MULTIPATH TCP VARIANCE IN 5G D2D NETWORK : COMPARISION BETWEEN MPTCP AND RFC 9438

Vinitha R , Varshini V G , Surya A
Department of Electronics and Communication Engineering
Kongu Engineering College
Erode -638060, India
Email:{vinitha0517@gmail.com, vgvarshini16@gmail.com, suryaa@gmail.com}

***Abstract*— Multipath TCP (MPTCP) is an extension of TCP that enhances data transmission by using multiple paths simultaneously, increasing throughput, redundancy, and efficiency in 5G Device-to-Device (D2D) communication. While MPTCP is well-suited for the high-speed, low-latency requirements of 5G networks, it faces significant challenges. Dynamic route changes, often caused by packet drops or external factors, are frequently misinterpreted as congestion, leading to unnecessary rate reductions and performance degradation. This paper evaluates the performance of MPTCP and RFC 9438 in handling such challenges in dynamic 5G D2D communication environments. RFC 9438 introduces enhanced mechanisms for path management and congestion control, aiming to address the limitations of MPTCP. A comparative analysis is conducted to assess their effectiveness in mitigating issues like signal fading and network instability. The findings provide insights into the strengths and limitations of each protocol and propose strategies for improving the efficiency and reliability of multipath communication in 5G networks.**

Keywords—MPTCP, RFC 9438, Metrices

I. INTRODUCTION

In recent years, there has been significant increase in demand for wireless network like in mobile network where 5G network communication which has features like high speed, low latency, etc. Device-to-Device (D2D) communication in 5G is a technology that allows direct communication between two mobile devices without any base station interference. The frequency range of 5G network is classified into three categories namely, Frequency range 1 (FR1) is from 450 MHz to 6 GHz [1], Frequency range 2 (FR2) is from 24.25 GHz to 52.6 GHz [2], Frequency range 3 (FR3) is from 7.125 GHz to 24.25 GHz [3]. These frequency ranges allow 5G networks to support various applications, including mobile broadband network, low-latency communication, and large company communications. However, the efficiency of data transmission depends upon the transport protocols, with TCP (Transmission Control Protocol) and MPTCP (Multipath TCP). TCP is responsible for ensuring reliable data transmission between devices. It uses a single path for communication, relying on mechanisms like congestion control, error detection, and retransmission to maintain end-to-end data transmission. While TCP has been highly effective and reliable, we need multiple path communication in 5G D2D communication to ensure high speed, congestion less path to deliver the data between two devices efficiently. That is why we are going for MPTCP. MPTCP is an extension of TCP but has multiple paths for data transmission simultaneously. By utilizing multiple paths, MPTCP can increase throughput, improve redundancy, and reduce congestion. This makes MPTCP particularly well-suited for 5G networks, where devices can connect through various frequency ranges and paths. In D2D communication, MPTCP's ability to manage multiple paths allows it to better handle the dynamic nature of wireless networks and offers much better performance than TCP. However, despite its advantages, MPTCP faces challenges in D2D communication. MPTCP uses dynamic route change if the data transmitted path (subflow) faces any issues like packet drop due to some external issues. So, this dynamic route changes in MPTCP can be misinterpreted as congestion, leading to unnecessary rate reductions and reduced throughput. This summarizes the need for new approach to congestion control in MPTCP, one that can distinguish between actual congestion and other factors affecting network performance, ensuring optimal data transmission despite varying network condition.

Fig.1 depicts Device-to-Device (D2D) communication, where devices exchange data directly with each other, enhancing speed and reducing latency in 5G networks. Fig.2 demonstrates MPTCP communication, which uses multiple network paths simultaneously to optimize bandwidth, fault tolerance, and overall network stability. Together, these figures highlight how D2D enables direct communication between devices, while MPTCP improves data transfer efficiency through multiple subflows.

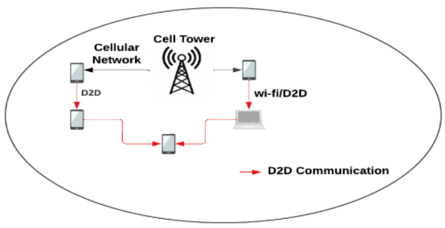

Fig. 1. Device-to-Device Communication

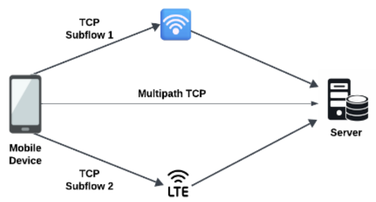

Fig. 2. Multipath TCP Communication

II. Background and Related Works

A. Multipath TCP for D2D Communication

5G Device-to-Device (D2D) communication using Multipath TCP (MPTCP) is a technology that enables direct communication between two mobile devices without the need for a base station. Multipath TCP (MPTCP) is a extension of the traditional TCP protocol designed to enhance performance, reliability in dynamic network environments by allowing a single connection to utilize multiple network paths simultaneously which can improve the reliability and throughput of D2D communication. In traditional TCP, data is transmitted through a single path between two client and server. MPTCP, however, establishes multiple subflows over different network interfaces, such as Wi-Fi, cellular data, or Ethernet, within a single session. These subflows achieves better performance and fault tolerance. This makes MPTCP ideal for 5G D2D communication, ensuring better connectivity and performance in dynamic and fading wireless environments.

B. Connection Establishment

During connection establishment phase, MPTCP extends the traditional TCP three-way handshake: SYN, SYN-ACK and ACK but it is added with additional options that support multipaths.If both ends supports MPTCP ,additional subflow connections can be established later over other interfaces or network paths, such as Wi-Fi, LTE, or Ethernet. Each new subflow also uses the three-way handshake to join the existing MPTCP session, allowing multiple paths to be connected under the same session ID. This enables the device to distribute traffic across different network links while appearing as a single connection to higher-layer protocols. New paths can be dynamically added to the MPTCP path which allows for fine-tuning of data packet transmission.

Once the connection is established, MPTCP distributes data across multiple subflows to optimize bandwidth and improve fault tolerance. Data transmission in MPTCP is managed by segmenting data packets into smaller packets, which are then distributed across the available subflows. Each segment is assigned a unique sequence number, allowing MPTCP to reassemble the data correctly at the receiver. MPTCP continuously monitors the performance of each path, including factors such as bandwidth, latency, and congestion, and dynamically adjusts the traffic distribution across subflows to optimize throughput and minimize delays. Fast, low-latency paths are prioritized for time-sensitive data, while larger data volumes are sent over higher-capacity paths. At the receiving end, Data Sequence Mapping ensures that the segments which are sent across different paths arrive in the correct order at the receiving end based on the global sequence number, ensuring uninterrupted data transfer

The termination of an MPTCP connection involves closing all active subflows. Each subflow performs a four-way handshake (FIN, ACK) to confirm that no more data will be sent. However, the MPTCP session as a whole is only considered terminated when all subflows are closed. If a subflow terminates unexpectedly due to network failure, MPTCP ensures the remaining subflows continue handling the data transfer, maintaining the connection’s integrity. Once the endpoints agree to close the session, the MPTCP session ID is released, and no further subflows can be added. MPTCP’s termination mechanism ensures that even in cases of partial network failure, the protocol closes gracefully without data loss or disruption. This robust termination process enhances MPTCP's reliability, ensuring seamless data transfer and minimizing the impact of network disruptions throughout the session lifecycle.

C. Related Works

MultiPath TCP (MPTCP) extends traditional TCP by allowing simultaneous data transmission over multiple paths, significantly improving bandwidth utilization and fault tolerance. This literature review and variants of MPTCP explores the development and applications of MPTCP, highlighting its advantages for mobile connectivity, data centers, and real-time services. Additionally, we examine the challenges it presents, including complexity and security concerns, while identifying key areas for future research and optimization

Reference [1] proposes a method to improve MPTCP efficiency in wireless networks by integrating a practical and robust scheduler. Reference [2] discusses TCP extensions for multipath operation, emphasizing the benefits of multiple addresses for optimizing network performance. Reference [3] explores CUBIC as an alternative for fast, long-distance connections, illustrating its advantages over traditional TCP implementations. Reference [4] focuses on enhancing MPTCP's performance in lossy wireless networks, highlighting its resilience to packet loss. Reference [5] presents an experimental study demonstrating MPTCP's effectiveness over heterogeneous wireless networks, confirming its capability for improved throughput. Reference [6] introduces a proactive scheduling approach for seamless handoff in diverse access networks, addressing the challenges of mobility. Reference [7] describes a software-defined MPTCP solution tailored for mobile tactical networks, showcasing its flexibility in dynamic environments. Reference [8] investigates the streaming of high-quality mobile video using MPTCP in heterogeneous networks, illustrating its role in improving user experience. Reference [9] examines constraint-based proactive scheduling to optimize MPTCP in wireless settings, focusing on resource allocation strategies. Finally, reference [10] proposes a stochastic optimal scheduler for MPTCP within software-defined wireless networks, emphasizing its potential for adaptive resource management. Collectively, these studies underscore the growing importance of MPTCP in enhancing network performance across various applications and environments.

In summary, recent studies highlight significant improvements in Multipath TCP (MPTCP) performance across various network environments. Proposed enhancements include robust scheduling methods to optimize efficiency in wireless networks and CUBIC's advantages for long-distance connections. Research confirms MPTCP's effectiveness in lossy and heterogeneous networks, enhancing throughput and user experience, particularly for mobile video streaming. Proactive scheduling techniques facilitate seamless handoffs in diverse access networks, while software-defined solutions improve flexibility. Collectively, these advancements underscore MPTCP's growing importance in optimizing network performance for diverse applications.

III. Comparision of MPTCP and RFC 9438

Multipath Transmission Control Protocol (MPTCP) (RFC 8684) and RFC 9438 both were extensions of Transmission Control Protocol (TCP) that is designed to improve the data transmission but both of them has different aim and approach that will help improve the throughput. Fig.3 illustrates the comparison between MPTCP and RFC 9438.

A. Handling Multiple Paths

MPTCP is designed for multi-homed environments, where multiple network interfaces are used simultaneously. It creates multiple TCP subflows over these interfaces and adjusts the paths according to network conditions. This enables data transmission over various networks, such as Wi-Fi and cellular, which increases throughput and provides redundancy in case one path fails. The equation for MPTCP throughput aggregation is defined as:

$$T_{MPTCP} = \sum_{i=1}^{n} \frac{CWND_i}{RTT_i} \quad (1)$$

Where $CWND_i$ is the congestion window for subflow i, and RTT_i is its round-trip time. This equation shows how MPTCP aggregates throughput across multiple subflows based on their respective congestion windows and RTTs.

RFC 9438, on the other hand, operates on a single path. It focuses on improving throughput by using an aggressive growth model for the congestion window. This enables faster recovery from congestion in high-bandwidth and high-latency environments. The congestion window for RFC 9438 grows as per the following equation:

$$W(t+1) = W(t) + \frac{\alpha}{W(t)} + \beta . W(t)^2 \quad (2)$$

Where W(t) is the congestion window at time t, α is the additive increase factor, and β is the multiplicative decrease factor. This equation describes the non-linear growth of the congestion window, enabling faster recovery and efficient throughput on a single path.

B. Congestion Control Mechanism

Multipath TCP employs a congestion control mechanism where each subflow has its own congestion window, and traffic is coordinated across all paths. The general congestion control equation for MPTCP ensures fairness and optimal utilization of available bandwidth by considering the RTTs of all paths. This is represented by:

$$\Delta CWND_i = \frac{RTT_{min}}{RTT_i} \cdot \frac{\Delta}{\sum_{j=1}^{n} \frac{RTT_{min}}{RTT_j}} \quad (3)$$

Where Δ represents the total congestion window increase, and RTT_{min} is the minimum RTT across all paths. This ensures fairness across subflows while maximizing throughput.

RFC 9438 uses a different approach to congestion control. It employs a RFC 9438 function to let window growth be aggressive in utilizing and thus recovering more quickly from the congested network. The non-linear growth phase of RFC 9438 makes it highly suitable for long-distance, high-capacity links, thus providing improvement over TCP algorithms, such as Reno or NewReno. This is mathematically expressed as:

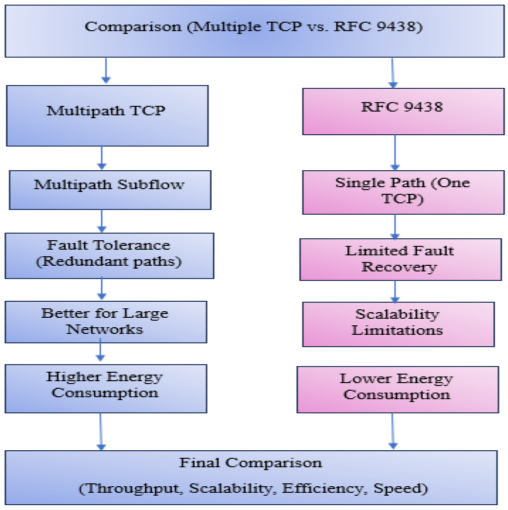

Fig.3. Comparision of MPTCP and RFC 9438

$$CWND_{new} = CWND_{old} + c \, . \sqrt{\frac{RTT_{min}}{RTT_{current}}} \quad (4)$$

Where c is a scaling factor that controls the aggressiveness of the window growth. This equation ensures that the window increases more quickly when the RTT is low, optimizing throughput under favorable conditions.

C. Throughput and Resource Utilization

MPTCP achieves higher overall throughput by balancing traffic across multiple subflows, thereby distributing data loads across multiple paths, this in turn lends to better resource usage especially where heterogeneous networks are in play with each path possibly varying in terms of characteristics like latency, bandwidth, etc. The throughput for each subflow is calculated as:

$$Rate_i = \frac{CWND_i}{RTT_i} \quad (5)$$

where $Rate_i$ is the rate for subflow i, $CWND_i$ is the congestion window for subflow i, and RTT_i is the round-trip time for that subflow. This equation ensures that each subflow transmits data proportionally to its congestion window and RTT, optimizing resource utilization across different network interfaces.

RFC 9438, however, focuses on optimizing throughput on a single path. The throughput on a single path is determined by the size of the congestion window and the RTT of that path, as shown in the equation:

$$Throughput = \frac{CWND}{RTT} \quad (6)$$

where CWND is the congestion window, and RTT is the round-trip time. This equation reflects how the throughput is directly influenced by the congestion window and RTT, optimizing performance on a single connection.

D. Scalability

MPTCP's scalability is a key advantage, as it can dynamically adjust by adding or removing paths based on network conditions. It is therefore ideal for applications in which failure-related considerations should weigh most: high data rates, fault tolerance, and load balancing. The total throughput for MPTCP across multiple paths is calculated by:

$$T_{total} = \sum_{i=1}^{n} \frac{CWND_i}{RTT_i} + \frac{CWND_{new}}{RTT_{new}} \quad (7)$$

where Ttotal is the total throughput, $CWND_i$ is the congestion window for each subflow i, and RTT_i is the round-trip time for that subflow. This equation illustrates how new paths contribute to overall throughput, ensuring scalability as new interfaces become available or existing ones fail.

RFC 9438, while efficient for single-path scalability in high-bandwidth, high-latency networks, does not inherently scale across multiple connections or interfaces. RFC 9438's reach for scalable functionality remains limited to single-path performance, offering reliable throughput for large-scale single-path applications but whose application within multipath frameworks-ready setups is ruled out. The scalability for RFC 9438 is represented as:

$$T_{MPTCP} = \frac{CWND_{max}}{RTT_{min}} \quad (8)$$

where T_{max} is the maximum throughput for RFC 9438, $CWND_{max}$ is the maximum congestion window, and RTT_{min} is the minimum round-trip time. This equation highlights how the throughput is maximized on a single path, making RFC 9438 suitable for environments where multi-path connectivity is not required.

IV. EVALUATION

A. Simulation Parameters

The simulation of Multipath Transmission Control Protocol (MPTCP) and RFC 9438 for Device-to-Device (D2D) communication are done using Network Simulator-3 (ns-3). Table 1 depicts the Simulation Parameters used in the simulation environment and Fig. 4 depicts the network topology generated using PyVis, showing interactions between nodes during the mobility test simulation.

TABLE 1. SIMULATION PARAMETERS

Parameters	Value
TCP	MPTCP, RFC 9438
Link	802.11n
No. of Clusters	10
No. of nodes in each cluster	20
MaxBytes	Unlimited (0 indicates no limit)
Max Window Size	64 bytes
Simulation Start Time	1 second
Simulation Stop Time	100 seconds
Data Packet Interval	1 second
Speed	0.5 m/s, 1 m/s, 1.5 m/s, 2 m/s
Output Visualization	NetAnim, Flow Monitor, PyVis

B. Simulation Metrices

The simulation of Multipath Transmission Control Protocol (MPTCP) and RFC 9438 for Device-to-Device (D2D) communication is performed using Network Simulator-3 (ns-3). The mobility test evaluates network

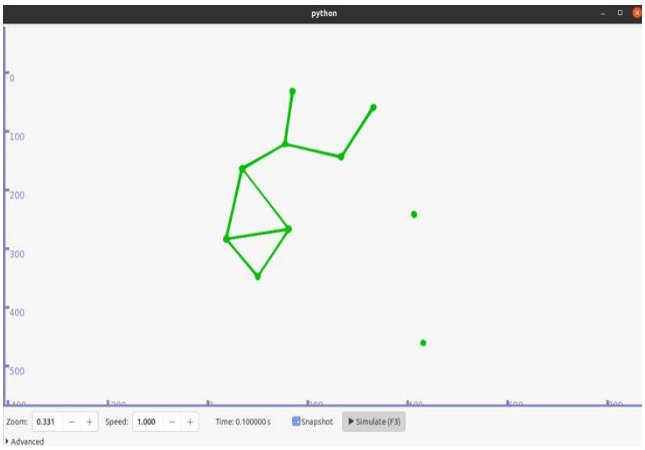

Fig. 4. Simulation Result in pyvis

performance under dynamic topology changes, with 4 key metrics such as throughput, latency, efficiency, and packet loss.

Throughput: Throughput measures the rate of successful data transfer over the network, expressed in megabits per second (Mbps). It indicates the network's capacity to handle traffic efficiently. Higher throughput reflects better network performance, especially for applications with high data transmission requirements.

Latency: Latency refers to the time delay experienced in the transmission of data packets from source to destination, measured in milliseconds (ms). Low latency is crucial for real-time applications like video streaming and gaming. Reducing latency improves the responsiveness of applications and enhances the overall user experience.

Efficiency: Efficiency represents the ratio of useful transmitted data to the total data sent, expressed as a percentage. It helps assess how effectively network resources are utilized. A high

Number of Packets Lost: This metric tracks the number of data packets that fail to reach their destination due to network issues such as congestion or signal interference. Monitoring packet loss is essential for diagnosing network issues and ensuring reliable communication in various applications.

C. Mobility Test

The mobility test simulation for MPTCP and RFC 9438 is done using Network Simulator-3, NetAnim, and PyVis, the focus is on evaluating how MPTCP (Multipath TCP) and RFC 9438 handle varying mobility conditions of data transmission. This test scenario aims to simulate realistic movement dynamics in scenarios such as mobile ad hoc networks or dynamic D2D (Device-to-Device) communication in 5G environments.

The test will measure key performance indicators such as throughput, latency, efficiency and packet loss under different mobility scenarios. As the speed of data transmission increases from 0.5 m/s to 2 m/s, the impact on MPTCP's ability to maintain stable connections and efficient data transfer will be observed. In conclusion, through the mobility test using ns-3, NetAnim, and PyVis, we aim to demonstrate how MPTCP adapts to dynamic network conditions influenced by varying speeds of node mobility.

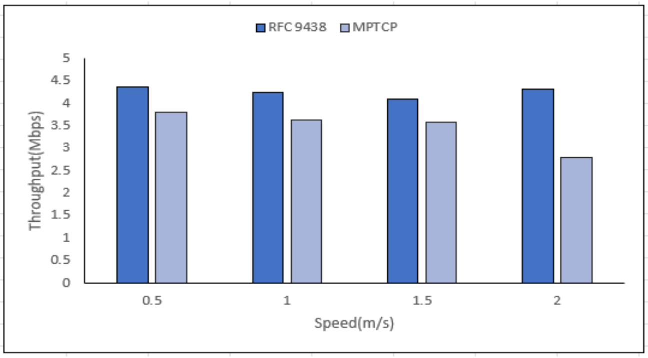

Fig. 5. Comparision of Throughput of RFC 9438 and MPTCP

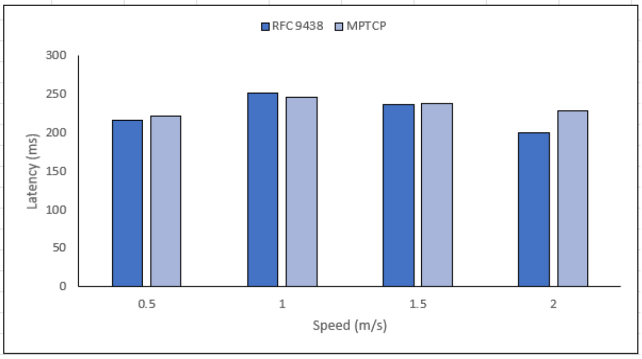

Fig. 6. Comparision of Latency of RFC 9438 and MPTCP

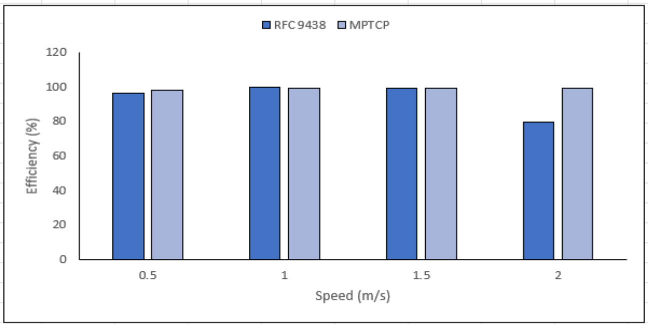

Fig. 7. Comparision of Efficiency of RFC 9438 and MPTCP

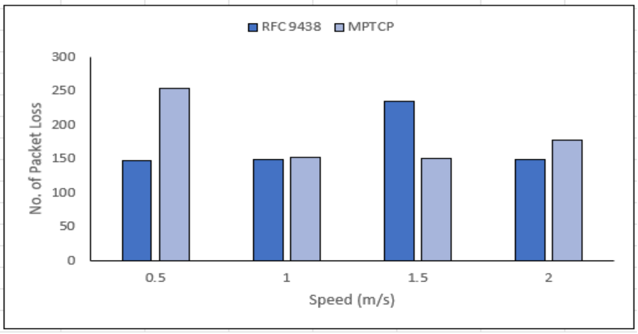

Fig. 8. Comparision of Packet Loss of RFC 9438 and MPTC

The mobility test will also evaluate the protocols' ability to adapt to changes in signal strength and network congestion as nodes move at varying speeds. MPTCP's capability to maintain stable connections through multiple paths will be compared to RFC 9438's single-path approach, with a focus on throughput and congestion handling. Real-time data visualizations through PyVis will track key metrics like packet loss and latency. By simulating different mobility scenarios, the test aims to provide insights into the protocols' performance under dynamic network conditions, helping to improve mobile connectivity protocols in future research.

1)Throughput

The throughput is measured by the Rx bitrate, which represents the rate at which data is received, expressed in kbps. This value provides insight into the overall data transmission performance between nodes. From the Flow Monitor output, we obtain the Rx bitrate directly, which is a key indicator of how well the network performs under different mobility conditions.

Fig.5 compares the throughput of MPTCP and RFC 9438 across varying mobility speeds, showing that RFC 9438 consistently achieves higher throughput. At slower speeds (0.5–1.5 m/s), RFC 9438 outperforms MPTCP by 14.8–16.7%, while at 2 m/s, the gap widens to 54.4% as MPTCP struggles with overhead and congestion control. On average, RFC 9438 delivers 24% higher throughput, making it ideal for high data transfer needs. MPTCP, though offering lower throughput, provides more stable performance, which may be advantageous in dynamic scenarios.

2)Latency

Latency is captured through the Mean delay, which measures the time taken for a packet to travel from the source to the destination, averaged over all packets in the flow. This value, measured in milliseconds, reflects the responsiveness of the network. Lower latency is preferred, especially for real-time applications. The Flow Monitor provides this mean delay, allowing us to compare how MPTCP and RFC 9438 handle packet delivery delays under different mobility scenarios.

Fig.6 compares MPTCP and RFC 9438 latency across different mobility speeds. At 0.5 m/s, RFC 9438 achieves 215.21 ms, 2.9% lower than MPTCP’s 221.90 ms, ensuring faster delivery. However, MPTCP shows slight improvements at 1 m/s and 1.5 m/s, with 2.4% and 0.7% lower latency, respectively.At 2 m/s, RFC 9438 records 199.37 ms, 12.4% better than MPTCP’s 227.71 ms. On average, RFC 9438 offers 1.4% lower latency, making it more suitable for real-time applications.

3)Efficiency

Efficiency is evaluated by comparing the rxPackets (received packets) to the txPackets (transmitted packets). The number of received packets indicates how efficiently the network handled the transmission without significant loss or retransmissions. A higher number of received packets relative to transmitted packets points to a more reliable network, especially in dynamic environments.

Fig.7 compares the efficiency of MPTCP and RFC 9438 in utilizing resources for data delivery in mobile environments. At 0.5 m/s, MPTCP reaches 98.4% efficiency, 2.07% higher than RFC 9438’s 96.4%. As speeds increase to 1 m/s and 1.5 m/s, both protocols show similar efficiencies, with MPTCP at 99.2% and 99.5%. However, at 2 m/s, MPTCP maintains a high efficiency of 99.34%, while RFC 9438 drops to 79.5%, a 24.98% difference. Overall, MPTCP demonstrates 5.5% higher efficiency on average, highlighting its adaptability in dynamic conditions.

4)Packet Loss

Packet loss is calculated by subtracting the rxPackets from the txPackets. It represents the number of packets that were transmitted but never successfully received. High packet loss can indicate issues like network congestion or mobility-related interruptions. By using the Flow Monitor data, we can precisely determine the packet loss for each flow and understand the resilience of both MPTCP and RFC 9438 under varying network conditions.

Fig.8 compares packet loss between MPTCP and RFC 9438, highlighting their effectiveness in data transmission. At 0.5 m/s, MPTCP suffers significant packet loss, with 254 packets lost compared to 148 for RFC 9438, a 71.6% increase. As speed increases, the gap narrows: at 1 m/s, MPTCP loses 152 packets, just 2% more than RFC 9438's 149. At 1.5 m/s, MPTCP shows improved management with 150 packets lost versus RFC 9438's 234. However, at 2 m/s, MPTCP records 178 lost packets, 19.5% more than RFC 9438. Overall, MPTCP experiences 23.5% more packet loss, indicating that while RFC 9438 manages packet loss better, MPTCP needs further optimization for reliability.

Overall, the test results show how both MPTCP and RFC 9438 perform under different mobility conditions. RFC 9438 works better for higher throughput and lower latency, while MPTCP does a better job of maintaining efficiency and handling changes in the network. These results highlight the strengths and weaknesses of both protocols: RFC 9438 is better for stable networks, while MPTCP is more reliable in mobile and changing environments.

V. Conclusion And Future Scope

In conclusion, the comparative analysis of MPTCP and RFC 9438 reveals the differences in their performance across various metrics, including throughput, latency, efficiency, and packet loss. RFC 9438 consistently outperforms MPTCP in throughput, with an average improvement of 24% across all tested speeds, making it more suitable for high-data-rate applications. While RFC 9438 achieves 1.4% lower latency on average, providing better performance for time-sensitive tasks, MPTCP demonstrates 5.5% higher efficiency, reflecting its adaptability and resource optimization in dynamic network environments. However, MPTCP's packet loss rate is 23.5% higher compared to RFC 9438, indicating challenges in maintaining reliability, especially at lower mobility levels. RFC 9438 maintains stable performance with fewer dropped packets, reinforcing its suitability for applications that demand reliable and consistent data transmission under varying speeds.

The future scope focuses on enhancing MPTCP performance through the optimization of its congestion control algorithms and the implementation of advanced path management techniques. These initiatives aim to improve data transmission efficiency, adaptability to varying network conditions, and overall reliability in real-world applications. First, refining MPTCP's congestion control algorithms will better manage fluctuations in network conditions. This includes developing new algorithms that can adapt dynamically in real time, ensuring smoother data transmission even in congested scenarios.

References

[1] Ahmadniai Khajekini, A., Amca, H., Ulusoy, A. H., & Ever, E. (2024). "Enhancing MPTCP performance in wireless networks with PRS-MPTCP." Pramana – Journal of Physics, vol. 162, pp. 103-560.

[2] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., & Paasch, C. (2020). "RFC 8684: TCP extensions for multipath operation with multiple addresses." RFC 8684, vol. 34, pp. 8684.

[3] Xu, L., Ha, S., Rhee, I., Goel, V., & Eggert, L. (2023). "CUBIC: RFC 9438 for fast and long-distance networks." RFC 9438, vol. 34, pp. 1-12.

[4] Nguyen, K., Villardi, G. P., Kibria, M. G., Ishizu, K., Kojima, F., & Shinbo, H. (2016). "An enhancement of multipath TCP performance in lossy wireless networks." In IEEE Local Computer Networks Conference (pp. 1-6). DOI 10.1109/LCNW.2016.36.

[5] Abdrabou, A., & Prakash, M. (2016). "Experimental performance study of multipath TCP over heterogeneous wireless networks." IEEE Transactions on Network and Service Management.

[6] Sinky, H., Hamdaoui, B., & Guizani, M. (2016). "Proactive multipath TCP for seamless handoff in heterogeneous wireless access networks." IEEE Transactions on Mobile Computing, vol. 15, no. 7, pp. 4754.

[7] Zhao, Q., Du, P., Gerla, M., Brown, A. J., & Kim, J. H. (2018). Software-defined multi-path TCP solution for mobile wireless tactical networks. IEEE, vol. 1013, pp. 1013–1018.

[8] Wu, J., Yuen, C., Cheng, B., Wang, M., & Chen, J. (2016). Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks. IEEE Transactions on Vehicular Technology, vol. 15, no. 9, pp. 2345.

[9] Oh, B. H., & Lee, J. (2015). "Constraint-based proactive scheduling for MPTCP in wireless networks." Computer Networks, vol. 19, pp. 548–563.

[10] Gao, K., Xu, C., Qin, J., Zhong, L., & Muntean, G.-M. (2019). "A stochastic optimal scheduler for multipath TCP in software defined wireless network." IEEE Transactions on Network and Service Management, vol. 19, pp. 978-1-5386-8088.

[11] Khalili, R., Gast, N., Popovic, M., & Le Boudec, J. Y. (2013). Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP. *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 760–771.

[12] Ford, A., Raiciu, C., Handley, M., Barre, S., & Bonaventure, O. (2013). TCP Extensions for Multipath Operation with Multiple Addresses. *RFC 6824*.

[13] Oh, B. H., & Lee, J. (2015). Constraint-based proactive scheduling for MPTCP in wireless networks. *Computer Networks*, vol. 19, pp. 548–563.