for efficient reordering strategies to ensure optimal performance in various environments. The findings from Figures 6–9 highlight the critical role of robust packet reordering mechanisms in maintaining stable and efficient tunnel throughput under varying delay conditions. In Figure 9, the 10 ms delay does not significantly affect throughput in the case of 10 Mbps and 20 Mbps transmission speeds because the MPT-GRE environment, when equipped with a properly configured packet reordering mechanism, is capable of effectively handling out-of-order packets introduced by such moderate delays. In this scenario, the buffer size, reorder window, and maximum buffer delay parameters are likely sufficient to compensate for the increased delay, preventing packet drops and minimizing disruptions in packet sequencing. As a result, the system maintains stable throughput despite the presence of a 10 ms delay.



**Figure 9.** Tunnel throughput: transmission speed and delay without MPT-GRE packet reordering enabled under Debian 11.7.

16 of 23

This paper quantifies the effects of increased delay and jitter on tunnel throughput within MPT-GRE networks by systematically introducing varying levels of these factors across multiple paths. The results clearly demonstrate how packet reordering mechanisms can mitigate these negative effects, providing valuable insights for configuring reordering parameters in both symmetric and asymmetric network scenarios.

#### 5.3. Impact of the Reorder Window Size

The investigation evaluated how the reorder window size influences the efficacy of the reordering mechanism in MPT-GRE, affecting its ability to aggregate throughput. The designated transmission speeds for the first and second paths were set at 1000 Mbps and 500 Mbps, respectively. The delay values remained consistent with the previously established parameters {1, 2, 3, 4, 5} ms. To comprehensively assess the impact of varying reorder window size values, tests were conducted utilizing a parameter exhibiting exponential growth (25, 50, 100, 200, 400, 800, 1600, 3200, 6400, and 10,000, the maximum allowed value). This method efficiently covered multiple parameter values, ensuring a systematic exploration of the parameter space. The results depicted in Figure 10 offer valuable insights into the influence of the reorder window size on the aggregation of MPT-GRE throughput in the Ubuntu 22.04 environment. The analysis shows that insufficient reorder window sizes compromise the efficiency of throughput aggregation. Optimal efficiency is achieved when the reorder window size is sufficiently large. Significantly, any additional changes in the size of the ordering window beyond the specified criteria do not provide any further benefits.



**Figure 10.** MPT-GRE Throughput Aggregation Performance under Different Reorder Window Sizes and Delays (Ubuntu 22.04).

These results confirm the importance of selecting an appropriate reorder window size, particularly in high-delay conditions. The data support the recommendation that in the current case, the reorder window size should be chosen judiciously within the range of 800 to 10,000 to maximize throughput aggregation efficiency under varying conditions.

Figure 11 shows how the size of the reorder window and the delay significantly affect MPT-GRE throughput aggregation. The figure indicates a clear pattern: smaller reorder window sizes result in lower throughput, while larger window sizes lead to higher throughput levels. The MPT-GRE can aggregate packets more effectively with an appropriate window size, reducing overhead and improving transmission efficiency. Additionally, the results demonstrate that lower delays are associated with higher throughput, while higher delays result in decreased throughput. These results highlight the critical role of latency in packet transmission and acknowledgment in MPT-GRE. Lower delays enable faster data exchange and overall improved network performance.



**Figure 11.** MPT-GRE Throughput Aggregation Performance under Different Reorder Window Sizes and Delays (Debian 11.7).

Tables 2 and 3 present a comparison of the duration of the transient period of the throughput conducted on Debian and Ubuntu systems. The difference is the time interval, given in seconds, in which the initial throughput was smaller than the throughput in the stable state. The higher transmission speeds and lower delays resulted in a shorter transient period in the Ubuntu system compared to Debian.

<b>Reorder Window</b>	Value of the Delays						
Size	0 ms	10 ms	20 ms	30 ms	40 ms	50 ms	
25	0	0	0	0	0	0	
50	0	0	0	0	0	0	
100	0	0	0	0	0	1	
200	0	0	1	1	3	3	
400	0	1	2	2	3	3	
800	0	1	2	3	4	7	
1600	0	1	2	3	6	8	
3200	0	1	3	4	5	7	
6400	0	1	3	4	5	8	
10,000	0	1	3	4	5	7	

**Table 2.** The length of the transient period under Debian (disabled RSS<sup>1</sup>).

<sup>1</sup> Receive-Side Scaling (RSS) is a mechanism that distributes incoming network traffic across multiple CPU cores to improve processing efficiency and network performance on multicore systems.

Table 3. The length of the transient period under Ubuntu (enabled RSS).

Reorder Window	Value of the Delays						
Size	0 ms	10 ms	20 ms	30 ms	40 ms	50 ms	
25	0	0	0	0	0	0	
50	0	0	0	0	0	0	
100	0	0	0	0	0	0	
200	0	0	0	0	0	0	
400	0	0	0	0	1	1	
800	0	0	0	0	1	1	
1600	0	0	0	1	1	1	
3200	0	0	0	1	1	1	
6400	0	1	3	4	5	8	
10,000	0	1	3	4	5	7	

A comparison of Tables 2 and 3, which detail transient periods under Debian with disabled RSS and Ubuntu with enabled RSS, reveals several important observations regarding the influence of reorder window size and delay on network stability. Table 2, representing Debian, illustrates a consistent progression in transient period length with increasing reorder window size and delay values. Higher values in Table 2 indicate lengthy adjustment periods before system stability is achieved, especially at reorder window sizes of 1600 and above, where delay values of 40 ms and 50 ms lead to transient periods as long as 8 s. In contrast, Table 3 for Ubuntu shows generally shorter or non-existent transient periods across a range of reorder window sizes and delay values. Smaller reorder window sizes maintain stable operation with zero transient periods observed under all delay conditions, while more significant reorder window sizes introduce minimal transient effects, especially noted in the 40 ms and 50 ms delay categories.

This difference underscores the impact of operating system configurations, specifically the presence or absence of RSS, on network behavior and transient period dynamics.

#### 5.4. Impact of the Delay and Jitter

The results of experiments examining the effects of delay and jitter on tunnel throughput within the MPT-GRE multipath network have provided essential insights into network performance. Analysis of the results in Figure 12 shows that Throughput1 experiences high delay and jitter, with jitter constituting 20% of the delay, leading to reduced tunnel throughput. This reduction is evident as tunnel throughput decreases with increasing delay and jitter. In contrast, for Throughput2 in the same figure, when only jitter is present, the impact on throughput is less apparent, although there is still a decreasing trend as jitter increases. These findings suggest that jitter alone has less influence on tunnel throughput than when combined with delay.



Figure 12. Impact of the Delay and Jitter on the MPT-GRE Tunnel Throughput in symmetric paths.

The analysis reveals a complex interaction between delay and jitter. Delays directly affect packet traversal times, while jitter causes variability in packet arrival times, leading to packet reordering and increased loss, which reduces overall throughput.

The MPT-GRE library's reordering packet mechanism has proven effective in improving tunnel throughput aggregation, reducing the effects of delay and jitter. Minimizing delay and jitter is critical for QoS mechanisms to ensure acceptable throughput levels. Increased delay and jitter can have significant implications for reordering buffer size, potentially leading to buffer overflow and excessive latency. Experimental results in Figure 12 demonstrate that while jitter alone has a relatively smaller impact on throughput, its combination with high delay causes significantly greater performance degradation. The MPT-GRE reordering mechanism effectively tackles these challenges by enhancing throughput aggregation and managing delay and jitter.

## 5.5. Round-Trip Time (RTT) Statistics Summary

We conducted an experiment scenario using transmission speeds of 1000 Mbps on both paths. We introduced delay and jitter at the middle node, with delay values ranging from 10 ms to 50 ms and corresponding jitter values ranging from 2 ms to 10 ms. The jitter-to-delay ratio remained consistent at 20% for each scenario. These adjustments were made using the traffic control command (tc) on both network interfaces. We controlled the delay and jitter in the network using the netem module with the delay option and a normal distribution.

Table 4 presents the MPT-GRE tunnel's RTT Statistics Summary. The RTT statistics summary for the MPT-GRE tunnel helped us understand the impact of delay and jitter on network latency. As delay and jitter increased, the RTT values increased and became more variable. For example, with delay set at 10 ms and jitter at 2 ms, the RTT statistics (min, average, max, standard deviation) were 13.592, 19.884, 28.069, and 2.659, respectively. Similarly, as the delay increased to 50 ms with jitter at 10 ms, the RTT values increased to 72.113, 102.164, 135.284, and 14.619, indicating longer round-trip times and greater variability in packet delivery.

	Value the Delays/Jitters (ms)						
KI I Statistics	10/2	20/4	30/6	40/8	50/10		
RTT Min	13.592	27.723	36.567	48.789	72.113		
RTT Average	19.884	40.156	61.840	79.206	102.164		
RTT Max	28.069	53.047	83.511	115.180	135.284		
Standard Deviation	2.659	5.138	9.723	11.234	14.619		

Table 4. Round-Trip Time (RTT) Statistics Summary of the MPT-GRE tunnel with Delay and Jitter.

In contrast, in Table 5, when delay was applied without jitter, the RTT statistics remained relatively consistent and minimal. For instance, with delay values ranging from 1 ms to 5 ms, the RTT min, average, max, and standard deviation values showed minimal fluctuations, indicating consistent and predictable packet delivery. A comparison of these results reveals the significant impact of jitter on RTT variability. While delay alone leads to predictable RTT values, adding jitter introduces fluctuations, resulting in higher average RTT values and increased variability. This finding highlights the importance of considering delay and jitter in network performance evaluations, especially for applications requiring consistent, low-latency communication, such as real-time applications.

Table 5. Round-Trip Time (RTT) Statistics Summary of the MPT-GRE tunnel with delay only.

	Value the Delays (ms)						
KIT Statistics	10 ms	20 ms	30 ms	40 ms	50 ms		
RTT Min	2.158	4.157	6.158	8.158	10.159		
RTT Average	2.163	4.163	6.163	8.163	10.163		
RTT Max	2.170	4.175	6.184	8.169	10.167		
Standard Deviation	0.002	0.002	0.003	0.002	0.002		

The relationship between RTT statistics and the effectiveness of the packet reordering mechanism highlights the challenges posed by delay and jitter in multipath networks. The increased RTT variability caused by jitter and delay necessitates careful management of reordering processes and buffer sizes to maintain network performance. The MPT-GRE reordering mechanism relies on buffering and sorting packets by their GRE sequence numbers. High RTT variability, influenced by jitter and delay, can increase reordering delays and potential buffer overflow. As RTT variability increases, the system must manage larger buffers and more complex reordering processes to ensure correct packet sequencing.

# 6. Practical Application

The MPT-GRE architecture has already been deployed and tested in practical scenarios. We have conducted real-world experiments to evaluate the performance of the MPT framework, particularly in enhancing real-time communication services. As evidence, we refer to our previously published work, which demonstrates the practical implementation and effectiveness of the MPT-GRE tunnel in real-time video streaming applications [33]. This study provides clear evidence of the architecture's applicability in real IP networks.

The MPT-GRE architecture is particularly well-suited for use cases that demand high reliability and bandwidth aggregation, such as real-time video conferencing. It is especially beneficial in environments with asymmetric paths, where multipath aggregation and packet reordering can significantly improve service quality. These practical evaluations confirm that the MPT-GRE solution is not only theoretically sound but also applicable and effective in real network deployments.

### 7. Conclusions

We conducted a detailed analysis of the packet reordering mechanism to understand how the MPT-GRE network encounters out-of-order packet arrivals due to delays in the middle path. We tested different scenarios to assess the efficacy of packet reordering. In the first scenario, we enabled the packet reordering technique; in the second scenario, we disabled it. Our study evaluated the impact of MPT-GRE throughput aggregation under various conditions, such as delays, jitters, and transmission rates. In the first scenario, the delay introduced in the second path caused a slight decrease in throughput.

In the second scenario, tunnel throughput was significantly reduced due to out-oforder packet arrivals, demonstrating the need for MPT-GRE's packet reordering mechanism. The study also assessed the critical role of reordering window size in the efficiency of packet reordering for throughput aggregation in the MPT-GRE network. The results showed that throughput aggregation is inefficient with a small reorder window, while optimal efficiency is achieved with a larger (appropriate) reorder window size. However, increasing the reorder window size beyond what is necessary provides no additional benefits and can lead to long adjustment periods, negatively impacting overall system performance. The packet reordering mechanism improves the overall performance of the MPT-GRE network, underscoring the importance of fine-tuning this parameter. Enabling packet reordering in MPT-GRE can be a strategic move, especially when delays may cause packets to arrive out of order. Implementing packet reordering mechanisms can reduce the negative effects of delays and out-of-order packets, leading to a more efficient and reliable user experience in multipath communication scenarios. These improvements can enhance the overall efficiency of the MPT-GRE network. Our research findings have practical applications that can benefit network engineers, system administrators, and researchers working on multipath communication and QoS mechanisms.

Our analysis reveals a complex relationship between delay and jitter. Delays directly impact the time packets travel, while jitter causes variation in packet arrival times. Arrival

time variations can lead to packet reordering and increased loss, reducing overall throughput. The reordering packet mechanism in the MPT-GRE library has effectively enhanced tunnel throughput aggregation, addressing the impacts of delay and jitter. This analysis is crucial in QoS mechanisms for managing delay and jitter and maintaining acceptable throughput levels.

# 8. Limitations and Future Work

While the MPT-GRE architecture offers significant benefits in throughput aggregation and resilience through its packet reordering mechanism, several limitations should be acknowledged. First, the system's performance is sensitive to the selection of the reorder window size and buffer delay parameters. Inappropriate tuning of these values can lead to increased latency or packet drops, especially under highly variable network conditions. Second, the current implementation relies on manual parameter configuration, which may not adapt optimally to dynamic network environments in real time. Future research will focus on developing adaptive reordering mechanisms and automatic parameter-tuning strategies based on real-time path monitoring. In addition, we plan to design an intelligent algorithm capable of predicting and mitigating out-of-order packet arrivals in real time. Another important direction is developing a mathematical model that generalizes the relationship between reorder window size, network delay, and throughput performance, thereby enabling more effective and dynamic optimization of MPT-GRE networks.

**Author Contributions:** All authors contributed to the study's conception and design. Material preparation and testbed building by N.A.-I., with valuable support from G.L. The first draft of the manuscript was written by N.A.-I., and all authors commented on previous versions of the manuscript. Conceptualization: G.L.; methodology: N.A.-I.; formal analysis and investigation: N.A.-I.; writing—original draft preparation: N.A.-I.; writing—review and editing: G.L. and N.A.-I.; resources: N.A.-I. and G.L.; supervision: G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

**Data Availability Statement:** The Public GitHub repository of the first author contains several configuration settings files of MPT and scripts that were used to build the experiments: https://github.com/NaseerAJabbar/MPT\_Connections\_files (accessed on 28 March 2023). The Python script to measure the throughput tunnel of MPT and execute the experiments over 100 s utilizing the iperf3 command: https://github.com/NaseerAJabbar/Packet-Reordering-Mechanism (accessed on 19 February 2025).

Acknowledgments: The measurements were conducted remotely using the facilities provided by NICT StarBED, located at 2–12 Asahidai, Nomi-City, Ishikawa 923-1211, Japan. The authors thank Shuuhei Takimoto for granting access to StarBED. The authors thank Bertalan Kovács from the Budapest University of Technology and Economics and Szabocs Szilágyi from the University of Debrecen for reading and commenting on the manuscript. The authors thank Brant von Goble from Széchenyi István University for proofreading the English-language version of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

#### References

- Ma, J.; Guo, Z.; Song, M. SSPRD: A Shared-Storage-Based Hardware Packet Reordering and Deduplication System for Multipath Transmission in Wide Area Networks. *Micromachines* 2024, 15, 1323. [CrossRef] [PubMed]
- Frommgen, A.; Erbshauser, T.; Buchmann, A.; Zimmermann, T.; Wehrle, K. ReMP TCP: Low Latency Multipath TCP. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016. [CrossRef]

- García-Torres, M.; Pinto-Roa, D.P.; Núñez-Castillo, C.; Quiñonez, B.; Vázquez, G.; Allegretti, M.; García-Diaz, M.E. Feature Selection Applied to QoS/QoE Modeling on Video and Web-Based Mobile Data Services: An Ordinal Approach. *Comput. Commun.* 2024, 217, 230–245. [CrossRef]
- 4. Xing, Y.; Xue, K.; Zhang, Y.; Han, J.; Li, J.; Liu, J.; Li, R. A Low-Latency MPTCP Scheduler for Live Video Streaming in Mobile Networks. *IEEE Trans. Wirel. Commun.* 2021, 20, 7230–7242. [CrossRef]
- 5. Almási, B.; Lencse, G.; Szilágyi, S. Investigating the Multipath Extension of the GRE in UDP Technology. *Comput. Commun.* 2017, 103, 29–38. [CrossRef]
- 6. Ford, A.; Raiciu, C.; Handley, M.; Bonaventure, O.; Paasch, C. *TCP Extensions for Multipath Operation with Multiple Addresses*; RFC8684; RFC Editor: Marina del Rey, CA, USA, 2020. [CrossRef]
- Fejes, F.; Katona, R.; Püsök, L. Multipath Strategies and Solutions in Multihomed Mobile Environments. In Proceedings of the 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw, Poland, 16–18 October 2016; pp. 79–84. [CrossRef]
- 8. Fejes, F. MPT-Multi-Path Tunnel. 2019. Available online: http://github.com/spyff/mpt (accessed on 9 October 2019).
- 9. Kovács, Á. Evaluation of the Aggregation Capability of the MPT Network Layer Multipath Communication Library and Multipath TCP. *Acta Polytech. Hungarica.* 2019, *16*, 129–147. [CrossRef]
- Almási, B.; Kósa, M.; Fejes, F.; Katona, R.; Püsök, L. MPT: A Solution for Eliminating the Effect of Network Breakdowns in Case of HD Video Stream Transmission. In Proceedings of the 2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Gyor, Hungary, 19–21 October 2015; pp. 121–126. [CrossRef]
- Szilágyi, S. The Effects of Different Congestion Management Algorithms over VoIP Performance. *Int. J. Adv. Comput. Sci. Appl.* 2015, *6*, 66–70. [CrossRef]
- Wang, J.; Wen, J.; Zhang, J.; Han, Y. TCP-FIT: An Improved TCP Congestion Control Algorithm and Its Performance. In Proceedings of the IEEE INFOCOM 2011—IEEE Conference on Computer Communications, Shanghai, China, 10–15 April 2011; pp. 2894–2902. [CrossRef]
- 13. Lencse, G.; Szilágyi, S.; Fejes, F.; Georgescu, M. MPT Network Layer Multipath Library. 2025. Available online: https://datatracker.ietf.org/doc/html/draft-lencse-tsvwg-mpt-11 (accessed on 23 March 2025).
- 14. Gao, C.; Ling, Z.; Yuan, Y. Packet Reordering Analysis for Concurrent Multipath Transfer. *Int. J. Commun. Syst.* **2014**, 27, 4510–4526. [CrossRef]
- 15. Szilágyi, S.; Bordán, I.; Harangi, L.; Kiss, B. Throughput Performance Analysis of the Multipath Communication Technologies for the Cloud. J. Electr. Electron. Eng. 2019, 12, 69–72.
- Al-Imareen, N.; Lencse, G. On the Impact of Packet Reordering in MPT-GRE Multipath Networks. In Proceedings of the 2023 46th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 12–14 July 2023; pp. 82–86. [CrossRef]
- 17. Szilágyi, S.; Fejes, F.; Katona, R. Throughput Performance Comparison of MPT-GRE and MPTCP in the Fast Ethernet IPv4/IPv6 Environment. *J. Telecommun. Inf. Technol.* **2018**, *2*, 53–59. [CrossRef]
- 18. Zinner, T.; Gia, P.T.; Tutschku, K.; Nakao, A. Performance Evaluation of Packet Reordering on Concurrent Multipath Transmissions for Transport Virtualisation. *Int. J. Commun. Networks Distrib. Syst.* **2011**, *6*, 322. [CrossRef]
- 19. Xu, Z.; Li, X.; Wang, Y.; Chen, M.; Zhou, H. DOFMS: DRL-Based Out-of-Order Friendly Multipath Scheduling in Mobile Heterogeneous Networks. *IEEE Trans. Mob. Comput.* **2024**, *23*, 8274–8288. [CrossRef]
- Almási, B.; Szilágyi, S. Multipath FTP and Stream Transmission Analysis Using the MPT Software Environment. Int. J. Adv. Res. Comput. Commun. Eng. 2013, 2, 4267–4272. Available online: https://irh.inf.unideb.hu/~szilagyi/wp-content/uploads/papers/ 22-h-Almasi\_Bela\_dr-multipath.pdf (accessed on 12 November 2013).
- Ma, J.; Guo, Z.; Pan, Y.; Zhang, M.; Zhao, Z.; Sun, Z.; Chang, Y. ORNIC: A High-Performance RDMA NIC with Out-of-Order Packet Direct Write Method for Multipath Transmission. *Electronics* 2025, 14, 88. [CrossRef]
- 22. Li, Z.; Huang, J.; Wang, S.; Wang, J. Achieving Low Latency for Multipath Transmission in RDMA-Based Data Center Network. *IEEE Trans. Cloud Comput.* 2024, 12, 337–346. [CrossRef]
- Yang, F.; Wang, Q.; Amer, P.D. Out-of-Order Transmission for In-Order Arrival Scheduling for Multipath TCP. In Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014; pp. 749–752. [CrossRef]
- 24. Xing, Y.; Xue, K.; Zhang, Y.; Han, J.; Li, J.; Wei, D.S.L. An Online Learning Assisted Packet Scheduler for MPTCP in Mobile Networks. *IEEE/ACM Trans. Netw.* **2023**, *31*, 2297–2312. [CrossRef]
- Chaturvedi, R.K.; Chand, S. An Adaptive and Efficient Packet Scheduler for Multipath TCP. *Iran J. Sci. Technol. Trans. Electr. Eng.* 2021, 45, 349–365. [CrossRef]
- Alheid, A.; Kaleshi, D.; Doufexi, A. An Analysis of the Impact of Out-of-Order Recovery Algorithms on MPTCP Throughput. In Proceedings of the 2014 IEEE 28th International Conference on Advanced Information Networking and Applications, Victoria, BC, Canada, 13–16 May 2014; pp. 156–163. [CrossRef]

- 27. Huang, J.; Lyu, W.; Li, W.; Wang, J.; He, T. Mitigating Packet Reordering for Random Packet Spraying in Data Center Networks. *IEEE/ACM Trans. Netw.* **2021**, *29*, 1183–1196. [CrossRef]
- 28. Szilágyi, S.; Bordán, I. Investigating the Throughput Performance of the MPT-GRE Network Layer Multipath Library in Emulated WAN Environment. *Acta Tech. Jaurinensis* **2022**, *15*, 7–14. [CrossRef]
- 29. Lyu, B.; Liu, C.; Yue, X. Integrated Predictor–Observer Feedback Control for Vibration Mitigation of Large-Scale Spacecraft with Unbounded Input Time Delay. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 4561–4572. [CrossRef]
- Liu, C.; Yue, X.; Zhang, J.; Shi, K. Active Disturbance Rejection Control for Delayed Electromagnetic Docking of Spacecraft in Elliptical Orbits. *IEEE Trans. Aerosp. Electron. Syst.* 2022, *58*, 2257–2268. [CrossRef]
- Al-Imareen, N.; Lencse, G. MPT Connections Files. 2023. Available online: https://github.com/NaseerAJabbar/MPT\_ Connections\_files (accessed on 28 March 2023).
- 32. Dugan, K.P.J.; Elliott, S.; Mah, B.A.; Poskanzer, J. Iperf3 Documentation. 2018. Available online: https://iperf.fr/iperf-doc.php (accessed on 15 August 2018).
- Al-Imareen, N.; Lencse, G. Enhancing Real-Time Video Streaming Quality via MPT-GRE Multipath Network. *Electronics* 2025, 14, 497. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.