

Design and Implementation of Dynamic Packet Scheduling with Waiting Time Aware: DPSW2A

K Raghavendra Rao¹

Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram-522502, Guntur, Andhra Pradesh, India

B N Jagadesh²

Professor, Department of CSE
Srinivasa Institute of Engineering & Technology
Amalapuram, Andhra Pradesh, India

Abstract—One of the principal goals of 5G is to enhance performance in connection with speed and delay curtailment. To accomplish this, IETF has proposed Multipath TCP to utilize the accessible interface for communication. The demand for mobile communication is escalating day by day. The predominant communication option for people is mobile. For giving better service for users, nodes are fitted out with multiple interfaces. Multiple interfaces are as well one of the benefits of 5G. Multi path protocols are used to load balancing and resilience to failure. When communicating with asymmetric interfaces, latency is an imperative factor. To attain low latency is hard when asymmetric interfaces are used. When communication happens using multiple interfaces, the scheduler plays a central role since it decides which interface needs to be used for the packet. In this article we spotlight on scheduling algorithms, how this schedule will play a vital role to transfer data to receiver nodes with low latency. In this paper, we emphasize on the Scheduler named DPSWWA with the objective of minimizing delay and effective usage of interfaces.

Keywords—MPTCP; scheduler; latency; delay; transport protocol; waiting time

I. INTRODUCTION

Mobile devices are one of the predominant communication devices to users. Mobile devices are steadily increasing day by day. Mobile devices are equipped with multiple interfaces like 3G and Wi-Fi interface for better communication. Many researchers are developing algorithms for multipath communication. Multipath TCP provides resilience to failure and better communication [1]. When compared to single path TCP, MPTCP will give better performance [2]. Mainly multipath TCP is useful in one of the scenarios is data centers [3]. Latency is more important for sensitive applications [4]. Latency is very important when the interfaces are asymmetric [28]. This paper mainly focused on scheduling data packets using MPTCP protocol to multiple interfaces of type asymmetric. Here we mentioned some of the widely used state-of-the-art MPTCP schedulers. Most of the schedulers are not effectively utilizing and reaching the goal of MPTCP in terms of using all interfaces. In this paper we propose an algorithm Dynamic Packet scheduling with Waiting time Aware (DPSWWA). We mainly focused on how to effectively utilize the interface and schedule the packets not to occur late at the receiver end. This paper covers both the implementation and the design of scheduling techniques in the Linux kernel. The result shows that DPSWWA fulfill its design goals.

Compared to existing default MPTCP schedulers DPSWWA is increasing the speed of communication between nodes.

This article is structured as follows: Section II presents background; Section III presents the state-of-the-art-of-the MPTCP scheduler, Section IV presents the state-of-the-art-of-the schedulers, Section V discussed about DPSWWA, Section VI evaluated through experiments with the goal of less latency. Throughout this scenario we use web traffic. In Section VII, we present an evaluation report. In Section VIII, we present conclusion and in Section IX is presented the future enhancement.

II. PARALLEL RESEARCH WORK

Smart phones are equipped with multiple interfaces. The advantage of multiple interfaces is easy to handover from one interface to another interface when a problem occurs. This advantage is given to mobile devices to grow like anything and make it the people predominant communication device. MPTCP was designed as an alternative to TCP for working on multiple interfaces at the same time. The available Transport protocols are not supported to multiple interfaces used at the same time. So we need new protocols to support multiple interfaces at the same time. IETF is standardized by MPTCP in RFC 6824[5]. There are commercial applications that use MPTCP; one of the MPTCP used is Apple Siri [6]. Fig. 1 shows how the MPTCP can transfer the data to multiple sub-flows, once it receives data from the application layer.

Fig. 1 shows how data is delivered from sender application layer to receiver network layer via MPTCP sub layer. Once data is available to the application layer it can send the data to the transport layer. Here MPTCP protocol comes into picture, by taking data from the upper layer it sends the packets to all available flows. MPTCP considers all the factors before sending the packet to sub-flow. Like RTT, when MPTCP contains more than one sub flow, then the scheduler will decide which sub-flow should send packets. Which sub-flow has lowest round trip time. The scheduler will select that sub-flow with consideration of the network properties.

Fig. 2 represents the MPTCP connection establishment from client to server or node/node. Connection establishment of MPTCP is same as to TCP. Here MPTCP can create a one connection first then he will establish another connection to server. Initially client sends a request packet to server, then server will accept that request and sends the acknowledgment. Once the client has received the acknowledgment it is send to

server segment. Once the first flow connection establishment is over then client can create another flow connection from client to server. In this manner a client can create multiple flow form client to server. The diagram clearly represents the connection establishment flow from source to destination or client to server or one node to another node. The comprehensive connection establishment is also known as MPTCP 3-way handshake.

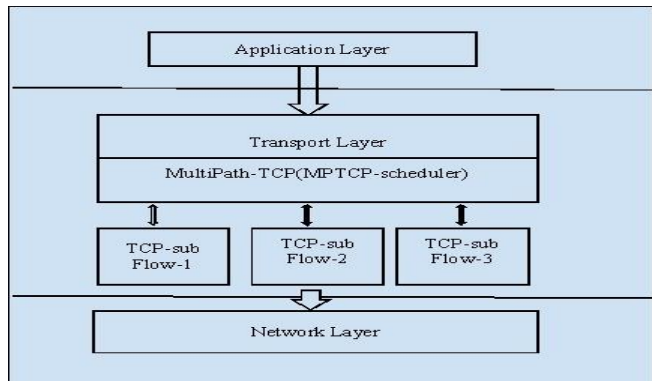


Fig. 1. Data flow from Application to Network via MPTCP.

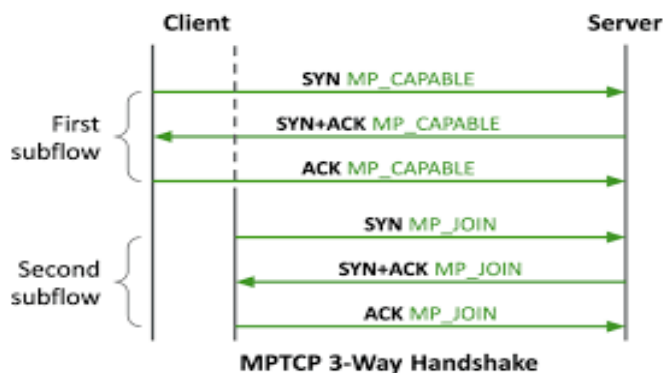


Fig. 2. MPTCP Connection Establishment.

Fig. 3 represents the connections/flows from sender to receiver or client to server or client host to server host. The above picture shows two flows from client to server. Here one flow is from Wi-Fi and another flow is 3G/4G/LTE connection. The above two flows are simultaneously used to transfer the data from client to server.

available flow. When the MPTCP buffer received the data from upper layer, then it can check all available resources like path characteristics and send the packet to flows. The above diagram represents two flows from sender to receiver. One flow is named as sub-flow-1 and another flow is named as sub-flow-2. The characteristics of flows are flow-1 rtt time is 10ms, cwnd is 30 bytes and flow-2 rtt is 30 ms and cwnd is 30 bytes.

Most of the communications requires data in the same order as what the sender sends. Multipath protocol splits data to different flows, so chances are there to receive the packets out of order on the receiver side. When data arrives out of order many problems may occur. Some of the well-known problems are discussed in [7].

Fig. 6 explains problems of transmitting data over asymmetric paths.

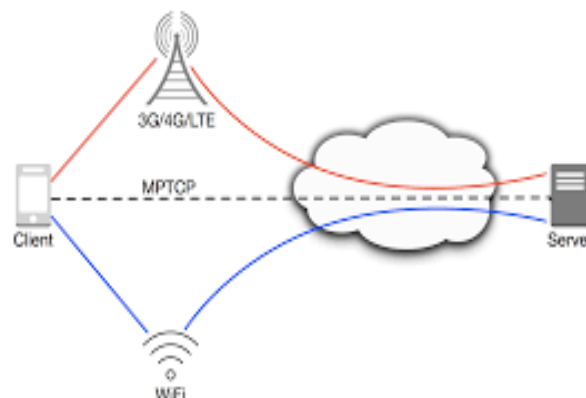


Fig. 3. MPTCP Connection Representation from Sender to Receiver.

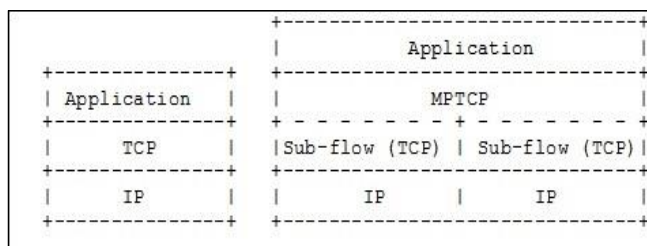


Fig. 4. Comparison of Standard TCP and MPTCP Protocol Stacks.

Fig. 4 presents protocol stack of TCP and MPTCP. In TCP we can make use of only one interface communication whereas MPTCP can use multiple interfaces at the same time using multiple interfaces. MPTCP have multiple interfaces to transfer the data from client to server / node to node.

Fig. 5 represents the functional flow from sender application layer to receiver buffer. Here data flow will start from multiple flows. More exactly once data is available from application layer to MPTCP send buffer, then MPTCP send buffer will apply scheduling algorithm to schedule the segments to available flows not to occur HOL problem, receiver buffer problem and out-of-order packets problem. Here scheduler plays a vital role to schedule the segments to

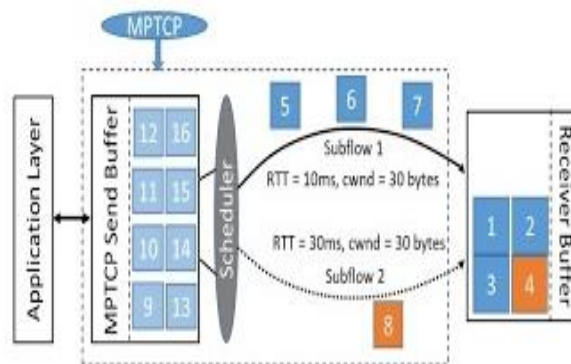


Fig. 5. Data flow from Sender Application Layer to Receiver Buffer.

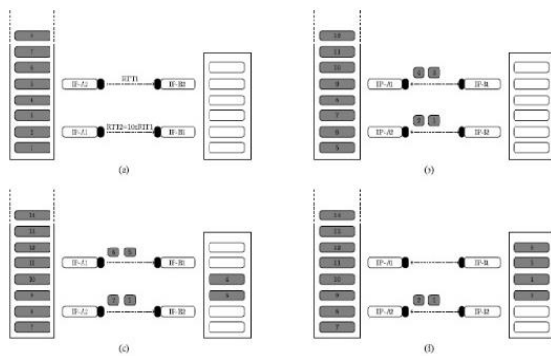


Fig. 6. Represents the Work Flow of Sender and Receiver, how the HOL Problem Occurs.

Here two nodes are communicating using two paths to share information. In the two paths path one has the RTT 10 times larger than the path two. Segments are transferred to 1, 2 using slow path and 3,4 using fast path Fig. 6b. Fig. 6c represents a well-known problem of HOL. Where 3 and 4 received and kept buffered until received 1,2. So it introduces delay in delivering data to applications explained in Fig. 6d. If the receiver buffer is filled with unordered packets then the sender will block them from sending. Then will face reduced throughput and increased delay. So to address this issue, we are very careful while selecting interfaces to use segment segments using multipath protocols. For work explained in this article, the Linux Kernel implementation. Some variants are available LowRTT [8]. Now take one example to send 15 packets using MPTCP. One flow is using 3G and another flow is used with WLAN. One of the interfaces is WLAN having RTT 10ms and another one having 100 ms in 3G of RTT. Now if we apply LRTT it immediately uses 10ms flow instead of 100ms flow. Initial congestion window will take 10 segments. First 10 will send in fast flow remaining 5 segments will send in slow sub flow. To address the buffer problem many researchers have put forward schedulers for MPTCP. The four state-of-the-art-of-the schedulers DAPS, ECF, OTIAS, BLEST will be explained in the next section.

S. Habib, [13] present a report on the need for a single path to multipath TCP. They offer multiple features like load balancing and reduced delay.

Y.-C. Chen [14] mentioned the advantages of multipath over single path; flow capacity and how scheduling will affect the performance.

M. Baerts, [15] has identified the gap between TCP and MPTCP; he implemented the MPTCP in android smartphones to understand the performance of heavy applications.

L. Ji, [16] studied multipath TCP; he mentioned the advantages of multipath TCP in mobile communication and gained less delay by considering the low rtt flows.

B. Briscoe [17] specifies the sources of reducing performance of multipath tcp in terms of latency and throughput.

S. Ferlin [4] finds the results of asymmetric links do not make MPTCP performance better than TCP.

A. Barnstorm, [18] mentioned his report on cloud based multipath mobile communication he mentioned how the latency has decreased and improved the performance of cloud based communication.

M. Wang [19] assessed the MPTCP in high quality video streaming, he assessed the video streaming with multiple interfaces facing issues. He mentioned data retransmission also.

A. Gurney, [20] specifies that low latency is not achieved by using multiple paths, but we can achieve low latency by using a good scheduler. He mentioned that scheduling plays a vital role in the MPTCP packet scheduling when having multiple interfaces.

S. Alfred son, [21], specifies that Network delay and latency plays a very important role in cellular networks. He studied a detailed report on delay in respect to network, application and transport application perspective.

J. Huang, [22], specifies that lower latency and higher bandwidth is attracting many users. He discovered various limitations in TCP over LTE. He proposed a novel and light weight algorithm to estimation of bandwidth. Based on his observation many tcp connections are under utilizing the available bandwidth.

J. Vehkaperä, [23], specifies the optimized scheduling based on the traffic. He suggests that for scheduling on different types of traffic different types of mechanism is needed. Making network aware applications play a vital role in scheduling segments.

G. Texier, [24], online video services are sensitive to overall quality of video stream and a more important factor is latency between video generation and video playback. He addresses some of the problems in video streaming. He proposed an implementation of multipath video delivery at the application level.

A. Alheid, [25] specifies that multipath TCP achieves higher bandwidth and higher resilience against network path failures. He mentioned how out-of-order packets affect the communication. He specifies how best we can re-order the packet technique.

N. Kuhn, [26] specifies that the increasing heterogeneity and asymmetry in network communication makes delay and quality of service to be a challenging task. He proposed a novel scheduling algorithm to reduce the receiver buffer blocking problem.

J. Rexford, [27], points out delay sensitive applications like video streaming, voice over IP and live streaming requires low end-to end delay. Recent year's popularity of low delay applications has increased. He proposed an algorithm that minimizes the end-to-end delay experienced by inelastic traffic.

III. STATE-OF-THE-ART-OF-THE MPTCP SCHEDULER

MPTCP default scheduler will allocate packets to the fast path by considering the congestion window with low RTT [8].

This will work fine in symmetric paths but not in asymmetric paths like 3G and WLAN.

1) *Delay aware packet scheduling*: The main goal of this scheduler is to send segments to the receiver in order. DAPS [9] can send segments to the receiver by considering only “not to block the receiver buffer”. It can send the data to sub flows. First it selects which sub-flow to use then it determines which segment to send which sub-flow and last it sends data segments according to the selected sub-flow.

2) *Out-of-order-transmission-for-in-order-arrival*: TIAS [10] sends the packets based on the low RTT. It uses simplification of $RTT/2$. Here with, it considers the one way delay. OTIAS differs from LRTT and DAPS. Every time OTIAS calculates the transfer time for every packet then it selects the flow which has very less transfer time flow.

3) *Earliest completion first*: ECF [11] addresses the issue of performance degradation problems that come from path asymmetric. It minimizes the fast path waiting time. ECF can consider the parameters $CWND$, RTT A and data available to send. By considering the factors ECF schedules the packets to sub-flow. ECF can wait and send data to the fast path.

4) *Block estimation scheduler*: BLEST [12], which aims to, reduce the buffer blocking of receiver buffers in heterogeneous networks. By increasing throughput and reducing spurious retransmission. It increases the good-put also.

IV. START-OF-THE-ART-OF-THE SCHEDULER EVALUATION AND DISCUSSION

For WLAN/3G, we observe the OTIAS is similar to LRF and DAPS, ECF worse than LRF. BLEST gives better performance compared to DAPS.

Fig. 7 is the functional flow diagram of default RTT algorithm and Fig. 8 is the proposed algorithm functional flow. Fig. 9 shows the good put of WLAN/3G.

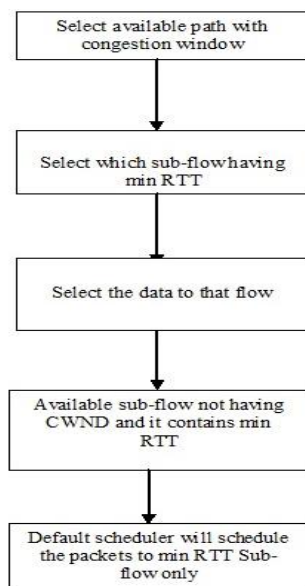


Fig. 7. Functional flow Diagram for Default Min-RTT Scheduler.

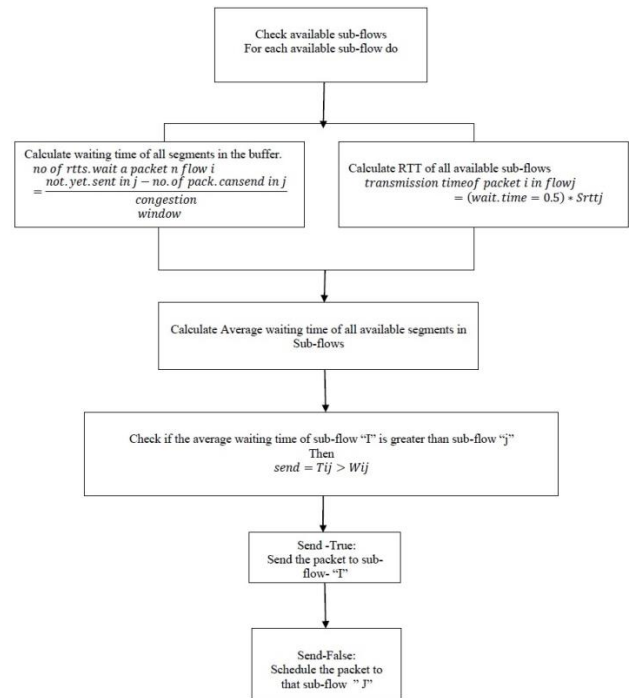


Fig. 8. Functional flow Diagram for Proposed DPSWWA Scheduler.

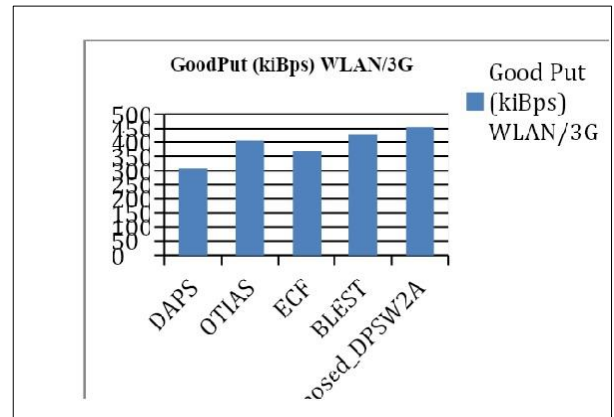


Fig. 9. Illustrates the Performance of State-of-the-art-of-the Scheduler using Bulk Transfer; we Consider the Good-put as a Metric.

V. PROPOSED METHODOLOGY

Waiting-Aware MPTCP-Scheduler: Design and Implementation.

Asymmetric interfaces flow, to avoid out-of-order delivery, delay can be reduced. However, state-of-the-art-of-the schedulers do not completely address this issue. To address this issue, we introduce the new scheduling algorithm: DPSWWA.

A. DPSWWA

The main idea of DPSWWA is to utilize all the available sub flows and reduce the delay in segment delivery to the receiver. Here we explained the design of DPSWWA, new metric to estimate the waiting time of sub flow that might result from scheduling the segment on an available flow. The DPSWWA algorithm 1 is present in

Algorithm 1: Proposed Algorithm: DPSW2A

Step1: Schedule the packet
Step2: Calculate waiting time of flow
Step3: waiting=RTT of fast path /2
Step4: if Fast path waiting time >= Waiting
Step5: then calculate the packet Arrival Time
Step6: if packet arrival time is equal to slow path of RTT/2
Step7: then
Step8: send the packet into available flow J else
Step9: Select the another path of available cwnd with min rtt to send
Step10: Calculate arrival time of the packet
Step11: Do

To address the issue of not utilizing all sub flows, we present a dynamic scheduling algorithm with waiting time awareness. MPTCP maintains the send window on connection level for each connection, which is necessary for multiplexing the data. DPSWWA assumes that the provided segment will occupy the space in the MPTCP window. We infer all packets in the window will wait at the same time.

We estimate the data that will send on flow during one RTT. For every RTT we increase by 1segment of the congestion window.

$$rtts=RTTSRTTF \tag{1}$$

$$x=MSSF(CWND+(rtts-1)2)*rtts \tag{2}$$

1) *Functional comparison of LRF and DPSWWA*: We will show how scheduling decisions are made by LRF and DPSWWA. The data shown in Fig. 4 was collected from emulation experiments using Linux with an enabled MPTCP kernel. We show how the scheduler act when a burst of 15 segments are sent by an application using tow flows. Paths RTT are p1=10ms and p2=100ms, the availability capacity for both paths have taken (0.5Gbits/s) and congestion window capacity is 10 segments.

First we consider the LRF scheduler 4a when scheduling first 6 packets are selected path p1 having 10ms of rtt. Then 6.7 are selected to 100ms flow. We allocate segments to all flows because of TSQ mechanism. After the two segments are scheduled to slow path, then remaining packets are scheduled to fast path, after the window full of fast flow schedulers schedule the packets to slow path.

The scheduling decision of DPSWWA is made with waiting time of flows. DPSWWA schedule the packets to fast flow once the window is full, then it moves to slow path equation taken from [10].

$$RTTOWAITJI=(NOTYETSENTJ-PACKCANSENDJCWNDJ)$$

Packets are scheduled based on the waiting time of fast path, then the scheduler will schedule the packets to slow path.

VI. EVALUATING DPSWWA

To assess the blocking and overuse of fast flow, we performed controlled experiments comparing the performance of LRF and DPSWWA. We evaluated latency and throughput

of the scheduler. The scheduler is evaluated in Linux kernel implementation and the results are reported in section

VII. EXPERIMENTAL SETUP

We implemented this experiment in Linux Kernel and NS-3 Implementation; setup is consisting of two machines. A client, two Wireless access points (WLAN/3G), a server accessing router equipped with two interfaces. We used Experiment, throughput and latency. Table mentioned the parameters used in the experiments.

To enable MPTCP communication between two nodes, the nodes were equipped with the MPTCP-enabled Linux kernel. The client used the MPTCP kernel and a server equipped with a modified DPSWWA MPTCP Linux kernel. For experiment, MPTCP was configured to use path one as the primary path. Table I has Network Parameters.

A. Throughput and Latency

We assess the scheduler’s latency and throughput; we compare transmission of two paths. (Path one has 10 to 50 ms and path 2 having 10 to 50ms).

Below Diagram shows the results of the experiment, the y-axis shows the transmission time from sender to receiver. X-axis shows the size of the segment to transfer the sender to receiver.

Fig. 10 will shows the transmission time of all algorithms, Fig. 11 has mentioned the page load time of algorithms with page load time. Table II has site information, objects size and size. Fig. 12 shows the path sharing of WLAN and 3G.

The difference among the scheduler, LRF scheduler with approximately 67% of the traffic over WLAN, BLEST a little more 72 % of traffic will use WLAN and DPSWWA is 79% of traffic use WLAN.

TABLE I. NETWORK CHARACTERISTICS, LATENCY AND THROUGHPUT EXPERIMENT

Path/Parameter	Latency and Throughput	
	Path-1	Path-2
Capacity [Mbps]	100	100
Delay[ms]	10...50	10...50
Loss[%]	0	0

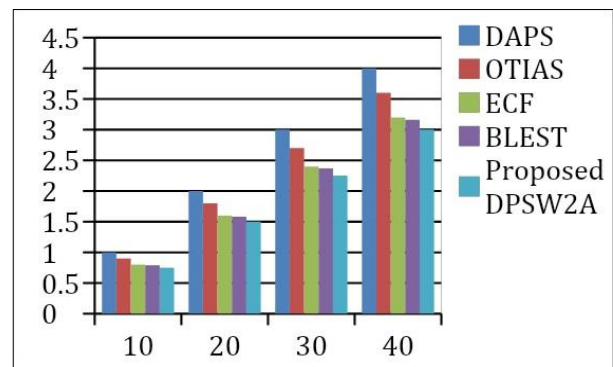


Fig. 10. Average Transmission Time for Burst of different Sizes over Sub-flows with Asymmetric Paths.

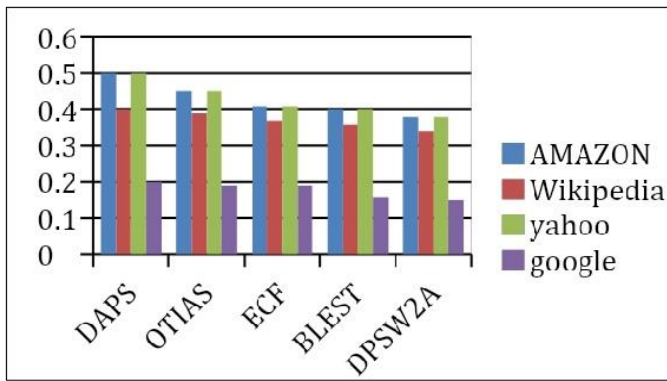


Fig. 11. Represent the Page Load Time of different Site of Schedulers.

TABLE II. WEB SITES USED IN THE EVALUATION

Web Site	Objects	Total Size[Kbits]
Amazon	10	300
Google	15	120
Wikipedia	19	40
Yahoo	22	250

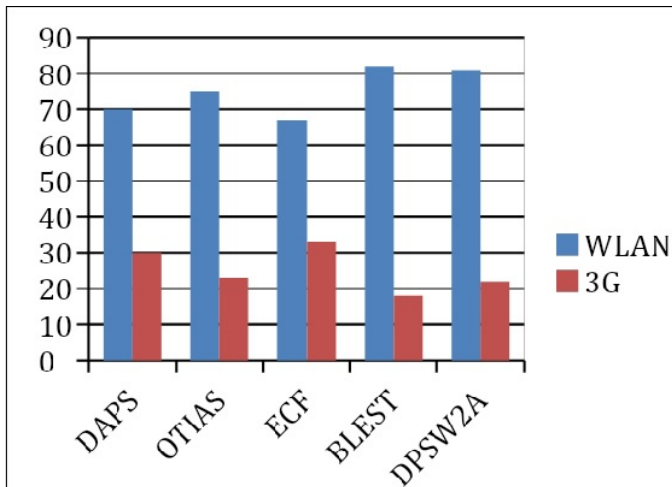


Fig. 12. Represents the Path Load Share between WLAN and 3G for Real Experiment.

VIII. CONCLUSION

The present-day devices are outfitted with multiple interfaces. Networks are often multipath. A node can extend to another node by utilizing multiple paths. For example, mobile phones have multiple interfaces and data centers having redundant paths to transfer the data. It has been shown to increase delay and throughput/output when using asymmetric interfaces. The rationale for poor performance can be encountered with the scheduler. If the scheduler will react dynamically to schedule the packet, then we achieve less delay and more throughputs. This paper provides in-depth analysis of state-of-the-art-of-the scheduler (DAPS, ECF, OTIAS, BLEST) for MPTCP to problems of asymmetric. DPSWWA is schedule the packet and reduce the delay and increase more throughputs with using all available paths.

IX. FUTURE ENHANCEMENT

Future work includes refinement of the DPSWWA to schedule packets in more network changes.

REFERENCES

- [1] Paasch, Christoph, and Olivier Bonaventure. "Multipath tcp." *Communications of the ACM* 57.4 (2014): 51-57.
- [2] Wischik, Damon, et al. "Design, Implementation and Evaluation of Congestion Control for Multipath TCP." *NSDI*. Vol. 11. 2011.
- [3] Raiciu, Costin, et al. "Improving datacenter performance and robustness with multipath TCP." *ACM SIGCOMM Computer Communication Review* 41.4 (2011): 266-277.
- [4] Ferlin, Simone, Thomas Dreiholz, and Özgü Alay. "Multi-path transport over heterogeneous wireless networks: Does it really pay off?." *2014 IEEE Global Communications Conference*. IEEE, 2014.
- [5] Ford, Alan, et al. "TCP extensions for multipath operation with multiple addresses." (2013).
- [6] "Advances in Networking" (WWDC), <https://developer.apple.com/videos/wwdc2017>.
- [7] Yedugundla, Kiran, et al. "Is multi-path transport suitable for latency sensitive traffic?" *Computer Networks* 105 (2016): 1-21.
- [8] Paasch, Christoph, et al. "Experimental evaluation of multipath TCP schedulers." *Proceedings of the 2014 ACM SIGCOMM workshop on Capacity sharing workshop*. 2014.
- [9] Sarwar, Golam, et al. "Mitigating receiver's buffer blocking by delay aware packet scheduling in multipath data transfer." *2013 27th international conference on advanced information networking and applications workshops*. IEEE, 2013.
- [10] Yang, Fan, Qi Wang, and Paul D. Amer. "Out-of-order transmission for in-order arrival scheduling for multipath TCP." *2014 28th international conference on advanced information networking and applications workshops*. IEEE, 2014.
- [11] Lim, Yeon-sup, et al. "ECF: An MPTCP path scheduler to manage heterogeneous paths." *Proceedings of the 13th international conference on emerging networking experiments and technologies*. 2017.
- [12] S. Ferlin and C. Paasch. (2017). MPTCP BLEST Linux Kernel Implementation.
- [13] Habib, Sana, et al. "The past, present, and future of transport-layer multipath." *Journal of Network and Computer Applications* 75 (2016): 236-258.
- [14] Chen, Yung-Chih, et al. "A measurement-based study of multipath tcp performance over wireless networks".
- [15] De Coninck, Quentin, et al. "Observing real smartphone applications over multipath TCP." *IEEE Communications Magazine* 54.3 (2016): 88-93.
- [16] Han, Bo, et al. "An anatomy of mobile web performance over multipath TCP." *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. 2015.
- [17] Briscoe, Bob, et al. "Reducing internet latency: A survey of techniques and their merits." *IEEE Communications Surveys & Tutorials* 18.3 (2014): 2149-2196.
- [18] Grinnemo, Karl-Johan, and Anna Brunstrom. "A first study on using MPTCP to reduce latency for cloud based mobile applications." *2015 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2015.
- [19] Wu, Jiyan, et al. "Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks." *IEEE Transactions on Mobile Computing* 15.9 (2015): 2345-2361.
- [20] Arzani, Behnaz, et al. "Impact of path characteristics and scheduling policies on MPTCP performance." *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2014.
- [21] Garcia, J., Alfredsson, S., & Brunstrom, A. (2015, June). Delay metrics and delay characteristics: A study of four Swedish HSDPA+ and LTE networks. In *2015 European Conference on Networks and Communications (EuCNC)* (pp. 234-238). IEEE.

- [22] Huang, Junxian, et al. "An in-depth study of LTE: Effect of network protocol and application behavior on performance." *ACM SIGCOMM Computer Communication Review* 43.4 (2013): 363-374.
- [23] Ojanperä, Tiia, and Janne Vehkaperä. "Network-assisted multipath DASH using the distributed decision engine." *2016 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2016.
- [24] Houze, Patrice, et al. "Applicative-layer multipath for low-latency adaptive live streaming." *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016.
- [25] Alheid, Amani, Angela Doufexi, and Dritan Kaleshi. "A study on MPTCP for tolerating packet reordering and path heterogeneity in wireless networks." *2016 Wireless Days (WD)*. IEEE, 2016.
- [26] Kuhn, Nicolas, et al. "DAPS: Intelligent delay-aware packet scheduling for multipath transport." *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014.
- [27] Javed, Umar, et al. "Multipath protocol for delay-sensitive traffic." *2009 First International Communication Systems and Networks and Workshops*. IEEE, 2009.
- [28] Hurtig, Per, et al. "Low-latency scheduling in MPTCP." *IEEE/ACM Transactions on Networking* 27.1 (2018): 302-315.