Proceedings of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 265–270 DOI: 10.15439/2024F1146 ISSN 2300-5963 ACSIS, Vol. 39

Congestion Control in Streaming Services with an On-Off MPTCP Algorithm

Łukasz Piotr Łuczak
0000-0003-0892-7276
Institute of Information Technology
Lodz University of Technology, Łódź, Poland
Email: lukasz.luczak@dokt.p.lodz.pl

Przemysław Ignaciuk
0000-0003-4420-9941
Institute of Information Technology
Lodz University of Technology, Łódź, Poland
Email: przemyslaw.ignaciuk@p.lodz.pl

***Abstract*—In this paper, by adopting the analytical framework of dynamical systems theory, a new congestion control (CC) algorithm for Multipath TCP (MPTCP) streaming services is developed. The proposed nonlinear algorithm, following an on-off principle, is formally demonstrated robust with respect to variable network conditions. It maintains stream consistency at the negotiated video rate despite *a priori* unknown delay fluctuations. Also, precise guidelines for buffer size allocation at the receiver are provided. The designed algorithm is compared against the established MPTCP CC algorithms: LIA, OLIA, BaLIA, and wVegas. The tests, conducted in open networks using real devices and equipment, show that the on-off controller reduces protocol delay, jitter, and Head-of-Line blocking, which is crucial for ensuring high-quality streaming in mobile networks.**

***Index Terms*—MPTCP, congestion control, streaming applications, tactile Internet.**

I. Introduction

THE RAPID expansion of Internet Protocol (IP)-based systems across a spectrum of applications underscores the need for network solutions that are both flexible and scalable. Despite their pervasive use, IP networks encounter dynamic resource allocation challenges that compromise Quality of Service (QoS), particularly affecting streaming services that demand a consistent, high-quality data transfer rate [1], [2]. By engaging multiple network paths, Multipath TCP (MPTCP) has been recognized as a promising solution to improve transmission reliability and efficiency. The core of MPTCP's effectiveness lies in its congestion control (CC) algorithms, which are crucial for managing data flow across multiple paths. However, the core MPTCP CC algorithms, namely LIA, OLIA, BaLIA, and wVegas, do not address the particular requirements of streaming applications in terms of minimizing delay and jitter [3].

Recent research in the development of CC algorithms for MPTCP has led to the proposal of several strategies aimed at improving network throughput and user fairness while efficiently managing network resources. A CC algorithm proposed by Kou et al. relies on packet loss to optimize network throughput and fairness among users, presenting an approach to balance resource allocation [4]. Similarly, the D-LIA algorithm, developed by Lubna et al., dynamically adjusts the congestion window decrease factor based on packet loss intervals, targeting throughput enhancement and packet loss reduction [5]. Additionally, Mudassir and Baig introduced the Modified Fast-Vegas-LIA Hybrid Congestion Control Algorithm (MFVL HCCA) for MPTCP, which operates in both uncoupled and coupled congestion control modes to adapt to network conditions, resulting in significant improvements in packet loss reduction and average goodput [6]. These contributions illustrate ongoing efforts to refine MPTCP CC algorithms to meet the requirements of modern network applications.

Our team has also contributed to this field through various publications. For instance, we explored the performance of MPTCP in industrial Internet of Things (IoT) applications and proposed a green multipath TCP framework to enhance energy efficiency and network performance [7]. Additionally, we conducted an experimental assessment of MPTCP congestion control algorithms specifically for streaming services in open Internet environments, highlighting the performance differences among LIA, OLIA, BaLIA, and wVegas algorithms [8]. In another study, we investigated appropriate control strategies for multipath transmission in Industry 4.0 applications, emphasizing the need for adaptive congestion control mechanisms [9].

This study introduces an On-Off CC MPTCP algorithm, designed to enhance streaming performance by adequately responding to the dynamic nature of network conditions. Unlike conventional approaches, the design of the On-Off algorithm, by adopting the analytical framework of dynamic systems theory, allows for a formal analysis of its properties. Following a rigorous mathematical argument, the proposed algorithm is demonstrated to be robust to variable networking conditions manifesting themselves in delay fluctuations. The receiver's data queue is shown to be finite with a precisely estimated upper limit that constitutes the required buffer capacity. Moreover, conditions for maintaining the preestablished transfer rate, and thus video quality, are formulated and proved.

The evaluation of the On-Off algorithm in a public network environment underscores its capacity to optimize congestion management, thereby directly addressing the limitations of existing MPTCP CC algorithms in maintaining high-quality

This work has been performed in the framework of a project "Robust control solutions for multi-channel networked flows" no. 2021/41/B/ST7/00108 financed by the National Science Centre, Poland

streaming services under fluctuating network conditions. Traditional MPTCP CC mechanisms, while proficient in handling multiple paths, were not specifically designed for scenarios where rapid adjustments to bandwidth fluctuations are essential to preserve streaming quality. This inadequacy manifests as increased latency and jitter, which are detrimental to real-time applications. The On-Off algorithm introduces a more dynamic and responsive approach to congestion control by design, effectively minimizing these adverse effects and enhancing the landscape of MPTCP CC solutions. This advancement promises improvements in the stability and reliability of streaming services, marking an enhancement over predecessors such as LIA, OLIA, BaLIA, and wVegas, which do not cater specifically to the high-demand scenarios of modern streaming applications [10], [11].

II. Related Work

To place this work in context, we review recent contributions in the area of MPTCP congestion control algorithms and highlight previous works by our team.

A. Multipath TCP Congestion Control Algorithms

Recent studies have made significant advancements in MPTCP congestion control. For instance, Kou et al. proposed a congestion control algorithm based on packet loss, which optimizes network throughput and fairness among users by balancing resource allocation [4]. Similarly, the D-LIA algorithm by Lubna et al. dynamically adjusts the congestion window decrease factor based on packet loss intervals, enhancing throughput and reducing packet loss [5]. Mudassir and Baig's MFVL HCCA algorithm adapts to network conditions by operating in both uncoupled and coupled modes, resulting in improved packet loss reduction and average goodput [6]. Additional research has explored other novel approaches such as a BBR-based congestion control scheme aimed at improving throughput in heterogeneous wireless networks [12], and a delay-based congestion control algorithm designed to optimize transmission performance by minimizing delay differences between paths [13]. The advanced MPTCP (AMP) protocol adjusts congestion detection based on subflow count, enhancing latency for small flows and throughput for large flows [14]. Another notable approach is the Shared Bottleneck-based Congestion Control scheme (SB-CC) utilizing ECN to detect shared bottlenecks among subflows, improving network performance [15].

B. Our Contributions

Our team has also contributed to this field through various publications. For instance, we explored the performance of MPTCP in industrial Internet of Things (IoT) applications and proposed a green multipath TCP framework to enhance energy efficiency and network performance [7]. Additionally, we conducted an experimental assessment of MPTCP congestion control algorithms specifically for streaming services in open Internet environments, highlighting the performance

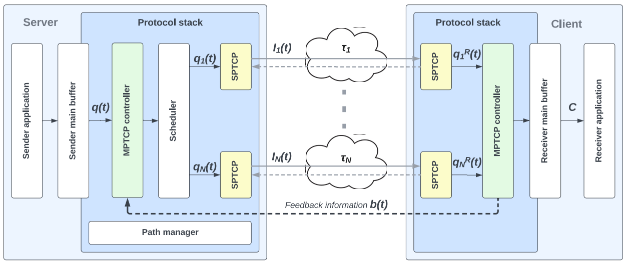


Fig. 1: Client-server interaction in an MPTCP framework.

differences among LIA, OLIA, BaLIA, and wVegas algorithms [8]. In another study, we investigated appropriate control strategies for multipath transmission in Industry 4.0 applications, emphasizing the need for adaptive congestion control mechanisms [9]. These works underscore our ongoing efforts to refine MPTCP congestion control to meet the evolving requirements of modern network applications.

Furthermore, the proposed On-Off CC MPTCP algorithm builds on our previous work by integrating dynamic systems theory to offer a more robust solution for streaming applications. This approach ensures better adaptability to fluctuating network conditions, thereby enhancing the quality and reliability of streaming services.

III. System Model

A. System Variables

Consider the scenario depicted in Fig. 1, where a user needs uninterrupted access to a live video broadcast at the negotiated rate C while on the move. To enhance reliability and counteract variable network conditions, the system employs independent connections. This multi-connection configuration is enabled by MPTCP, a protocol that enhances Internet connectivity by allowing the simultaneous use of multiple links.

In this model, the intervals $t = 0, 1, 2, \dots$ represent discrete time points for video transmission. The function $s_i(t)$ denotes the number of segments transmitted through subpath i at time t. These segments experience varying delays $\tau_i(t)$ as they traverse the network. The number of successfully delivered segments is expressed as

$$s_i^R(t) = s_i(t - \tau_i(t)), \tag{1}$$

and the queue length in the receiving buffer evolves according to:

$$b(t+1) = b(t) + \sum_{i=1}^{N} s_i(t - \tau_i(t)) - b_0. \tag{2}$$

Assuming zero initial conditions, i.e., an empty receiver buffer and no in-flight data at $t = 0$, the accumulation of segments in the receiver buffer for any $t > 0$ evolves as:

$$b(t) = \sum_{i=1}^{N} \sum_{k=0}^{t-1} s_i(k - \tau_i(k)) - Ct. \tag{3}$$

B. Uncertainty Model

The delay $\tau_i(t)$ varies randomly within the range:

$$(1-\delta_i)T_i \leq \tau_i(t) \leq (1+\delta_i)T_i, \tag{4}$$

where T_i represents the nominal delay on subpath i, and $\delta_i \in [0,1)$ indicates the uncertainty tolerance. The delay variability reflects inherent network fluctuations, such as traffic load or routing changes.

To quantify the impact of delay variations on data delivery, we introduce a function $\eta(t)$:

$$\eta(t) = \eta_+(t) - \eta_-(t), \tag{5}$$

where $\eta_+(t)$ accounts for segments that arrive sooner than expected, and $\eta_-(t)$ for those delayed beyond the expected timeframe. These components are calculated as:

$$\eta_+(t) = \sum_{i=1}^{N} \sum_{j\in(0,\delta T_i):\tau_i(t+j)\leq T_i-j} s_i(t-T_i+j) \tag{6}$$

representing the accumulation of segments that outpaced the anticipated delay, while

$$\eta_-(t) = \sum_{i=1}^{N} \sum_{j\in(0,\delta T_i):\tau_i(t-j)>T_i+j} s_i(t-T_i+j) \tag{7}$$

captures those delayed beyond the expected timeframe.

Thus, the receiver queue length evolution is given by

$$b(t) = \sum_{i=1}^{N}\sum_{k=0}^{t-1} s_i(k-T_i) + \eta(t) - Ct. \tag{8}$$

Considering the maximum amount of data that can be injected into path i in a time unit S_i, the magnitude of $\eta(t)$ does not exceed a maximum value, H, estimated as:

$$H = \sum_{i=1}^{N} \delta_i T_i S_i. \tag{9}$$

C. Congestion Control Algorithm

After evaluating the network's available resources and in-flight data, the sender prepares a set of S data segments for the transmission across N subpaths. The paths are allocated on a dynamic basis according to:

$$S_i(t) = \lambda_i(t)S, \tag{10}$$

where $S_i(t)$ is the amount of data forwarded onto path i, and $\lambda_i(t) \in [0,1]$, reflects the allocation strategy. A complete allocation is assumed, i.e.,

$$\sum_{i=1}^{N} \lambda_i(t) = 1. \tag{11}$$

The default MPTCP scheduler allocates the bandwidth evenly across all the subpaths, i.e., $\lambda_i = \frac{1}{N}$.

In the proposed On-Off algorithm, the decision to transmit S data segments at time t is guided by the following criteria:

$$s(t) = \begin{cases} S, & \text{if } \mathrm{B}_{ref} - b(t) - I(t) > 0 \\ 0, & \text{if } \mathrm{B}_{ref} - b(t) - I(t) \leq 0 \end{cases}, \tag{12}$$

where, B_{ref} denotes a queue reference level, indicating a desirable buffer fill level, and $I(t)$ represents the amount of in-flight data, i.e., the data that have been sent but not yet acknowledged. $I(t)$ can be calculated as:

$$\begin{aligned} I(t) &= \sum_{k=0}^{t-1} s(k) - \sum_{i=1}^{N}\sum_{k=0}^{t-1} s_i^R(k) \\ &= \sum_{k=0}^{t-1} s(k) - \sum_{i=1}^{N}\sum_{k=0}^{t-1} s_i(k-\tau_i(k)), \end{aligned} \tag{13}$$

This formula encapsulates the aggregate of all dispatched segments that are pending acknowledgment, providing a real-time snapshot of in-flight data.

IV. On-Off Algorithm Formal Analysis

The properties of the proposed algorithm are articulated through two theorems, each rigorously proven. The first theorem demonstrates that, despite the randomness of delays, the data accumulation in the receiver's buffer is bounded by a specific limit. This limit sets the required buffer capacity to effectively manage incoming data and prevent drops. The second theorem outlines how to set the reference queue length to prevent data starvation and sustain the desired streaming rate C.

For the proofs, we introduce the concept of the channel uncertainty function, which measures the actual volume of in-flight data compared to the anticipated amount:

$$\begin{aligned} I(t) &= \sum_{k=0}^{t-1} s(k) - \sum_{i=1}^{N}\sum_{k=0}^{t-1} s_i(k-T_i) - \eta(t) \\ &= \sum_{k=0}^{t-1} s(k) - \sum_{i=1}^{N}\sum_{k=0}^{t-T_i-1} s_i(k) - \eta(t). \end{aligned} \tag{14}$$

The uncertainty function calculates the volume of in-flight data by considering the total segments sent, subtracting those successfully delivered (accounting for network delays and variability), and adjusting for early or late arrivals via the $\eta(t)$ term.

Theorem 1. *Applying Algorithm (12) to system (2) results in a finite queue length at the receiver:*

$$\underset{t\geq 0}{\forall}\, b(t) \leq B, \tag{15}$$

with the upper bound

$$B = B_{ref} + S + H. \tag{16}$$

Proof. Initially, the receiver buffer is empty, and remains so for any $t \leq t_{\min} = \min\{(1-\delta_i)T_i\}$. For any time $p > t_{min}$, two cases can be distinguished:

Case 1*:* When $B_{ref} > s(p) + I(p)$, following from equation (14) we have:

$$\begin{aligned} b(p) &< B_{ref} - \sum_{k=0}^{p-1} s(k) + \sum_{i=1}^{N} \sum_{k=0}^{p-T_i-1} s_i(k) + \eta(p) \\ &= B_{ref} - \sum_{i=1}^{N} \sum_{k=p-T_i}^{p-1} s_i(k) + \eta(p) \leq B_{ref} + \eta(p). \end{aligned} \tag{17}$$

Given that $\eta(t)$ does not exceed its maximum expected value H for any $t \geq 0$, it follows that $b(p) < B_{ref} + H$, establishing the upper bound in this case.

Case 2*:* In the case where $B_{ref} \leq s(p) + I(p)$, we identify the most recent interval p_1 before p where I was less than $B_{ref} - b(p_1)$. Given the initial conditions, such an interval p_1 exists. The buffer level $b(p)$ during p_1 adheres to an inequality similar to equation (17), i.e.,

$$b(p) < B_{ref} - \sum_{k=0}^{p_1-1} s(k) + \sum_{i=1}^{N} \sum_{k=0}^{p_1-1} s_i(k - \tau_i(k)). \tag{18}$$

The data volume $b(p)$ can be expressed in terms of $b(p_1)$ as:

$$b(p) = b(p_1) + \sum_{i=1}^{N} \sum_{k=p_1}^{p-1} s_i(k - \tau_i(k)) - C(p - 1 - p_1). \tag{19}$$

Considering that C is nonnegative, by applying (18) to (19), we deduce:

$$\begin{aligned} b(p) &< B_{ref} - \sum_{k=0}^{p_1-1} s(k) + \sum_{i=1}^{N} \sum_{k=0}^{p_1-1} s_i(k - \tau_i(k)) \\ &+ \sum_{i=1}^{N} \sum_{k=p_1}^{p-1} s_i(k - \tau_i(k)) \\ &= B_{ref} + \sum_{i=1}^{N} \sum_{k=p_1}^{p-T_1-1} s_i(k) + \eta(p) \end{aligned} . \tag{20}$$

Given the data was last sent before p at p_1, and taking into account the uncertainty bound H,

$$b(p) \leq B_{ref} + S + H. \tag{21}$$

□

Theorem 2. *When algorithm (12) is applied to system (2), and S exceeds C, then the reference queue length:*

$$B_{ref} > S\Lambda + S + H, \tag{22}$$

where

$$\Lambda = \max_t \sum_{i=1}^{N} \sum_{k=t-L_i}^{t-1} \lambda_i(k), \tag{23}$$

ensures that for any $t \geq t_{\max} = \max_i \{(1+\delta_i)T_i\} + B/(S-C)$, the receiver queue length is strictly positive.

Proof. Consider a period $p \geq t_{max}$. Two scenarios are distinguished: when $I(p) < B_{ref} - b(p)$, and when $I(p) \geq B_{ref} - b(p)$.

Case 1*:* In the scenario where $I(p) \geq B_{ref} - b(p)$, following from (14), it is evident that:

$$\begin{aligned} b(p) &\geq B_{ref} - \sum_{k=0}^{p-1} s(k) + \sum_{i=1}^{N} \sum_{k=0}^{p-T_i-1} s_i(k) + \eta(p) \\ &= B_{ref} - \sum_{i=1}^{N} \sum_{k=p-T_i}^{p-1} s_i(k) + \eta(p) \\ &= B_{ref} - \sum_{i=1}^{N} \sum_{k=p-T_i}^{p-1} \lambda_i(k) s(k) + \eta(p). \end{aligned} \tag{24}$$

The volume of data that can be dispatched at any time within the interval $[p - T_i, p - 1]$ equals S, yet,

$$\sum_{i=1}^{N} \sum_{k=p-T_i}^{p-1} \lambda_i(k) s(k) \leq S \sum_{i=1}^{N} \sum_{k=p-T_i}^{p-1} \lambda_i(k) \leq S\Lambda. \tag{25}$$

Given that $\eta \geq -H$, it follows from equation (24) that:

$$b(p) \geq B_{ref} - S\Lambda - H. \tag{26}$$

Using assumption (25), we deduce that $b(p) > 0$, thus concluding the first part of the proof.

Case 2*:* Consider the scenario where the in-flight data volume $I(p) < B_{ref} - b(p)$. Identify the most recent time $p_1 < p$ when I was greater than or equal to $B_{ref} - b(p)$. According to Theorem 1, the data volume at the receiver does not exceed B. Given the buffer depletion rate C, the maximum duration for continuous data reception is $B/(S - C)$, confirming the existence of period p_1. It is inferred from the theorem's conditions that by period p_1, the initial data packets from all channels have been received, despite any delay variations.

The volume of in-flight data $I(p_1) \geq B_{ref} - b(p_1)$, and by employing reasoning similar to equations (24) we conclude:

$$b(p) \geq B_{ref} - \sum_{k=0}^{p_1-1} s(k) + \sum_{i=1}^{N} \sum_{k=0}^{p_1-1} s_i(k - \tau_i(k)) > 0. \tag{27}$$

Thus,

$$\begin{aligned} b(p) &= b(p_1) + \sum_{i=1}^{N} \sum_{k=p_1}^{p-1} s_i(k - \tau_i(k)) - C(p - 1 - p_1) \\ &\geq B_{ref} - \sum_{k=0}^{p_1-1} s(k) + \sum_{i=1}^{N} \sum_{k=0}^{p_1-1} s_i(k - \tau_i(k)) \\ &+ \sum_{i=1}^{N} \sum_{k=p_1}^{p-1} s_i(k - \tau_i(k)) - C(p - 1 - p_1) \\ &= B_{ref} + \sum_{i=1}^{N} \sum_{k=p_1}^{p-1} s_i(k) - \sum_{i=1}^{N} \sum_{k=p-L_1}^{p-1} s_i(k) + \eta(p) \\ &- C(p - 1 - p_1). \end{aligned} \tag{28}$$

TABLE I: Performance metrics of MPTCP CC algorithms

		LIA	OLIA	BALIA	wVegas	On-Off
Protocol jitter [ms]	θ_{av}	38.45	38.53	41.11	28.13	15.45
	θ_{max}	329.32	256.80	241.83	227.68	123.87
Protocol delay [ms]	υ_{av}	324.82	266.53	277.23	151.80	106.26
	υ_{max}	588.04	480.88	560.29	418.02	225.54
HoL Degree [ms]	ζ_{av}	251.89	195.19	205.48	83.86	36.66
	ζ_{max}	518.86	410.46	391.52	350.40	157.20
SRTT path 1 [ms]	τ^1_{av}	2.17	2.14	1.85	1.72	1.87
	τ^1_{max}	3.86	4.31	3.39	2.30	4.07
SRTT path 2 [ms]	τ^2_{av}	72.93	71.33	71.75	67.94	69.60
	τ^2_{max}	111.98	136.83	120.29	100.49	111.51
Mean drop rate [seg/s]	δ^1	2.90	2.90	2.90	2.90	2.90
	δ^2	1.53	1.90	0.37	0.50	0.57
Throughput [Mbps]	φ_{av}	5.07	5.07	5.12	2.57	4.90
	φ_{max}	8.48	8.09	7.85	5.49	7.16

At time p_1, $I(p_1)+b(p_1) \geq B_{ref}$, marking the last instance when no new data request is made. Subsequently, a volume S of data is dispatched to the receiver. Given the complete allocation (11), the first sum in equation (28) equates to $S(p-1-p_1)$. Following a logic similar to (25), the second sum is approximated as:

$$\sum_{i=1}^{N}\sum_{k=p-T_i}^{p-1} s_i(k) = \sum_{i=1}^{N}\sum_{k=p-T_i}^{p-1} \lambda_i(k)s(k) \leq S\Lambda. \quad (29)$$

The buffer occupancy level described in equation (28) fulfills the inequality:

$$b(p) > B_{ref} + S(p-1-p_1) - S\Lambda + \eta(p) - C(p-p_1). \quad (30)$$

This denotes the maximum data shortfall due to overly delayed data packets. Therefore, using the theorem assumption (22), it can be stated that

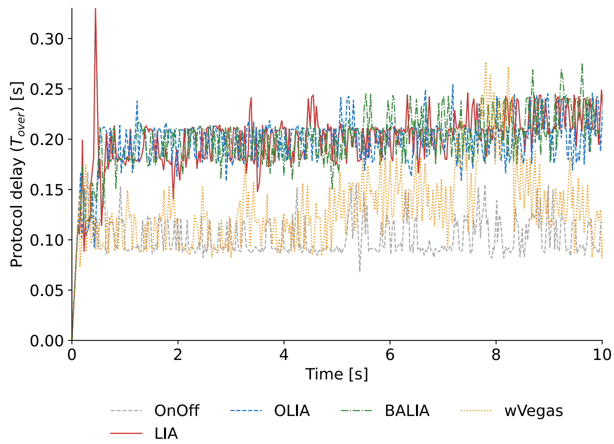
$$b(p) > S + S(p-1-p_1) - C(p-p_1) > 0. \quad (31)$$

Under the stipulated conditions, the buffer at the receiver maintains a strictly positive level, ensuring continuous data flow.

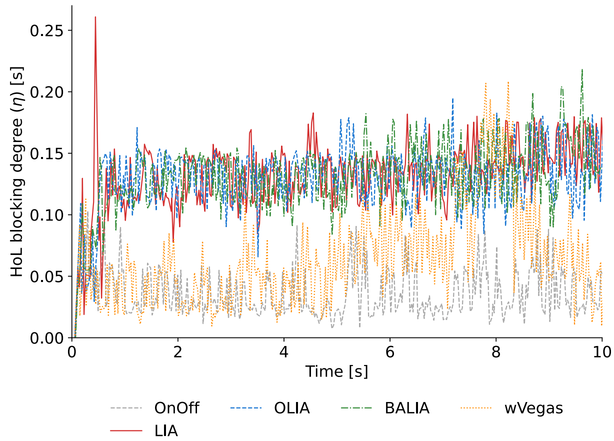
□

V. Experimental Setup

The experimental setup aimed to assess the performance of various congestion control (CC) algorithms in delivering streaming content over Multipath TCP (MPTCP). This setup replicates a typical data transmission scenario where a client fetches content from a server situated in a remote data center accessible via a public IP address. Both the client and server systems operated on Linux OS version 4.19, customized to support MPTCP version 0.95. For streaming content generation, we employed VLC media player, an open-source multimedia framework, player, and server. VLC was selected due to its reliability and widespread use in streaming "Big Buck Bunny," a popular open-source animated film, ensuring our testing environment accurately reflects real-world streaming conditions. The client setup included two communication interfaces: one connected to an LTE router through Ethernet and the other connected to the same router via Wi-Fi 802.11bgn, utilizing two distinct LTE networks. Each test session lasted for 10 seconds and was repeated 30 times to ensure the results were statistically significant.



(a) Protocol delay



(b) HoL blocking degree

Fig. 2: Measured transmission properties.

VI. Tests and Results

In our analysis, the On-Off algorithm was compared with several well-known CC algorithms: LIA, OLIA, BaLIA, and wVegas. These algorithms were evaluated based on metrics important for streaming applications, including protocol delay, jitter, and Head-of-Line (HoL) blocking degree. The results are summarized in Table I, and their graphical representation is shown in Fig. 2.

The On-Off algorithm demonstrated an improvement in average protocol delay, recording 106.26 ms, which is lower than the delays observed with the other algorithms. This lower delay suggests that the On-Off algorithm is more efficient in managing the transmission of data packets, thereby reducing latency. In terms of protocol jitter, the On-Off algorithm exhibited an average jitter of 15.45 ms with a variance of 0.30, indicating less fluctuation in transmission times. This stability is important for streaming applications, as it ensures a more consistent delivery of data packets, minimizing interruptions and buffering events. Regarding HoL blocking, the On-Off algorithm achieved an average of 36.66 ms. HoL blocking can lead to delays in packet delivery, and the lower HoL blocking value for the On-Off algorithm indicates its effectiveness in maintaining a steady flow of data, which is essential for high-quality streaming.

These metrics underscore the On-Off algorithm's potential to enhance the user experience by ensuring smoother and more consistent streaming in variable network conditions. Additionally, the comparison of Smoothed Round-Trip Time (SRTT) values and drop rates further validates the algorithm's efficiency in handling multi-path data flow for seamless streaming.

VII. Conclusion

In this study, we evaluated the On-Off algorithm for Multipath TCP (MPTCP) congestion control (CC), aimed at enhancing streaming services via multiple network paths. Performance comparisons were made with established protocols—LIA, OLIA, BaLIA, and wVegas—highlighting key metrics such as protocol delay, jitter, and Head-of-Line (HoL) blocking. The research involved the development and analysis of a mathematical model based on dynamical systems theory, designed to predict the On-Off algorithm's performance across various network conditions. This model outlines an approach to MPTCP CC by focusing on reducing protocol delay and jitter, which are vital for delivering reliable streaming content.

Empirical tests, conducted in an Open Internet environment using real devices and equipment, validated the On-Off algorithm's effectiveness in lowering protocol delay and jitter, as well as reducing HoL blocking. These results, which align with the mathematical model's predictions, demonstrate the algorithm's capacity to enhance the streaming experience by ensuring continuous content delivery. The findings affirm the On-Off algorithm's potential to not only improve streaming quality but also to decrease receiver buffer size requirements and bolster connection stability. This positions the On-Off algorithm as an effective MPTCP congestion control strategy, particularly suitable for mobile networks.

Through both analytical and empirical evaluations, this study confirms the On-Off algorithm's utility, laying the groundwork for further research and its prospective implementation in public network applications. The On-Off algorithm's robust performance in varying network conditions highlights its potential for improving user experiences in streaming services, ensuring smoother and more reliable content delivery.

Acknowledgment

This work has been completed while the first author was the Doctoral Candidate in the Interdisciplinary Doctoral School at the Lodz University of Technology, Poland

References

[1] "Cisco annual internet report (2018-2023) white paper," 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] K. Hatakeyama, Y. Osana, M. Tanabe, and S.-i. Kuribayashi, "Proposed congestion control method reducing the size of required resource for all-ip networks," in *2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. Victoria, BC, Canada: IEEE, 8 2009. doi: 10.1109/PACRIM.2009.5291413. ISBN 978-1-4244-4560-8 pp. 1–4, [Online; accessed 2024-04-03]. [Online]. Available: http://ieeexplore.ieee.org/document/5291413/

[3] C. Raiciu, M. Handley, and D. Wischik, "Coupled congestion control for multipath transport protocols," RFC Editor, Tech. Rep., 10 2011, dOI: 10.17487/rfc6356. [Online]. Available: https://www.rfc-editor.org/info/rfc6356

[4] L. Kou, R. Wang, and S. R. Chen, "A mptcp congestion control algorithm based on packet loss," in *Frontiers of Manufacturing Science and Measuring Technology III*, ser. Applied Mechanics and Materials, vol. 401. Trans Tech Publications Ltd, 12 2013. doi: 10.4028/www.scientific.net/AMM.401-403.1760 pp. 1760–1765.

[5] T. Lubna, I. Mahmud, and Y.-Z. Cho, "D-LIA: Dynamic congestion control algorithm for MPTCP," *ICT Express*, vol. 6, no. 4, pp. 263–268, 2020. doi: https://doi.org/10.1016/j.icte.2020.03.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959519304229

[6] M. U. Mudassir and M. I. Baig, "Mfvl hcca: A modified fast-vegas-lia hybrid congestion control algorithm for mptcp traffic flows in multihomed smart gas iot networks," *Electronics*, vol. 10, no. 6, 2021. doi: 10.3390/electronics10060711. [Online]. Available: https://www.mdpi.com/2079-9292/10/6/711

[7] M. Morawski and P. Ignaciuk, "A green multipath tcp framework for industrial internet of things applications," *Computer Networks*, vol. 187, p. 107831, 3 2021. doi: 10.1016/j.comnet.2021.107831

[8] Ł. Łuczak, P. Ignaciuk, and M. Morawski, "Experimental assessment of mptcp congestion control algorithms for streaming services in open internet," in *Annals of Computer Science and Information Systems*, 10 2023. doi: 10.15439/2023F9991 pp. 359–364, [Online; accessed 2024-04-03]. [Online]. Available: https://annals-csis.org/proceedings/2023/drp/9991.html

[9] M. Morawski and P. Ignaciuk, "Choosing a proper control strategy for multipath transmission in industry 4.0 applications," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3609–3619, 6 2022. doi: 10.1109/TII.2021.3105499

[10] M. O. Farooq, T. Kunz, and M. St-Hilaire, "Differentiated services based congestion control algorithm for wireless multimedia sensor networks," in *2011 IFIP Wireless Days (WD)*. Niagara Falls, ON, Canada: IEEE, 10 2011. doi: 10.1109/WD.2011.6098182. ISBN 978-1-4577-2028-4 pp. 1–6, [Online; accessed 2024-04-03]. [Online]. Available: http://ieeexplore.ieee.org/document/6098182/

[11] C. Lee, "Hcca for wireless mobile coverage networks," *Indian Journal of Science and Technology*, vol. 9, no. 1, pp. 1–8, 1 2016. doi: 10.17485/ijst/2016/v9i46/107198

[12] W. Wei, K. Xue, J. Han, Y. Xing, D. S. L. Wei, and P. Hong, "Bbr-based congestion control and packet scheduling for bottleneck fairness considered multipath tcp in heterogeneous wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 330–345, 1 2021. doi: 10.1109/TVT.2020.3047877

[13] H. Li, Y. Wang, R. Sun, S. Guo, and H. Wang, "Delay-based congestion control for multipath tcp in heterogeneous wireless networks," *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1–6, 4 2019. doi: 10.1109/WCNCW.2019.8902835

[14] J. Ye, L. Feng, Z. Xie, J. Huang, and X. Li, "Fine-grained congestion control for multipath tcp in data center networks," *IEEE Access*, vol. 7, pp. 30 495–30 504, 3 2019. doi: 10.1109/ACCESS.2019.2902860

[15] W. Wei, K. Xue, J. Han, D. S. L. Wei, and P. Hong, "Shared bottleneck-based congestion control and packet scheduling for multipath tcp," *IEEE/ACM Transactions on Networking*, vol. 28, pp. 1380–1395, 2 2020. doi: 10.1109/TNET.2020.2970032