Investigating the Effect of Different Subflow MTUs on MPTCP Throughput

Research Proposal

SCS4224 Final Year Project in Computer Science

Shamalka Manorathne

20001118

June 19, 2024

Declaration

The Project Proposal is my original work and has not been submitted previously for any examination/evaluation at this or any other university/institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name : C.S. Manorathne
Registration Number : 2020/CS/111
Index Number : 20001118

Signature & Date

This is to certify that this project proposal is based on the work of Mr. C.S. Manorathne under my supervision. The project proposal has been prepared according to the format stipulated and is of acceptable standard.

Supervisor Name : Dr. C.I. Keppitiyagama

Signature & Date

Co-Supervisor Name : Mr. T.N.B. Wijethilake

Signature & Date

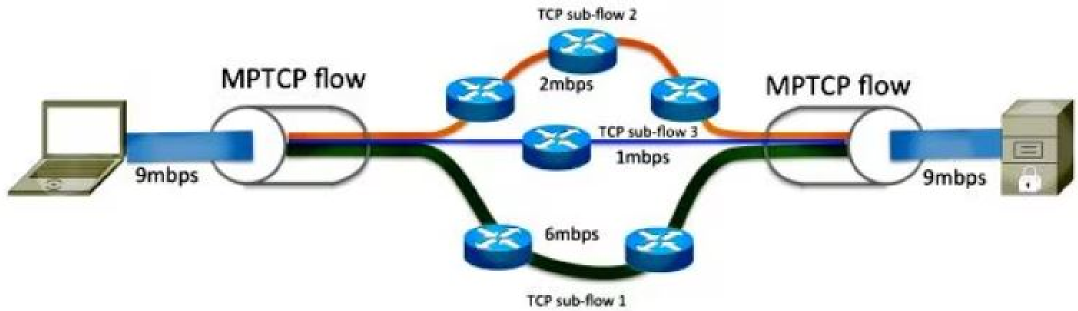
Table of Contents

Introduction

In the past, end hosts were usually only connected to the Internet through one interface. Each interface was given a unique IP address that all host interactions went through. However, the modern Internet doesn't work like this model used to. Devices like computers and tablets now have more than one interface and both wired and wireless interfaces, making Ethernet and Wi-Fi connections possible. Similarly, smartphones now have more than one interface. Even in specialized places like data centers, where many servers talk to each other for tasks like data replication and distributed computing, a fast network is necessary. This is why redundant infrastructure is used to spread traffic across multiple paths and make the network more resilient in case a link or node fails.

The Internet Engineering Task Force (IETF) standardized the Multipath Transmission Control Protocol (MPTCP). It gives the application layer a single TCP (Transmission Control Protocol) connection and allows two hosts to use several Internet paths efficiently (Ford et al. 2020). The idea behind Multipath TCP, or MPTCP, was to solve these problems. The goals that went into making MPTCP are these:

- It should work with current programs just like regular TCP.
- It should be able to use more than one network path for a single connection.
- Normal TCP shouldn't have to struggle to find enough network paths, so it should be able to use them at least as well



Because of Multipath TCP, a host can support a single TCP connection through more than one interface and address. By combining all the available resources, Multipath TCP can improve the service that the apps receive in terms of speed and reliability[2]. To find losses and see if each path is working right or not, the sender has to quickly and accurately measure the Round-Trip Time (RTT) numbers for each path. It takes milliseconds to measure the RTT, which is the time it takes to send a data packet and receive a signal acknowledging that it was sent (Paasch et al. 2014).

The efficiency of MPTCP is dependent upon the packet scheduler. A scheduler allocates the packets to the available pathways. Incorrect scheduling decisions can decrease MPTCP performance in both homogeneous and heterogeneous networks, causing reduced throughput and longer download times. The presence of heterogenous paths results in a higher number of packets being delivered out of order, which subsequently leads to the problem of Head of Line (HOL) blockage, caused by limitations in the receiver's window. An improved packet scheduler will utilize all available paths to minimize the occurrence of out-of-order packets, hence enhancing throughput and performance

The Maximum Transmission Unit (MTU) is the largest packet size that can be sent over a network path without needing fragmentation. When a host wants to send data across an interface, it consults the MTU of the interface to determine the maximum amount of data that can be included in each packet. Ethernet ports typically feature a default MTU of 1500 bytes, excluding the Ethernet header or trailer. as well as most datacenter networking hardware, can support jumbo frames which is 9000 bytes (Julaihi 2011). In practice, when a host wants to transmit a TCP data stream, it would usually allocate the initial 20 bytes out of the total 1500 bytes for the IP header, the subsequent 20 bytes for the TCP header, and utilize the remaining 1460 bytes for the data payload as required. By encapsulating data in packets of maximum size, it minimizes the consumption of bandwidth caused by protocol overhead.

Path MTU (Figure 1) discovery is the method used to determine the MTU of

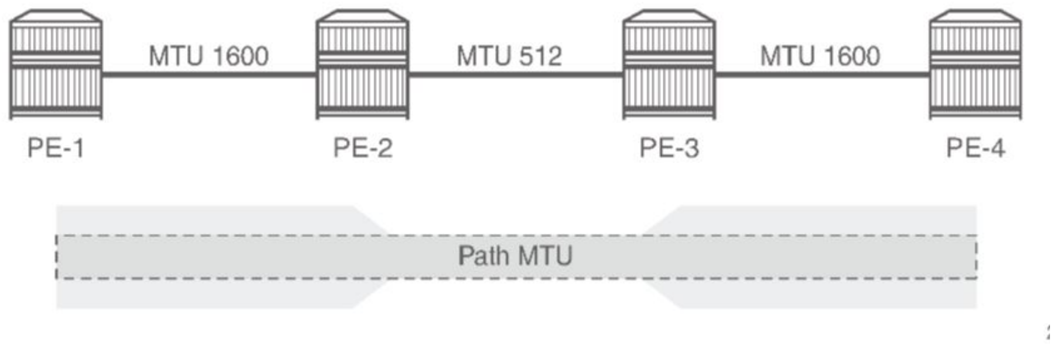


Figure 1: Path MTU

each connection that a packet may traverse (Mogul & Deering 1990). In order to optimize the use of a path, hosts need to determine the path MTU, which refers to the smallest MTU among all the links in the path to the end host. For example, if two hosts communicate across three routed links that each have an MTU of 1500, 800, or 1200 bytes, the end host must assume the smaller MTU (800 bytes) to avoid fragmentation (Custura et al. 2018).

Motivation

Implementing Multipath TCP (MPTCP) with heterogeneous subflows presents significant challenges. Achieving optimal performance with MPTCP necessitates near-homogeneous network conditions across all subflows (Adarsh et al. 2019). To maximize the potential of MPTCP, it needs to adopt a comprehensive view of each path performance, which should include, at a minimum, considerations of path Maximum Transmission Unit (MTU).

Unfortunately, not all links which compose the Internet have the same MTU. Different paths may have different MTUs due to variations in the underlying physical media type or configured encapsulation (Asiri 2021). If the packet size exceeds the MTU of link or interface, then it must be fragmented into smaller pieces to transmit it as two (or more) individual pieces, each within the link.

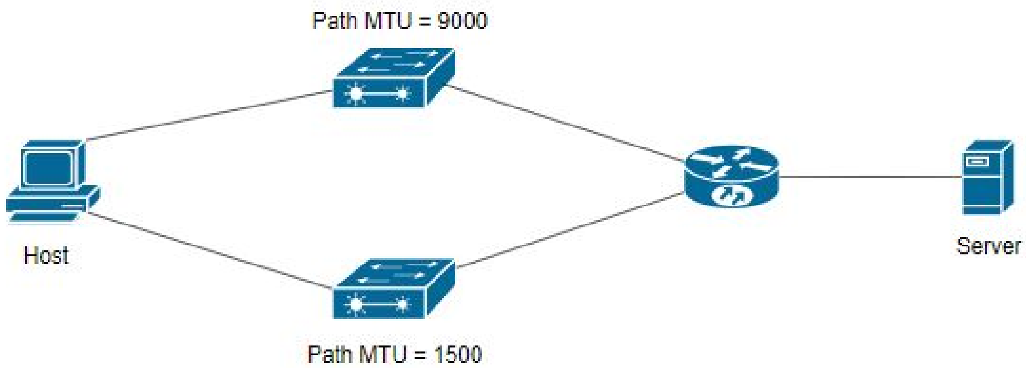


Figure 2: Two subflows having different path MTU

Fragmentation is a costly process since it requires the use of hardware resources and extra bandwidth (Feng et al. 2022). This is because new headers must be created and attached to each fragment.

In MPTCP implementation, the MTU of each subflow is not directly considered when scheduling packets across subflows. This consideration is unnecessary in homogeneous environments where all subflows have the same MTU, as they can use that consistent value directly (Asiri 2021). By considering the path MTU sizes of each subflow (Figure 2), when making scheduling decisions, it may be possible to improve the efficiency of packet scheduling across subflows, thereby increasing the overall performance and throughput of MPTCP in heterogeneous network environments.

Aims and Objectives

Aim

Exploring the behavior of Multipath TCP on subflows with different Path MTU values to ensure continuous network connectivity, evaluating the throughput and conducting a comparative analysis with standard TCP.

Objectives

- Create an emulation: Analyze the behavior of MPTCP on subflows with varying path MTU values using current implementations.
- Analyze Path MTU Variations: Investigate the variations in path MTU across different subflows and how these differences impact MPTCP performance.
- Measure Throughput: Evaluate the throughput of MPTCP when operating over subflows with different path MTU values compared to standard TCP.
- Assess Fragmentation Effects: Analyze the impact of fragmentation on MPTCP performance when subflows have differing path MTUs.
- Evaluate Scheduler Efficiency: Investigate how the MPTCP scheduler handles segments of different sizes and its effect on overall network performance and connectivity.
- Analyze and identify any performance bottlenecks or inefficiencies in MPTCP when dealing with subflows of different path MTUs. If that is so suggest and validate potential optimization techniques to enhance MPTCP performance with different path MTU values.

Research Gap

Different path MTUs in subflows mean that data should be sent in different sizes on each path, depending on the specific subflow. Consequently, each path must process a different Maximum Segment Size (MSS) based on its MTU, excluding headers such as TCP and IP. This results in the scheduler needing to handle segments of different sizes. If the MSS is the same for each segment, then the network layer must decide on fragmentation to send data through subflows with both large and small path MTUs. The lager datagram received from the transport layer cannot be sent via small path MTU subflow, there is an additional fragmentation process that needs to be done and resulting that fragmentation overhead and might affect performance as well. Efficient scheduling has the ability to decrease head-of-line blocking and latency, especially in heterogeneous environments, allowing streaming applications to minimize the buffer requirement. However, it remains unclear how different path MTU subflows should take into account for scheduling packets across them. Furthermore, there is no analysis of throughput in such an environment. Current techniques do not emphasize the involvement of managing subflows with different path MTUs and the selection of appropriate MSS values for sending data in each subflow. Examples of existing packet scheduling techniques are Min-RTT, FPS and DPSAF. Min-RTT is the default schedular in current MPTCP implementation on Linux which selects the path with the lowest RTT. FPS predicts scheduling values without considering packet loss or bandwidth. DPSAF adjusts scheduling based on packet loss rates and TCP SACK feedback but ignores bandwidth. It uses a more complex analyzing model that involves a significant level of computational complexity (Maxwell 2023). The default path manager and scheduler focus more on balancing load and optimizing latency rather than MTU-specific optimizations. Unlike RTT and bandwidth, the integration of MTU considerations has not been thoroughly explored in literature. By addressing this gap, it may be possible to achieve more efficient, reliable, and high-performance data transmission in heterogeneous situations of the network. Studies could focus on,

Fragmentation Overhead: When packets are larger than the path MTU of a subflow, they must be fragmented. This adds additional processing overhead at both the sender and receiver.

Suboptimal Resource Utilization: Using a fixed path MTU may cause some paths to be underutilized if their actual MTU is higher than the used one, leading to inefficiencies in bandwidth usage.

Research questions

How does Multipath TCP adjust to subflows with different Path MTU values and improve the packet scheduling to enhance network performance, compared to standard TCP? Additionally, what impact does fragmentation overhead have on subflow packet scheduling in the context of different Path MTU?

Significance of the Research

The research on Multipath TCP behavior with subflows having different Path MTUs is significant because it aims to improve network performance and efficiency in diverse, real-world network environments. By understanding and improving how MPTCP handles subflows with different MTUs, this research can enhance throughput, ensure better quality of service, and improve resource utilization efficiency in devices.

Planned Research Approach

This exploratory research investigates the performance of Multi-Path TCP when subflows have different path MTUs. To achieve this, the research will follow these steps:

- Setup Testing Environment: Establish a controlled testing environment to emulate network conditions with different path MTU values across the different subflows.
- Run Tests: Execute the test scenarios using MPTCP to observe behavior under different path MTU configurations.
- Analyze Results: Collect and analyze the data from the tests to identify existing MPTCP scheduling techniques to understand their approach to scheduling packets among subflows with different path MTUs.

- Correlate Path MTU Impact: Determine the correlation between subflows with different path MTU values and overall throughput, assessing how variations affect MPTCP performance.

- Performance Evaluation: Compare the performance of MPTCP under subflows with different path MTU conditions with that of standard TCP to evaluate efficiency and reliability.

Scope

In Scope

- Analyze the existing MPTCP scheduling algorithms: Assess the current MPTCP implementations and their ability to handle subflows with different path MTUs.

- Comparative Analysis: This study aims to evaluate and contrast the efficiency of Multipath TCP with normal TCP in comparable circumstances. It will specifically highlight situations in which MPTCP demonstrates notable benefits. In addition, the study will evaluate the resilience and flexibility of MPTCP in response to changes in different network conditions.

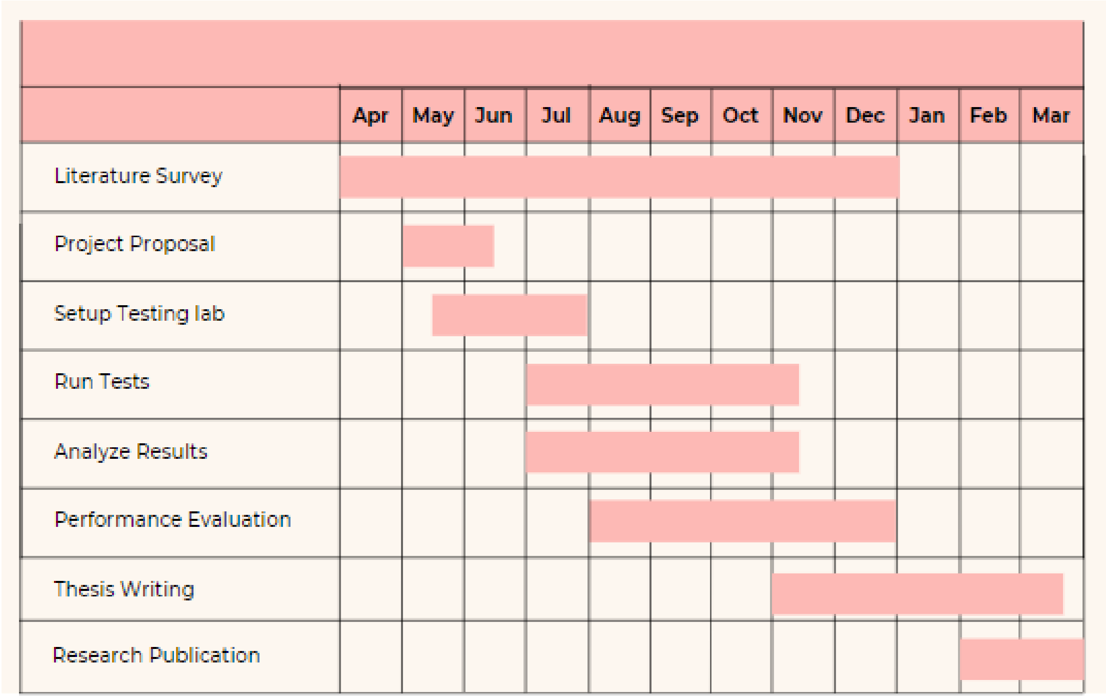
- Emulation and Lab Testing: Conduct emulations to model various path MTU configurations, followed by real-world tests to validate these emulation results and gather empirical data. Diverse testing environments, including lab setups and live network conditions, will be utilized to ensure a comprehensive evaluation of MPTCP's performance over subflows have different path MTU.

Out Scope

- Non-MTU Related Scheduling factors: This research will not thoroughly explore other aspects that affect scheduling, such as the allocation of bandwidth, path reliability, or cost considerations, unless they are directly related to MTU challenges.

- Detailed Congestion Control Mechanisms: Although congestion control is a critical aspect of MPTCP performance, this research will not focus on developing new congestion control algorithms. Existing mechanisms will be used as a basis for evaluating MTU-aware scheduling.
- Security and Encryption Concerns: While security is crucial for any network protocol, this research will not address specific security or encryption issues related to MPTCP, unless they directly impact MTU handling.
- Non-TCP Multipath Protocols: The study will be limited to MPTCP and will not cover other multipath protocols such as Multipath QUIC or SCTP.
- Application on All Platforms: The main focus will be on how MPTCP works on Linux. This study does not cover changes that need to be made for other operating systems or for custom hardware solutions.

Project Timeline and Plan



References

Adarsh, V., Schmitt, P. & Belding, E. (2019), Mptcp performance over heterogenous subpaths, *in* '2019 28th International Conference on Computer Communication and Networks (ICCCN)', IEEE, pp. 1–9.

Asiri, M. (2021), 'Novel multipath tcp scheduling design for future iot applications'.

Custura, A., Fairhurst, G. & Learmonth, I. (2018), Exploring usable path mtu in the internet, *in* '2018 Network Traffic Measurement and Analysis Conference (TMA)', IEEE, pp. 1–8.

Feng, X., Li, Q., Sun, K., Xu, K., Liu, B., Zheng, X., Yang, Q., Duan, H. & Qian, Z. (2022), Pmtud is not panacea: Revisiting ip fragmentation attacks against tcp.

Ford, A., Raiciu, C., Handley, M. J., Bonaventure, O. & Paasch, C. (2020), 'TCP Extensions for Multipath Operation with Multiple Addresses', RFC 8684.
URL: *https://www.rfc-editor.org/info/rfc8684*

Julaihi, A. A. (2011), A Fragmentation Control Approach in Jumbo Frame Network, PhD thesis, Universiti Teknologi Malaysia.

Maxwell, C. N. (2023), 'Multipath tcp, and new packet scheduling method', *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal* **10**(1), 7.

Mogul, J. C. & Deering, S. E. (1990), 'Rfc1191: Path mtu discovery'.

Paasch, C. et al. (2014), 'Improving multipath tcp', *Diss. Universit'e catholique de Louvain (UCL), London* .