

Enhancing Maritime Communication Connectivity Using MPTCP-Based Link Aggregation

Vignesh Nagamuthu
Engineering Cluster

Singapore Institute of Technology
Singapore

nagamuthu.vignesh@singaporetech.edu.sg

Neelakantam Venkatarayalu
Engineering Cluster

Singapore Institute of Technology
Singapore

n.venkat@singaporetech.edu.sg

Nalam Venkata Abhishek
ICT Cluster

Singapore Institute of Technology
Singapore

venkata.abhishek@singaporetech.edu.sg

Redi Ramli

Engineering Cluster

Singapore Institute of Technology
Singapore

redi.ramli@singaporetech.edu.sg

Abstract—Reliable maritime connectivity is essential for supporting emerging communication services in coastal and offshore environments. This work presents a real-world investigation of maritime multi-link connectivity using Multipath TCP (MPTCP)-based bandwidth aggregation over commercial 4G/5G cellular networks in Singapore's southern waters. Drive tests conducted along ferry routes reveal operator-dependent coverage variations, highlighting the need for multi-operator aggregation to improve connectivity resilience. Measurement results show that MPTCP aggregation significantly enhances uplink throughput stability compared with conventional single-link TCP connectivity. In addition, a queue-aware Low Round Trip Time (LRTT) scheduler is proposed to improve latency-aware path utilization and regulate retransmission behavior under dynamic maritime channel conditions. The findings demonstrate the practical potential of MPTCP-based multi-link aggregation for improving the reliability and performance of maritime communication systems.

Index Terms—Multi-link connectivity, Bandwidth Aggregation, Maritime Communications.

I. INTRODUCTION

Reliable internet connectivity at sea is becoming increasingly critical for modern maritime operations. In Singapore, one of the world's busiest shipping hubs, there is a significant push toward maritime digitalization, particularly for autonomous and remotely operated vessels. These applications require a high uplink data bandwidth from ship to shore. The current terrestrial 5G connectivity infrastructure faces two primary hurdles. First, most base stations are deployed inland with optimized antenna downtilt for ground users [1]. Offshore coverage is incidental rather than purposefully engineered. Second, the strength and quality of the signal vary significantly with the position of the vessel. Different mobile network

operators (MNOs) exhibit distinct spatial coverage footprints. Given this heterogeneity, relying on a single access link is suboptimal. Instead, aggregating multiple heterogeneous links offers increased aggregate bandwidth, resilience to single-link failure, smoother throughput variation, and seamless failover [2]. Multipath TCP (MPTCP) provides a transport-layer mechanism to utilize multiple network interfaces concurrently under a single connection abstraction [3]. However, its effectiveness critically depends on the choice of scheduler. This work investigates maritime-optimized scheduling for MPTCP, with an emphasis on queue dynamics and retransmission control, and recovery handling.

Recent Multipath TCP (MPTCP) scheduling research has increasingly focused on making smarter use of available paths. Rather than selecting paths based on speed alone, newer approaches consider factors like queue occupancy, packet loss, and buffer blocking. For instance, Queue-Aware (QAware) [4] jointly considers delay and driver queue state, while Loss-Aware Multipath Scheduler (LAMPS) [5] factors in loss rates to avoid wasting bandwidth. Machine learning has also been applied to this problem, where Online Learning Assisted Packet Scheduler (OLAPS) [6] predicts future congestion from past path behavior, a Blocking Probability (BLP)-based scheduler [7] learns when to hold back packets to prevent buffer overflow to preserve Quality of Service (QoS). These works indicate a shift toward capacity-aware and condition-aware scheduling, where packets are assigned to the path best suited to carry them at a given time. However, most existing MPTCP schedulers have been evaluated in general heterogeneous or mobile network settings, with limited attention to real maritime environments. Maritime cellular connectivity is highly dynamic due to vessel movement, offshore coverage gaps, changing signal quality, and operator-dependent spatial coverage differences. Therefore, there is a need to evaluate MPTCP-based aggregation and scheduling under real maritime channel conditions.

This research is supported by the National Research Foundation, Singapore (NRF), Maritime and Port Authority of Singapore (MPA) and Singapore Maritime Institute (SMI) under its Maritime Transformation Programme (Project No. SMI-2024-MTP-01). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of NRF, MPA and SMI.

Motivated by this gap, a real-world measurement campaign was conducted across Singapore’s southern waters to investigate operator-dependent coverage variability and its impact on maritime uplink performance. The results motivate the use of multi-link aggregation for improving maritime communication resilience. In this paper, we implement and evaluate two scheduling strategies: (a) Default Linux scheduler, (b) Queue-aware LRTT with a $2 \times$ Bandwidth-Delay Product (BDP) threshold [8] [9] [4].

II. MEASUREMENT CAMPAIGN

To understand the practical variability of offshore cellular connectivity, a measurement campaign was conducted across Singapore’s southern waters covering St. John’s Island, Kusu Island, and Lazarus Island. Commercial user equipment devices equipped with SIM cards from two different mobile network operators were deployed onboard a ferry. Devices were positioned on the open upper deck to minimize structural signal obstruction. Using a professional cellular testing tool, key metrics including Reference Signal Received Power (RSRP), Signal-to-Interference-plus-Noise Ratio (SINR), and serving cell identifiers were logged continuously throughout the ferry route which is around 80min.

The User Equipment (UE) devices (Samsung S22s) with mobile network testing tool QualiPoc (from Rohde & Schwarz) was used to monitor the LTE network KPIs of two different mobile operators (A and B) along the ferry route. The network performance evaluation was conducted using iPerf3 executed from the end device to measure transport-layer throughput and TCP retransmission behavior over the communication link. The end device operated as an iPerf3 client, transmitting traffic to a remote iPerf3 server hosted in AWS in this case, over a specific test duration and reporting interval. The measured metrics included instantaneous throughput and the number of retransmissions, which provide an indication of packet loss, congestion, or wireless link impairments. By analyzing throughput together with retransmission statistics, the efficiency and reliability of the network connection under varying channel and mobility conditions can be quantitatively assessed.

The TCP network throughput achieved while accessing the AWS server through the the service from Operator A is shown in Fig. 1(a). A wide variation in the network throughput is observed over the duration of the ferry route with average throughput of 4.8Mbps uplink is observed. The independently and simultaneously conducted RSRP measurement for the LTE link from operator A and operator B is shown Fig. 1(b) and 1(c) respectively. The mobile network measurement tool allows to monitor the modulation index as well and in Fig. 1(b) and 1(c) its clearly evident that the higher order modulation schemes 64-QAM is used only in locations or periods when the RSRP is high. When the RSRP is low, the link switches to QPSK. Furthermore a visual correlation of RSRP and the TCP link throughput is observable for Operator A from Fig. 1(a) and 1(b).

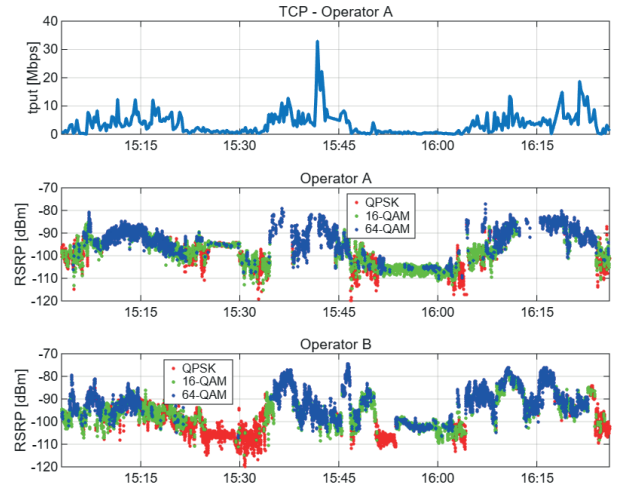


Fig. 1: (a) Throughput with TCP link using Operator A (b) Measured RSRP of Operator A (c) Measured RSRP of Operator B over the duration of the ferry journey

The RSRP measurements revealed clear spatial variability in signal strength between operators. Near the shoreline, both operators provided comparable RSRP levels. However, as the vessel traversed offshore regions, the dominance shifted periodically between operators. In certain segments of the route, Operator B delivered significantly stronger signal strength, while in other segments Operator A provided superior coverage. These variations were consistent with differences in base station placement and antenna orientation. Given that achievable uplink throughput is strongly correlated with signal strength and link quality, these findings indicate that no single operator provides consistently optimal performance throughout the route. Consequently, a multi-operator aggregation approach is highly beneficial in maritime contexts. Such aggregation not only increases peak bandwidth but also enables seamless failover when one operator’s link quality deteriorates. These observations directly motivate the use of MPTCP for maritime communications. This serves as a motivation for us to develop a MPTCP-based multilink solution that aggregates links across multiple cellular connections with an objective to enhance the overall throughput.

A testbed setup as illustrated in Fig. 2 was developed and used in the ferry to carryout throughput measurements and network performance investigations. Two smartphones acting as mobile Wi-Fi hotspots provide the MPTCP subflows for this measurement campaign. The first path utilises a Phone 1 (Google Pixel device) with LTE link to one of the Operator networks, while the second path uses Phone 2 (Samsung Galaxy S23 device) connected to another operator network. Both networks operate within Singapore. The end-devices used are two laptops. One laptop is connected to Phone 1 through a TCP link. The second laptop is connected via MPTCP link with two interfaces to both Phone 1 and Phone 2 simultaneously. The laptop with MPTCP link has a custom Linux kernel built from the mptcp-net-next Git repository and

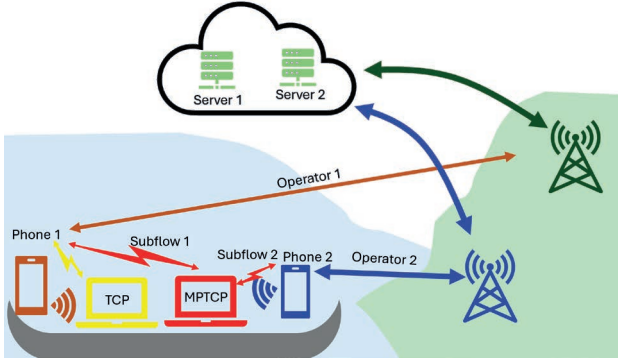


Fig. 2: Illustration of MPTCP-based communication link setup on the vessel

leverages on the default network scheduler.

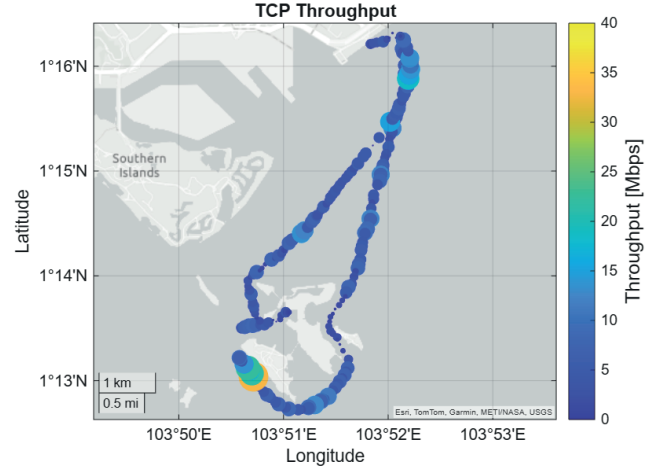
The measured throughput of a) the TCP link and the b) the MPTCP link along the ferry route is shown in Fig. 3(a) and 3(b) respectively, where evidently the superior throughput performance of the MPTCP-based link with aggregated throughput across both the subflows 1 and 2 is seen. Fig. 4 shows the empirical CDF of the throughput measured across the entire duration of 80min of the vessel. It is seen that while for 50% of the journey the TCP link has more than 2.5Mbps uplink throughput, the MPTCP link has more than 7.5Mbps uplink throughput. By transmitting over multiple links concurrently, the system is able to combine the available bandwidth from different operator, resulting in higher and more stable throughput. Aggregation also provides resilience because if one operator does not serve a particular location, the application traffic will not break. Instead, it will seamlessly continue over the other operator, enabling smooth failover and uninterrupted connectivity.

III. NETWORK SCHEDULER

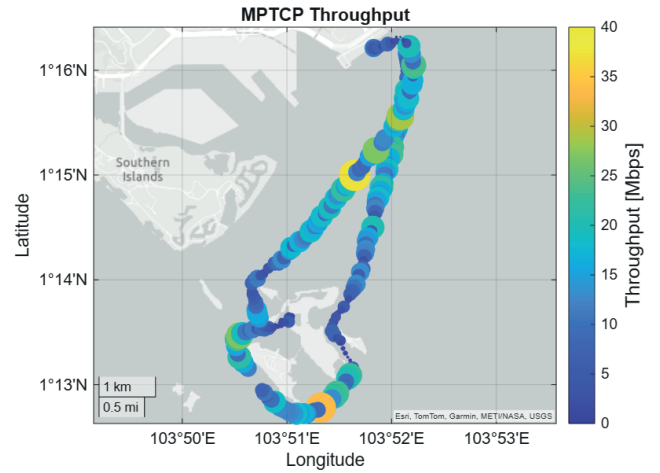
A custom Linux kernel was built from the open-source mptcp-net-next repository, which extends the Linux networking stack with the latest MPTCP features [10]. This kernel exposes the MPTCP scheduler framework, enabling control over subflow selection. The custom LRTT schedulers was implemented and integrated directly into the kernel networking stack.

Early default scheduler implementations used a minRTT scheduler, which selected the subflow with the lowest RTT. [11] The current Linux kernel default scheduler replaces RTT based selection with a linger time metric [10]. The default scheduler selects subflows based on a linger-time metric, which estimates how long each subflow's send queue will take to drain given its current queue occupancy and pacing rate. For each subflow i , the scheduler computes a linger-time estimate $\ell_i = w_i / \bar{p}_i$, where w_i is the send buffer occupancy and $\bar{p}_i$ is a smoothed pacing rate updated as a weighted average:

$$\bar{p}_i^{(t+1)} = \frac{\bar{p}_i^{(t)} \cdot w_i + p_i^{(t)} \cdot B}{w_i + B} \quad (1)$$



(a) TCP Link



(b) MPTCP Link

Fig. 3: Measured Throughput along the ferry route using (a) TCP link and (b) MPTCP link.

where $p_i^{(t)}$ is the instantaneous pacing rate and B is the burst size committed in the current scheduling round. The subflow with the minimum ℓ_i is selected for the next transmission burst. The scheduler dynamically adapts to instantaneous network conditions. For example, when one subflow accumulates queued data, its linger time ℓ_i increases. Therefore, subsequent transmissions will be redirected to the less loaded subflows. This self balancing property allows the scheduler to distribute traffic across heterogeneous links without explicit configuration. For retransmissions, the scheduler opportunistically selects a subflow with an empty send queue, so that lost packets can be resent without queuing behind pending data. If no such subflow is available, retransmission falls back to TCP's per-subflow retransmission mechanism on the original path.

The queue-aware Lowest RTT (qaLRTT) scheduler selects subflows based on the smoothed round-trip time (SRTT), denoted as srtt_us_i for subflow i . At each scheduling decision,

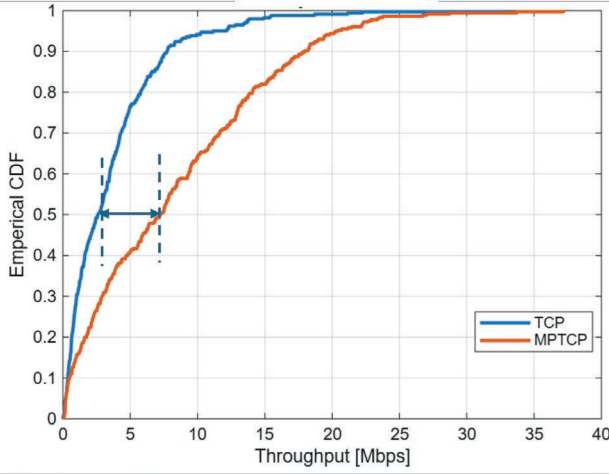


Fig. 4: Empirical CDF of throughput over the entire duration of the ferry journey

the scheduler iterates over all subflows and selects the one with the minimum $srtt_{us_i}$, subject to two conditions:

- 1) The subflow must be active, have available stream memory, and pass TCP's send readiness check.
- 2) The unsent queue length must not exceed twice the BDP of that path, preventing excessive buffering on the lowest RTT subflow [12].

BDP is the amount of data that can be in flight on a subflow at any given time. This queue-awareness mechanism ensures that the scheduler does not blindly send data to the lowest RTT path when that path is already congested, thereby preventing a single subflow from becoming a bottleneck due to excessive buffering. When the preferred low RTT subflow exceeds this threshold, the scheduler selects the next eligible subflow, effectively distributing load away from saturated subflow without abandoning the RTT first selection policy entirely.

If no subflow satisfies these criteria, the scheduler falls back to the kernel's default subflow selection helper and retransmissions follows the same logic as the default scheduler. Additionally, a netdevice notifier is registered to detect interface up and recovery events, triggering subflow re-establishment on recovered links.

The default scheduler is RTT blind as it will treat high latency satellite link and a low latency LTE link equally as long as their queue occupancies and pacing rates are similar. This can cause data to be sent on a high latency path simply because the queue is momentarily empty, resulting in head-of-line (HOL) blocking at the receiver [8].

The qaLRTT scheduler addresses this limitation by explicitly selecting the subflow with the lowest smoothed RTT R_i , ensuring that lower-latency paths are always preferred. This makes qaLRTT better suited for latency sensitive marine traffic where achieving minimal end-to-end delay is very crucial. To prevent the lowest RTT path from accumulating excessive backlog, an anti-overbuffering threshold is applied. Unlike the

default scheduler, qaLRTT does not continuously rebalance data load based on the subflows' queue length. It selects an alternate path only when the overbuffering threshold is exceeded or a lower RTT path is detected.

IV. SCHEDULER PERFORMANCE

The tested illustrated in Fig. 2 is extended with three laptop devices on board the vessel. While the first laptop remains as TCP link to connect to Phone 1, the other two laptops use a MPTCP based link to both Phone 1 and Phone 2, and use either the default scheduler or the qaLRTT based scheduler. Similar measurement campaigns as discussed in Sec. II, were carried out with three scenarios including a) TCP link, b) MPTCP link with default scheduler and c) MPTCP link with qaLRTT scheduler. In these measurements along with the uplink throughput from iPerf3, the number of retransmissions were also captured across the entire ferry route for all 3 scenarios.

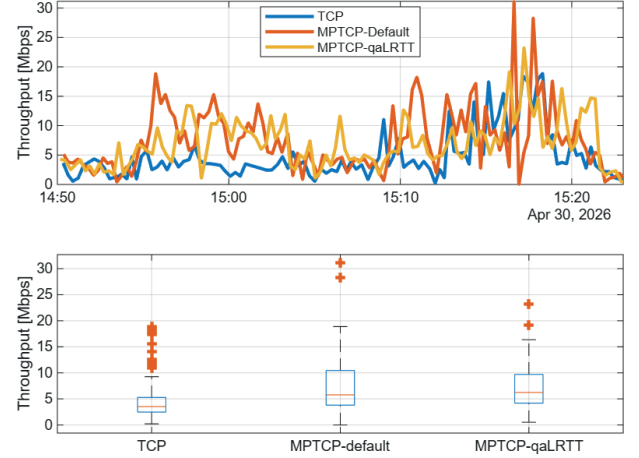


Fig. 5: Measured Throughput for the three different connectivity links, over time, across the ferry route

As shown in Fig 5, both MPTCP variants outperform standard TCP across the observation window. Default achieves the highest peak throughput, reaching approximately 30 Mbps around 15:20, but exhibits high variability throughout the session. qaLRTT achieves comparable throughput with a slightly lower peak of approximately 22 Mbps, but maintains a more consistent distribution. The box plots confirm that default and qaLRTT have similar medians of approximately 5 to 6 Mbps, both substantially higher than TCP at approximately 4 Mbps. However, default shows a wider interquartile range, indicating greater throughput variability, while qaLRTT achieves a tighter distribution suggesting more stable utilisation of the available links.

As shown in Fig 6, qaLRTT has significantly lesser retransmissions compared to both TCP and default. The box plot shows that qaLRTT has the lowest median retransmissions, with a tighter interquartile range. This indicates that with qaLRTT, a more reliable and consistent data transfer can be achieved. Default exhibits the highest median retransmissions

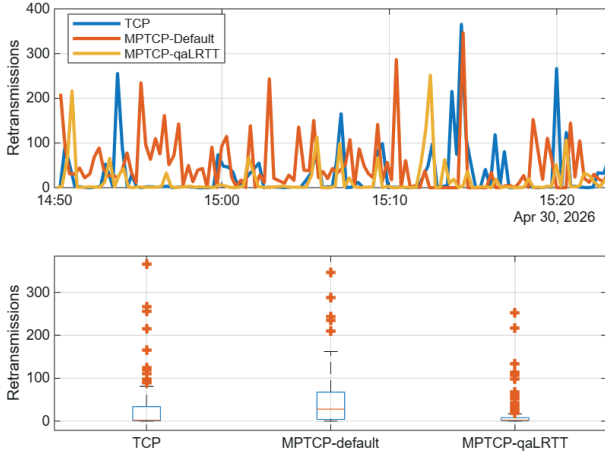


Fig. 6: Measured number of retransmissions for the three different connectivity links, over time. across the ferry route

among the three. This suggests that even though default achieves better throughput, it undergoes excessive buffering leading to increased number of retransmissions.

The CDF plots in Fig 7 corroborate the earlier findings in Fig 5 and Fig 6. In the throughput CDF, both MPTCP variants are shifted significantly to the right of TCP, with median throughput approximately doubling that of TCP. In the retransmissions CDF, qaLRTT rises more steeply than default, with 90% of samples falling below approximately 100 retransmissions compared to 150 for default. Therefore, qaLRTT achieves a favourable trade-off by achieving comparable throughput to default and having lower retransmissions making it suitable for rapidly changing maritime environment.

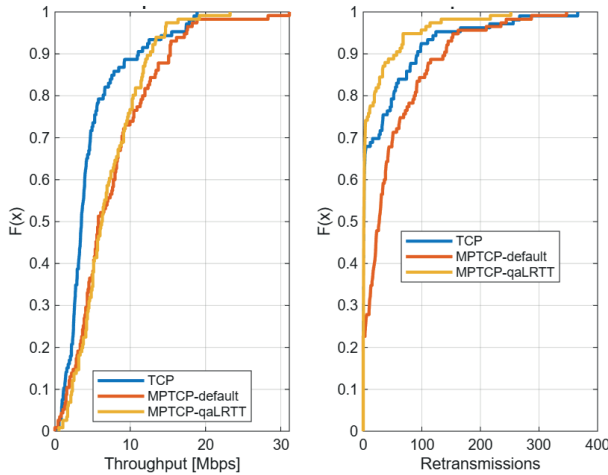


Fig. 7: Measured number of retransmissions for the three different connectivity links, over time. across the ferry route

V. CONCLUSION

Overall, the presented work demonstrates the practical feasibility and effectiveness of MPTCP-based multi-link maritime connectivity using existing terrestrial cellular infrastructure,

providing an important step toward resilient and high-capacity communication systems for future smart and autonomous maritime operations.

ACKNOWLEDGMENT

The authors would like to thank Dr. Pei Yiyang, Dr. Tay Chuan Beng from Singapore Institute of Technology, for the fruitful discussions and support in the measurement campaigns. The authors would also like to sincerely thank Kannan Navaneethan for his valuable assistance and contributions to the experimentation, and testing involved in this work.

REFERENCES

- [1] V. Nagamuthu, N. Venkatarayalu, A. P. C. P. Yiyang, and N. V. Abhishek, "5g cellular network testing based maritime radio channel characterization," in *2025 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, 2025, pp. 1–5.
- [2] D. Dimopoulos, A. K. Salkintzis, D. Tsolkas, N. Passas, and L. Merakos, "Evaluating the impact of packet scheduling and congestion control algorithms on mptcp performance over heterogeneous networks," *arXiv preprint arXiv:2511.14550*, 2025.
- [3] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses," Internet Engineering Task Force (IETF), RFC 8684, Mar. 2020, standards Track. Obsoletes RFC 6824. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc8684>
- [4] T. Shreedhar, N. Mohan, S. K. Kaul, and J. Kangasharju, "Qaware: A cross-layer approach to mptcp scheduling," in *2018 IFIP Networking Conference (IFIP Networking) and Workshops*, 2018, pp. 1–9.
- [5] E. Dong, M. Xu, X. Fu, and Y. Cao, "Lamps: A loss aware scheduler for multipath tcp over highly lossy networks," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE, 2017, pp. 1–9.
- [6] Y. Xing, K. Xue, Y. Zhang, J. Han, J. Li, and D. S. WeiMember, "An online learning assisted packet scheduler for mptcp in mobile networks," *IEEE/ACM Transactions on Networking*, vol. 31, no. 5, pp. 2297–2312, 2023.
- [7] S. Das, J. Guo, K. Parsons, Y. Nagai, T. Sumi, N. Sakaguchi, P. Orlik, and S. Kalafatis, "Learning based scheduling and adaptive congestion control for multipath quic," in *2025 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2025, pp. 830–836.
- [8] S. Ferlin, Alay, O. Mehani, and R. Boreli, "Blest: Blocking estimation-based mptcp scheduler for heterogeneous networks," in *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, 2016, pp. 431–439.
- [9] B. Felix, I. Steuck, A. Santos, S. Secci, and M. Nogueira, "Redundant packet scheduling by uncorrelated paths in heterogeneous wireless networks," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00498–00503.
- [10] Multipath TCP Development Team, "mptcp_net-next," GitHub, 2025, [Online]. Available: https://github.com/multipath-tcp/mptcp_net-next. [Accessed: May 29, 2026].
- [11] Multipath TCP Development Team, "mptcp," GitHub, 2025, [Archive]. Available: <https://github.com/multipath-tcp/mptcp>. [Accessed: May 29, 2026].
- [12] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "Bbr: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time," *Queue*, vol. 14, no. 5, pp. 20–53, 2016.