Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 1, pp. 337–348, DOI 10.12694/scpe.v26i1.3512

NEW PATH SCHEDULING APPROACH FOR MULTIPATH TRANSMISSION CONTROL PROTOCOL

DEEPIKA SINGH KUSHWAH* AND MAHESH KUMAR†

Abstract. A popular transport layer protocol that makes extensive use of multi-path communication technology is called Multi-Path Transmission Control Protocol (MPTCP). Making use of the end-hosts multi-interface capabilities are MPTCP's primary goal. However, the protocol's main weakness is lower channel usage suffers from a sharp decline in throughput due to its sudden congestion window (cwnd) growth policy. Furthermore, the performance of MPTCP is significantly reduced when there are many possible channels for transmission because of different wireless path parameters such as bandwidth, loss rate (due to congestion), and delay necessitate higher re-ordering at the receiver end. As a result, it causes a significant increase in buffer-blocking and needless retransmissions. Also, a strategy that efficiently schedules and distributes the load over accessible, qualifying channels have been proposed. However, our proposed policy takes all these parameters into consideration and the performance has improved significantly can be seen by the graphs presented in the later section of the paper.

Key words: *Multi-Path Transmission Control Protocol, Stream Control Transmission Protocol, Linked Increase Algorithm, Balanced Link Adaptation Algorithm*

1. Introduction. Over the last few years, many applications have been adopting the single-path communication paradigm when utilizing the Transport layer (Layer-4) protocols, such as Transmission Control Protocol (TCP) [1] and User Datagram Protocol (UDP) [2]. Unfortunately, previous Internet technologies have blindly focused on single-way communication paradigm because of its simplicity and minimal network overhead, even though the wide path variety has undoubtedly been accessible on the Internet. However, single-path Layer-4 protocols lack the necessary stability and proficiency to maintain their performance in the face of constantly increasing traffic. Furthermore, single-path communication protocols are unable to provide significant fault tolerance requirements to a variety of applications with resource constraints (such as BW and throughput), nor can they provide flexible support for Quality of Experience (QoE) and Quality of Service (QoS). The Internet can gain enhanced customer fulfillment and reliability by implementing the contemporary multi-path communication paradigm [3]–[5]. Multi-pathing is rapidly gaining attraction in the present day thanks to its amazing qualities of increased network resilience, performance, and dependability. As a result, multi-homing and multi-streaming are becoming more and more common in Internet Protocol (IP) networks. Additionally, real-time online applications like video conferencing, online gaming, live streaming of videos and live sports broadcasting present new difficulties due to resource limitations, which encourages routing through several possible network paths.

However, the problems with multi-path communication technology are related to routing over multiple paths (e.g., what basis should computation and packet forwarding be carried out for multiple paths) and traffic splitting (e.g., what basis should flow stripping be scheduled over multiple available paths). In addition to these difficulties, there are further concerns like Fairness: equal distribution of resources (such as BW) at the bottleneck; packet reordering: this resolves the issue of receiver buffer blocking caused by packets being routed across numerous pathways. If several routes are chosen and share the same bottleneck link, the background traffics [6].

In recent times, there has been a significant surge in industrial interest and absorption of extensive research on multi-path transmission from research societies and standardized bodies such as IEEE and IETF. The traditional protocols used in heterogeneous network interface utilization includes MP-TCP [7] and Stream

*CSE Department, Jaypee University of Engineering and Technology, Guna, India (deepika.kush09@gmail.com)
†CSE Department, Jaypee University of Engineering and Technology, Guna, India (Corresponding author, mahesh.chahar@gmail.com)

Control Transmission Protocol (SCTP) [8]. Multi-homing is used by SCTP and its Concurrent Multi-path Transfer extension (CMTSCTP) [9] to build transmission associations that include multiple paths between two end hosts. MP-TCP differs from SCTP in an established connection by systemizing many TCP sub-flows at once. This is what MP-TCP does to support both multi-path transmission and retains backward compatibility with traditional TCP. According to Xu et al. [10] congestion control strategies for multi-path Layer-4 protocols was presented.

TCP is used in some way by the network applications and infrastructures that are being used nowadays. However, SCTP solutions are not compatible with TCP, which makes them more expensive to deploy in more practically based network environments [11]. The previously mentioned reason is what drives the most recent procedures and initiatives aimed at improving MP-TCP performance. Additionally, MP-TCP functions flawlessly with the middle-box integrations found in modern Internet architecture [5] [12]. Furthermore, when data segments are shredding middle-boxes in the traditional Internet architecture, MP-TCP gives significantly better performance (e.g., when compared with regular SCTP and TCP). Thus, with modern Internet infrastructure, MP-TCP enables superior deployment skill.

The paper is organized as follows. Related Work is discussed in section 2. Proposed Work is explained in subsection 2.3. Our new algorithm is in subsection 3.5, experimental results are in subsection 4.2, and the conclusions follow in section 5.

1.1. Motivation. However, a significant amount of ongoing research is currently being done with a focus on various aspects of multi-path communication strategy. The round-robin scheduling approach is a standard mechanism used by the traditional scheduling policies to schedule data over many accessible network paths. This competition is round-robin data packets for the transport layer protocols are dynamically divided into equal-sharing segments over all accessible network channels by the scheduling strategy. In the meantime, this scheduling policy ignores the different attributes (such as available BW, path quality, and delay) of several open network paths and schedules the transfer without any thought. As a result, there is no denying that the traditional round-robin data scheduling strategy significantly impairs application-level throughput performance and results in significant out-of-order delivery to the destination. Out-of-order delivery thus results in duplicate congestion window reductions and needless quick retransmissions as well. It also causes the network's buffer-blocking problem.

Regarding the multi-path communication policy, the MP-TCP protocol then enters the picture. The fundamental design standard of MP-TCP is predicated on the joint capabilities of TCP and CMT-SCTP. In contrast to SCTP, MP-TCP expertly establishes many concurrent TCP sub-flows within a connection that has already been created. Specifically, the primary goal of MP-TCP is to enhance throughput performance. However, MP-TCP has also suffered from the severe problem caused by packet re-ordering at the receiver's end. Nevertheless, a serious issue brought on by packet reordering at the recipient's end has also affected MP-TCP. This is due to the different qualities and characteristics of the pathways between the two end hosts. The end host must buffer the earlier received packets in order to guarantee reliable data packet delivery to the upper (i.e., application) layer (receiver), after which it must wait for the remaining packets to be transmitted across sluggish pathways. It thus causes the overall transmission performance to be compromised. Additionally, the buffer-blocking problem impedes packet exchange and exacerbates the idle connection problem, which raises spurious retransmissions and average End-to-End (ETE) latency in addition to decreasing the network's overall throughput performance. Aside from this, the receiver buffer occasionally runs out of room to accommodate incoming data packets. In this scenario, the receiver buffer finally notifies the sender of its zero window value. This stops the sender from sending more, which forces it to enter the persist phase and cause a significant decrease in application-level throughput performance. Nonetheless, a lot of the suggestions have sought to provide effective traffic scheduling while attempting to mitigate the aforementioned problems depending on certain estimates of path and connection quality [13]–[15].

1.2. Contributions and Highlights of this Paper. To improve multi-path data transmission performance by utilizing delay variations, this research suggests an approach technique to MP-TCP. When scheduling data over numerous paths, our approach effectively accounts for varied path characteristics and handles unjustified cwnd growth modifications. We have taken into consideration delay variation as a statistic that, in the end, shows the capacity of the resources for every path. In addition, we have proposed an adaptive rapid retransmission policy based on delay variation that further controls the suitable transmission rate for every path. As a matter of fact, our approach effectively predicts congestion during the rising phase by periodically evaluating the increasing delay variation of the data segments that have been transferred over a path. Then, our approach modifies the cwnd growth scheme for the congested path.

- We propose policy that is adaptive and flexible multi-path data transfer policy for MP-TCP.
- Also proposes delay variations based adaptive data packet scheduling and adaptive fast retransmission policy to MP-TCP.
- We confirmation of the effectiveness of the proposed approach by conducting extensive experiments and by doing comprehensive evaluation. The validation of our approach has been carried out on ns-2.
- Our offers Improvement of 46 percent, 19 percent, and 34 percent better in throughput performance for varied packet loss rate, BW, and path delay cases respectively. Additionally, the approach offers improved file transfer time as compared to MP-TCP.

2. Related Work. The ultimate goal of this section is to outline for the reader the main concerns affecting various SCTP and MP-TCP based approaches. Additionally, in this area, new concerns related to these difficulties have been included. In addition, we will comprehend in this section how study endeavors have been undertaken to address all these problems. Additionally, even if after resolving all of these difficulties, what new issues have surfaced? There are two types of multi-path Layer-4 protocols: connection-oriented and connectionless (also known as multi-path real-time protocols). Still, this section will focus exclusively on connection-oriented protocols.

When TCP was first designed, end hosts could only connect using single interface support that is, with the same IP address. However, the increased usage of complex networking devices has led to the creation of end-hosts that can handle numerous interfaces. In the modern era of communication, cellphone networks and Wi-Fi are extensively accessible in urban areas. It would be desirable for users of smartphones and tablets to be able to start a connection, especially a TCP connection in a Wi-Fi hotspot and continue it later to their 3G interface [16] [17].

As a result, extremely intelligent middle-boxes are developed, thereby dispelling the need for ETE design. Moreover, these sophisticated middle-boxes have provided a host of benefits, including speed enhancements, security (firewalls), usage (load balancers), etc. But the rapid collapse of these clever middle-boxes also resulted in increased complexity when it came to Layer-4's design protocols. However, SCTP's adoption is quite challenging in the current Internet architecture since it relies on the very simple assumption that the received data segments remain identical. As of now, MP-TCP is the only protocol that was created with middle-box functionality in mind. However, under a few unique circumstances, MP-TCP's performance might be considerably impacted when working with certain middle-boxes as well. Consequently, a large amount of study is needed to handle middle-boxes efficiently [3] [18].

2.1. Multiple path based Proposals. Researchers were certain that multi-pathing was inevitable because new designs are always striving to develop more and more workable solutions. This in turn led to the development of MP-TCP. Originally, several congestion control strategies that immediately extended TCP New Reno for the aim of creating the MP-TCP policy, also known as the Linked Increase Algorithm (LIA) [19], i.e., recommended policies, directly initiate the TCP New Reno's capability on each sub-path independently. When the available network paths share bottleneck links with the network paths used by MP-TCP users the network's single-path TCP users may experience significant unfairness as a result of this direct extended version. To structure an effective multi-path Layer-4 protocol that is specifically compatible with the conventional TCP (i.e., Coupled Congestion Control (CCC) algorithm), several academics have proposed several techniques [20]. These suggested methods in [21]–[24] are most suited for situations where similar or minimal differences in Round Trip Times (RTTs) have been recorded, as they typically only use the best paths that are available to the users. However, these algorithms experience floppiness and lower responsiveness.

First of all, these algorithms are not always able to adjust quickly. Specifically, they are not able to explore the paths that have a greater channel and congestion-induced loss probability, which results in a significantly lower level of responsiveness. Second, the network exhibits extreme floppiness according to these techniques. Peng et al. [25] suggested that the half-coupled congestion control approach is inhospitable to single-path TCP users. It inflates during different RTT iterations of every feasible sub-path.

The authors have subsequently defined design standards that provide assurance of existence, stability, and uniqueness of the system. Their method, known as the Balanced Link Adaptation Algorithm (BALIA) [26], was primarily concerned with performance measures including cwnd fluctuations, TCP friendliness, and receptiveness. Additionally, Lubna et al. [27] proposed the Dynamic OLIA (DOLIA) policy, which successfully restrained MP-TCP's aggression about the cwnd expansion. Many adaptive scheduling policies that guarantee in-order data delivery have already been proposed to mitigate the buffer-blocking issue. In the meantime, researchers have previously done a great deal of research on topics like path heterogeneity [28]–[30], congestion control techniques [31]-[34], enhanced traffic management schedulers [21][24][35].

The MP-TCP SBD, or Shared Bottleneck Detection based Coupled Congestion Control [36] is an issue to be put light on. The performance of other sub-flows is impacted by the extra loss and delayed sub-flow, as demonstrated by the writers. As a result, the throughput performance is eventually hampered and the bottleneck of many MP-TCP connections develop. The authors recommended using fountain codes to effectively manage the variable attributes of numerous pathways to mitigate these impacts.

For multipath transmission, Thomas et al. [37] proposed a hybrid congestion control policy. The Multi-flow Congestion Control with Network Assistance (MFCCNA) the policy has been provided by the authors; their approach dynamically examines the available topological information about the network. The ultimate goal of MFCCNA is to improve network resource consumption without sacrificing friendliness. Nevertheless, the harshness of the window growth program was not taken into account by MFCCNA thus causes a decline in performance about packet losses. Until its cwnd is entirely filled with data, the current scheduler (default), minimal RTT (RTTmin) first, distributes data packets to the fastest available sub-flow. The leftover packets are then suitably assigned to other available sub-flows. Although this method outperforms the traditional round robin data scheduling scheme in the majority of circumstances, new research indicates that the RTTmin approach still needs more thorough assessment and consideration when the available network pathways have different attributes.

Thus, this / it further contributes to the Head-of-Line (HoL) [38] [39] blocking problem, which raises the likelihood of receiving data packets in a disordered manner at the receiver buffer and, in turn, lowers the throughput performance at the application level. As a result, many improved dynamic scheduling techniques have been created in the past for efficiently dividing up data packets among several open network channels. While the most recent delay-aware scheduling policies can enhance MP-TCP performance, they are unable to provide reliability assurances. However, because of the diverse environment, delay aware scheduling strategies need to handle the problem of long retransmissions and highly unreliable features of multi-path sessions.

2.2. Security, Reliability and compatibility of MPTCP. Firstly security, because multipath routing fragments cross-path data, firewalls and virus scanners become ineffective when they can only able to track traffic on one path. While MPTCP doesn't enforce encryption on its own, it can be used in conjunction with secure encryption protocols such as TLS (Transport Layer Security) or IPsec (Internet Protocol Security). When data is encrypted using these protocols, it remains confidential and secure against eavesdropping attacks, ensuring that data transmitted over MPTCP connections is protected [40].

Secondly reliability, MPTCP enhances reliability by providing fault tolerance, increasing throughput, adapting to network fluctuations, supporting seamless handover, implementing selective retransmission, and employing efficient congestion control mechanisms.

Thirdly for compatibility with existing protocols, a thorough study is being done before implementing MPTCP. The middleboxes play a very important role in the performance of the protocol. Outside protocols see MPTCP is an inside part of TCP only. Therefore our introduced changes are also not affecting other protocols that deal with the transmission of data in the network.

2.3. The Proposed Method. When evaluating the wired network scenario during multi-homed communication, there is typically a chance that the available channel parameters (i.e., path latency, BW, and loss rate) will vary greatly. Therefore, if MPTCP uses the entire available network interfaces for multi-path transmissions, it will surely result in the problem of packet reordering in the framework. As a result, this strategy severely degrades the network's throughput performance. In order to guarantee efficient support for MP-TCP, this article introduces the assorted path scheduling approach. Abrupt traffic management (i.e., scheduling) schemes and ridiculously incorrect congestion window growth adjustments are guaranteed to be efficiently handled by our proposed policy. To achieve this, our policy takes into account the quality of any network path that is participating in data transmission and also modifies the data scheduling approach.

The scalability of the proposed work is wherever MPTCP is being used; the proposed approach can be used i.e. on more complex and large networks too. The proposed approach takes RTT variations and congestion window capacity as its metric to decide where to schedule next to. Calculating RTT each time and scheduling desired traffic is a bit complex procedure but ns-2 does this at ease. Also, while scaling Multipath TCP to larger networks or more complex environments present challenges related to overhead, path management, resource constraints, interoperability, security, privacy, and management, these challenges can be addressed through optimizations in MPTCP implementations, efficient path management strategies, resource optimization techniques, interoperability efforts, enhanced security measures, and effective management and monitoring tools. By addressing these considerations, MPTCP can be effectively scaled to meet the demands of diverse networking environments, providing improved performance, reliability, and flexibility for data transmission over multipath networks.

Delay variations, an essential metric that effectively illustrates the representation of resource availability for a candidate path, is what our policy has employed for this. In addition, an adaptive rapid retransmission policy based on the delay variations metric is presented in this study for the same. Our approach bears the obligation to control an effective transmission management plan in line with the available network pathways varying delay.

$$\text{Packets\_sent(on\_a \_ path)} = \text{cwnd\_currentpath -unacknowledged\_packets} \tag{2.1}$$

$$\text{Waiting\_ time} = \text{immediately\_ sent\_ packets - packets\_ not\_sent} \tag{2.2}$$

3. Problem Identification. Indeed, a packet scheduler is required to enable the use of various accessible pathways in MP-TCP. Selecting which data packet should be sent via which open path the primary duty of the packet scheduler is. On the other hand, MP-TCP has to transmit packets at the receiver's end to guarantee system reliability in the same order in which they are sent from the sender's end. Specifically, the features of the dissimilar pathways, such as packet loss rate, delay, and BW availability, will consistently result in data being received in an erratic manner at the recipient's end. Nonetheless, the receiver buffer included in the standardized MP-TCP eventually helps the protocol reorder data packets originating from all of the accessible interfaces. However, the MP-TCP receiver buffer capacity is extremely constrained; as a result, the buffer becomes blocked when the frequency of receiving increases and the amount of unordered data packets increases. The MP-TCP sender maybe forced by this circumstance to send the new data packets throughout the stable network pathways that are open for transmission. The reason for this is that there are insufficient buffer spaces at the receiver's end for the incoming fresh data packets. This indicates that unstable paths (i.e., paths with greater delay variability) equally, disrupt the available stable routes inside the MP-TCP connection for the delivery of data packets. Consequently, the MP-TCP application-level throughput performance may be significantly decreased by unstable pathways utilized for multi-path data transfer.

3.1. Proposed Solution. The suggested approach's primary goal is to estimate unstable pathways promptly. As a result, our approach can effectively handle irrational congestion window growth adjustments in addition to scheduling transmissions via alternative stable pathways. More specifically, our suggested method arranges the transmission load over various available stable sub-paths suitably and estimates the paths with higher delay fluctuations. According to Equation 3, the propagation delay (Prop-Delay), processing delay (Proc-Delay), queueing delay (Queue-Delay), and transmission delay (Trans-Delay) generally make up the path delay (Path-Delay).

We quickly discuss the aforementioned network delays that ultimately led to the overall packet delay to illustrate the impact of path delay. Trans-Delay: - The amount of time required by the sender's Layer-1, or Physical Layer, to transfer every bit onto the wire, Prop-Delay: - The flight time of bits over the connected channel is how it is defined. Queue-Delay: - A packet must wait an endless period of time in a router's queue before it can be processed; this time is dependent on the router's traffic load. The precise information about the level of network congestion is provided by Queue-Delay. Proc-Delay: - A packet experiences a continuous delay at both the sender's and the recipient's end.

$$PATH\_Delay(i) = PRO\_Delay(i) + TRANS\_Delay(i) + QUEUE\_Delay(i) + PROC\_Delay(i) \quad (3.1)$$

This delay may be caused by the time it takes for analog data to be converted to digital form at the sender's end and then packetized through multiple layers of operating protocols before the data are sent to the Physical Layer for transmission.

Similar to this, the behavior of the underlying hardware, the Operating System (OS), and the kind of application data being considered may have an impact on this delay at the receiver level. However, we use path delay in our proposed approach to explicitly evaluate the delay fluctuations. Therefore, it is not necessary to evaluate the aforementioned delay kinds on the proposed policy in detail.

3.2. Validity of Proposed Work. This section explains the validation of the proposed policy to mitigate losses and increase bandwidth utilization. Our proposed policy anticipates congestion in the network and timely assessment of dealing with that occurrence of congestion. The assessment is done by checking the variations in RTT of each path; if some sudden changes are there in RTT, it means congestion may occur. So this is the Congestion Avoidance phase. Our approach suggests an improvement in the CA phase only so that the utilization of the bandwidth is done and overall performance of the network is being improved. Therefore it is easily implementable for other researchers also. The Proposed algorithm is also there to help understand the proposed approach and simulated results.

3.3. Generalizability and scaling Multipath TCP (MPTCP). MPTCP have broad applicability across various network configurations, traffic patterns, and application scenarios. Here's an exploration of the generalizability of these findings: By addressing scalability challenges and optimizing MPTCP deployments for diverse use cases, organizations can leverage the benefits of multipath communication to enhance performance, reliability, and flexibility in their networks and applications.

1. *Network Configurations:* MPTCP's scalability solutions apply to different network configurations, including local area networks (LANs), wide area networks (WANs), data center networks, and wireless networks. Regardless of the specific topology or technology used in the network, MPTCP's ability to leverage multiple paths simultaneously can improve performance and reliability by utilizing available resources efficiently and mitigating congestion.
2. *Traffic Patterns:* MPTCP's scalability considerations are relevant for diverse traffic patterns, including bursty traffic, streaming media, interactive applications, and large-scale data transfers. By dynamically adapting to changes in traffic conditions and network capacity, MPTCP can optimize data transmission to meet the requirements of different applications and traffic patterns, ensuring reliable communication and efficient resource utilization.
3. *Application Scenarios:* MPTCP's scalability solutions are applicable to various application scenarios, such as cloud computing, content delivery, real-time communication, and mobile networking. Whether it's distributing workload across multiple servers in a cloud environment, delivering content to geographically distributed users, maintaining low-latency communication in real-time applications, or supporting seamless mobility for mobile devices, MPTCP's multipath capabilities can enhance performance, reliability, and flexibility across a wide range of use cases.
4. *Heterogeneous Environments:* MPTCP's scalability findings are relevant for heterogeneous environments comprising different network technologies, devices, and operating systems. Whether it's integrating MPTCP support into legacy systems, ensuring interoperability between diverse network components, or accommodating heterogeneous network conditions, the proposed solutions can help overcome compatibility challenges and promote the adoption of MPTCP across heterogeneous environments.
5. *Future Trends:* As networking technologies continue to evolve, including the proliferation of Internet of Things (IoT) devices, the deployment of 5G networks, and the emergence of edge computing, MPTCP's scalability solutions will remain relevant for addressing the scalability, performance, and reliability requirements of future network architectures and applications. By adapting to emerging trends and technologies, MPTCP can continue to provide value in an ever-changing networking landscape.

3.4. Proposed Policy. Each TCP sub-flow in MP-TCP maintains an exclusive congestion window. Additionally, if three duplicate ACKs (dupACKs) are received, the congestion window is reduced automatically by half. On the other hand for MPTCP, coupled congestion control algorithm (CCCA) is used to regulate the augment of the congestion window of every sub-flow at the MP-TCP flow level. CCCA also does resource utilization at each available TCP subflows and try to deliver fairness among them. The congestion window growth of each path is different and availability of each path leads to delay variation in congestion windows which estimates false congestion sometimes. This difference in RTT of available path is one of the reasons why the performance of MPTCP degrades. The goal of this approach is to combine the higher BW of several accessible links while avoiding or avoiding the more aggressive MP-TCP protocol than regular TCP flows on each link that is used. This is done to prevent MP-TCP from using excessive amounts of resources and ensure TCP friendliness. Moreover, packet loss caused by duplicate ACKs is the only factor in this technique that lowers the congestion window. Consequently, there may be a significant variation in the congestion window of each path. As a result, there maybe a significant path delay difference or variation, which is detrimental to the throughput performance at the application level.

In our approach, the capacity and delay of each available path is being monitored continuously which gives the actual state of each path so that the transmission of data is done accordingly. In other words, we can get the picture of stable and unstable paths for further transmission of data. Growth of each congestion window is done and explained with the help of equation 3.2 -3.3. Hence in our approach, the paths have been identified with actual window capacity and despite following traditional RR policy; data transmission is done according to delay and capacity of each path. Hence data is transmitted fast and efficiently. More precisely, the Slow-Start (SS) of MP-TCP is designed to detect the bandwidth availability of the path in order to prevent network congestion. The window growth mechanism moves from the SS phase to the Congestion Avoidance (CA) phase as soon as the congestion window growth reaches a certain extent (SSThresh). The sender then transitions to the CA phase by modifying its SSThresh as soon as feasible, as detected by the timeout mechanism, should a packet loss occur. When in slow start, the window increases after receiving an acknowledgement each time as follows:

$$\min((\alpha/cwnd\_total\_paths), 1/cwnd\_current\_path) \tag{3.2}$$

where α can be calculated as:

$$\alpha = \frac{cwnd\_total\_paths(\max(cwnd\_current\_path/rtt\_current\_path^2)}{\sum(cwnd\_current\_path/rtt\_current\_path)} \tag{3.3}$$

Eq. 3.2 and Eq. 3.3 can be simplified when substituting value of alpha from Eq. 3.2.1 to Eq. 3.2 and is given as follows:

$$\alpha = cwnd\_total\_paths \cdot (\max(cwnd\_i/(cwnd\_current\_path)^2) \tag{3.4}$$

The evaluation of the suggested policy's cwnd growth policy has been conducted in accordance with the existing delay variations, which equal a path. The optimal congestion window growth behavior has been found, fully using the likely channel use. It guides performance improvement in relation to bandwidth and total number of network timeouts. Let the pathways be Pi = (P1, P2, P3 … Pn). Di = (D1, D2, D3 … Dn) indicates the RTT latency of every path − its recurrence. The first measured RTT must be used to initialize the minimum RTT, or ”RTTMin.”

3.5. Algorithm. See Algorithm 1.

4. Assessment of performance. This section conducts extensive experiments in Network Simulator-2 (ns-2) to validate our proposed technique and evaluates its effectiveness against current multi-path techniques.

4.1. Experimental design and simulation setup. The simulation has been run entirely on NS-2.35.

About real world testing: for implementing MPTCP we need more than two systems having configuration that supports our protocol. Out of which windows still don't support MPTCP protocol so Linux and IOS

Algorithm 1 An algorithm for scheduling of Data

Requirement: RTT_curr(i), RTT_min_delay(i), αThresh, ssthresh(Segment)
Begin: Calculate initial ssthresh in terms of Bytes
1. ssthresh$_i$(Bytes) $\leftarrow$ ssthresh(Segment) * Maximum Segment Size (MSS)
2. For every SACK received (at the sender side for each destination):
3. Calculate delay
4. IF cwnd$_i$ <ssthresh$_i$
5. cwnd$_{i+1}$ $\leftarrow$cwnd$_i$+1 // (SSphase)
6. ELSEIF $\mu_i > \alpha$Thresh
7. cwnd$_{i+1}$ $\leftarrow$ cwnd$_i$
8. ELSE
9. path_order$_i$ $\leftarrow$cwnd$_i$
10. RTT_curr(i) $\leftarrow$RTT_min(i) // (Congestion Avoidance Phase)

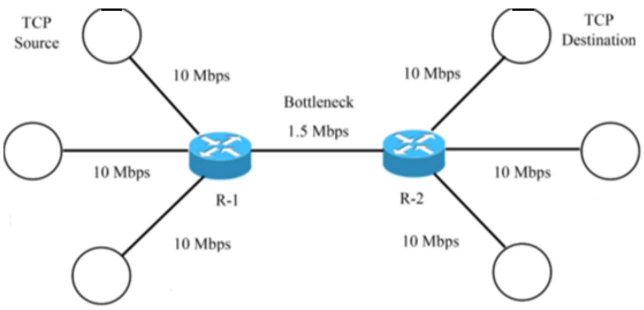

Fig. 4.1: Simulation topology

should be configured in the system. The simulator we used to perform test cases is ns-2.34 which is widely accepted and known for running almost every networking protocol.

The network topology employed in the simulation is displayed in Figure 4.1. One source is used in the topology with one destination with two network interfaces. Figure 4.1 displays the bandwidth and delay of each link with background traffic source and destination. One TCP source is used as a background traffic generator to check the performance of the proposed approach while traffic is there. Each of the nodes are connected to Router 1 with 10 Mbps bandwidth and with Router 2 also same 10Mbps bandwidth while both of the routers R1 and R2 are connected and forming a bottleneck situation with 1.5 Mbps bandwidth. The sender side has background traffic attached to it as TCP Source and UDP Source creating a bottleneck at the middle of the topology with routers R-1 and R-2.

4.2. Simulation and Results. Results of the work are being compared and presented graphically in this section. We have implemented different network configurations, and traffic patterns. The clear comparison of results is done on the topology being mentioned in Figure 4.1. As far as traffic patterns we have used TCP for background traffic we have attached another figure showing throughput in case of UDP as background traffic now. Only TCP and UDP can be used as background traffic for MPTCP.

The change in throughput (Kbps) during the course of the simulation is displayed in Figure 4.2 and 4.3. When a receiver buffer is employed, the MP-TCP and suggested policy results show the traffic across PATH-1 and PATH-2, respectively. Because both MP-TCP and the proposed policy promptly investigate the available channel capacity, their throughput initially climbs significantly. The suggested method transfers more data on a path with lower latency and higher bandwidth by using the path capacity and latency as a consideration in data chunk scheduling.

Additionally, the Slow Start algorithm repeatedly doubles the size of the cwnd. In particular, packet loss and recovery cause a rapid change in throughput by its half. As a result, both policies recover from

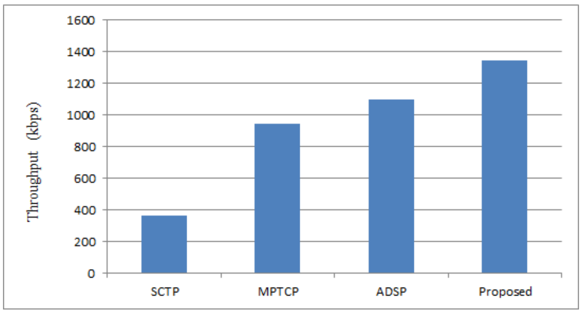

Fig. 4.2: Throughput comparison of SCTP, MPTCP and Proposed Approach

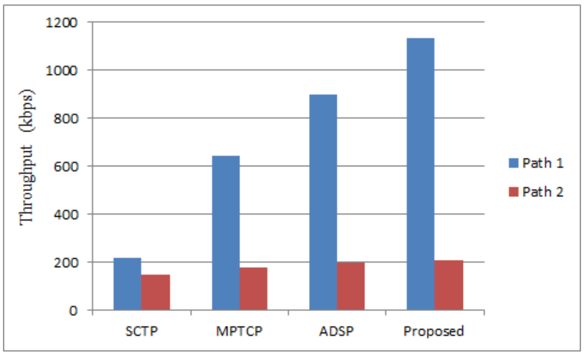

Fig. 4.3: Throughput comparison of SCTP, MPTCP and Proposed Approach

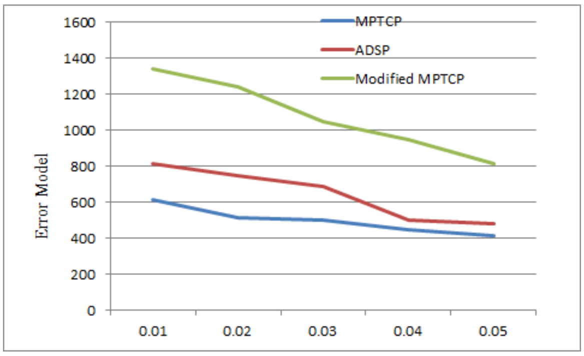

Fig. 4.4: Packet Loss Ratio of MPTCP and our Proposed Approach

losses following appropriate cwnd modifications and quick retransmissions. As a result, the proposed one uses adaptive rapid retransmission policy, RTT variations, and strong packet loss and unordered delivery mitigation to provide higher throughput performance.

The fluctuations in average throughput (Kbps) during the simulation period are displayed in Figure 4.4. The purpose of this experiment is to confirm that every simulated strategy for effectively managing packet losses, which significantly affect average throughput. The average throughput for all the simulated techniques continuously drops as the PLR grows, as illustrated in Figure 4.4. The retransmission timeout is then examined when path-1 packet loss rate is 1 percent and path-2 packet loss rate ranges from 1 percent to 10 percent in

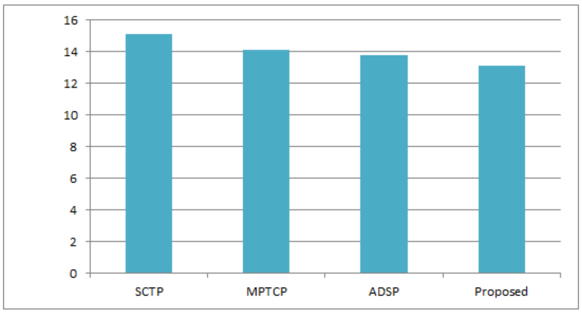

Fig. 4.5: Retransmission time in MPTCP and our Proposed Approach

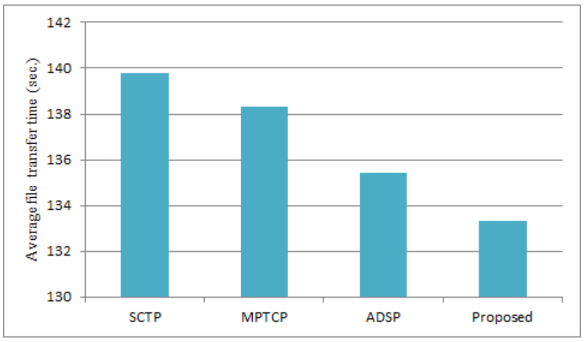

Fig. 4.6: Average file transfer time in SCTP, MPTCP and Proposed Approach

Figure 4.5. The throughput is directly impacted by the retransmission timeout. Out of all the methods utilized for comparison, the suggested approach has the lowest timeout. Consequently, the suggested approach has a greater average usage than SCTP and MPTCP as shown in Figure 4.5. Different multipath protocols have different file transfer times are displayed in Fig. 4.6, with file sizes ranging from 10MB to 90MB. It indicates that the file transfer time grows together with the file size. The average file transfer time improvement using the suggested method is 4.6 percent when compared to MPTCP and SCTP. Therefore, compared to the previously discussed methods, the suggested solution performs better overall in a symmetric packet loss situation.

The suggested technique transmits data over several paths while optimizing network use by using bandwidth and delay-aware scheduling. In contrast to SCTP and MPTCP, the suggested technique transmits files of any size faster. In Figure 4.7, it is illustrated that file transfer times are different for files ranging in size from 10MB to 90MB therefore it demonstrates that file transfer times rise in tandem with file sizes. The data pieces are being distributed equally across several pathways by both strategies. The suggested approach factors in route latency and bandwidth while arranging data chunks. Thus, a lot of data is scheduled on the high bandwidth and minimum delay line by the suggested strategy.

5. Conclusion. This study introduced an approach that is bendable and adaptive multi-path data transmission scheme, in response to the constraints posed by various path characteristics in concurrent multi-path transmission design. MP-TCP transfer scheme. Rather than arbitrarily allocating traffic across several viable network channels without taking into account the differences in path characteristics, the proposed one considers each path's quality and adjusts the scheduling criteria accordingly. The proposed policy handles the buffer-blocking issue at the receiver side and the sudden congestion window growth adjustments issue at the transmitter side with its RTT variation-based adaptive scheduling and rapid retransmission technique. Using the idea of evaluating RTT variations to determine the traffic state of a path, the RTT variation-based adaptive scheduling technique operates. Additionally, the adaptive rapid retransmission technique based on RTT variations controls the volume of traffic (transmission rate) in relation to RTT variations. The simulation results

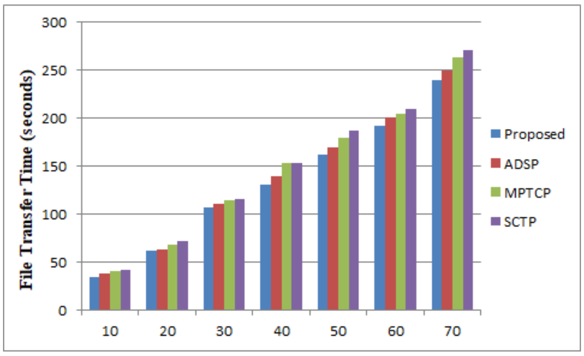

Fig. 4.7: File transfer time in SCTP, MPTCP and Proposed Approach

show how effective our proposed approach is in terms performance and average throughput. The proposed approach gives 56 percent, 19 percent, and 36 percent greater throughput than MP-TCP, respectively and the proposed approach provides 50 percent better FTT.

REFERENCES

[1] J. Postel, Transmission Control Protocol, Darpa Internet Program, September 1981. RFC 793. [Online]. Retrieved: https://datatracker.ietf.org/doc/html/rfc793.
[2] J. Postel, User Datagram Protocol, RFC768, The Internet Engineering Task Force, California, USA, 2007. RFC 768. [Online]. Retrieved: https://datatracker.ietf.org/doc /html/rfc768.
[3] Habib, S., Qadir, J., Ali, A., Habib, D., Li, M., and Sathiaseelan, A. (2016). The past, present, and future of transport-layer multipath. Journal of Network and Computer Applications 75, 236–258.
[4] Khalili, R., Gast, N., Popovic, M., and Le Boudec, J.Y. (2013). MPTCP is not pareto-optimal: Performance issues and a possible solution. IEEE/ACM Transactions on Networking 21, 1651–1665.
[5] Raiciu, C., Paasch, C., Barre, S., Ford, A., Honda, M., Duchene, F., Bonaventure, O., and Handley, M. (2012). How hard can it be? Designing and implementing a deployable multipath TCP. In Proceedings of NSDI 2012: 9th USENIX Symposium on Networked Systems Design and Implementation, (USENIX Association), pp. 399–412.
[6] Honda, M., Nishida, Y., Eggert, L., Sarolahti, P., and Tokuda, H. (2009). Multipath Congestion Control for Shared Bottleneck. International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT).
[7] Paasch, C., and Enture, O.B. (2014). Multipath TCP. Communications of the ACM 57, 51–57.
[8] Verma, L.P., Sharma, V.K., and Kumar, M. (2018). New delay-based fast retransmission policy for CMT-SCTP. International Journal of Intelligent Systems and Applications 10, 59–66.
[9] Shailendra, S., Bhattacharjee, R., and Bose, S.K. (2011). MPSCTP: A simple and efficient multipath algorithm for SCTP. IEEE Communications Letters 15, 1139–1141.
[10] Xu, C., Zhao, J., and Muntean, G.M. (2016). Congestion Control Design for Multipath Transport Protocols: A Survey. IEEE Communications Surveys and Tutorials 18, 2948–2969.
[11] Ferlin, S., Dreibholz, T., and Alay, Ö. (2014). Multi-path transport over heterogeneous wireless networks: Does it really pay off? In 2014 IEEE Global Communications Conference, GLOBECOM 2014, (Institute of Electrical and Electronics Engineers Inc.), pp. 4807–4813.
[12] Ha, S., Le, L., Rhee, I., and Xu, L. (2007). Impact of background traffic on performance of high-speed TCP variant protocols. Computer Networks 51, 1748–1762.
[13] Noura, H.N., Melki, R., and Chehab, A. (2022). Network coding and MPTCP: Enhancing security and performance in an SDN environment. Journal of Information Security and Applications 66.
[14] Ha, T., Masood, A., Na, W., and Cho, S. (2023). Intelligent Multi-Path TCP Congestion Control for video streaming in Internet of Deep Space Things communication. ICT Express 9, 860–868.
[15] Li, M., Lukyanenko, A., Tarkoma, S., Cui, Y., and Ylä-Jääski, A. (2013). Tolerating path heterogeneity in multipath TCP with bounded receive buffers. In Performance Evaluation Review, pp. 375–376.
[16] Lin, J., Cui, L., Zhang, Y., Tso, F.P., and Guan, Q. (2019). Extensive evaluation on the performance and behaviour of TCP congestion control protocols under varied network scenarios. Computer Networks 163.
[17] Wang, Z., Zeng, X., Liu, X., Xu, M., Wen, Y., and Chen, L. (2016). TCP congestion control algorithm for heterogeneous Internet. Journal of Network and Computer Applications 68, 56–64.
[18] Bonaventure, O., Handley, M., and Raiciu, C. (2012). An Overview of Multipath TCP. USENIX Login 17–23.
[19] Lubna, T., Mahmud, I., and Cho, Y.Z. (2020). D-LIA: Dynamic congestion control algorithm for MPTCP. ICT Express 6,

263–268.
[20] Verma, L.P., Sharma, V.K., Kumar, M., and Mahanti, A. (2022). An adaptive multi-path data transfer approach for MP-TCP. Wireless Networks 28, 2185–2212.
[21] Ferlin, S., Alay, O., Mehani, O., and Boreli, R. (2016). BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks. In 2016 IFIP Networking Conference (IFIP Networking) and Workshops, IFIP Networking 2016, (Institute of Electrical and Electronics Engineers Inc.), pp. 431–439.
[22] Dong, P., Yang, W., Tang, W., Huang, J., Wang, H., Pan, Y., and Wang, J. (2018). Reducing transport latency for short flows with multipath TCP. Journal of Network and Computer Applications 108, 20–36.
[23] Chen, D., Gao, D., Jin, L., Quan, W., and Zhang, H. (2023). ADAS: Adaptive Delay-Aligned Scheduling for Multipath Transmission in Heterogeneous Wireless Networks. Peer-to-Peer Networking and Applications 16, 1583–1595.
[24] Lim, Y.S., Towsley, D., Nahum, E.M., and Gibbens, R.J. (2017). ECF: An MPTCP path scheduler to manage heterogeneous paths. In CoNEXT 2017 - Proceedings of the 2017 13th International Conference on Emerging Networking EXperiments and Technologies, (Association for Computing Machinery, Inc), pp. 147–159.
[25] Peng, Q., Walid, A., Hwang, J., and Low, S.H. (2016). Multipath TCP: Analysis, Design, and Implementation. IEEE/ACM Transactions on Networking 24, 596–609.
[26] Kato, T., Diwakar, A., Yamamoto, R., Ohzahata, S., and Suzuki, N. (2019). Experimental analysis of MPTCp congestion control algorithms; Lia, olia and Balia. In Multi Conference on Computer Science and Information Systems, MCCSIS 2019 - Proceedings of the International Conferences on Big Data Analytics, Data Mining and Computational Intelligence 2019 and Theory and Practice in Modern Computing 2019, (IADIS Press), pp. 135–142.
[27] Lubna, T., Mahmud, I., Kim, G.H., and Cho, Y.Z. (2021). D-olia: A hybrid mptcp congestion control algorithm with network delay estimation. Sensors 21.
[28] Li, M., Lukyanenko, A., Tarkoma, S., Cui, Y., and Ylä-Jääski, A. (2013). Tolerating path heterogeneity in multipath TCP with bounded receive buffers. In Performance Evaluation Review, pp. 375–376.
[29] Vo, P.L., Le, T.A., Lee, S., Hong, C.S., Kim, B., and Song, H. (2014). Multi-path utility maximization and multi-path TCP design. Journal of Parallel and Distributed Computing 74, 1848–1857.
[30] Ferlin, S., Dreibholz, T., and Alay, Ö. (2014). Multi-path transport over heterogeneous wireless networks: Does it really pay off? In 2014 IEEE Global Communications Conference, GLOBECOM 2014, (Institute of Electrical and Electronics Engineers Inc.), pp. 4807–4813.
[31] Zhou, D., Song, W., and Cheng, Y. (2013). A study of fair bandwidth sharing with AIMD-based multipath congestion control. IEEE Wireless Communications Letters 2, 299–302.
[32] Carofiglio, G., Gallo, M., and Muscariello, L. (2016). Optimal multipath congestion control and request forwarding in information-centric networks: Protocol design and experimentation. Computer Networks 110, 104–117.
[33] Tao, Y., and Huang, P. (2014). MPTCP congestion control algorithm based on the fairness of bottleneck. In Applied Mechanics and Materials, (Trans Tech Publications), pp. 3995–4000.
[34] Mahmud, I., Lubna, T., Song, Y.J., and Cho, Y.Z. (2020). Coupled multipath BBR (c-MPBBR): An efficient congestion control algorithm for multipath TCP. IEEE Access 8, 165497–165511.
[35] Yang, W., Dong, P., Tang, W., Lou, X., Zhou, H., Gao, K., and Wang, H. (2018). A MPTCP scheduler for web transfer. Computers, Materials and Continua 57, 205–222.
[36] Wei, W., Xue, K., Han, J., Wei, D.S.L., and Hong, P. (2020). Shared Bottleneck-Based Congestion Control and Packet Scheduling for Multipath TCP. IEEE/ACM Transactions on Networking 28, 653–666.
[37] Thomas, Y., Xylomenos, G., and Polyzos, G.C. (2020). Multipath congestion control with network assistance. Computer Communications 153, 264–278.
[38] Ha, T., Masood, A., Na, W., and Cho, S. (2023). Intelligent Multi-Path TCP Congestion Control for video streaming in Internet of Deep Space Things communication. ICT Express 9, 860–868.
[39] Choi, K.W., Cho, Y.S., Aneta, Lee, J.W., Cho, S.M., and Choi, J. (2017). Optimal load balancing scheduler for MPTCP-based bandwidth aggregation in heterogeneous wireless environments. Computer Communications 112, 116–130.
[40] Popat, K., Kapadia, V.V. (2021). Multipath TCP Security Issues, Challenges and Solutions. In: Bhattacharya, M., Kharb, L., Chahal, D. (eds) Information, Communication and Computing Technology. ICICCT 2021. Communications in Computer and Information Science, vol 1417. Springer, Cham. https://doi.org/10.1007/978-3-030-88378-22.
[41] Zhao Baosen, Yang Wanghong, Du Wenji, Ren Yongmao, Sun Jianan, Wu Qinghua, Zhou Xu. (2024). A multipath scheduler based on cross-layer information for low-delay applications in 5G edge networks. Computer Networks vol 224. Elsevier. https://https://doi.org/10.1016/j.comnet.2024.110333.
[42] Amend, M., and Rakocevic, V. (2024). Cost-efficient multipath scheduling of video-on-demand traffic for the 5G ATSSS splitting function. Computer Networks 242.

Edited by: Manish Gupta
Special issue on: Recent Advancements in Machine Intelligence and Smart Systems
Received: Mar 13, 2024
Accepted: Jun 7, 2024