Technical Comparison between MPTCP and TCP in Heterogeneous Networks

https://doi.org/10.3991/ijim.v16i19.35299

Ahmed Saleem Abbas
College of Information Technology, University of Babylon, Babylon, Iraq
Ahmed_saleam@uobabylon.edu.iq

Abstract—Multipath TCP (MPTCP) is a project that has proposed by IETF group to allow packet streams to be separated across multiple sub-flow paths. MPTCP brings a great benefit in throughput, reliability, and fairness. However, MPTCP can be implemented in a little Linux distribution where it requires compiling and installing to become available for an actual and simulation scenario. In the paper, the performance analysis and evaluation results in both TCP and MPTCP are achieved through PC computer connected to Wi-Fi access point and 3G in heterogeneous networks. The simulation results prove that MPTCP performs better than single-path TCP for heterogeneous networks. All experiments over MPTCP and TCP leads to same results in terms of the throughput in heterogeneous networks. In this paper, four congestion control methods are implemented in the kernel and compared with each other understand the behaviors of those methods. The result of the experiment proves that Linked Increased Algorithm (LIA) has the advantage over the others in light data traffic while Balanced LIA (BLIA) has the advantage the others when there is heavy data traffic.

Keywords—TCP, MTCP, IETF, CWNDSim, BLIA, OLIA

1 Introduction

Transmission Control Protocol is the most broadly applied on the networks in the present time. To transfer data streams through Internet, TCP can be considered as a significant protocol which carry data from transmitter to receiver by using reliable mechanisms such as retransmission packets and check sum. This protocol has evolving frequently for past five decodes to be more reliability. The early design decision of TCP is still frustrated for all users; the isolation between the network and transport layers has not been perfectly apparent. Nevertheless, to distinguish between single stream of data through arriving traffic, de-multiplexes packets according to the four fields by the receiver; destination IP, protocol identifier, source IP, and the source port. These four fields also are called socket. Consequently, each connection of TCP has a one path and limited to the one socket from the transmitter also from receiver [1]. That leads to a significant concept, where if the TCP connection is established from side, and as a result of changing IP address in the transmitter or receiver from another side, then the entire connection will not work.

The current networks are enabled to be multipath. Most of the network devices such as servers, routers and computers are multi-homed in addition to the smartphone devices have several wireless interfaces for instance GSM, Wi-Fi and data centers which have many excessive links. The excessive links need to a modern design of TCP to convert it to work on multiple paths; this named as MPTCP or multipath TCP [2]. MPTCP is a key adjustment of TCP that permits multipath to be exploited concurrently by the particular TCP connect. MPTCP has suggested through IETF [3]. It has evident advantages in the reliability especially in terms of link errors and loading balance in case of data centers and servers. It achieves the best robustness, and more performance than an individual TCP while performing maintenance on the actual data traffic. That can be achieved through supplying a standard TCP interface for apps and leave the fourth layer alone (transportation) to handle multiple path connections. A sub layer under TCP represent a part of MPTCP, It distributes data flow across multiple paths known as sub-flows [3]. There is set of requirements that can be influenced on MPTCP including the compatibility and networks [4]. Any application implement on TCP must work correctly on MPTCP without changes, while the compatibility refers to that MPTCP must work with any environment that TCP can operate. The core component of TCP is the congestion control method that permits to TCP alter data rates due to the congestions state. Every TCP connection handles an important factor which is congestion window (cwnd). Cwnd controls the volume of data that the transmitter send them without needing to wait the acknowledgments from the destination. The same method can be implemented on MPTCP where each sub-flow has its cwnd and it is adjusted adaptively depending on the congestion in the single path that forwarding to the receiver.

A little MPTCP congestion control methods have developed like uncoupled congestion control (LIA) [5], Opportunistic Link Increase Congestion control (OLIA)[6,7], Balance link adaptive congestion control(BALIA) [8] and latency based congestion control (Wvegas)[9]. The key objective of the article is the analyses and evaluation of the modern developing MPTCP protocol (on multi-path and single links). The data rates of various scenarios is measurement by Linux distributions. The outcomes of the whole experiments will be compared with the others in addition to the performance of various MPTCP congestion control methods are compared with each other.

2 Related work

MPTCP is a modern method toward the best balance loading, which it can balance at the end hosts in network as a part of the processing. It is illustrated in RFC 6897 [4], an activate work groups in IETF. Set of tires are achieved to execute the MPTCP in Linux distributions. Researchers in ref [10] have made efforts about the kernel to be customized and tuned by calling the variables. With the potential of exploiting multipath for TCP, there is concern has generated about congestion control methods about the paths of data. The most challenge of MPTCP is implementing robust and efficient congestion control methods which can use all the paths without aggressive the original

TCP. Ref [11] has proved that executing congestion control methods of MPTCP can cause harm.

Ref [12] showed set of states for these harmful methods for example TCP-EWTCP which used equally weighted of TCP. This version of MPTCP distributes the data equally comparing to the default TCP. Although that this version of MPTCP has advantage in terms of equally data flows through paths, it considers not effective because the traffic data did not divide equally through the available paths. There is a kind of congestion control method can achieve better decision making than TCP-EWTCP and can solve fairness challenge which called coupled manner [5]. The impairment of this manner is clear when the existing paths have various data rates like the case with GSM and WiFi. It does not equal the round trip time (RTT) the reason of that is; this manner tends to forward all its data on the low congestion path, that leads to low throughput path inefficient. To solve these issues, Ref [12] was proposed through using semi-coupled instead of coupled manner. In semi-coupled, the implementation of the congestion control algorithm depends on the objectives of design such as balance congestive, compatibility, available default TCP and path selection.

Ref [13] represents the real experiment MPTCP congestion control method which focused on maximizing data rates and balanced congestion. in this Ref, the authors compared the developed method with three congestion control methods which are : coupled TCP, uncoupled-TCP and LIA through using the simulation CWNDSim. In Ref [13], it was offered that the developed algorithm can be successful to keep the whole data rate near the target data rate, and data traffic is forwarded into the low congestion path. Ref [2] has provided total functional of MPTCP and the results achieved over Linux. In this Ref, authors overviewed the most cases of networks in MPTCP including data centers and mobiles device. The data rates transmit measurements of the smart devices like mobiles focused over the typical modes for operation where the devices are communicated to 3G, then connection decline and the devices changed to Wi-Fi. The results analyzed and made comparison to regular TCP then presented a smoothing support with MPTCP. The reason is that data continues flowing regardless the interfaces switched. The another scenario was balanced loading in the data centers where the evaluation was achieved across regular TCP, three data flow MPTCP, and four flow MPTCP while experiments over EC-20 test with 45 instances. Authors in Ref [2] shows MPTCP with 4 paths better performance than both 3 paths MPTCP and standard TCP.

3 Experimental results

Experiment evaluating of MPTCP is achieved by Ubuntu and the kernel supports MPTCP [10]. Several use cases are achieved through laptop which has 3G and Wi-Fi interfaces. The network topology of this scheme is described in Figure 1. This topology is designed on ns-3 to obtain results of the experiments. The network interfaces have speed which reach to 5Mbps.

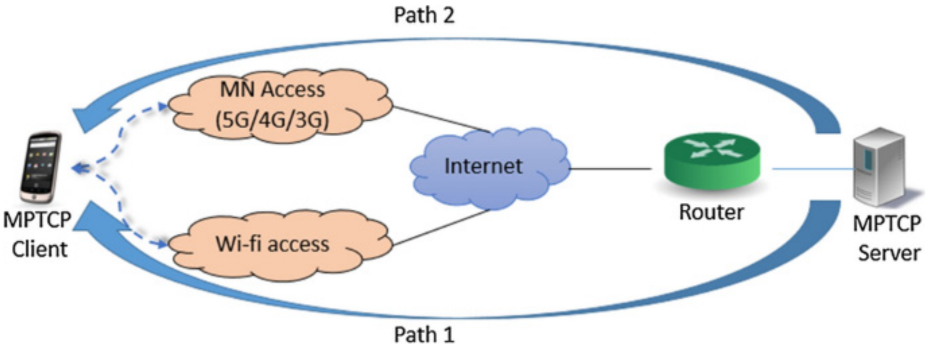

Fig. 1. Network topology

3.1 Comparison between MPTCP and default TCP

In this experiment, 1GB file is downloaded from the server by different scenarios of Internet. The 1st scenario includes default TCP on Wi-Fi link, while the second scenario uses default TCP on 3G interface. The third scenario uses MPTCP on 3G and Wi-Fi simultaneously. Figure 2 shows the results of the all scenarios including one criteria which is throughput. In this figure, it is very clear that throughput has best performance comparing with default TCP weather TCP over 3G or Wi-Fi. MPTCP can enhance the performance remarkably because two of interfaces are active during the simulation. It also displays the throughput by using MPTCP is practically the summation of every path. It is not a condition that throughput equal to the sum of two interfaces because in some times the experiment is achieved on link with shared some hosts in the same network. Throughput is observed by using simulation tools such as if stat [14]. The aggregation of downloads for every interface is collected and plotted in average number for every 5 readings to make the graph more sooth. It is very clear the Wi-Fi interface better than 3G in terms of stability which leads to best performance of MPTCP. To prove that, Figure 2 shows the weired behavior at 160 310 and 560 second.

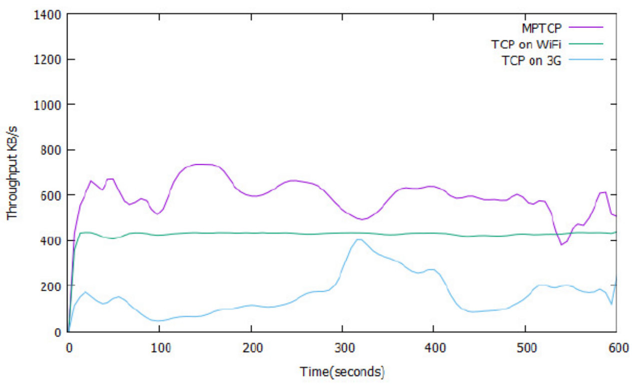

Fig. 2. Comparison between MPTCP and TCP over Wifi and 3G in terms of throughput (light traffic)

3.2 Heavy load comparison

In this scenario, FTP is downloaded with 1 GB file by MTCP over 3G and Wi-Fi interfaces. Whereas the file is downloaded, the video file is starting to consume the available bandwidth at 310 seconds. This paper conducted the same experiment with default TCP. Figure 3 displays the throughput for both scenarios. As it expected, the video file obtains best throughput with MPTCP. The server did not support MPTCP, therefore video file download uses Wi-Fi link. The whole capacity of 3G link can be used in FTP and Wi-Fi will be shared by FTP and video download. In the time 150 second, the throughput for MPTCP is not stability as a result of the sharing link capacity. While video file at 270 second, the throughput declined to 74 kilobyte/second for default TCP and 174 kilobyte/second for MPTCP. This demonstrates how multipath improves MPTCP in terms of throughput without impacting the attitude of default TCP at link bottleneck.

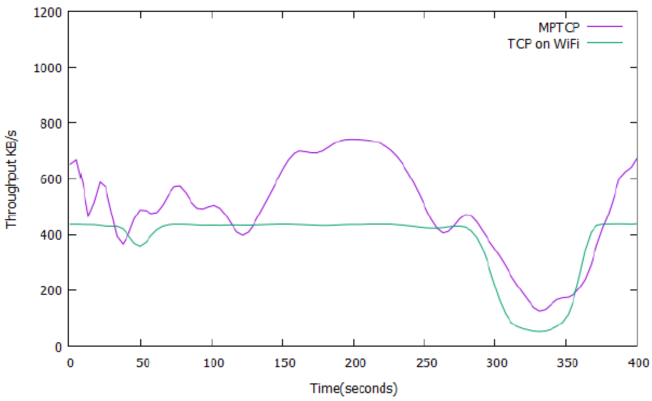

Fig. 3. Comparison the throughput between MPTCP and default TCP over WiFi (a huge amount of traffic)

3.3 The MPTCP sub flows on one link

The throughput during download in this scenario observed by MPTCP with single sub flow on Wi-Fi link. MPTCP has compared with throughput of couple sub-flow on the same link. The simulation Results presents a great improving on data rates when increasing the number of sub-flow. Figure 4 displays the downloading speed of MPTCP with couple of sub flows is 72 KB/s as an increasing value while it compared with MPTCP that has one sub flow.

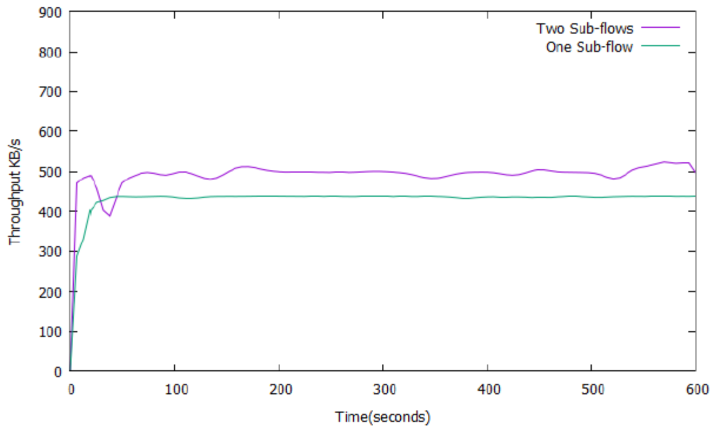

Fig. 4. Comparison between two and one sub-flow in terms of throughput

3.4 The application layer handover

Within application layer handover the data rates of default TCP is observed and evaluated in comparison to MPTCP. The data rates forTCP is observed when download a one GigaByte file from a server by TCP connect over WiFi interface. The Wi-Fi link is locking through the download. The apps layer perceived the WiFi link declined and reconnect to 3G link with a TCP connection (changed). The similar experiment has done repeatedly besides enabling MPTCP. The simulation results in Figure 5 displayed from two cases, the handover happens between 102- 132 second through MPTCP and data traffic continues pushing in spite of the losing of the links.

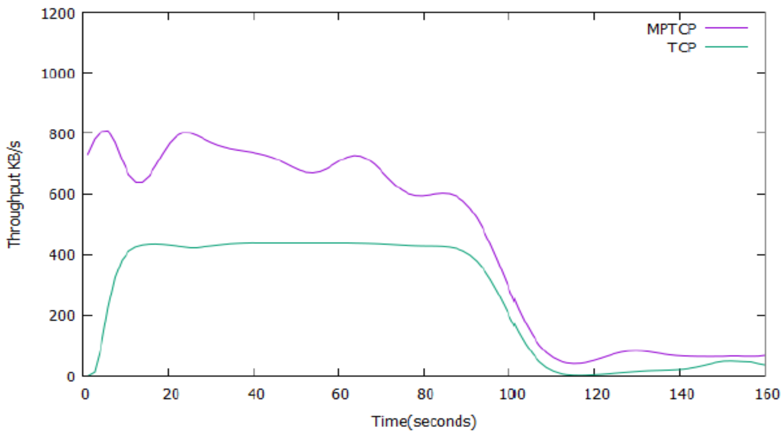

Fig. 5. Comparison between MPTCP and TCP in terms of throughput

As it is presented the handover time, the download did not reach the zero in the occurrence of MPTCP. For TCP, for all handover durations before a connection is established, data rates drop to zero.

4 Congestion control methods of MPTCP

MPTCP deploys the loading through creating several sub-flows across possible paths between transmitter and receiver. It is important to notice that congestion control of MPTCP and default TCP are not similar. The easy method is implementing the default TCP congestion control method over each one of sub-flow. This solution considers not efficiency because it presents the multiple flows which is more than its fair that share several sub-flows through the same bottleneck. Additionally, it is desirable that the transmitter with multipath carry more data traffic using the less congested path. Besides that, the transmitter achieves the resource pooling which means that set of links behave similarly one sharing link within high capacity [14].

Each one of MPTCP congestion control method should meet the three different objectives [15-25]. The first objective is enhancing the throughput; multipath connection must achieve not less than one-path TCP on the best path. In this case, the worst case of MPTCP has throughput equal to default TCP. The second objective is that MPTCP must not harm;a multiple sub-flows must not possess the bandwidth more than default TCP over the same link, this confirm that multiple sub-flows does not harm others. Both first and second objectives guarantee the fair at link, while the third objective performs resource pooling approach. However, several congestion control methods of MPTCP were designed and executed in Kernel Linux. The four methods of MPTCP congestion control listed below will be implemented in several Linux distributions [10].

4.1 Linked Increasing Algorithm (LIA)

This algorithm joins the congestion control methods that were implementing over various sub-flow through linking their increasing functions and dynamically adapt the cwnd. LIA implemented only to increasing part of congestion avoidance stage. LIA results is a fair and good to default TCP on links and at exact time shifting data traffic exit from paths of congestion. LIA follows the concept of Additive Increment/ Multiplicative decrement (AIMD). According to AIMD, the LIA can represented like the following:

All received ACK by the transmitter on each sub-flow i, the cwnd i in Additive Increase [5]:

$$Min\left\{\frac{\alpha.B_{ack}.Mss_i}{\sum_{i=0}^{n} cwnd_i},\frac{B_{ack}.Mss_i}{cwnd_i}\right\} \tag{1}$$

α: A value which defines the aggressive of the multiple flows.

Back: Refers to how many acknowledged bytes.

Mssi: Refers to Max segment size on single sub-flow i.

n: Refers to how many sub flows.

For every packet loss over sub flow i, the cwnd in multiplicative decrease is:

$$cwnd = cwnd / 2$$

Due to the first equation in (1), the entire throughput for a multiple flows based on α, Mssi, and RTT of their paths. to do the first objective, it is incredible to select a one value of α which performs the target through-put at every time. Where, α is calculated depends on monitored behavior of every paths as illustrated in equation (2), a α alpha is resulting through equalization the rates of multiple flows and TCP flow which are executed over the similar path.

$$(\sum_{i=0}^{n} cwnd_i)\frac{Max\left\{\frac{cwnd_i}{RTT_i^2}\right\}}{(\sum_{i=0}^{n}\frac{cwnd_i}{RTT_i})^2} \quad (2)$$

Where:
$\left(\sum_{i=0}^{n}\frac{cwnd_i}{RTT_i}\right)$: The sum of entire potential values for the paths
$\left(max\left\{\frac{cwnd_i}{rtt_i^2}\right\}\right)$: Max value of any potential path

4.2 Opportunistic Linked Increasing Algorithm (OLIA)

By measurements and analyses in Ref [7], it was demonstrated that the implementations of the current MPTCP-LIA is forcing on trade-off between responsiveness from side and optimum resource pooling from another side. Both objectives cannot be performed simultaneously, which leads to the fairness issue for TCP users. If the users changed from default TCP to MPTCP, this may shrink a throughput of another user with no any profit to users of MPTCP, and as a result of that violating the third objective (resource pooling). OLIA is presented as alternative for MPTCP-LIA. OLIA joins the increasing of cwnd and uses the same method in multiplicative decrease when there is packet loss event in network. OLIA is just implemented to additive increment of congestion avoidance stage, while the slow started phase has similar to the used in default TCP, in addition to a small adjustment in status of multipath is used [6]. The incremental stage illustrated as the following:

Every ACK received by the transmitter on each sub-flow *i*, the *cwnd i* in Additive Increase [7]:

$$\frac{\frac{cwnd_i}{RTT^2}}{((\sum_{i=0}^{n} cwnd_i).(\frac{cwnd_p}{RTT_p}))^2} + \frac{\alpha_i}{cwnd_i} \quad (3)$$

Where:
cwndp: The size of congestion window for the path p with biggest cwnd
RTTp: Round trip time (delay) for path p with biggest cwnd.
αi: referes to adjust the parameter of the path i.
n: Refers to how many sub flows
For every packet loss over sub flow i, the cwnd in multiplicative decrease is:

$$cwnd = cwnd / 2$$

The first side in formula (3) provide the optimum resources pooling, where TCP compatible version which compensate for various RTT. The 2nd side with αi assurances the no flappiness and responsiveness of OLIA.

4.3 The Balanced Opportunistic Linked Increasing Algorithm (BLIA)

Current MPTCP congestion control methods such as MPTCP-LIA and MPTCP-OLIA suffering from both unfair to the TCP of single path or inability to respond to conditions of network particularly when the entire paths used in MPTCP have the same latency. The tradeoff between unresponsiveness and unfairness problems is unavoidable. BALIA wisely balances the tradeoff between two issues through striking a fair balance between responsiveness and friendliness [8]. The additive increasing and multiplicative decreasing stages can be illustrated as the following:

Each ACK received by the transmitter on each sub-flow *i*, the *cwnd i* in Additive Increase [8]:

$$\left(\frac{x_i}{RTT_i(\sum_{k=0}^{n} x_k)^2}\right).\left(\frac{1+\alpha_i}{2}\right).\left(\frac{4+\alpha_i}{5}\right) \tag{4}$$

Where:

$x_i = \frac{cwnd_i}{Rtt_i}$ $\quad a_i = \frac{Max(x_k)}{x_i}$

n: Refers to how many sub flows

For every packet loss over sub flow i, the cwnd in multiplicative decrease is:

$$\left(\frac{cwnd_i}{2}\right).Min\{a_i, 1.5\}$$

If the network has one available path then a_i must be 1 and the addition and multiplicative formula will be mitigated to default TCP algorithm which is either TCP-NewReno or TCP-Cubic.

4.4 The Delay Based Congestion Control (Wvegas)

This method called Wvegas [9]. Contrasting of MPTCP-LIA that depends on packet drop event, this algorithm exploits packet queue latency as an indicator for existing the congestion in network. Comparing with LIA, OLIA and BALIA described previously which depend on packet loss event, wVegas is very sensitive to the changing when the congestion state occurs in the network, and it achieves quicker convergence and more data traffic moving timely. The following formula should be achieved at the finish of every transmission:

In each sub_flow *i*, calculating the different between the actual data rate and expected rate [9]:

$$diff_i = \left(\frac{cwnd_i}{base\_RTT_i} - \frac{cwnd_i}{RTT_i}\right).base\_RTT_i \tag{5}$$

Where,
RTTi: represents the mean delay over the latest round on every sub flow i,.
Base_RTTi represents delay for sub flow i in case of the path is uncongested.
In the avoidance of congestion, if the *diff i* isn't less than unfair i, and the data rate should be changed.

$$rate_i = \frac{cwnd_i}{rtt_i} \tag{6}$$

$$weight_i = \frac{rate_i}{total\ rate\ of\ all\ i} \tag{7}$$

$$a_i = weight\ i\ .total\_a \tag{8}$$

5 The comparison of MPTCP congestion control methods

The comparison about the throughput while transferring a one-Gigabyte file achieved in this scenario, through the four various MPTCP congestion control methods illustrated in the previous sections (LIA, OLIA, BLIA and wVegas). The file is downloading four times, each time with various congestion control methods are used. Both interfaces 3G and Wi-Fi are used with MPTCP congestion control methods at normal data traffic load. The results confirm that MPTCP-LIA outperforms the other three MPTCP congestion control in terms of the throughput at time (152s – 482s) as illustrated in Figure 6. MPTCP-BALIA outperforms the other three MPTCP congestion control methods in the interval between 0 322 second in terms of throughput.

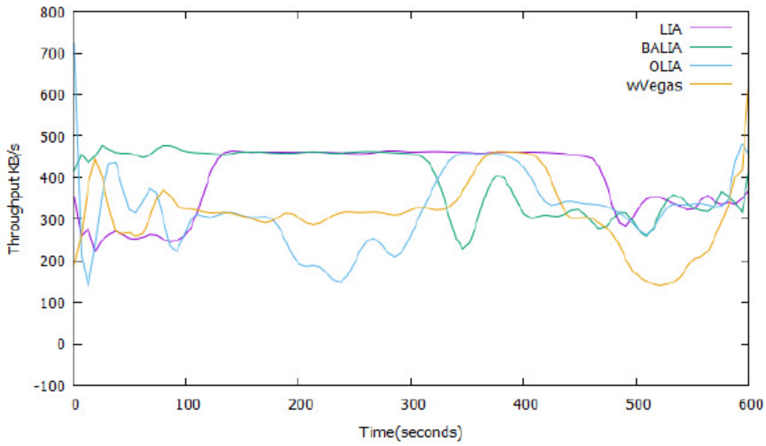

Fig. 6. Comparison between MPTCP congestion control methods

6 Conclusions

MPTCP exploits multi interfaces such as 4G, mmWave,Wi-Fi to generate several subflows which use multi paths for one connection. The concept of multipath for heterogeneous network enhance throughput, exploit the resources of network, and confirm the robustness. In this paper, there is set of experiments which demonstrated that MPTCP outperforms single path TCP in terms of throughput. In normal traffic, the MPTCP has the preferences in terms of the throughput when comparing with single path TCP which uses Wi-Fi or 3G (Figure 2). For a heavy traffic, MPTCP enhanced the throughput without aggressing the default TCP when bottleneck of network is congestion (Figure3).

When comparing MPTCP with single subflow on Wifi link, the results confirm that there is enhancing in throughput because increasing one sub-flow comparing with single path (Figure 4). With application layer, it is clear that MPTCP outperformed single path TCP through handover time. The throughput of MPTCP did not reach zero while the TCP has zero throughput and can resume after new connection (Figure 5). Regarding MPTCP congestion control methods, it is a clear from the comparison that LIA has the best results in terms of the throughput during the light data traffic while BLIA has the best throughput during the heavy data traffic. The future work will be computing energy consuming in MPTCP and comparing with the single path.

7 References

[1] H. Si, C. Sun, B. Chen, L. Shi and H. Qiao, "Analysis of Socket Communication Technology Based on Machine Learning Algorithms Under TCP/IP Protocol in Network Virtual Laboratory System," in IEEE Access, vol. 7, pp. 80453-80464, 2019. https://doi.org/10.1109/ACCESS.2019.2923052

[2] Bonaventure, Olivier, Mark Handley, and Costin Raiciu. "An overview of Multipath TCP.”; login: 37, no. 5 (2012): 17-23.

[3] Ford, C. Raiciu, and M. Handley. "TCP Extensions for Multipath Operation with Multiple Addresses" draft-ietf-mptcpmultiaddress (work in progress)." (2010).

[4] Scharf, Michael, and Alan Ford. Multipath TCP (MPTCP) application interface considerations. No. RFC 6897. 2013. https://doi.org/10.17487/rfc6897

[5] Raiciu, C., M. Handley, and D. Wischik. "RFC 6356, Coupled Congestion Control for Multipath Transport Protocols." (2011). https://doi.org/10.17487/rfc6356

[6] Khalili, Ramin, Nicolas Gast, and Miroslav Popovic. "Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP." (2013).

[7] Khalili, Ramin, Nicolas Gast, Miroslav Popovic, Utkarsh Upadhyay, and Jean-Yves Le Boudec."MPTCP is not pareto-optimal: performance issues and a possible solution." In Proceedings of the 8th international conference on Emerging networking experiments and technologies, pp. 1-12. ACM, 2012. https://doi.org/10.1145/2413176.2413178

[8] Walid, A., Q. Peng, J. Hwang, and S. Low. "Balanced Linked Adaptation Congestion Control Algorithm for MPTCP." Working Draft, IETF Secretariat, Internet-Draft draft-walid-mptcpcongestion-control-03, July (2015).

[9] Cao, Yu, Mingwei Xu, and Xiaoming Fu. "Delay-based congestion control for multipath TCP." In Network Protocols (ICNP), 2012 20th IEEE International Conference on, pp. 1-10. IEEE, 2012.
[10] "MPTCP Linux Kernel Implementation." MultiPath TCP. Accessed sep. 01, 2022. http://mptcp.info.ucl.ac.be/.
[11] Wischik, Damon, Costin Raiciu, Adam Greenhalgh, and Mark Handley. "Design, Implementation and Evaluation of Congestion Control for Multipath TCP." In NSDI, vol. 11, pp. 8-8. 2011.
[12] Honda, Michio, Yoshifumi Nishida, Lars Eggert, Pasi Sarolahti, and Hideyuki Tokuda. "Multipath congestion control for shared bottleneck." In Proc. PFLDNeT workshop, pp. 19-24. 2009.
[13] Raiciu, Costin, Damon Wischik, and Mark Handley. "Practical congestion control for multipath transport protocols." University College London, London/United Kingdom, Tech. Rep (2009).
[14] C. Huang, J. Zhang and T. Huang, "Objective-Oriented Resource Pooling in MPTCP: A Deep Reinforcement Learning Approach," 2020 3rd International Conference on Hot Information-Centric Networking (HotICN), 2020, pp. 175-181. https://doi.org/10.1109/HotICN50779.2020.9350854
[15] I. A. Aljazaery, and A. H. M. Alaidi, "Encryption of Color Image Based on DNA Strand and Exponential Factor," iJOE, vol. 18, no. 03, p. 101, 2022. https://doi.org/10.3991/ijoe.v18i03.28021
[16] H. Salim, T. H. H. Alrika "Enhanced Data Security of Communication System using Combined Encryption and Steganography," International Journal of Interactive Mobile Technologies, vol. 15, no. 16, pp. 144-157, 2021. https://doi.org/10.3991/ijim.v15i16.24557
[17] A. H. M. Alaidi, A. S. Abdalrada, and F. T. Abed, "Analysis the Efficient Energy Prediction for 5G Wireless Communication Technologies," International Journal of Emerging Technologies in Learning (iJET), vol. 14, no. 08, pp. 23-37, 2019. https://doi.org/10.3991/ijet.v14i08.10485
[18] I. A. Aljazaery, J. S. Qateef, A. H. M. Alaidi, and R. a. M. Al_airaji, "Face Patterns Analysis and Recognition System Based on Quantum Neural Network QNN," International Journal of Interactive Mobile Technologies, vol. 16, no. 8, 2022. https://doi.org/10.3991/ijim.v16i08.30107
[19] N. Alseelawi, and H. T. Hazim, "A Novel Method of Multimodal Medical Image Fusion Based on Hybrid Approach of NSCT and DTCWT," iJOE, vol. 18, no. 03, p. 115, 2022. https://doi.org/10.3991/ijoe.v18i03.28011
[20] H. T. H. Haider TH. Salim ALRikabi, "Secure Chaos of 5G Wireless Communication System Based on IOT Applications," International Journal of Online and Biomedical Engineering(iJOE), vol. 18, no. 12, pp. 89-102, 2022. https://doi.org/10.3991/ijoe.v18i12.33817
[21] A. H. M. Alaidi, R. a. M. Al_airaji, H. T. ALRikabi, I. A. Aljazaery, and S. H. Abbood, "Dark Web Illegal Activities Crawling and Classifying Using Data Mining Techniques," International Journal of Interactive Mobile Technologies, vol. 16, no. 10, 2022. https://doi.org/10.3991/ijim.v16i10.30209
[22] A. SALAH, and R. S. Khairy, "The Detection of Counterfeit Banknotes Using Ensemble Learning Techniques of AdaBoost and Voting," International Journal of Intelligent Engineering and Systems, vol. 14, no. 1, pp. 326-339, 2021. https://doi.org/10.22266/ijies2021.0228.31
[23] A. S. Hussein, R. S. Khairy, S. M. M. Najeeb, "Credit Card Fraud Detection Using Fuzzy Rough Nearest Neighbor and Sequential Minimal Optimization with Logistic Regression," iJIM, vol. 15, no. 5, 2021. https://doi.org/10.3991/ijim.v15i05.17173

[24] L. F. J. Ban Hassan Majeed, Haider Th.Salim Alrikabi, "The impact of teaching by using STEM approach in the Development of Creative Thinking and Mathemati-cal Achievement Among the Students of the Fourth Sci-entific Class," International Journal of Interactive Mobile Technologies (iJIM), vol. 15, no. 13, pp. 172-188, 2021. https://doi.org/10.3991/ijim.v15i13.24185
[25] B. Y. L. Kimura, D. C. S. F. Lima and A. A. F. Loureiro, "Packet Scheduling in Multipath TCP: Fundamentals, Lessons, and Opportunities," in IEEE Systems Journal, vol. 15, no. 1, pp. 1445-1457, March 2021. https://doi.org/10.1109/JSYST.2020.2965471

8 Author

Assistant Prof. Dr. Ahmed Saleem Abbas, Head of Software Department, College of Information Technology, University of Babylon, Iraq. All Publications are 40 (16 Paper in my Scopus Profile) (Email: Ahmed_saleam@uobabylon.edu.iq).

Article submitted 2022-08-13. Resubmitted 2022-09-16. Final acceptance 2022-09-18. Final version published as submitted by the author.