International Conference on Intelligent Technologies & Science - 2025
(ICITS-2025)

Network Performance Enhancement under Multi-Path TCP Approach by Parameter Optimization Scheme

Saurabh Bhutani
Electronics & Communication Engineering,
School of Engineering,
BBD University, Lucknow, India
saurabhbhutani2412@gmail.com

Nitin Jain
Electronics & Communication Engineering,
School of Engineering,
BBD University, Lucknow, India

ABSTRACT- Multi-homing, the process of connecting a host device to numerous networks, is increasingly being used with contemporary Multipath TCP (MPTCP) techniques to improve transmission performance, bandwidth efficiency, and dependability. Traditional TCP, which operates over a single path, frequently exhibits a reduction in performance during brief data transfers due to path-sharing and TCP's aggressive start behavior. This drop happens because data packets are processed independently, causing significant traffic consequences. This study optimizes MPTCP performance by parameter adjustment, with the goal of improving important metrics. Timeouts, retry events, and wait durations, all of which effect packet loss and delays, have been recognized as influential parameters. MATLAB-based simulation identifies ideal parameters in text-based MPTCP applications, minimizing the risk of congestion and preventing loss. The results show that the proposed approach considerably minimizes delays while maintaining high throughput.

Keywords— Multipath TCP, completion time, short flow. packet loss, optimization

I. INTRODUCTION

Integration of radio access technologies with heterogeneous networks supported the growth of multiple network interfaces facility at mobile terminals. It helps to transmit data simultaneously through different interfaces. Recent communication protocols unfortunately use single interfacing simultaneously even though multiple interfaces are present. Theoretical solutions under transmission for exploiting multiple interfaces for available network resource aggregation [1] is proposed for providing better performance under better connectivity and higher bandwidth schemes. However, due to the dissimilar characteristics of path followed for data transmission through multiple path requires final reordering of packets [2] generates latency, packet losses, etc. and degrades the performance in real-time.

In recent years, methods [3] for multi-interface terminals have been developed to improve concurrent transfers across multiple paths, addressing issues like as sharing of load, delivery order, and fairness. These techniques implemented at multiple stages to optimize packet scheduling [4], [5]. Among these, the transport layer is ideal for efficient exploitation of lower-layer services and ensuring application transparency. Notably, TCP is unique among transport layer protocols in its ability to adapt swiftly to congestion across several pathways. Furthermore, it provides significant characteristics such as dependable delivery and fairness assurances.

Multipath transmission uses numerous TCP flows to control data transport. Packets are scheduled and transmitted over many data flows inside a single connection. When each TCP flow is granted its own path, many pathways can be used at the same time. However, differences in latency over these channels might pose problems. Packets having lower sequence numbers transmitted through slower channels may arrive on recipient end after packets having higher sequence numbers send via quicker routes. This causes out-of-order packet delivery, necessitating considerable storage to keep these packets temporarily. Such requests might quickly deplete the minimal buffer space available.

While raising the buffer size causes wasteful memory consumption. A more intelligent solution involves tackling the problem at the sender by providing a scheduling mechanism with little computing complexity. A large variety of TCP-based multipath transmission protocols have been thoroughly researched [6], with some being further described by the IETF Working Group (WG). Multipath TCP (MPTCP) approach adds an extra layer above TCP, which is implemented as a TCP option, to manage many pathways between destinations effectively. A scheduling system is also necessary to guarantee that delivery orders are followed correctly. In MPTCP, the original data stream is segmented and transferred across several TCP connections, known as "TCP subflows."

Although intelligent scheduling techniques at the sender are intended to decrease out-of-order packets, they ignore the effects of packet loss, rendering them ineffective in lossy and diverse networks. Wu et al. [7] investigated packet loss, streaming coding, and energy usage, however their research was largely concerned with mobile video transmission. Network coding [8] mitigates the impact on loss of packet, but not resolve packets that are out-of-order. These techniques omit critical feedback information in acknowledgments, which might give insight into the correctness of past scheduling decisions. As a result, the sender lacks the necessary knowledge to make timely changes to the schedule for the following cycle. This causes scheduling to be done individually for each round, resulting in the accumulation of mistakes across successive rounds.

In this research, we present a technique for identifying the best server and client settings for MPTCP. This approach assigns network settings to each TCP subflow to reduce packet loss and transmission latency. The appropriate settings are chosen based on the TCP properties of each subflow in an MPTCP connection. Furthermore, the technique combines the notion of TCP modeling while accounting for packet loss, making it more adaptable and suited for lossy and heterogeneous networks. The primary aim is to identify an ideal parameter configuration that efficiently avoids packet losses while minimizing transmission delays. The main work contributions in this article summarized below:

1) Implanted a multipath architecture MPTC. Under each subflow, here utilized text data to be transmitted and account for both time offset & packet loss rate.

2) On the basis of above scheme, identified the server and client side parameters that impact transmission response based on accumulated in the simulation results.

3) The Taguchi approach is used to construct experiments with an orthogonal array of network parameter settings, resulting in maximal coverage with the smallest array size. This technique determines the ideal parameter set in fewer trials, providing more exact estimation and decreasing algorithm complexity when compared to the method presented in [9]. It offers an overview of the effects of network parameter changes, allowing for more efficient performance analysis. This is especially beneficial for assessing MPTCP's upgraded architecture under new simulation situations, such as random packet losses and mobile scenarios. The remainder of this paper is arranged as follows: Section II presents a assessment of related literature of several scheduling methods utilized in multipath transmission, with a special emphasis on transport layer approaches. Section III describes the recommended technique. The simulation results under the performance comparisons are reported in section IV. Finally, Section V concludes the article.

II. Related Works

This section summarizes previously suggested MPTCP congestion control algorithms (MPTCP-CCAs), emphasizing their merits and limitations. These algorithms may be divided into three categories: delay-based, loss-based and hybrid approaches. In loss-based CCA, packet loss is the primary determinant of rate of transmission. These algorithms interpret packet losses as a indicator of congestion, which occurs when the bottleneck buffer exceeds capacity. To relieve network congestion, they narrow the congestion window (CWND), slowing data transmission and relieving buffer pressure. Several loss-based CCAs have been built using MPTCP's main design concepts, including LIA [10], OLIA [12], BALIA [13], and D-LIA [11].

Raiciu et al. invented LIA [10], which successfully moves traffic from busy to less congested routes, improving both throughput and fairness. However, its stringent devotion to fairness restricts its capacity to make full use of the underlying network resources. Khalili et al. found a trade-off in LIA in responsiveness and congestion balancing optimization. They created OLIA [12], which combines both features. OLIA continues to suffer with underutilization of its available network resources.

Peng et al. investigated OLIA further and discovered that it was insensitive to network changes in some cases. They proposed BALIA [13], a modified method that aims to increase responsiveness. However, like with its predecessors, BALIA has issues in fully using the underlying network.

Previous MPTCP-CCAs struggled to fully leverage the underlying network, result in inferior throughput [17]. Further, the algorithm concentrated over congestion window (CWND) growing process. To overcome this restriction, D-LIA [11], a loss-based CCA, proposed a dynamic strategy for changing the CWND reduction mechanism. In reaction to packet loss, D-LIA adjusts the CWND reduction factor based on the time between loss events. This method allowed D-LIA to attain better throughput than LIA, BALIA & OLIA. However, it led to an hike in packet losses.

MPTCP-CCAs based on delay take a proactive approach, using delay as an indicator of congestion to avoid queue buildup at bottleneck buffers. These methods decrease round-trip time (RTT) optimizing throughput, making them suited for applications for low-latency. Notable CCAs based on delay management in MPTCP are wVegas [15] & C-MPBBR [14].

Yu et al. created wVegas [15], a MPTCP-CCA with delay-based inspired by TCP Vegas. This approach achieves fine balancing of load by using packet queuing delay as a indicator of congestion. However, rather of conforming to MPTCP's three fundamental design concepts, wVegas focuses on the "congestion equality principle," resulting in a failure to completely achieve MPTCP-CCA design objectives. To solve this issue, a delay-based MPTCP CCA termed as C-MPBBR [14] was developed. Built on single-path TCP BBR [11], C-MPBBR aims to achieve MPTCP's basic design goals while maintaining low latency, high throughput and better fairness. However, the underlying single-path TCP-CCA BBR [11] still has some serious flaws that require additional investigation.

As a result, more advancement to C-MPBBR are expected in the future. Hybrid MPTCP-CCAs, which blend loss- and delay-based methods, have the ability to deliver high throughput while still achieving MPTCP-CCA design requirements. Phuong et al. introduced MCompound [16], a hybrid MPTCP-CCA intended for long-delay, high-speed networks. This approach extends the previously described Compound TCP [18] to multipath settings, demonstrating increased performance in such contexts. However, it falls short of meeting MPTCP's primary design goals, notably fairness with single-path TCP traffic. As a result, there is presently no standardized MPTCP-CCA that meets all design criteria.

III. Motivation

This section describes the motive behind proposed MPTCP for further improvements to handle MPTCP sub-flows successfully. First one is the issue of Fairness that is observed while conducting experiment at different parameter sets. Two primary separate paths (R1 & R2) are taken that are further has two possible paths (R3 & R4) from sender to receiver (Figure 1). Thus, there are four possible path of data subflow. For each path the network parameters are varied, the bottleneck BW is taken at 10 Mbps, to examine delay and packet loss rate. The system is first setup with all accessible pathways between the sender and recipient. After the MPTCP synchronization procedure, which begins at the server node (S) and ends at the client node (C), fairness among subflows is achieved to allow simultaneous usage of the underlying network across all pathways. Another issue driving this research is the Condition for Network Delay, which occurs if bottleneck queue becoming full and packets are dropped. If the queue is empty, the remaining network capacity is underused. Hence, an optimum network parameter setting is necessary to estimate the condition with least delay and packet losses. In the modern state of the Internet, precisely assessing the appropriate network settings to improve MPTCP performance is difficult. The third motivation for our study is the lack of a suitable hybrid MPTCP solution that fits the demands of high-speed networks by integrating loss-based & delay-based approaches. This hybrid method would enable MPTCP to gain a deeper understanding of the underlying network and effectively feed the required amount of data into it. Therefore, this work focuses on developing a hybrid MPTCP that integrates both delay-based and loss-based approaches, resulting in improved performance.

IV. Methodology

In this software instrumentation control toolbox is used for applying functionality using TCP client and server function commands [18, 19]. Two different codes are written for defining operation of client end and server end. The two instances of independent running MatLab is launched on same PC. In this work a MPTCP model is implemented using simulation design on MatLab software. One instance used to run server side operation and another instance for client side in parallel. In the figure 1 the implemented topology is shown. Here S is the server and C is the client. The multipath is established by providing four routers R1, R2, R3 and R4. In figure 1 the subflow of text data is carried out through the four possible routes S-R1-R3-C, S-R1-R4-C, S-R2-R3-C and S-R2-R4-C. The data that is transferred from server to client communication is the text data. In each subflow 10 alpha numeric characters are sent randomly from any of the four paths. In the figure 2 the source side data in text format is shown. This text data is sent from server through router R1 and R4 randomly. In figure 3 the text data subflow from routers R2 and R3. Here we may also observe that packet id is shown in title. Total 20 packets are sent through multipath approach. In this proposed work different parameter set are considered at server and client side for maintaining best QoS by maintaining minimum delay and packet lossees. The parameters that are selected are (1) S1: server waiting time for client to connect (2) S2: delay inserted prior to next retry by server (3) C1: client waiting time for client to connect (4) C2 : delay inserted prior to next retry by client and the response are TS_{req}: Time delay in accepting the request of server by the client, TS_{Write},: Time delay in writing packet from S to C P_{Loss}: Packet loss (0/1), TC_{Delay}: time delay in transmitting data in propagation from S to C and TC_{Read}: Time delay at the client end to read the data. All the response are observed at different combination of parameter set {S1,S2,C1,C2} as shown in table 1. There are nine different orthogonal combinations of S1, S2 C1 and C2 MPTCP opted using Taguchi Method. These are minimum number of combination that cover maximum domain of parameter values for finding the optimum set. S1 and C1 is taken ➔[1.5,2.5,3.5] in sec, S2 and C2 ➔[2,3,4]. In this way 3 values of S1, 3 values of S2 and 3 values of C1 and 3 values of C2 are considered that may create 3x3x3x3=81 combinations but using Taguchi methods only 9 combinations as design of experiment set are generated as orthogonal combination. This may help in finding the optimum solution in faster way by consider small number of solution set that covers maximum search space domain.

Design Table (L9 Orthogonal Array): Below is the experimental plan:

Experiment	S1	S2	C1	C2
1	1.5	2	1.5	2
2	1.5	3	2.5	3
3	1.5	4	3.5	4
4	2.5	2	2.5	4
5	2.5	3	3.5	2
6	2.5	4	1.5	3
7	3.5	2	3.5	3
8	3.5	3	1.5	4
9	3.5	4	2.5	2

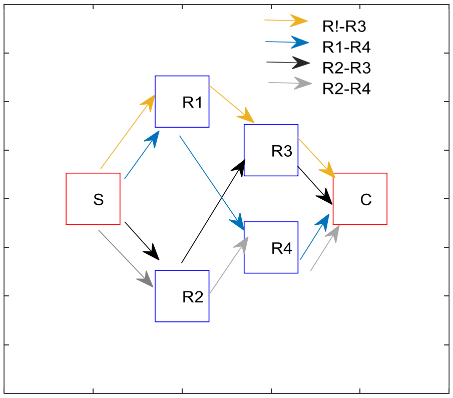

Fig1. MPTCP network architecture representing paths for data subflow

In the figure 4 the text mesg data suflow is shown that are forwarded from server S to client C through route R1 (top) or R2 (bottom). The events with packet loss may be seen as the absent text message at different routes. During the data subflow the performance is measured and shown in figure 5 in terms of request time, write time, read time and propagation delay at different packet id under different experiment performed at specific combination of network parameter set. It may be observed that request time and propogation delay is varying from 0 to $5x10^{-5}$ sec and red write time varying from 0

to 0.01 secs. The variation in performance is further observed in more elaborated way in figure 6 as the boxplot to visualize maximum,
minimum and average value under different experiment id. The lowest sum of T_{req}, T_{write}, T_{read} & T_{delay} is observed for experiment id 9 for the parameter set {S1,S2,C1,C2}➔{3.5, 4, 2.5, 2} . It shows that rrequest and propagation delay is negligible main time consumed in read/write operation. Minimum packet loss of 24% is observed for experiment S5.

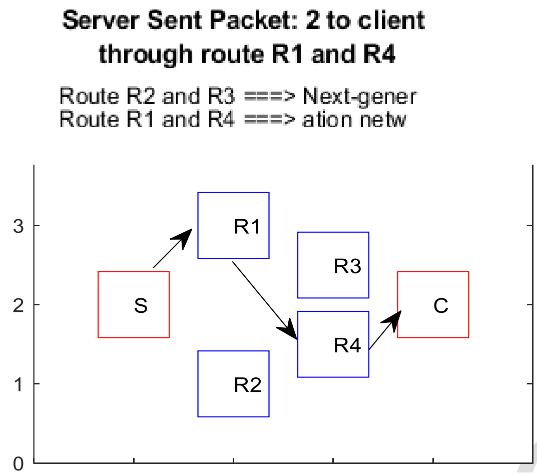
Server Sent Packet: 2 to client
through route R1 and R4
Route R2 and R3 ===> Next-gener
Route R1 and R4 ===> ation netw
3
2
1
0
S
R1
R2
R3
R4
C

Fig2. Transmission of text data packet id 2 from path S-R1-R4-C

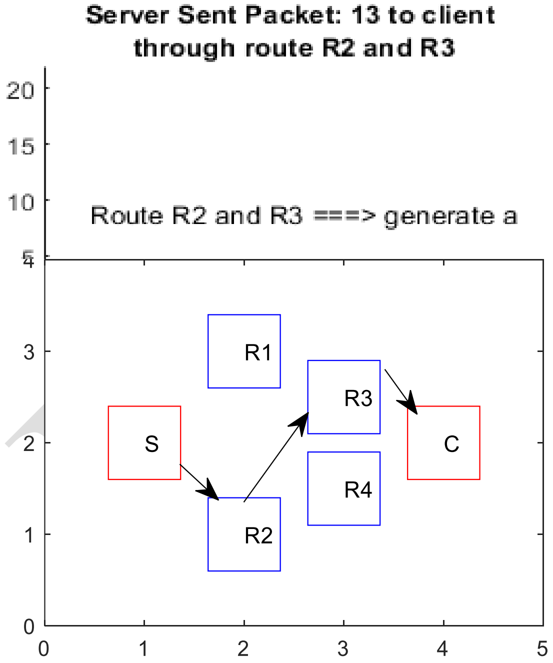
Server Sent Packet: 13 to client
through route R2 and R3
20
15
10
5
Route R2 and R3 ===> generate a
4
3
2
1
0
0
1
2
3
4
5
S
R1
R2
R3
R4
C

Fig3. Transmission of text data packet id 13 from path S-R2-R3-C

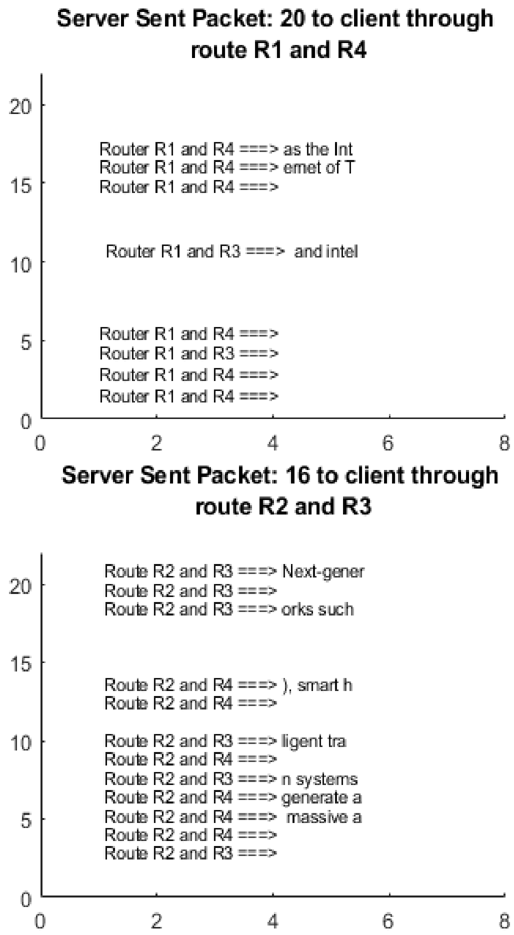

Fig4. Transmission records of text data through router R1 (top) and router R2 (bottom)

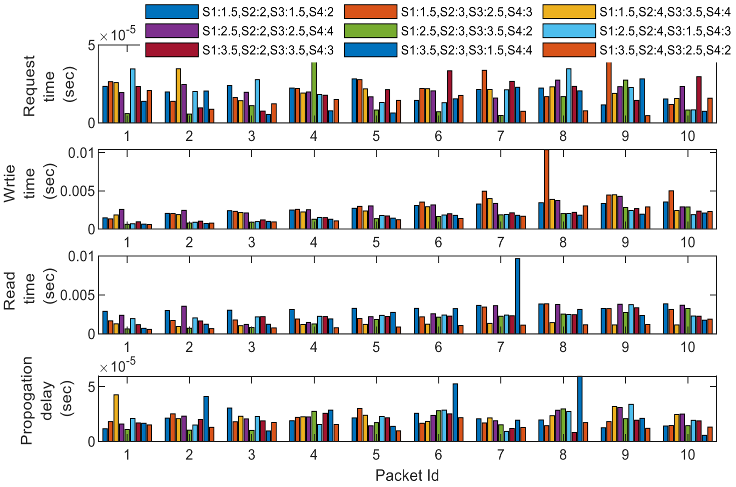

Fig5. Time taken during request, write, read and propagation delay under MPTCP for each packet id under different experiments.

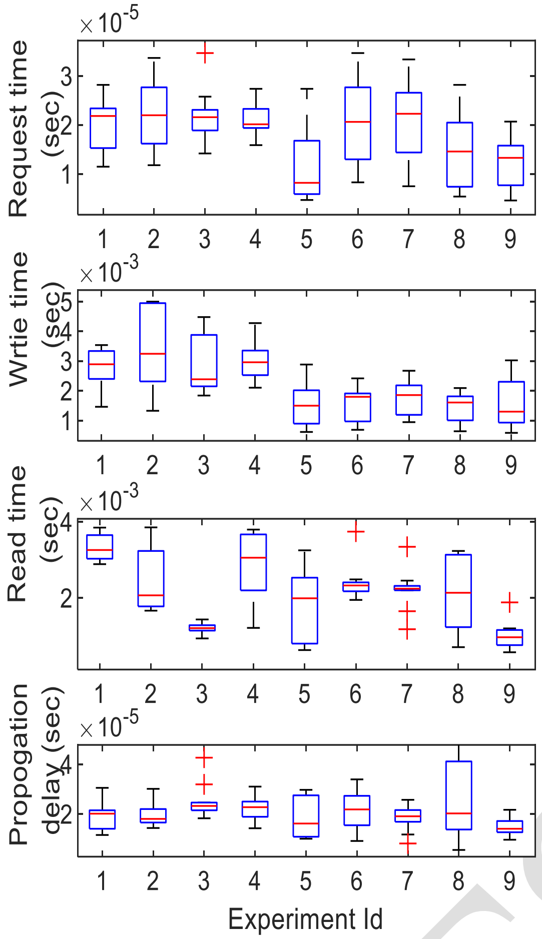

Fig6. Boxplot for showing maximum, minimum range of request, write, read and propagation delay time

V. Conclusions

MPTCP transmission associated packet loss and delay are observed to be uncoupled and independent behavior but causes congestion and reduce the throughput on data share under common bottleneck. In this paper, network parameter identification and optimization performed using orthogonal array selection via Taguchi method for finding under client and server side behavior to minimize issue behind QoS degradation. Using information about path status appropriate parameter set is identified for smoothing the data transmission under simulation platform on MatLab. The strategy reduces the aggressive growth in congestion inside MPTCP, eliminating buffer overflow, which can result in packet losses and timeouts. MatLab simulation results show that the suggested strategy significantly reduces MPTCP transfer completion time by minimizing the number of retransmissions and retransmission timeouts (RTOs).

References

[1] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "Effective delay-controlled load distribution over multipath networks," IEEE Trans. Parallel Distrib. Syst., vol. 22, no. 10, pp. 1730–1741, Oct. 2011.

[2] H. Im, C. Joo, T. Lee, and S. Bahk, "Receiver-side TCP countermeasure to bufferbloat in wireless access networks," IEEE Trans. Mobile Comput., vol. 15, no. 8, pp. 2080–2093, Aug. 2016.

[3] K. Habak, K. A. Harras, and M. Youssef, "Bandwidth aggregation techniques in heterogeneous multi-homed devices: A survey," Comput. Netw., vol. 92, pp. 168–188, 2015.

[4] X. Zheng, Z. Cai, J. Li, and H. Gao, "A study on application-aware scheduling in wireless networks," IEEE Trans. Mobile Comput., vol. 16, no. 7, pp. 1787–1801, Jul. 2017.

[5] P. Sharma, S. Lee, J. Brassil, and K. Shin, "Aggregating bandwidth for multihomed mobile collaborative communities," IEEE Trans. Mobile Comput., vol. 6, no. 3, pp. 280–296, Mar. 2007.

[6] S. Shailendra, R. Bhattacharjee, and S. K. Bose, "MPSCTP: A simple and efficient multipath algorithm for SCTP," IEEE Commun. Lett., vol. 15, no. 10, pp. 1139–1141, Oct. 2011.

[7] J. Wu, C. Yuen, B. Cheng, Y. Yang, M. Wang, and J. Chen, "Bandwidth-efficient multipath transport protocol for quality-guaranteed real-time video over heterogeneous wireless networks," IEEE Trans. Commun., vol. 64, no. 6, pp. 2477–2493, Jun. 2016.

[8] C. Xu, Z. Li, L. Zhong, H. Zhang, and G.-M. Muntean, "CMT-NC: Improving the concurrent multipath transfer performance using network coding in wireless networks," IEEE Trans. Veh. Technol., vol. 65, no. 3, pp. 1735–1751, Mar. 2016.

[9] D. Ni, K. Xue, P. Hong, and S. Shen, "Fine-grained forward prediction based dynamic packet scheduling mechanism for multipath TCP in lossy networks," in Proc. 23rd Int. Conf. Comput. Commun. Netw., 2014, pp. 1–7.

[10] C. Raiciu, D. Wischik, and M. Handley, "Practical congestion control for multipath transport protocols," 2009. [Online]. Available:

[11] K. T. J. Song, Q. Zhang, and M. Sridharan, "Compound TCP: A scalable and TCP-friendly congestion control for high-speed networks," in Proc. PFLDnet 2006, Beijing, China, Feb. 2–4, 2006.

[12] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," Commun. ACM, vol. 60, pp. 58–66, 2017.

[13] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "MPTCP is not Pareto-optimal: Performance issues and a possible solution," IEEE/ACM Trans. Netw., vol. 21, pp. 1651–1665, 2013.

[14] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath TCP: Analysis, design, and implementation," IEEE/ACM Trans. Netw., vol. 24, pp. 596–609, 2014.

[15] T. Lubna, I. Mahmud, and Y.-Z. Cho, "D-LIA: Dynamic congestion control algorithm for MPTCP," ICT Express, vol. 6, pp. 263–268, 2020.

[16] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath TCP," in Proc. 20th IEEE Int. Conf. Netw. Protocols (ICNP), Austin, TX, USA, Oct. 30–Nov. 2, 2012, pp. 1–10.

[17] B.-P. Ha, B.-Y. Tran, T.-A. Le, C.-H. Tran, and P. L. Vo, "A hybrid multipath congestion control algorithm for high-speed and/or long-delay networks," in Proc. 2014 Int. Conf. Adv. Technol. Commun. (ATC 2014), Hanoi, Vietnam, Oct. 15–17, 2014, pp. 452–456.

[18] R. Lübben and J. Morgenroth, "An odd couple: Loss-based congestion control and minimum RTT scheduling in MPTCP," in Proc. 2019 IEEE 44th Conf. Local Comput. Netw. (LCN), Osnabrueck, Germany, Oct. 14–17, 2019, pp. 300–307.

[19] S. Sahay, A. Banoudha, and R. Sharma, "On the use of ANFIS for ground water level forecasting in an alluvium area," Int. J. Res. Dev. Appl. Sci. Eng., vol. 2, no. 1, 2012.