Evaluation of MPTCP Schedulers in Diverse Scenarios

Vidya S. Kubde and Sudhir Sawarkar

Abstract — Multipath Transmission Control Protocol (MPTCP) is an extension of TCP meant for multihomed devices, which uses all the available interfaces for a single connection. MPTCP was evolved for Bandwidth aggregation and re-silience to network failure. The wireless networks of multihomed devices are of different characteristics, when used together decreases Quality of Service (QoS). MPTCP schedulers tried to fill this gap with different approaches. In this paper we tried to study these schedulers in different network scenarios and came with the findings that to achieve good throughput and decrease download time, only fast paths are preferred.

Index Terms — MPTCP, Path Utilization, Download Time, Throughput.

I. INTRODUCTION

Multipath TCP is (MPTCP) evolved from TCP, with the aim of Bandwidth aggregation and re-silience to network failure. In multihomed devices, the single path TCP protocol fails, if any network failure. In order to overcome this, IETF [1] has designed MPTCP, which uses all the available paths for single connection. As of today, the Linux Kernel MPTCP implementation [3] is one of the most widely used MPTCP implementations besides Apple's implementation for the cloud-based assistant system Siri [4].

TCP is mainly designed for wire-line networks. MPTCP is built over TCP and is designed for smart home devices, which mostly use wireless networks. A smart phone is having two wireless interfaces, Wi-Fi and Cellular networks. Thus, MPTCP has to deal with the wireless channel impairments. This is because packet loss, network delay, roundtrip time variation is very probable in wireless networks. Indeed, the MPTCP have to deal with more than one wireless networks with different network characteristic. Thus, to aggregate, the throughput of the multiple paths in MPTCP of different characteristics is a challenging issue.

Consequently, we have verified the behaviour of MPTCP schedulers in different network scenarios. The comparison is done with respect to parameters which affect Quality of Service, likewise Throughput, Download Time and Path utilization rate. The results signified that Round Robin performed the best in homogeneous networks, also have utilized both the paths to their full potential. But fails to perform in heterogeneous environment. On the contrary Blest can with stand heterogeneity by reducing out of order packets. Blest prioritizes fast path only so all the paths are not utilized.

This paper makes the following contributions. First the experiment study for the comparison between the throughput

of MPTCP schedulers. Secondly, we have observed the MPTCP schedulers download time for different data volumes. Lastly, we have analyzed the effect of scheduling policies on path utilization rate.

II. BACKGROUND AND MOTIVATION

In single-path TCP if packets are not lost or not re-transmitted, then they will arrive in order [5]. In MPTCP as packets are going to traverse through multiple paths, with different characteristics causing out of order at the receiver end [6]. This results in Head of Line Blocking decreasing the throughput. The functional goal of the scheduler is all the packets should reach the receiver on time. A wrong decision taken by a scheduler impacts the performance of MPTCP and may also make additional path useless. If both the paths are heterogeneous in nature as depicted in Fig. 1 then packets will reach the receiver in out of order. For example, packet no 1 will reach late, as compared to packets 2-4. Receiver cannot process packets 2-4 have to wait for packet 1. Thus, session is blocked due to wrong scheduling decision. Hence a scheduler plays a vital role in deciding the performance of MPTCP.

Fig. 1. MPTCP Scheduler.

A. Background and Related Work

The MPTCP architecture is introduced in RFC [1]. Over the past years, there has been a lot of research on MPTCP implementation, design, and performance issues. The scheduler deals with the selection of path, in order to increase the throughput compared to single-path TCP. the scheduler is invoked, either when a new packet has arrived from the application layer or acknowledgment is received. The scheduler will acquire the path characteristics, round trip time (RTT), signal strength, and throughput, loss rate. The transmission performance of the paths is evaluated with these parameters. The best path is selected to establish the connection.

A wealth of research has been done to resolve the unsolved issues of packet scheduling in MPTCP. Some works implement Round Robin [7]. This scheduler selects the paths one after another in turn. It could not perform in

Submitted on January 06, 2021.

Published on April 23, 2021.

Vidya S. Kubde, Datta Meghe College of Engineering, Airoli, Navi-Mumbai, India.

(e-mail: vidya.kubde@dmce.ac.in)

Dr. Sudhir Sawarkar, Datta Meghe College of Engineering, Airoli, Navi-Mumbai, India.

(e-mail: sudhir_sawarkar@yahoo.com)

heterogeneous networks. To face the heterogeneity, MinRTT was evolved which selects the path with the lowest RTT. MinRTT [8], is the default scheduler of MPTCP to date. The amount of data on this selected path is decided by its congestion window.

This scheduler worked well, except on memory-constrained devices that use small receive window. This problem was identified by Costin Raiciu [9] to avoid the head of the line blocking issue caused by a limited receiver window. Likewise, DAPS [10] replaced RTT by the forward delay that is sending time plus in flight time to estimate the time taken by the packet to reach the destination. Although the work added more precision, this delay aware scheduler is advantageous only when the values of RTT and congestion window remains stable for the whole duration of the schedule. Thus, DAPS is not able to deal with network failures.

Further in a study by yang [11], the authors propose an OTLAS algorithm that tries to resolve this issue by using current data. Its working is based on the idea of scheduling more segments on a sub-flow than what it can currently send. Lim in his work ECF [12] believes that the fast path is not utilized to its full extent. It aims to minimize the periods where a fast sub-flow becomes idle. Blest [13] worked with this same idea that instead of using slow paths. It prioritizes only fast paths. CP [14] contributes by, blocking the slow path. If the slow path is causing performance issues than CP

prefers to block that path. In STTF [7] hurting believes that in default scheduler unavailable paths are sometimes a better choice. The idea behind STTF is very simple; for each segment to schedule, calculate its transmission time considering data already in flight. STMS [15] deals with out of order sending of packets for in order receiving.

Alternative Scheduling Decisions for Multipath TCP [16] proposed policies for different types of applications sorting the paths, one of the policy deals with sending rate of the flows, the other policy uses the highest available space in the congestion window. In Qaware [17] Shreedhar was motivated with the fact that the particular sub-flow which is used more frequently tends to increase its end-to-end delay gradually, making it less attractive to use. They have used end to end delay means local device driver queue occupancy plus end-to-end delay measurements for path selection. Remp [18] is a different category of the schedulers, where all the paths are used but the same data is transferred on both the paths. It does not enhance the throughput but only is useful in latency- sensitive applications.

DEMS [19] believes that by strategically planning the scheduling, one can reduce the download time. It achieves this by decoupling both the paths mean sending the data in a forward direction on one path and in reverse direction on the other path, it performs data re injection, to a very small extent to reduce the download time. A detailed analysis of the schedulers is described in Table I.

TABLE I: ANALYSIS OF EXISTING SCHEDULERS

SCHEDULER	Working Principles	ISSUES
Round Robin	Optimal load balancing	Unknown fairness
MinRTT	To increase the throughput	It cannot work with asymmetric paths.
DAPS	Tries to maximize the probability of in-order arrival	Cannot deal with network failures It builds schedules runs being unable to react to network changes It does not apply scheduler re-injections
ECF	aims to minimize completion time.	Can be hurt by Head of Line Blocking
OTIAS	Schedule's data to minimize transmission time	1)It assumes symmetric forward delays (OWD = RTT/2) 2)It does not apply scheduler re-injections (re-transmissions)
DEMS	Reduced download time	Rely on exact knowledge of data chunk boundary for efficient scheduling
CP	Out of order packets	Requires network assistance
STMS	Out of order packets	RTT error sensitivity
Qaware	Limits the HOL issue	Requires network assistance
MPTP Scheduler		
Blest	Reduces out of order packets	Prefers Fast path only

III. EXPERIMENT STUDY AND EXPERIMENT SETUP

To compare the performance of four schedulers, the multihomed client and server nodes are connected to each other through different wireless networks as depicted in Fig.3 The experimental setup consists of a Dell Laptop as the client and installed Ubuntu Linux 16.04 which built the MPTCP version on the Linux kernel. The server also built with MPTCP kernel version 0.95 on Ubuntu Linux 16.04 on a general PC. The experiment network topology is shown in Fig. 2.

A. Network Characteristics of Sub flows

In our test bed the client and server are connected by a two path MPTCP connection. These paths are setup on different wireless networks WIFI, LTE, and ETH. Our aim is to study the performance of MPTCP schedulers in heterogeneous networks, that is two paths in a connection with different network characteristics like RTT and Bandwidth shown in

Table II. For this purpose, we have used Linux Transfer Control to command. To study the MPTCP scheduler performance with respect to web traffic, we have used different file size for measurement 64 Kb,100 Kb, 500 Kb and 1000 Kb.

As we were interested in finding the path utilization rate, we have monitored the traffic sent on each interface by each scheduler. Also, we have measured throughput by running Iperf in client Server mode and used -n flag in Iperf to specify the transfer size. Each set of experiments is run 50 times for every file size. This traffic is monitored by Socket Statistics SS command [20].

TABLE II: NETWORK SPECIFICATIONS

Interface	Rate	RTT
Ethernet	90 mbps	20 ms
LTE	26 mbps	80 ms
Wi-Fi	25 mps	70 ms

Fig. 2. System Topology.

IV. PERFORMANCE COMPARISON BETWEEN MPTCP SCHEDULERS

In this section, our results from multiple experiments are presented. We were interested in analyzing MPTCP performance with respect to four schedulers Default, Round robin, Blest and Redundant.

A. MPTCP Performance Comparison in Different Network Scenarios

We have explored the effect of heterogeneity on the performance of MPTCP schedulers. In related to this. Specifically, we have focused on throughput of MPTCP in Homogeneous (LTE-LTE), nearly homogeneous (LTE-Wifi) and heterogeneous.

Primitively the author worked with homogeneous networks Fig. 3 (a), it was observed that Round robin has shown better performance compared with the other three schedulers. It provides a throughput rise of 50% over other schedulers. The Fig. 3 (b) depicts that if heterogeneity is added by using LTE-WIFI.

Round robin performance decreases nearly equal to other schedulers. As we move on to purely heterogeneous networks Fig. 3 (c) WI-FI-Eth scenario, Blest provides 8% increase over Default and 35% over Redundant and 50% over Round robin. In short, the Round robin scheduler works best for homogeneous networks but cannot withstand heterogeneity. Blest works best in heterogeneous networks.

Fig. 3 (a). Throughput Comparison of Schedulers in Homogeneous Networks (LTE-LTE) for 20 secs.

Fig. 3 (b). Throughput Comparison of Schedulers in Heterogeneous Networks (Wifi-LTE) for 20 secs.

Fig. 3(c). Throughput Comparison of Schedulers in Heterogeneous Networks (Wi-Fi-Eth) for 200 secs.

B. MPTCP Scheduler Performance for Different Data Volumes

In this part we have transferred different data volumes (64 Kb, 100 Kb, 500 Kb, 1000 Kb) Iperf traffic and noted the download times of four schedulers in Wi-Fi-LTE scenario.

Fig. 4. Download Time of Schedulers with different data volumes.

Fig. 4 depicts that for 64 Kb of data, Blest provides 29% increase in download time than Default, Redundant and Round robin Schedulers. For 200Kb and 500Kb data segment, Blest provides 5% fast download time as compared to all the three schedulers. In 1000Kb of data transfer results are reversed round robin provides 7% fast download time compared with Blest.

We can summarize that for small data volumes (64 Kb) nearly all the schedulers perform the same as there is no need of second subflow. Till the time second subflow is established the data transfer is done. As the data volume increases above 100Kb the performance of MPTCP varies with different schedulers as the issues of full congestion window, out of order packets, receiver buffer full arises. In this regard the scheduler capable of handling these issues can perform better.

C. Path Utilization in MPTCP Schedulers

In order to analyze the bandwidth aggregation performed by each scheduler we have calculated the traffic sent on each Path.

As depicted in Fig. 5 (a) Default and Blest scheduler have used Wi-Fi sub-flow for less than 10% in both the scenarios Wi-Fi-Ethernet and Wi-Fi-LTE. This is because the algorithm with which these schedulers initiate path selection gives higher priority to the path with lower RTT. On the other side Fig. 5 (b) in Round-robin and Redundant, Wifi and Ethernet contribute 46% and 54% in redundant and 61% and 39% in Round-robin with Wi-Fi-Ethernet networks. In the Wi-Fi scenario shown in Fig 5. b Wi-Fi and LTE networks contribute equally.

Fig. 5 (a). Path Utilization Rate in WiFi-Eth.

Fig. 5 (b). Path Utilization Rate in WiFi-LTE.

V. CONCLUSION

In this work we have studied the effect of different networks on four schedulers. The comparison was done in terms of Path utilization rate, Download time and Throughput. The results depicts that the scheduler which uses all the available paths, works best in Homogeneous networks but fails to perform in Heterogeneous networks. On the contrary the schedulers which perform uses fast paths only are able to perform in Heterogeneous networks as they deal with out of order packets. It was observed that Round robin provides best throughput in homogeneous networks but cannot perform in heterogeneous networks. The performance of round robin is inversely proportional to heterogeneity of paths. This was proved that in LTE-LTE scenario it performed the best, in Wi-Fi it was better and worst in ETH-Wi-Fi. On the other side Blest and Default performs best in heterogeneous networks fast path is not available in short Bandwidth aggregation is not possible in both of these schedulers, as they use only one path at a time.

In related to download time, it was analyzed that for small data volumes performance of all the schedulers is same. This is because for small data volumes the primary flow bandwidth is sufficient for data transfer. For large data volumes, in WiFi-LTE scenario Blest adds additional delay for waiting for faster sub-flows, achieve more download time for medium size file, 1000 Kb of file in comparison with Round Robin.

In terms of path utilization Round robin and Redundant both the paths contributed equally but to deal with out of order packets caused by heterogeneity, Blest and Default schedulers avoid use of slow paths.

REFERENCES

- [1] Ford, M. Raiciu, and Handley, 2010.
- [2] O. Bonaventure, M. Handley, and C. Raiciu, pp. 17–23, 2012.
- [3] Paasch and S. Barre. [Online]. Available: <http://www.multipath-tcp.org>.
- [4] Apple, “Advances in networking,” *Proc. Worldwide Developers Conf. (WWDC)*, 2017.
- [5] J. Postel, “RFC793: Transmission Control Protocol. USC,” *Information Sciences Institute*, vol. 27, pp. 123–150, 1981.
- [6] Paasch, “Experimental evaluation of multipath TCP schedulers,” *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*, 2014.
- [7] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, O. Alay, and N. Kuhn, “Low-Latency Scheduling in MPTCP,” *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 302–315, 2019. [Online]. Available: [10.1109/tnet.2018.2884791](https://doi.org/10.1109/tnet.2018.2884791); <https://dx.doi.org/10.1109/tnet.2018.2884791>.
- [8] J. Hwang and J. Yoo, “Packet scheduling for multipath TCP,” *Seventh International Conference on Ubiquitous and Future Networks*, 2015.
- [9] Raiciu, “How Hard Can It Be? Designing and Implementing a Deployable Multipath {TCP},” *9th {USENIX} Symposium on Networked Systems Design and Implementation*, 2012.
- [10] N. Kuhn, “DAPS: Intelligent delay-aware packet scheduling for multipath transport,” *IEEE International Conference on Communications (ICC)*, 2014.
- [11] Yang, Q. Wang, and P. D. Amer, “Out-of-order transmission for in order arrival scheduling for multipath TCP,” *28th International Conference on Advanced Information Networking and Applications Workshops*, 2014.
- [12] Y.-S. Lim, “ECF: An MPTCP path scheduler to manage heterogeneous paths,” *Proceedings of the 13th International Conference on emerging Networking Experiments and Technologies*, 2017.
- [13] S. Ferlin, “BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks,” *IFIP Networking Conference (IFIP Networking) and Workshops*, 2016.
- [14] B.-H. Oh and J. Lee, “Constraint-based proactive scheduling for

- MPTCP in wireless networks,” *Computer Networks*, vol. 91, pp. 548–563, 2015. [Online]. Available: [10.1016/j.comnet.2015.09.002](https://doi.org/10.1016/j.comnet.2015.09.002); <https://dx.doi.org/10.1016/j.comnet.2015.09.002>.
- [15] H. Shi, “{STMS}: Improving {MPTCP} throughput under heterogeneous networks,” *{USENIX} Annual Technical Conference*, 2018.
- [16] Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, “Alternative Scheduling Decisions for Multipath TCP,” *IEEE Communications Letters*, vol. 21, no. 11, pp. 2412–2415, 2017. [Online]. Available: [10.1109/lcomm.2017.2740918](https://doi.org/10.1109/lcomm.2017.2740918); <https://dx.doi.org/10.1109/lcomm.2017.2740918>.
- [17] T. Shreedhar, “QAware: A cross-layer approach to MPTCP scheduling,” *IFIP Networking Conference (IFIP Networking) and Workshops*, 2018.
- [18] Frommgen, “ReMP TCP: Low latency multipath TCP,” *IEEE International Conference on Communications (ICC)*, 2016.
- [19] Y. Guo and Ethan, “DEMS: Decoupled multipath scheduler for accelerating multipath transport,” *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, 2017.
- [20] 2020. [Online]. Available: <http://man7.org/linux/man-pages/manss.8.html>.