International Journal of Computational and Experimental Science and ENgineering (IJCESEN)
Vol. 11-No.2 (2025) pp. 2400-2407
http://www.ijcesen.com
ISSN: 2149-9144

Research Article

Enhancing Efficiency with Adaptive Optimized Balancing Factor for MPTCP Congestion Control Using Deep Deterministic Policy Gradient: EE-AOBF-MPTCP-DDPG

K. Raghavendra Rao[1,*], Ruth Ramya Kalangi[2]

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur District, Andhra Pradesh, India
* **Corresponding Author Email:** raghava514.k@gmail.com, - **ORCID:** 0009-0007-6940-0431

[2]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Green Fields, Vaddeswaram, Guntur District, Andhra Pradesh, India
Email: ramya_cse@kluniversity.in - **ORCID:** 0000-0002-7399-0186

Article Info:

DOI: 10.22399/ijcesen.1409
Received : 21 January 2025
Accepted : 13 March 2025

Keywords :

Congestion window,
Threshold,
Multipath,
Machine Learning,
Balancing factor,
Sub-flow.

Abstract:

MPTCP is rapidly emerging as one of the most advanced networking protocols. Standardized by the IETF as an extension of TCP, it enables seamless communication across multiple interfaces from source to destination. Despite its potential, existing multipath congestion control mechanisms face significant challenges due to the diverse QoS characteristics of heterogeneous interfaces. While recent algorithms primarily emphasize enhancing the growth dynamics of the congestion window (CWND), the reduction mechanisms remain largely overlooked. Furthermore, conventional congestion control approaches often rely on manual adjustments, which are insufficient in highly dynamic network environments. Given the demonstrated success of machine learning algorithms across industries such as IoT, video streaming, and autonomous vehicles, this study introduces the Deep Deterministic Policy Gradient Multi-Path (DDPG-MP) framework. This innovative approach dynamically optimizes congestion control using a balancing factor, enabling adaptive and efficient performance in multipath networking environments.

1. Introduction

Bluetooth, Wi-Fi, and 4G/5G are integral communication interfaces in contemporary devices. However, the conventional Transmission Control Protocol (TCP) operates over a single interface, resulting in suboptimal utilization of network resources [1]. To address this limitation, the Internet Engineering Task Force (IETF) introduced the Multipath TCP (MPTCP) standard. MPTCP extends the capabilities of traditional TCP by enabling simultaneous utilization of multiple interfaces for a single application, thereby improving network robustness, efficiency, and overall performance [2].
The primary approach to designing and implementing multipath communication, which has garnered considerable attention in recent years, is congestion control. Among the proposed methods for MPTCP congestion control is the Linked Increases Algorithm (LIA). LIA [3], the default congestion control mechanism in MPTCP, evaluates available interfaces based on metrics like round-trip time (RTT) and packet loss. However, LIA's behavior is relatively aggressive compared to single-path TCP (SP-TCP), prompting the development of alternative algorithms such as OLIA [4], BALIA [5], and DLIA [6] to improve efficiency and fairness. These algorithms are fundamentally based on the Additive Increase Multiplicative Decrease (AIMD) principle, which serves as the backbone of TCP congestion control. Unlike DLIA, most modern algorithms focus on expanding mechanisms derived from traditional TCP approaches. The congestion window (CWND) plays a vital role in maximizing bandwidth utilization and shows a strong correlation with packet loss. A larger CWND generally leads to better bandwidth usage over time. Ideally, the CWND after a packet loss event should be close to its pre-loss value to ensure optimal performance. However, conventional methods that halve the

CWND upon packet loss often reduce it to a suboptimal size, leading to inefficient utilization of network resources. [7] He mostly focused on fairness and how it would be influenced. He extends fairness from a single road to multiple paths. [8] has concentrated on coupled congestion control algorithms.

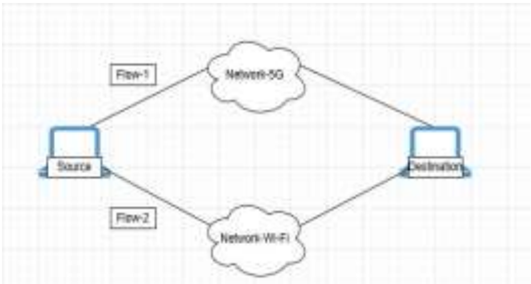

Figure. 1. *MPTCP connections from source to destination.*

Figure 1 depicts the simple architecture of the MPTCP. There are two flows that connect the source and destination. One flow will send data over one interface, such as LTE, while another interface will transfer data over another interface, such as Wi-Fi.

Raiciu et al [9] established three fundamental design objectives for the MPTCP congestion control algorithm to achieve enhanced performance and equitable resource utilization:

1. Maximizing Throughput: A multipath flow should deliver performance that is at least equivalent to a single-path flow on the most optimal available paths. This ensures an economic rationale for adopting multipath transmission.

2. Ensuring Non-Disruption: A multipath flow must not consume more bandwidth on any individual path than a single-path flow would on that route. This guarantees minimal interference with other network flows.

3. Congestion Balancing: Provided the first two criteria are met, a multipath flow should strive to redistribute traffic away from heavily congested paths, promoting efficient use of network resources.

2. Back Background and Problem Formulation

2.1 Background

MPTCP has been widely studied to develop efficient congestion control mechanisms. In [7], an uncoupled congestion control (CC) approach was introduced, where each flow adjusts its CWND independently, without awareness of other flows. However, this method does not address the problem of traffic redistribution from highly congested paths to less congested ones. To tackle this limitation, [8] proposed a fully connected model, which resolved the initial concern but led to a new issue: continuous traffic shifting between flows. The LIA approach was then introduced to address this challenge, though it tends to exhibit greater aggressiveness compared to single-path TCP.As shown in Figure 2, the MPTCP protocol stack is divided into sub-flows. It can transport data over several flows, but from outside the application, it appears to be a single flow. The transport layer has two internal layers: the MPTCP connection level and the sub-flow level.

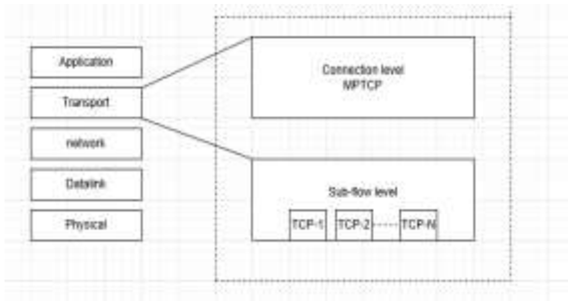

Figure. 2. *Protocol stack of MPTCP.*

The limitations of the LIA approach were identified by Khalili [3], who observed its aggressive behavior when compared to single-path TCP. To address this, he proposed Opportunistic LIA (OLIA) as a potential solution [4], demonstrating that MPTCP could achieve Pareto optimality. However, OLIA also faces challenges when network conditions fluctuate. To mitigate these issues, BALIA [5] was introduced, offering an improved approach. Building on this, DLIA [6] further proposed a method to reduce congestion based on dynamic value.

Farinaz et al. delved deep into the world of MPTCP, exploring the intricate dynamics of both coupled and uncoupled congestion control (CC) algorithms. His research redefined the concept of fairness, examining it not just at the flow level but also across the entire network. With a sharp focus on performance, he meticulously analyzed how coupled CC algorithms adapt and thrive in diverse network conditions, uncovering insights that bridge efficiency and equity in data flow management [10]. According to Wei et al., the current congestion control mechanism is suboptimal since it employs different congestion controls at various bottlenecks. They did this by using loss correlation and delay correlation between two flows. He has suggested utilizing an explicit congestion notification technique to discover common bottlenecks [11].Wenzhong et al. explored the challenges of congestion control arising from path heterogeneity and proposed a learning-based framework to address these complexities. Their solution employs machine learning algorithms [12] to adaptively manage the diverse characteristics of network paths. However, the approach does not account for the critical aspect of TCP-friendliness, leaving room for further refinement. Conducted a comprehensive evaluation of three congestion control algorithms to determine their suitability for high-speed railway networks [13-15]. The study examined the performance of both coupled and uncoupled congestion control mechanisms. The findings highlighted that among the evaluated approaches, equally weighted congestion control achieved the most favorable results, demonstrating its effectiveness in such environments. Soheil at el [16] noted the shortcomings of stale learning-based congestion control and suggested a hybrid approach to deal with the challenges he saw, such as overhead, network performance issues, and convergence problems.

Kefan et al. [17]. The current wireless/wired scenarios are becoming increasingly complex; the assumptions offered by existing tcp variations may no longer be valid, hence we developed a model-free congestion control technique based on Deep Reinforcement Learning. The model will learn from its previous experiences in the form of measurable attributes. Maliha et al. [18] provided an analysis and comparison of traditional and contemporary methods for congestion control and scheduling mechanisms in MPTCP. While MPTCP has demonstrated superior performance compared to TCP, it also has its own limitations. To address these challenges, the study explored and proposed potential solutions for improving MPTCP.Taeyun et al. [19] highlighted the inefficiency of existing congestion control techniques in addressing the demands of IoDST, where reliable communication is critical. Traditional algorithms regulate data transmission rates using predefined rules, which limit their adaptability and effectiveness. To enhance throughput and optimize data transfer in IoDST, the study proposed an innovative and intelligent communication approach. According to Shiva et al. [20], the current implementations of MPTCP, such as LIA, OLIA, and BALIA, are solely focused on congestion control and do not incorporate packet scheduling. However, Shiva has suggested Deep Q learning, which can handle scheduling and, in the event of congestion, regulate it. He uses the Policy Gradient technique as his methodology.

2.2 Contribution

Identify applicable sponsor/s here. If no sponsors, delete this text box (sponsors).

Building on the prior analysis, our approach prioritizes efficient resource utilization by implementing a dynamic reduction factor for the CWND during packet loss events. This method emphasizes maintaining the CWND close to its optimal value, resulting in enhanced throughput and more effective bandwidth utilization.

3. Deep Reinforcement Algorithm

3.1 Markov Decision Process

This section introduces the development of the system's learning process, beginning with the formulation of a Markov Decision Process (MDP). We then employ the Deep Deterministic Policy Gradient (DDPG) algorithm to compute the optimal congestion balancing factor. Traditional congestion control techniques rely heavily on complex, manual configurations, often resulting in inefficient resource utilization while attempting to maintain the CWND at its optimal value. Drawing inspiration from the advancements in machine learning for congestion control, this work adopts a more efficient approach. By leveraging Deep Reinforcement Learning, we dynamically manage the CWND reduction factor, optimizing the balancing factor to achieve adaptive and resource-efficient congestion control.

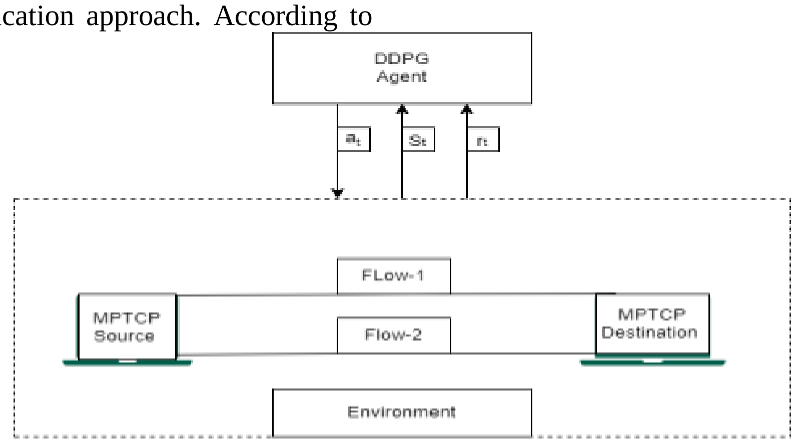

Figure. 3. *Markov Decision Process model.*

At each step, as illustrated in Figure 3, the system diligently monitors the environmental state St and computes the congestion management balancing factor based on the proposed methodology. By leveraging this balancing factor, the system dynamically adjusts and reduces the CWND, ensuring an optimized response. Upon execution of this action, the intelligent system is immediately rewarded. The congestion window is characterized through parameters including the current CWND, historical CWND, prior threshold, congestion ratios, and threshold metrics, providing a comprehensive depiction of network behavior.

Definition: the components of the Intelligent system

State: The state represents the current network conditions:

S = {Bandwidth (B), Delay (D), Loss Rate (L), CWND_curr, CWND_prev, U, Thresh_prev}

Action (A):

The action adjusts the Congestion Window (CWND) using the balancing factor θ:

A=θ

Reward (R):

The reward incentivizes high throughput, low delay, and low packet loss:

$$R = \text{Throughput} - \alpha_1 \cdot \text{Delay} - \alpha_2 \cdot \text{Loss Rate}$$

Where α1 and α2 are weighting factors.

3.2 Deep Deterministic Policy Gradient

A continuous action space is crucial for effectively addressing congestion control in real-world environments. Recent advancements in Deep Reinforcement Learning (DRL) algorithms have showcased significant success in tackling complex challenges across various domains. In this research, we adopt the Deep Deterministic Policy Gradient (DDPG) methodology to compute the optimal balancing factor, facilitating rapid convergence to the ideal congestion window (CWND) value with precision and efficiency.

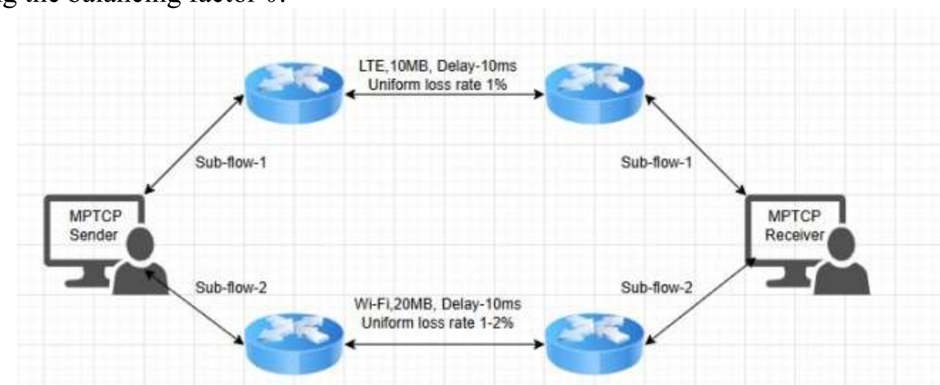

Figure. 4. *MPTCP topology used in the system.*

Figure 4 depicts a network having a source and a destination connected by two flows. Each flow has unique properties. This was the topology we utilized throughout the experiment. TCP behavior indicates that consecutive congestion events are likely to occur when network conditions change. In such scenarios, the Congestion Window (CWND) must be rapidly reduced to identify a new optimal CWND that aligns with the altered network state. Conversely, in the presence of congestion, TCP must ensure efficient bandwidth utilization to achieve an optimal CWND. The congestion window plays a critical role in determining the updated CWND. To address these challenges, we propose a Reinforcement Learning-based Balancing Factor, termed the **Optimized Balancing Factor for MPTCP Congestion Control (OBF2-MPTCP-CC-DDPG).** This approach leverages Deep Reinforcement Learning to dynamically adjust the CWND, optimizing MPTCP performance in dynamic network environments.

Furthermore, for the increase mechanism, we employ the default MPTCP CC method, LIA.

For each sub-flow r, we raise the CWND wr per ACK by

$$w_r = \min\left\{\left(w_r + \frac{\alpha}{w_r}\right), \left(w_r + \frac{1}{w_r}\right)\right\} \quad (1)$$

Where

$$\alpha = w\left(\frac{\frac{\max_r \sqrt{w_r}}{RTT_r}}{\sum_r \frac{w_r}{RTT_r}}\right)^2$$

Additionally, we suggest reducing wr after every loss occurrence by multiplying by a balancing factor. For example, on each sub-flow wr, on each loss event,

1. Compute α:

$$\alpha = \frac{\mathrm{CWND}_{\mathrm{curr}}}{\mathrm{CWND}_{\mathrm{prev}}} \quad (2)$$

2. Compute β:

$$\alpha = \frac{\mathrm{CWND}_{\mathrm{curr}}}{\mathrm{T}hresh_{\mathrm{prev}}} \quad (3)$$

3. Compute γ:

$$\gamma = |\alpha - \beta| \quad (4)$$

4. Utilization Ration U:

$$U = \frac{\text{Achieved Throughput}}{\text{Available Bandwidth}} \quad (5)$$

5. Compute θ:

$$wr = \max\left\{\theta, \frac{wr}{2}\right\} \quad (6)$$

6. Update wr:

$$wr = \max\left\{\theta, \frac{wr}{2}\right\} \quad (7)$$

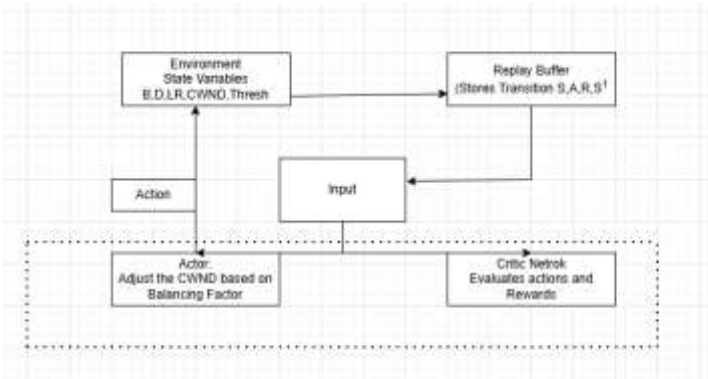

Figure. 5. *Flow representation of DDPG algorithm.*

As shown in Figure 5, the intelligent system transfers the data to the replay buffer after evaluating the variables and monitoring the environment's state. The input field feeds data to both the actor and critic networks. The actor adjusts the CWND based on the balancing factor, influencing the next state through its actions. Meanwhile, the critic evaluates the agent's actions, assigning rewards positive or negative accordingly.

Algorithm 1 Deep Deterministic Policy Gradient (DDPG) Algorithm for Congestion Control using Balancing Factor.

1. Initialization:

- Initialize the Actor network μ(s) and Critic network Q (s, a) with random weights.
- Initialize target networks μ'(s) and Q'(s, a) as copies of the Actor and Critic networks.
- Initialize the replay buffer R.
- Set hyperparameters: learning rates, discount factor γ, target update rate τ, and exploration noise.

2. Training Loop:

For each episode:

1. Reset the environment and observe the initial state S.

2. For each step within the episode:

- Select an action A:
  A = μ(s) + exploration noise

- Execute the action and observe the next state S', reward R, and any packet loss.

- Compute the balancing factor θ:

a) Compute α:

$$\alpha = \frac{\mathrm{CWND}_{\mathrm{curr}}}{\mathrm{CWND}_{\mathrm{prev}}}$$

b) Compute β:

$$\alpha = \frac{\mathrm{CWND}_{\mathrm{curr}}}{\mathrm{T}hresh_{\mathrm{prev}}}$$

c) Compute γ:

$$\gamma = |\alpha - \beta|$$

Utilization Ratio (U) (Ratio of achieved throughput to available bandwidth):

$$U = \frac{\text{Achieved Throughput}}{\text{Available Bandwidth}}$$

Ensure 0 ≤ U ≤ 1, i.e., the value should be between 0 and 1.

d) Compute θ:

$$\theta = \left(\frac{CWND_{curr}}{2}\right) + \left(\gamma \bmod \frac{\mathrm{CWND}_{\mathrm{curr}}}{2}\right) + (\gamma.U)$$

$$wr = \max\left\{\theta, \frac{wr}{2}\right\}$$

- Store the transition (S, A, R, S') in the replay buffer.
- If the replay buffer contains enough samples:

a) Sample a mini-batch of transitions (S, A, R, S') from the replay buffer.
b) Compute the target value y:

$$y = R + \gamma . Q'(S', \mu'(S'))$$

c) Update Critic Network:
a. Minimize the loss:

$$L = \frac{1}{N}\sum(y - Q(S, A))^2$$

d) Update Actor Network
a. Using the Policy gradient

$$V_{\phi}^{\mu} J = \frac{1}{N}\sum \nabla_a Q(S, A) \nabla_{\phi}^{\mu} \mu(S)$$

Perform a soft update of the target networks:
e) Perform Soft Updates:
a. Update Critic Target Network:

$$\phi_{Q'} \leftarrow \tau . \phi_Q + (1 - \tau) . \phi_{Q'}$$

b. Update Actor Target Network

$$\phi_{\mu'} \leftarrow \tau . \phi_\mu + (1 - \tau) . \phi_{\mu'}$$

Repeat Until Convergence:

- Continue for a fixed number of episodes or until the policy achieves optimal performance.

4. Performance Eveluation

4.1 Emulation scenario and setup

We integrated the OBF2-MPTCP-CC-DDPG algorithm into the NS3 simulator and the Linux kernel for comprehensive evaluation. The algorithm was assessed and compared against LIA, OLIA, BALIA, and DLIA in two distinct emulation scenarios. In the first scenario, the evaluation focused on OBF2-MPTCP-CC-DDPG's utilization of the underlying network, while the second scenario assessed its fairness compared to other congestion control algorithms (CCAs). Two sub-flows were considered, designated as path 1 and path 2. Metrics such as CWND, alpha, theta, and gamma were captured using a TCP probe. Additionally, both the total number of retransmitted segments and the segment distribution across sub-flows were monitored using ifstat and netstat tools. The scheduling algorithm employed is based on the "Design and Implementation of Dynamic Packet Scheduling with Waiting Time Aware (DPSW2A) [14]."The emulator experiment was executed over a duration of 360 seconds. For each CCA type, a minimum of 30 emulation runs were conducted for every scenario to ensure comprehensive evaluation. For author/s of only one affiliation (Heading 3): To change the default, adjust the template as follows.

5. Results and discussion

We conducted extensive emulation experiments to evaluate the performance of the proposed OBF2-MPTCP-CC-DDPG algorithm. The values of γ and U were derived based on the dynamically obtained α and β parameters during the controlled tests. Both α and β exhibited variability across experiments, with other parameters also fluctuating dynamically throughout each test. The γ and U parameters play a critical role in preventing the congestion window from dropping to the current CWND value.
We are currently evaluating the performance of OBF2-MPTCP-CC-DDPG against advanced congestion control algorithms such as LIA, OLIA, BALIA, and DLIA. The "Aggregate Benefit (Ag_Bf)" metric, as defined in [15], has been utilized for this comparison.

$$Ag\_Bf = \begin{cases} \dfrac{G - C_{\max}}{\sum_{i=1}^{n} C_i - C_{\max}} & , if \;\; \mathrm{G} \geq \mathrm{C}_{\max} \\ \dfrac{G - C_{\max}}{C_{\max}} & , if \;\; \mathrm{G} < \mathrm{C}_{\max} \end{cases}$$

Figures 6, 7, and 8 present a comparison between the proposed methodology and state-of-the-art algorithms. Experimental results demonstrate that the proposed congestion control algorithm outperforms existing methods across all evaluated metrics, including throughput, aggregate benefit, and Jain's Fairness Index.

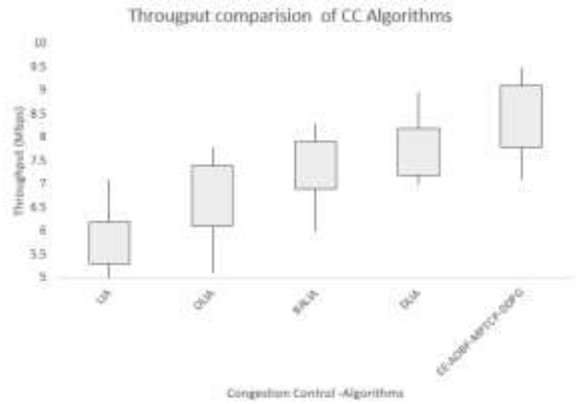

Figure. 6. *Throughput performance of LIA, OLIA, BALIA, DLIA & Proposed CC algorithms.*

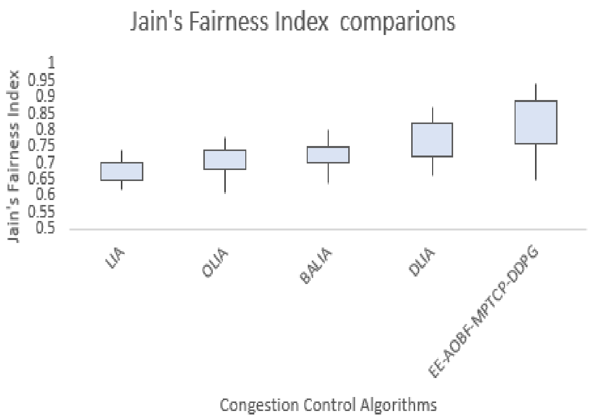

Figure. 7. *Jain's Fairness Index performance of CC algorithms.*

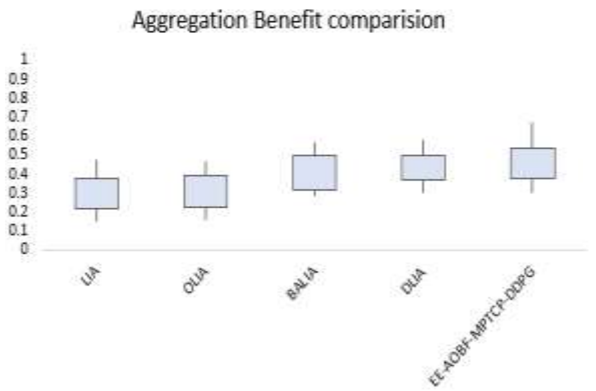

Figure 8. *Performance of state-of-the-art and proposed algorithms Aggregation Benefit.*

6. Conclusion

In this work, we introduce a novel congestion control technique leveraging a machine learning model, designed to seamlessly integrate with existing MPTCP implementations. By dynamically adjusting the congestion window (CWND) reduction factor based on real-time network metrics, we aim to approach an optimal value without compromising the performance benefits of the system. The proposed OBF2-MPTCP-CC-DDPG method demonstrated significant improvements in throughput and fairness during emulation, outperforming classical congestion control algorithms such as LIA, OLIA, BALIA, and DLIA, while preserving their key advantages. This methodology is adaptable to any congestion control framework.
Given the inherently dynamic nature of modern networks, further measurements are required to identify the most optimal values. Future work will extend the proposed approach by incorporating additional metrics to enable a more comprehensive comparison with state-of-the-art methodologies, providing deeper insights into its potential for broader applicability.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] Barré, S., Paasch, C. and Bonaventure, O., (2011). Multipath TCP: from theory to practice. In *NETWORKING 2011: 10th International IFIP TC 6 Networking Conference, Valencia, Spain, May 9-13, 2011, Proceedings, Part I 10* (pp. 444-457). Springer Berlin Heidelberg.

[2] Raiciu C, Paasch C, Barre S, Ford A, Honda M, Duchene F, Bonaventure O, Handley M. How hard can it be? designing and implementing a deployable multipath {TCP}. *In9th USENIX symposium on networked systems design and implementation* (NSDI 12) 2012 (pp. 399-412).

[3] Wischik, D., Raiciu, C., Greenhalgh, A. and Handley, M., (2011). Design, implementation and evaluation of congestion control for multipath {TCP}. In *8th USENIX Symposium on Networked Systems Design and Implementation (NSDI 11).*

[4] Khalili, Ramin, Nicolas Gast, Miroslav Popovic, and Jean-Yves Le Boudec. (2013). MPTCP is not Pareto-optimal: Performance issues and a possible solution." *IEEE/ACM Transactions On Networking* 21(5); 1651-1665.

[5] Peng, Qiuyu, Anwar Walid, and Steven H. Low. (2013) Multipath TCP algorithms: Theory and design." *ACM SIGMETRICS Performance Evaluation Review* 41(1): 305-316.

[6] Lubna, T., Mahmud, I. and Cho, Y.Z., 2020. D-LIA: Dynamic congestion control algorithm for MPTCP. *ICT Express*, *6*(4),;263-268.

[7] Becke, M., Dreibholz, T., Adhari, H. and Rathgeb, E.P., (2012), June. On the fairness of transport protocols in a multi-path environment. In *2012 IEEE*

International Conference on Communications (ICC) (pp. 2666-2672).

[8] Raiciu, C., Handley, M. and Wischik, D., (2011). *Coupled congestion control for multipath transport protocols* (No. rfc6356).

[9] Raiciu, Costin, Damon Wischik, and Mark Handley (2009). Practical congestion control for multipath transport protocols. *University College London, London/United Kingdom, Tech. Rep*

[10] Jowkarishasaltaneh, Farinaz, and Jason But. (2022). An Analysis of MPTCP Congestion Control. In *Telecom*, vol. 3, no. 4, pp. 581-609. MDPI.

[11] Wei, W., Xue, K., Han, J., Wei, D.S. and Hong, P., (2020). Shared bottleneck-based congestion control and packet scheduling for multipath TCP. *IEEE/ACM Transactions on Networking*, *28*(2);653-666.

[12] Li, Wenzhong, Han Zhang, Shaohua Gao, ChaojingXue, Xiaoliang Wang, and Sanglu Lu. (2019) SmartCC: A reinforcement learning approach for multipath TCP congestion control in heterogeneous networks. *IEEE Journal on Selected Areas in Communications* 37(11): 2621-2633.

[13] Yu, Chengxiao, Wei Quan, Nan Cheng, Shihua Chen, and Hongke Zhang. (2019). Coupled or uncoupled? Multi-path TCP congestion control for high-speed railway networks. In *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 612-617. IEEE.

[14] Rao, K. Raghavendra, and B. N. Jagadesh. (2021). Design and Implementation of Dynamic Packet Scheduling with Waiting Time Aware: DPSW2A." *International Journal of Advanced Computer Science and Applications* 12(11).

[15] Paasch, Christoph, Ramin Khalili, and Olivier Bonaventure. (2013). On the benefits of applying experimental design to improve multipath TCP. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pp. 393-398.

[16] Abbasloo, Soheil, Chen-Yu Yen, and H. Jonathan Chao. (2020). Classic meets modern: A pragmatic learning-based congestion control for the internet. In *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, pp. 632-647.

[17] Xiao, K., Mao, S. and Tugnait, J.K., 2019. TCP-Drinc: Smart congestion control based on deep reinforcement learning. *IEEE Access*, *7;*11892-11904.

[18] Maliha, Maisha, Golnaz Habibi, and Mohammed Atiquzzaman. (2023). A Survey on Congestion Control and Scheduling for Multipath TCP: Machine Learning vs Classical Approaches. In *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 49-61. IEEE,

[19] Ha, Taeyun, Arooj Masood, Woongsoo Na, and Sungrae Cho. (2023) Intelligent multi-path TCP congestion control for video streaming in internet of deep space things communication. *ICT Express* 9, no. 5: 860-868.

[20] Maliha, Maisha, Golnaz Habibi, and Mohammed Atiquzzaman. (2023).A Survey on Congestion Control and Scheduling for Multipath TCP: Machine Learning vs Classical Approaches. In *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 49-61.