World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:16, No:11, 2022

Hybrid Multipath Congestion Control

Akshit Singhal, Xuan Wang, Zhijun Wang, Hao Che, Hong Jiang

Abstract—Multiple Path Transmission Control Protocols (MPTCPs) allow flows to explore path diversity to improve the throughput, reliability and network resource utilization. However, the existing solutions may discourage users to adopt the solutions in the face of multipath scenario where different paths are charged based on different pricing structures, e.g., WiFi vs. cellular connections, widely available for mobile phones. In this paper, we propose a Hybrid MPTCP (H-MPTCP) with a built-in mechanism to incentivize users to use multiple paths with different pricing structures. In the meantime, H-MPTCP preserves the nice properties enjoyed by the state-of-the-art MPTCP solutions. Extensive real Linux implementation results verify that H-MPTCP can indeed achieve the design objectives.

Keywords—Congestion control, Network Utility Maximization, Multipath TCP, network.

I. Introduction

MPTCPs that split a flow into multiple subflows to be sent via different paths towards the destination explore path diversity to improve flow throughput, network reliability and utilization. They play an important role in today's Internet, especially for end-to-end MPTCPs that do not explicitly involve the Internet routers for the control, as evidenced by the standardization of such protocols by IETF, e.g., [2]. Like end-to-end TCP, end-to-end MPTCPs can be deployed quickly at scale. Hence, the work in this paper aims at developing an end-to-end MPTCP.

A widely accepted design objective for MPTCP is three-pronged [1]: (i) the overall flow rate for an MPTCP flow should be at least as high as the highest flow rate a single-path TCP can achieve using any of the sub-flow paths of the MPTCP flow, and the single-path TCP flow that achieves this flow rate is called the best single-path TCP flow; (ii) to be fair to a single-path TCP, the total flow rate for any number of subflows of an MPTCP flow sharing a bottleneck link with a single-path TCP flow should not exceed the flow rate of the single-path TCP flow; and (iii) an MPTCP flow must be able to balance the load among subflow paths [1]. Notable MPTCPs that meet the three-pronged objective include Semi-coupled, LIA, OLIA and Balia [1], [3]-[5]. In this paper, we generally call these MPTCPs Equal bandwidth shared MPTCP (EMPTCP), because they tend to distribute the flow rates evenly among all flows, regardless how many subflows there are in a flow.

In this paper, we argue that the above three-pronged design objective should be augmented with yet another aspect, making it a four-pronged design objective, i.e. to provide a mechanism to incentivize users to use the protocol. Although the first of the three prongs of the above objective does provide some incentive for a user to use EMPTCP over single-path TCP, an EMPTCP may end up discouraging users from using it. This is simply because different paths used by an EMPTCP may be owned by different Internet service providers based on different pricing structures. For example, mobile devices, such as cell phones, are equipped with both WiFi connectivity and cellular data service (e.g., 3G/4G/5G) most of the time. Usually, the WiFi connection is without usage fee and the mobile data service may charge a usage fee based on the amount of data sent/received. Considering the scenario of a social gathering, e.g., a family party, where many guests may want to use their mobile phones to browse the Internet, watch online videos, sending/receiving messages and so on, via the host's WiFi network. This may result in low flow rates seen by and hence, poor Internet experiences for individual guests. When this happens, a guest with a cellular data service may be tempted to turn on an EMPTCP that meets the three-pronged design objective, thinking that by doing so, his/her cellular data service may help prop up the bandwidth needed to gain good experience at the cost of paying a small amount of cellular data service fee. In reality, however, he/she may end up with using the cellular data service almost entirely and receiving a hefty bill later. This may well be the case because an EMPTCP may attain the desired flow rate using the cellular connection only, hence giving up much of the free/low-cost bandwidth on the WiFi side. Obviously, this resource allocation is unfair to the guest who uses EMPTCP, and hence, would discourage him/her from using EMPTCP again in the future. This example clearly indicates that to incentivize users to use MPTCP in the face of multiple paths with different pricing structures, the three-pronged design objective should be augmented to a four-pronged one, with the fourth one being subflow path pricing structure aware.

A naive solution is to simply use a weighted single-path TCP for each subflow and assign a heavier weight to a subflow with a lower price. In this paper, we call this kind of MPTCP weighted MPTCP (WMPTCP). Note that an earlier MPTCP protocol, known as equal-weight MPTCP [3] is a special case of WMPTCP, which assigns the same weight to all the subflow paths. Again, using the above scenario as an example, one may assign, e.g., 75% and 25% weights to subflows using WiFi and cellular, respectively. By doing so, the guest who uses WMPTCP can get 75% of the single-path TCP flow rate from the WiFi side for sure. In the meantime, it can still compete for the bandwidth on the cellular side, but much less aggressively. Unfortunately, however, WMPTCP does not meet the first and the third prongs of the three-pronged design objective, meaning that it may lead to the overall flow rate lower than that of the best single-path TCP flow and cannot balance the load.

Akshit Singhal, Xuan Wang, Zhijun Wang, Hao Che, and Hong Jiang are with the Computer Science and Engineering Department, University of Texas at Arlington, Arlington, TX, United States (e-mail: akshit.singhal2@mavs.uta.edu).

In this paper, we propose an MPTCP that meets the four-pronged design objective. The idea is to combine an EMPTCP with a WMPTCP to come up with a hybrid MPTCP (H-MPTCP). H-MPTCP leverages the ability of WMPTCP to allow price-aware subflow path rate allocation and the ability of EMPTCP to meet the three-pronged objective, hence resulting in its ability to achieve the four-pronged design objective. We implement the proposed protocol in Linux Kernel based on the open source MPTCP source codes [6]. Extensive test results demonstrate that H-MPTCP can indeed achieve the four-pronged design objective. In the meantime, it outperforms EMPTCP, WMPTCP and some well-known MPTCP protocols (i.e., LIA [1] and Balia [4]) in terms of throughput and responsiveness.

II. Background and Motivations

A straightforward but naive approach to end-to-end MPTCP design is to simply run independent end-to-end TCP flows as subflows on different paths. This approach, however, is too aggressive and unfair to single-path TCP flows, and cannot balance the loads among subflow paths. This leads to the design of EWTCP [3], a member in the WMPTCP family. EWTCP attempts to achieve TCP fairness by modifying the previous approach, i.e., reducing the TCP window increase rate by a factor of $w_l = 1/S^2$ for all S TCP-based subflows. However, besides the lack of load balancing capability inherited from the previous approach, EWTCP may lead to inefficient use of networks and flow rate lower than the best single-path TCP. The shortcomings of EWTCP further lead to the design of a coupled congestion control algorithm, also known as Linked Increases Algorithm (LIA) [2], [1]. LIA is purposely designed to meet the three-pronged design objective and standardized by IETF. OLIA [5] improves over LIA in terms of the Pareto optimality. More recently, semi-coupled and Balanced linked adaptation (Balia) algorithm [4] were proposed to strike a good balance between TCP-friendliness, responsiveness and window oscillation, especially to further improve responsiveness when network condition changes. All these solutions (i.e., LIA, OLIA, semi-coupled, and Balia) meet the three-pronged design objective and hence, are members in the EMPTCP family. However, neither the WMPTCP family nor the EMPTCP family is capable of achieving the four-pronged design objective. To demonstrate this, we take a closer look at the family party example. Consider the network topology shown in Fig. 1. Adam is a MPTCP user who can use both WiFi free of charge and cellular with 10 cents per unit bandwidth per hour charge. Ken is a single-path TCP user who only uses WiFi for free. So, we have two users where Adam and Ken can both use WiFi service and Adam can also use cellular service. Assume that the maximum data rates for both Adam's and Ken's applications are 100 bandwidth units. The link bandwidths for both WiFi and cellular connections represent the capacity of the bottleneck link bandwidths of the connections. Now, we consider two different scenarios, where in the first scenario, both WiFi and cellular connections can support up to 100 units of bandwidth and in the second scenario where WiFi has 100 units of bandwidth but cellular has only 5 units of bandwidth. For both scenarios, EMPTCP will attempt to equalize the flow rate allocation between both users, whereas WMPTCP will attempt to allocate flow rates in proportion to their weights. For WMPTCP, we further assume that the weights assignments are 0.75 and 0.25 for WiFi and cellular, respectively. This will allow the subflow using the WiFi network to be allocated roughly 75% of the flow rate of a single-path TCP flow on the same network, resulting in a potentially much reduced flow rate needed for the subflow on the cellular network side.

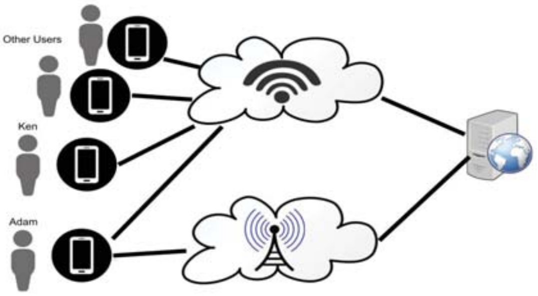


Fig. 1 A network example

Table I gives the flow rate allocation and costs for both Adam and Ken as well as the best single-path TCP flow when Adam uses either EMPTCP or WMPTCP for both scenarios. For the first scenario on the left, EMPTCP for Adam equalizes the flow rate allocation by fully utilizing the cellular connection, yielding to the single-path TCP flow for Ken completely on the WiFi connection. This results in Adam paying for the full usage fee of the cellular link of $10 per hour, whereas Ken enjoys the same flow rate performance for free. In contrast, WMPTCP that has a weight of 0.75 for WiFi and 0.25 for cellular is able to rip 43% of the WiFi link bandwidth and 57% of cellular link bandwidth, resulting in the same total flow rate of 100 units, as the cost of $5.7 per hour, more than 40% lower than the case of EMPTCP. In other words, in this scenario, WMPTCP offers better incentive to users to use MPTCP than EMPTCP by reducing the usage costs. These weights may be set as a function of the pricing models of individual subflow paths to further incentivize users to use it.

For the second scenario on the right in Table I, to equalize the flow rate allocation, EMPTCP has to take up a much bigger chunk of the bandwidth from WiFi connection, i.e. 47.5, to be exact. So, EMPTCP is able to achieve a total bandwidth of 52.5 units. In contrast, with the weight of 0.75, WMPTCP still can only grab 43% of the bandwidth from WiFi connection, leaving it with a total flow rate, 48 units, lower than both EMPTCP and the flow rate of 50 for the best single-path TCP.

The above example clearly demonstrates the need for a new MPTCP, which motivates us to propose H-MPTCP. As we shall demonstrate, in scenario 1, H-MPTCP will automatically select WMPTCP over EMPTCP, whereas in scenario 2, it will automatically select EMPTCP. In either case, H-MPTCP results in better than the best single-path TCP performance at

TABLE I
Flow Rate Allocations for the Example

Scenario	**WiFi=Cellular=100**				**Cellular=5, WiFi=100**			
	Adam			Ken	Adam			Ken
Subflows	Cellular	WiFi	Total	WiFi	Cellular	WiFi	Total	WiFi
Best-case TCP	100($10)	N.A.	100	100	N.A.	50	50	50
EMPTCP	100($10)	0	100	100	5($0.5)	47.5	52.5	52.5
WMPTCP	57 ($5.7)	43	100	57	5($0.5)	43	48	57

a lower cost than EMPTCP, hence, meeting the four-pronged design objective.

Based on the discussion so far and the performance data to be presented in the later sections, Table II provides a summary of the features for some notable end-to-end MPTCP protocols as well as H-MPTCP. As one can see, H-MPTCP possesses the most desirable features among all MPTCPs.

Finally, we note that there are other works that focus on specific aspects of the MPTCP protocol design challenges, e.g., bottleneck detection [12]-[14] and packet scheduling [8]-[11]. However, they are not concerned with new end-to-end MPTCP protocol design, the main focus of this paper.

III. Hybrid Multipath Congestion Control

In this section, we first briefly describe EMPTCP and WMPTCP. Then we introduce H-MPTCP.

A. EMPTCP

For EMPTCP, the congestion window size (W_l) for subflow (l) in each RTT in the slow start phase is the same as that in TCP Reno. The congestion window change for subflow l in each RTT in the congestion avoidance phase is,

$$\Delta W_l = \begin{cases} \frac{W_l}{\tau_l \sum_{j=1}^{S} W_j/\tau_j} & \text{if } cg = 0 \\ -\frac{W_l}{2} & \text{if } cg = 1. \end{cases} \quad (1)$$

where W_j and τ_j are the congestion window size and RRT for subflow j ($j = 1, ..., S$). In this paper, we choose semicoupled mentioned in Balia [4] for EMPTCP, but in general, we can use any members in the EMPTCP family.

From (1), we know that the subflow increase rate is proportional to the ratio of the subflow rate with the overall flow rate, i.e., x_l/x while its decrease rate is the same as that in TCP Reno. It means that the subflow rate increase is slower for a multipath flow with higher flow rate, or the network tries to evenly allocate the flow rate to all flows. For single path TCP, $W_l = W$, then the congestion window change is degenerated to single path TCP Reno.

B. WMPTCP

For WMPTCP, the change in congestion window size (W_l) for subflow (l) in each RTT in the slow start phase is same as that in TCP Reno in the slow start phase. The congestion window change for subflow l with weight (ω_l) in each RTT in the congestion avoidance phase is,

$$\Delta W_l = \begin{cases} \omega_l & \text{if } cg = 0 \\ -\frac{W_l}{2} & \text{if } cg = 1. \end{cases} \quad (2)$$

From (2), we know that the rate increase for a subflow l is proportional to the weight ω_l. If $\omega_l < 1$, the subflow increase rate is smaller than that in a single path TCP flow. If we set $\omega_l < 1$ for any subflow l, then each subflow obtains no more flow rate than that of a single path TCP in a shared link. WMPTCP can allocate more bandwidth than the best single path TCP in some cases, but it may not guarantee the rate of WMPTCP flow always be no less than the best single TCP flow rate as shown in Table I. If $\omega_l = 1$ for any subflow l, the congestion window change of each subflow is the same as in TCP Reno. In this case, a multipath flow is just the combination of S individual single TCP flows.

C. H-MPTCP

To meet the four-pronged design objective, we now design H-MPTCP. We set the weight $1/S \leq 1$ for each subflow of an MPTCP flow with S subflows. With such weight assignment, each subflow can get no more flow rate than that of a single TCP flow rate for a shared link. But the overall flow rate of an MPTCP may have chance to get more rate than its best single path TCP flow. In slow start phase, H-MPTCP behaves the same as TCP-Reno. In the congestion avoidance phase, H-MPTCP selects the larger subflow increase rate from the increase rates of EMPTCP and WMPTCP, i.e.,

$$\Delta W_l = max(\Delta W_l^{Equal}, \Delta W_l^{Weighted}), \quad (3)$$

where ΔW_l^{Equal} and $\Delta W_l^{Weighted}$ are the congestion window increases in each RTT defined in (1) and (2) in case of no congestion, respectively.

Under ”normal” situation, e.g., scenario one in the example given in Section II, WMPTCP is likely to be automatically selected because it is more responsive/aggressive than EMPTCP [4]. EMPTCP will take over only under ”abnormal” situation, e.g., scenario two in Section II, when WMPTCP fails to reach flow rate equal to or higher than the best single-path TCP. In this case, EMPTCP will ensure that flow rate will be balanced to further improve the flow rate performance.

In summary, in the slow start phase, we have,

$$\Delta W_l = \begin{cases} W_l & \text{if } cg_l = 0 \\ -\frac{W_l}{2} & \text{if } cg_1 = 1. \end{cases} \quad (4)$$

and in the congestion avoidance phase, we have,

$$\Delta W_l = \begin{cases} max(\omega_l, \frac{W_l}{\tau_l \sum_{j=1}^{S} W_j/\tau_j}) & \text{if } cg = 0 \\ -\frac{W_l}{2} & \text{if } cg = 1. \end{cases} \quad (5)$$

TABLE II
MPTCP DESIGN OBJECTIVES AND PERFORMANCE PARAMETERS

		Design Objectives			**Performance Evaluation**
Solutions	Best-case TCP	TCP Friendliness	Load Balancing	Adoption Incentive	Responsiveness
EWTCP [3]	No	Yes	No	Yes	High
LIA [1]	Yes	Yes	Yes	No	Low
Balia [4]	Yes	Yes	Yes	No	Medium
H-MPTCP	Yes	Yes	Yes	Yes	High

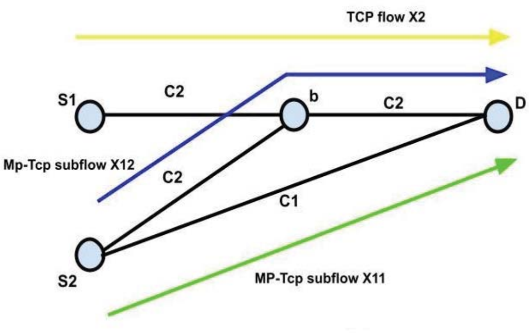


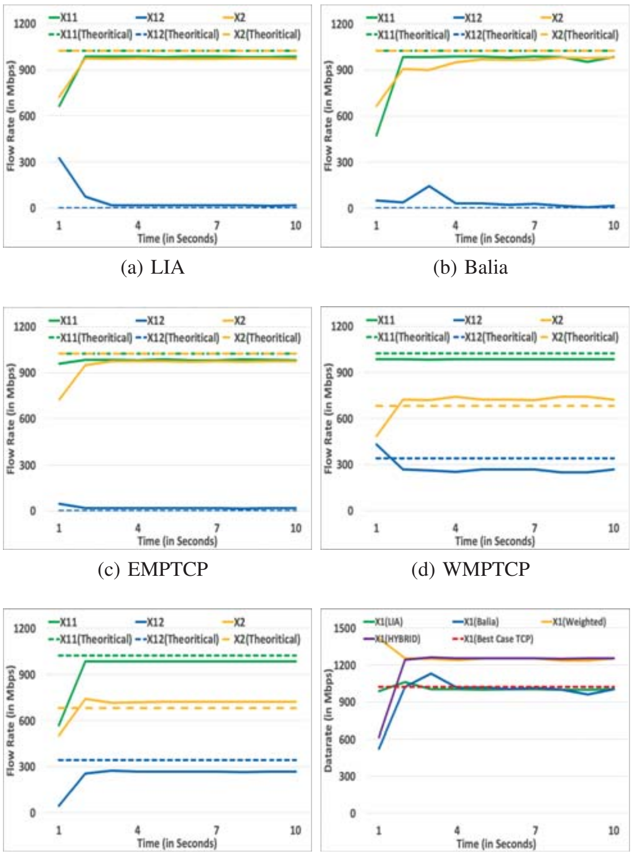
Fig. 2 Network Topology for Performance Evaluation

IV. PERFORMANCE EVALUATION

In this section, we evaluate the proposed H-MPTCP compared to the EMPTCP and WMPTCP and to the existing MPTCP solutions LIA [1] and Balia [4] in Linux kernel implementation on a testbed. The hosts (i.e., S_1, S_2 and D) are Dell Poweredge servers, each equipped with 8-core processors with 10 GB memory and running Linux 16.04. Node b is a Dell N4032F switch with multiple 1 Gbps Ethernet interfaces running Ubuntu 16.04.1 LTS (Linux kernel 4.19.98). The link bandwidth for all the links can be configured at any rate lower than or equal to 1 Gbps through the networking interface traffic control command tc, allowing for the testing of the MPTCP responsiveness to sudden link bandwidth changes. H-MPTCP is implemented by modifying the open source Linux kernel codes of LIA and Balia [6].

We use the network topology as shown in Fig. 2 for both implementations. In the network, source S_1 has a single path TCP flow x_2 running TCP Reno, and source S_2 has an MPTCP flow with two subflows x_{11} and x_{12}. Subflow x_{12} representing Wi-Fi link shares the link from node b to destination D with flow x_2. We set different bandwidth combination of C_1 and C_2 to test the flow rate allocation in different MPTCP solutions. This network topology is also adopted in Balia [4] for the testing of their solution.

We test the performance of the proposed protocol in three different cases, each has a different network setup (i.e., different bandwidth C_1 and C_2). For the test, each source has a huge data to send to the destination D, and each flow can be viewed as an infinite data flow during our test (i.e., the test is finished before the data are fully sent out). In this case, MPTCP user is trying to get the maximum bandwidth available to improve the flow completion time. Each flow reaches its stable rate quickly and we present the first 10-second results in the first two cases and the first 20-second for the third case. The weight values in WMPTCP are set as 1/2 for both subflows x_{11} and x_{12}. Flow x_2 and the two subflows of x_1 (i.e., x_{11} and x_{12}) start at the same time in each test.



(a) LIA (b) Balia

(c) EMPTCP (d) WMPTCP

(e) H-MPTCP (f) MPTCP flow rate comparison

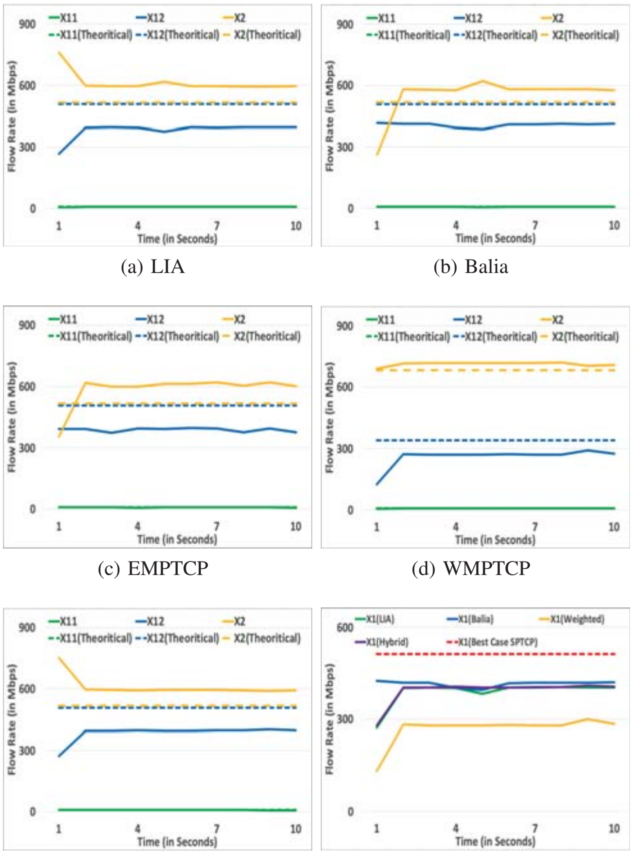
Fig. 3 Performance for MPTCP in Case 1

Case 1: First we test the performance of MPTCP solutions in a symmetric network by set $C_1 = C_2 = 1024$ Mbps (i.e. 1 Gbps link). With this setup, the rate allocation for EMPTCP are $x_{11} = 1024$ Mbps, $x_{12} = 0$ Mbps (i.e., the overall flow rate $x_1 = 1024$ Mbps) and $x_2 = 1024$ Mbps and for WMPTCP are $x_{11} = 1024$ Mbps, $x_{12} \approx 341$ Mbps (i.e., the overall flow rate $x_1 \approx 1365$ Mbps) and $x_2 \approx 683$ Mbps, respectively.

Fig. 3 shows the results of rate allocation (the average rate in a second) in the proposed protocol compared with EMPTCP, WMPTCP, LIA and Balia. The rate of each flow/subflow is also presented using dashed line with the same color as the real measured rate in an MPTCP solution. As LIA and Balia have equally sharing of the bandwidth, they achieve similar rate allocations as EMPTCP, and hence the theoretical rates in EMPTCP is also listed in LIA and Balia results for comparison. From the results, we know that the rate allocations in EMPTCP and WMPTCP closely match to their corresponding theoretical rates with little lower rates. The results indicate that EMPTCP and WMPTCP can really achieve the rate allocation. The lower rate is due to the discrete time control and network condition feedback delay, these make the protocols unable to achieve full bandwidth usage. LIA and Balia also achieve their design objective, i.e., sharing network bandwidth to all flows as even as possible. In this case, WMPTCP allows the MPTCP flow x_1 to obtain higher flow rate than the best single path TCP flow. As WMPTCP has higher flow rate than EMPTCP, H-MPTCP selects WMPTCP and hence achieves the same flow rate allocation as that in WMPTCP (see Fig. 3 (e)).

The overall flow rate of the MPTCP flow (i.e., x_1) and its theoretical best single path TCP flow rate (denoted as dashed red line) are shown in Fig. 3 (f). From the results, we can see that all EMPTCP, LIA, and Balia just achieve close to the theoretical best single path TCP flow rate. Although they make flow x_2 to get higher flow rate, they may have no incentives by usage of MPTCP. WMPTCP/H-MPTCP achieves higher MPTCP flow rate than the best single flow rate. It also benefits flow x_2 in case that x_1 was using a single path TCP as congestion control and chooses the path x_{12}. Hence WMPTCP/H-MPTCP gives more incentives to users to apply MPTCP.

Case 2: Now we test the performance of MPTCP solutions in an asymmetric network by set $C_1 = 8$ Mpbs and $C_2 = 1024$ Mbps. With this setup, the rate allocation for EMPTCP are $x_{11} = 8$ Mbps, $x_{12} = 508$ Mbps (i.e., the overall flow rate $x_1 = 516$ Mbps) and $x_2 = 516$ Mbps and for WMPTCP are $x_{11} = 8$ Mbps, $x_{12} \approx 341$ Mbps (i.e., the overall flow rate $x_1 \approx 349$ Mbps) and $x_2 \approx 683$ Mbps, respectively. In this case, x_1 in WMPTCP has lower flow rate than that of the best single path TCP flow. As EMPTCP can achieve higher flow rate, H-MPTCP selects EMPTCP increase rate and hence achieves the same flow rate allocation as that in EMPTCP, as shown in Figs. 4 (c) and (e). Fig. 4 shows the results of rate allocation in the MPTCP solutions. Again, we see that the rate allocations of EMPTCP/H-MPTCP and WMPTCP closely match to their corresponding theoretical rates. From Fig. 4 (f), we can see that all the protocols can still get more rates (516 vs. 512 Mbps theoretically) than that of the best single path TCP. Although the path S_2 to D has very limited bandwidth but EMPTCP, H-MPTCP, LIA and Balia can still balance the rates and benefit the single path flow x_2. This indicates that MPTCP can be useful to balance the traffic and increase the network utilization. From both case 1 and case 2, we know that H-MPTCP can really guarantee an MPTCP flow to achieve at least the best single path TCP flow rate as that in EMPTCP while it can try to obtain higher flow rate as many as possible and hence can encourage users to use it.



(a) LIA

(b) Balia

(c) EMPTCP

(d) WMPTCP

(e) H-MPTCP

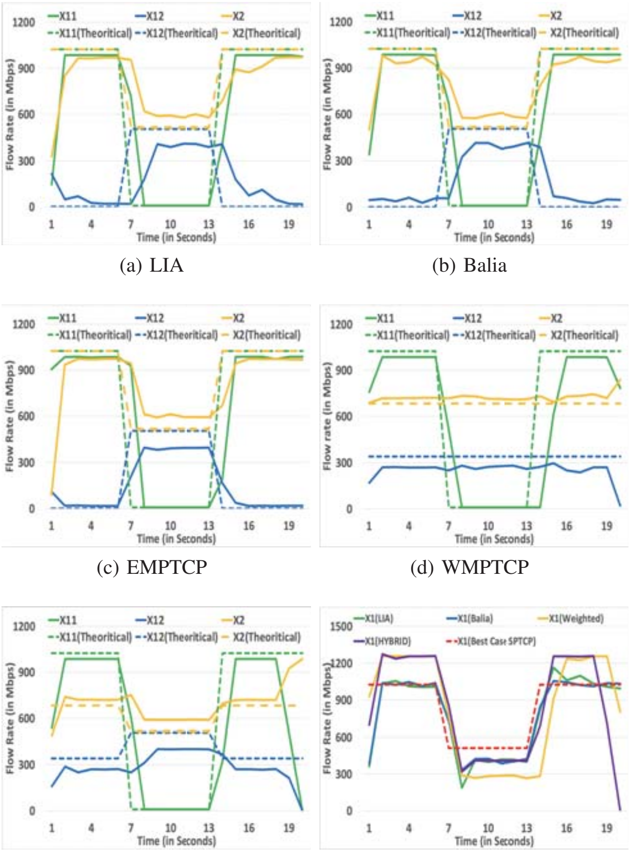
(f) MPTCP flow-rate comparison

Fig. 4 Performance for MPTCP in Case 2

Case 3: Finally, we test the performance of the proposed protocol in a dynamic network environments to see the responsiveness to network changes. In this case, we set the link bandwidth $C_1 = C_2 = 1024$ Mbps in the first 6 seconds and change the bandwidth $C_2 = 8$ Mbps in the next 7 seconds (i.e., from second 7 to second 13) and then C_2 is switched back to 1024 Mbps. As this setup, the rate allocation in the proposed protocol during the first 6 seconds and after the second 13 are the same as that in the case 1, and the rate allocation during the time period between second 6 to second 13 is the same as that in the case 2. For H-MPTCP, it always selects the higher flow from EMPTCP and WMPTCP, and hence it selects WMPTCP congestion control in the first 6 seconds and after second 13 and chooses EMPTCP during the time period from second 6 to second 13.

Fig. 5 shows the results of rate allocation in the five MPTCP solutions. From the results, we see that EMPTCP, WMPTCP and H-MPTCP can closely match to their rates in this dynamical network environment. Now let us look at the flow rate change during the network bandwidth changes.

First, we note that the subflow rate x_{12} in WMPTCP has no change, because the weight for the subflow is only depend



(a) LIA

(b) Balia

(c) EMPTCP

(d) WMPTCP

(e) H-MPTCP

(f) MPTCP flow-rate comparison

Fig. 5 Performance for MPTCP in Case 3

on the number of subflows, and hence the flow rate allocation in the shared link does not change.

Second, in EMPTCP, LIA and Balia, we can see that the subflow rate of x_{11} (denoted as green line) drops to 8 Mbps almost the same time in all the three protocols, this is because subflow x_{11} does not compete the bandwidth with any other flow/subflow. As the subflow rate x_{11} is reduced, the rate of subflow x_{12} is then increased in all the three protocols, and hence the flow rate x_2 should be reduced because of the shared bandwidth with x_{12}. From Figs. 5 (a)-(c), we can see that the transition time for flow x_2 (i.e., yellow line) dropped to its new balanced rate has almost the same time in the three protocols.

Third, from Figs. 5 (a)-(c), we know that the transition time for subflow x_{12} (i.e., blue line) reaching its new balanced rate is different in the three protocols. In LIA and Balia, the transition time takes about 3 seconds from second 6 to about second 9 while in EMPTCP, the transition time is about 2 seconds from second 6 to about second 8. This indicates that EMPTCP can be more quickly to catch up the network condition change to reach the new balanced rate than LIA and Balia. This is because semicoupled has higher responsiveness as compared to LIA and Balia as mentioned in the Balia paper [4].

Fourth, H-MPTCP switches from WMPTCP at second 6 and switches back to Weighted-MPCTP at second 13. The transition time in H-MPTCP is similar as that in EMPTCP as shown in Fig. 5 (d).

From the results, we can also see that H-MPTCP always choose the higher flow rate from EMPTCP and WMPTCP, and hence benefits the user to applying MPTCP. The results show that EMPTCP/H-MPTCP can quickly response to the network condition change and reaches to its new balanced state.

Through the three-case studies, we conclude that the proposed H-MPTCP solution is ready to be applied in today's Internet and can meet the MPTCP design goals and give more incentives for users to apply the MPTCP solutions.

V. Conclusion

In this paper, we propose a Hybrid-MPTCP (H-MPTCP) that always achieves higher flow rate from EMPTCP and WMPTCP and provides a built-in mechanism that can encourage users to apply it over other MPTCPs as well as single-path protocols with different pricing structures. Extensive real Linux implementation test results verify that the proposed H-MPTCP can indeed achieve the design objectives.

References

[1] Wischik, Damon and Raiciu, Costin and Greenhalgh, Adam and Handley, Mark, *Design, Implementation and Evaluation of Congestion Control for Multipath TCP*, Proceedings of NSDI, 2011.

[2] Raiciu, Costin and Handley, Mark and Wischik, Damond, *Coupled Congestion Control for Multipath Transport Protocols*, RFC 6356, 2011.

[3] Honda, Michio and Nishida, Yoshifumi and Eggert, Lars and Sarolahti, Pasi and Tokuda, Hideyuki, *Multipath congestion control for shared bottleneck*, Proceedings of PFLDNeT workshop, 2009.

[4] Peng, Qiuyu and Walid, Anwar and Hwang, Jaehyun and Low, Steven H, *IEEE/ACM Transactions on Networking (TON)*, RFC 6356, 2016.

[5] Khalili, Ramin and Gast, Nicolas and Popovic, Miroslav and Le Boudec, Jean-Yves, *MPTCP is not Pareto-optimal: Performance issues and a possible solution*, IEEE/ACM Transactions on Networking, 2013.

[6] C. Paasch, S. Barre, *Multipath TCP in the Linux Kernel*, http://www.multipath-tcp.org/, 2019.

[7] Kheirkhah, Morteza and Wakeman, Ian and Parisis, Goerge, *Multipath-TCP in ns3*, Proceedings of ACM Workshop on ns-3, 2014.

[8] Mirani, Farhan H and Boukhatem, Nadia and Tran, Minh Anh, *A data-scheduling mechanism for multi-homed mobile terminals with disparate link latencies*, IEEE 72nd Vehicular Technology Conference-Fall, 2010.

[9] Wei, Wenjia and Xue, Kaiping and Han, Jianping and Wei, David S. L. and Hong, Peilin, *Shared Bottleneck-Based Congestion Control and Packet Scheduling for Multipath TCP*, IEEE Transaction on Networking, 2020.

[10] Xu, Changqiao and Li, Zhuofeng and Li, Jinglin and Zhang, Hongke and Muntean, Gabriel-Miro, *Cross-layer fairness-driven concurrent multipath video delivery over heterogeneous wireless networks*, IEEE Transactions on Circuits and Systems for Video Technology, 2014.

[11] Xue, Kaiping and Han, Jiangping and Ni, Dan and Wei, Wenjia and Cai, Ying and Xu, Qing and Hong, Peilin, *DPSAF: Forward prediction based dynamic packet scheduling and adjusting with feedback for multipath TCP in lossy heterogeneous networks*, IEEE Transactions on Vehicular Technology, 2017.

[12] Hassayoun, Sofiane and Iyengar, Janardhan and Ros, David, *Dynamic window coupling for multipath congestion control*, 19th IEEE International Conference on Network Protocols, 2011.

[13] Ferlin, Simone and Alay, Özgü and Dreibholz, Thomas and Hayes, David A and Welzl, Michael, *Revisiting congestion control for multipath TCP with shared bottleneck detection*, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, 2016.

[14] Wei, Wenjia and Wang, Yansen and Xue, Kaiping and Wei, David SL and Han, Jiangping and Hong, Peilin, *Shared bottleneck detection based on congestion interval variance measurement*, IEEE Communications Letters, 2018.