IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 June 2025, accepted 26 June 2025, date of publication 10 July 2025, date of current version 17 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3586446

RESEARCH ARTICLE

Providing QoS and Resiliency for Video Streaming Over MPTCP

SHADI BIKAS[1] AND MÜGE SAYIT[2,3], (Senior Member, IEEE)
[1]Computer Engineering, TOBB ETÜ University, 06510 Ankara, Türkiye
[2]Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K.
[3]International Computer Institute, Ege University, 35040 İzmir, Türkiye

Corresponding author: Müge Sayit (muge.sayit@essex.ac.uk)

This work was supported by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) Electric, Electronic and Informatics Research Group (EEEAG) under Grant 121E033.

ABSTRACT The performance of Internet applications can be improved by using multipath transport protocols such as Multipath TCP (MPTCP), which leverages aggregated bandwidth from multiple paths for improved throughput. MPTCP can be utilized to apply Quality of Service (QoS) and address network resiliency in order to increase the Quality of Experience (QoE) of video streaming applications. This paper introduces a novel approach for providing service differentiation to video streaming clients across different classes, while ensuring robust communication to mitigate link failures. The proposed method aims to deliver seamless streaming sessions by enhancing resilience and maintaining QoS under challenging network conditions. We evaluate the proposed approach using Dynamic Adaptive Streaming over HTTP (DASH), a widely used video streaming application that delivers high-quality media content over the Internet. The aim is to evaluate the performance of the proposed approach by focusing on user groups belonging to different service classes, each with varying QoE requirements. Additionally, we assess the resiliency of the proposed algorithm in scenarios involving subflow failures within the MPTCP connection. The results demonstrate that the proposed approach outperforms alternative methods in maintaining network connectivity and sustaining a high level of QoE, even in the event of subflow disconnections.

INDEX TERMS Adaptive streaming, genetic algorithm, HAS, DASH, SDN, quality of service, routing.

I. INTRODUCTION

As one of the most widely used internet applications, video streaming requirements evolve alongside recent advancements in network technologies and protocols. The increasing quality of videos, such as 8K streaming and immersive technologies, results in higher bitrates, thereby necessitating network conditions capable of providing high bandwidth. Moreover, seamless streaming requires stable network conditions.

Multipath TCP (MPTCP), which has been introduced as an extension to the traditional TCP protocol, can be seen as an alternative to meet these requirements as MPTCP enhances throughput and reliability by enabling the use of multiple paths [1]. This increased throughput can enhance the Quality of Experience (QoE) for video streaming applications, thereby meeting stringent application requirements. For video streaming over MPTCP, the characteristics of the paths used for packet transmission between the server and the clients significantly influence the overall performance. Features of MPTCP subflows, such as the bandwidth, congestion rate, delay difference of the subflows, and more, can introduce challenges—such as the Head-of-Line (HoL) blocking issue. HoL blocking occurs when data packets arriving out of order on different subflows delay delivery to the application until all preceding data is received and reassembled, reducing performance. This issue becomes critical in video streaming scenarios, where consistent playback quality is essential [2]. HoL blocking that is caused by delay differences between subflows can lead to outages lasting more than 20% of the total video session duration, which significantly affects the overall QoE [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

Numerous studies have addressed the challenges mentioned above by proposing various path selection strategies for MPTCP subflows. Software-Defined Networking (SDN) has emerged as a promising technology for enabling third-party path selection at the network layer. By providing centralized control and a global view of the network, SDN facilitates dynamic traffic routing based on real-time network conditions, enabling intelligent path selection and service differentiation tailored to diverse requirements.

In this study, we focus on providing service differentiation for video streaming users, where video packets are transmitted using MPTCP. Maintaining the required service can be challenging due to the unstable nature of the network connections. The streaming paths may experience disruptions due to various factors, such as link disconnections, hardware malfunctions, physical damage, or routing anomalies. Therefore, choosing paths that provide reliable and seamless transmission is essential. The primary objective of this study is to ensure reliable service delivery to different client classes under challenging network conditions. Specifically, we aim to enhance the performance of video streaming applications while ensuring QoS and resiliency by integrating advanced technologies and protocols.

The proposed approach considers scenarios where one of the most widely used video streaming technologies, Dynamic Adaptive Streaming over HTTP (DASH), operates over an SDN based edge network. One potential use case involves cloud-hosted video streaming services that leverage MPTCP over SDN to optimize the delivery of video content to end-users. In multi-tenant data centers, MPTCP over SDN can be employed to improve the video streaming experience for clients accessing various services. Another relevant use case involves video streaming clients operating in wireless network environments. These clients may utilize multiple network interfaces, including WiFi, 5G, 6G, or even satellite connections. The proposed approach is well-suited for all these use cases, providing flexibility and robustness across diverse network scenarios. Unlike prior works that treat QoS and resiliency separately, this paper uses a Genetic Algorithm (GA) to jointly optimize path selection across multiple service classes and failure scenarios.

The key contributions of this study can be listed as follows:

- A GA based path selection approach for MPTCP subflows, designed to deliver guaranteed bitrates and differentiated services—based on QoE parameters—to DASH video clients with varying service class requirements;
- A resiliency-aware approach to ensure high stability and maintain QoS levels even in the presence of link failures;
- The evaluation of the performance of the GA based service differentiation in terms of guaranteed service and resiliency across different network topologies.

The rest of the paper is organized as follows: Section II gives background on GA and multipath transmission, and encloses studies on the QoE enhancement with path selection, QoS and resiliency approaches where multipath transmission is utilized. Section III introduces the strategy to guarantee QoS and offer resiliency by presenting the details of the proposed architecture and GA. Section IV presents the comparative performance evaluation. The paper concludes in section V, followed by the references.

II. BACKGROUND AND RELATED WORK

A. GENETIC ALGORITHM AND MULTIPATH TRANSMISSION

GA is a population based optimization technique that employs selection, crossover, and mutation operators [4]. This algorithm is widely used to solve complex problems by mimicking the principles of natural selection and genetics to identify high-quality or near-optimal solutions. This approach, which would be time-consuming using traditional methods, begins with generating a random population of potential solutions, each known as a chromosome, with each bit within a chromosome referred to as a gene. To evaluate the accuracy of each solution, a fitness function is determined, representing how well each solution addresses the problem. Following this, parent selection occurs, where solutions in the current population, known as parents, are chosen to produce the next generation. These selected parents are combined through a crossover process, creating new solutions known as children. Parent selection is crucial as it enables the generation of progressively better populations. Multiple parents are chosen during crossover, and various methods are applied to produce one or more children. The final stage of the GA is called mutation, where a set of children is randomly selected, and one or more genes are altered based on a certain probability. The stages of parent selection, crossover, and mutation are repeated for a certain number of cycles, leading to the development of the population over time. Eventually, the chromosome with the highest fitness value is selected as the solution [5].

Due to their efficiency in optimizing search performance within a limited time-frame, GAs have been employed in multipath transmission. For instance, the authors in [6] utilized a GA to minimize delay differences among MPTCP subflows in heterogeneous wireless networks. In [7], the GA minimizes path sizes and maximizes line usage diversity to ensure consistent traffic flow in a two-phase multipathing scheme. The authors in [8] utilize a GA to generate disjoint paths with minimal overlap to maximize network resource utilization, aiming to optimize the multipathing process in data center networking. A GA is developed to provide a near-optimal solution for distributing traffic rates across multiple subflows in MPTCP, balancing energy consumption and throughput [9]. In [10], a GA is employed for packet scheduling across multiple Virtual Communication channels (VCs). In [11], a GA for multipath routing optimization in Mobile Ad Hoc Networks (MANETs) is implemented to optimize routes based on metrics like shortest distance, residual energy, and congestion. The authors developed a GA to select fully disjoint and reliable paths with lower latency, aiming to optimize multipath transmission in heterogeneous networks for smart city networks [12]. In [13], the authors proposed a multipath routing strategy for Elastic Optical Networks (EONs) using a GA combined with distance-adaptive modulation formats to reduce blocking probability under dynamic traffic conditions. The proposed algorithm combines a two-stage multi-population GA with adaptive routing to improve real-time scheduling in Industrial Internet of Things (IIoT) networks. It supports multipath transmission for critical flows and optimizes routing and scheduling based on flow criticality, resource utilization, and transmission success rate [14]. Authors in [15] propose a GA based multipath routing mechanism for emergency video delivery in vehicular fog computing environments. None of these works use the GA to provide service differentiation, improve QoE, or ensure resiliency, which are the main focus of our current work.

In our previous study, we proposed a GA based path selection approach for improving the QoE of video streaming applications [3]. In the study, the GA aimed at enhancing the video streaming performance using the MPTCP transmission protocol by selecting the optimal paths. Network conditions such as bandwidth, delay difference of the subflows and disjointness of the links were the criteria considered in the GA development. In this paper, we enhanced our previous work focusing on service differentiation for client classes and developing a resiliency-aware path selection scheme.

B. QUALITY OF EXPERIENCE ENHANCEMENT WITH PATH SELECTION

SDN is a new paradigm to networking that provides centralized control, easy programming, and flexible network management [16]. By decoupling the control plane from the data plane, SDN enables dynamic routing configurations, which help optimize traffic flow and significantly improve QoE for end users by reducing delays, packet loss, and service interruptions. Indeed, routing within an SDN domain plays a critical role in achieving these goals, as it directly influences performance, reliability, and resource utilization. In this context, a variety of routing strategies and optimization techniques have been explored widely in the literature. While earlier work on SDN path selection primarily focused on selecting routes based on metrics such as bandwidth [17], or QoS metrics such as packet loss rate and delay [18], more recent studies have explored orchestration frameworks aimed at managing multimedia services and enhancing QoE [19]. In [20], the authors proposed a machine learning based approach that collects Key Performance Indicators (KPIs) to determine SDN routes for various application scenarios, such as video streaming and VoIP. Another study [21] proposed determining SDN routes by considering both QoE-related parameters gathered from clients and underlying link characteristics.

Recently, video streaming over multipath and dynamic path selection in SDN has attracted increasing research interest. In [22], the authors proposed a method to identify an alternative path within the SDN domain, which can be used when the primary path becomes critical or unavailable, in order to sustain QoE. Other studies have investigated the simultaneous use of multiple paths to improve QoE. In [23], the multipath selection problem is formulated as a Multi-Commodity Flow Problem (MCFP), with the goal of ensuring QoE fairness among clients. The authors in [24] proposed using Multipath Generic Routing Encapsulation (MPT-GRE) — a GRE-over-UDP based tunneling mechanism — to transmit high-resolution video streams simultaneously over multiple network paths, aiming to improve video quality and resilience compared to traditional single-path streaming. MPTCP subflow paths are selected by considering bandwidth and delay differences in our previous work to improve QoE [3]. Several frameworks have been proposed that enable the selection and utilization of multiple paths using SDN technology, based on available bandwidth, to enhance QoE in applications such as video conferencing [25] and remote surgery [26].

While selecting the multiple paths with SDN technology for improving QoE is studied in these recent works, providing service differentiation and sustainable QoE and seamless video streaming in a faulty network environment is still an open question and not studied well in multipath video streaming scenarios. Our work differs from literature from this perspective.

C. QUALITY OF SERVICE AND RESILIENCY IN MULTIPATH TRANSMISSION

While providing QoS is one of the areas that has been studied extensively, there are a limited number of studies that address QoS with multipath transmission. In [27], the authors proposed a DRL based path selection and traffic allocation for providing QoS by selecting multiple paths in an SDN domain. In [28], the authors utilize both path selection and service differentiation. After eliminating paths with a higher delay than a certain threshold, paths with higher bandwidth are assigned to clients. The number of MPTCP subflows is dynamically adjusted to perform service differentiation based on the network condition and the guaranteed service. In one of our previous work [29], we propose a novel architecture that improves the QoS and QoE for video streaming applications compared to traditional throughput maximization approaches. The OpenFlow priority mechanism is utilized in [30] to assign priorities and optimize resource allocation, enabling service differentiation in SDN environments. Similarly, the authors in [31] focus on service differentiation by employing multipath transmission in SDN, where services are provided on a per-user basis.

Only a few studies in the literature focus on the MPTCP path selection problem considering resiliency. Furthermore, there is no study focus on providing consistent QoE with resilient and service differentiated MPTCP path selection. The authors in [32] focus on evaluating the resilience of MPTCP in environments with prevalent link failures. The study delivers a comprehensive view of how MPTCP can enhance the resilience of network communications in the face of link failures. In [33], the authors emphasized the effect of MPTCP on enhancing the resilience and efficiency of internet connections for IoT devices. In [34], authors developed a scalable SD-WAN solution known as WAN-aware MPTCP (WaMPTCP) by integrating LAN virtualization at end systems with WAN virtualization at SD-WAN gateways. While improving resilience is a critical component of WaMPTCP, the overall design and implementation also focus on enhancing bandwidth utilization, traffic balance, and network performance across heterogeneous WAN links. Another study [35] focused on enhancing the resilience and reliability of the OpenFlow channel in SDN by utilizing MPTCP. They propose using both in-band and out-of-band paths to ensure continuous communication between the SDN controller and the network switches, even during link failures. The solution aims to improve network robustness by dynamically rerouting traffic through alternative paths when errors occur. Multiple link failures is considered in [36], where alternative switches are selected before the failure in an hybrid IP/SDN network.

While prior studies have explored QoS and resiliency in multipath transmission, several critical limitations remain. Existing approaches focus primarily on optimizing bandwidth or minimizing delay but often fail to integrate these objectives into a unified framework that considers both service differentiation and resilience against link failures. Furthermore, GAs have been utilized in MPTCP for traffic optimization and congestion control; however, their application has been largely limited to addressing individual performance metrics such as delay or energy efficiency. None of the studies mentioned above addressed the selection of network paths to sustain QoE by considering link failures. Together, these threads of research underscore the need for integrated solutions that can intelligently manage routing, optimize QoS, and maintain resiliency goals that our proposed GA based SDN-MPTCP framework aims to achieve.

Our approach stands out because it combines several important aspects—service differentiation, QoE guarantees, and resiliency against link failures—into a single, unified solution. By using a GA that adapts to changing network conditions and user requirements, we provide a novel method for managing video streaming in real-world networks. This fills an important gap in current research, where these elements are often treated separately.

III. PROPOSED APPROACH

A. MPTCP PATH SELECTION WITH GENETIC ALGORITHM CONSIDERING SERVICE CLASSES

In this work, the server and the clients are connected using MPTCP in an SDN network, with the SDN controller responsible for managing the network and measuring the traffic using the modules implemented on it.

The proposed path selection algorithm is designed as a third party application that communicates with the SDN controller through its northbound API. Based on the application's output, the controller sends the relevant rules of the selected paths to the switching devices. After the transmission paths are determined, data transmission from the server to the client begins. The system is illustrated in Fig. 1. According to the figure, there are three clients belonging to different service classes. The client with the highest service class transfers the video over three different paths to achieve a higher aggregated bandwidth. In the figure, the links between routers are not shown for simplicity.

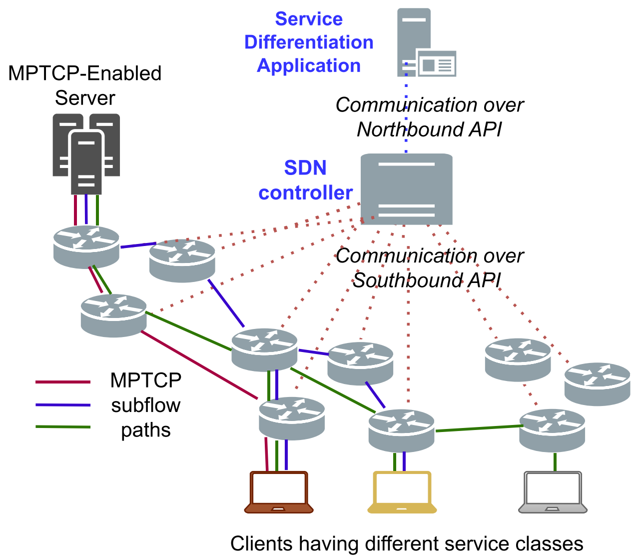


FIGURE 1. Example topology illustrating service differentiation for clients.

Three different service classes were defined to offer varying levels of service guarantees on the network. In these service classes, namely for Gold, Bronze, and Best Effort, a particular bitrate guarantee is defined for Gold-type users and Bronze-type users. The guaranteed bitrate for Gold-type users is higher than for Bronze-type users. When a client joins the system, the SDN controller determines which class the user belongs to. The controller manages this task by depending on the user's IP address, assuming the user agreements will be within the network operator's knowledge. After determining the user's service class, the SDN controller runs the GA application to obtain the path set determined for that user. Furthermore, to ensure that the bandwidth guarantees for new clients do not adversely affect existing flows, we incorporated an additional traffic management module into the SDN controller. This module applies limitations on the network interfaces that belong to the switches of the paths assigned to clients. This approach maintains the guaranteed service quality of all users, without performing kernel-level modification.

B. GENETIC ALGORITHM DETAILS

The general outline of the GA is given in Algorithm 1. The algorithm starts by generating the initial population. Details on the fitness calculation, selection, crossover, and mutation mechanisms are provided in the following subsections.

1) GENERATING INITIAL POPULATION

A GA based approach was proposed to optimize path selection by considering critical parameters such as path bandwidth and subflow delay differences in our previous work [3]. This GA approach allows for efficient and dynamic path selection, enhancing network performance by balancing load and minimizing latency. In this work, we enhanced this algorithm by considering the bitrate guarantees provided to the clients. For this purpose, each step of the algorithm needs to be changed from the beginning, starting from initial population generation.

The SDN controller runs the algorithm when a new client joins the system. The process starts with the SDN controller's path determination module, which lists all network paths and assigns a unique ID to each one. Each path ID represents a gene in this algorithm, while a chromosome is defined as a combination of these genes. In the proposed method, a chromosome is described as a $1 * m$ vector, where m is the maximum number of subflows that can be established. For this study, the maximum number of subflows for any client is limited to three, resulting in chromosomes that are combinations of three genes each.

The initial population of chromosomes is formed that consists of various subflow combinations for the newly joined client. Hence, the members of the new initial population are determined, as shown in the first step of Algorithm 1. However, the bitrate guarantees should be considered during the initial population generation. For this purpose, the *generate_initial_population* function takes the bitrate threshold that is defined for that user class, $minthr_{br}^{h}$, as an argument, as well as n, pre-determined population size. To form the GA's initial population, each chromosome's cumulative bandwidth is evaluated concerning this threshold, ensuring that only chromosomes with sufficient bandwidth proceed to the next stage. This check uses Equation (1). In the formula, $minthr_{br}^{h}$ represents the guaranteed bitrate for clients in class h. Chromosomes that do not meet this condition are eliminated from the population.

$$\frac{abw_p}{|H_p|+1} \geq minthr_{br}^{h} \tag{1}$$

2) CALCULATING FITNESS

The next step after initializing the population is fitness calculation, which is given in the 4^{th} line of the algorithm. Each gene has a specific bandwidth and delay value, which is used as input in the fitness function. The fitness function calculates the fitness of each chromosome by the sum of the bandwidth and delay differences of the subflows as given in function in the 5^{th} line of the algorithm. The network operator can determine the values of α and β according to the current network policy. The $\zeta(.)$ and $\xi(.)$ functions are given in formulas (2) and (3), respectively. The $(2)^{nd}$ equation represents the summation of the total bandwidth of the paths in a chromosome. The $(3)^{st}$ equation calculates the delay differences between the selected paths. This information is used to minimize the effects of HoL. In the formulas, P_c represents the set of all paths assigned to the client c; d_p represents the delay of the path p; H represents the set of clients currently in the system; H_p represents the clients using the path p, and abw_p represents the bandwidth of the path p. Using these formulas, the delay difference and the total bandwidth of the possible path sets for the newly connected client are calculated in the fitness function. The values of these functions are normalized with the min-max normalization to a range between 0 and 1 before calculating the formula. This normalization ensures that the value ranges are distinct and that no single value dominates or suppresses the others. This design of the fitness function both maximizes the bandwidth and minimizes the delay difference. This way, the GA favors chromosomes that have a balance between high total bandwidth and minimum delay differences. This technique ensures that the selected paths not only provide better QoS through increased throughput but also address the HoL problem by reducing the subflow delay differences.

$$\sum_{p=1}^{|P_c|} \frac{abw_p}{\sum_{h\in H_p} +1} \tag{2}$$

$$\sum_{p=1}^{|P_c|-1} \sum_{r=p+1}^{|P_c|} |d_p - d_r| \tag{3}$$

3) PERFORMING SELECTION, CROSSOVER, AND MUTATION

The following steps in the GA are selection, crossover, and mutation. These steps are implemented to create a new generation from the initial population. Tournament selection, which is summarized between 6^{th} and 9^{th} lines of the algorithm, is employed for selecting chromosomes from the population, referred to as parents, presented as M_x. In this process, a subset of chromosomes is randomly chosen from the population, and those with higher fitness are selected as parents for the next generation, defined as $P_{current}$ in the algorithm. A predefined proportion of the next-generation members are selected from the previous generation. This is ensured by repeating the selection of parents as many times as the r ratio of the new generation members, as seen in the 7^{th} line of the algorithm.

Subsequently, crossover operations were performed using a two-point crossover technique which is presented between 10^{th} and 17^{th} lines of the algorithm. This method involves selecting two crossover points on the parent chromosomes (M_1 & M_2). The genes between these two points are swapped between the two parent chromosomes to produce two new offspring (O_1 & O_2).

The final step of the algorithm is mutation, which is detailed in Algorithm 2. In the mutation process, each gene in a chromosome has can change to a different value within the available range with the probability of μ. The mutation probability is set to 25% in this setting which indicates a 25% chance of altering a gene. The mutation process ensures that the mutated chromosome maintains the bitrate guarantee for the client. To ensure this, the algorithm checks if the mutated chromosome's total bandwidth meets the bitrate threshold, as specified in line 3 of Algorithm 2. If this condition is satisfied, the mutation is performed. Otherwise, the algorithm searches for another chromosome which meets bandwidth requirement after mutation.

The chromosomes form a new population after going through selection, crossover, and mutation steps. In this study, these steps are repeated 30 times.

Algorithm 1 Service Differentiation based Path Assignment

Data: M_x: x^{th} parent chromosome
O_x : x^{th} offspring chromosome
P_{new}: the generated new population
$P_{current}$: the set of temporary population
f_i: fitness value of i^{th} chromosome
Input: n: Population size
h: new client
$minthr_{br}^{h}$: Minimum guaranteed bitrate for new client h
r: parent selection ratio for next generation

P_{new} = *generate_initial_population*(n, $minthr_{br}^{h}$);
while *no of generations* $\leq$ *MaxGenerationCount* **do**
  $P_{current} \leftarrow \emptyset$
  // *Fitness Calculation*:
  $f_i = \alpha.\zeta(.) - \beta.\xi(.)$;
  // *Tournament selection*:
  while $|P_{current}| \leq n * r$ **do**
    M_1, M_2 = Randomly select two chromosomes in P_{new}
    Add the chromosome with higher fitness value to $P_{current}$
  // *Two-point crossover*:
  while $|P_{current}| \leq n$ **do**
    foreach $M_1 \in P_{new}$ & $M_2 \in P_{new}$ **do**
      O_1, O_2 = *twoPointCrossover* (M_1, M_2);
      if *total_bandwidth*(O_1) $\geq minthr_{br}^{h}$ **then**
        $P_{current} = P_{current} \cup O_1$;
      if *total_bandwidth*(O_2) $\geq minthr_{br}^{h}$ **then**
        $P_{current} = P_{current} \cup O_2$;
  // *Mutation*:
  foreach *chromosome* $P_i \in P_{current}$ **do**
    mutation(chromosome P_i, $minthr_{br}^{h}$*)*
  $P_{new} \leftarrow P_{current}$
  no of generations++;

Algorithm 2 Mutation Algorithm with Guaranteed Bitrate Check: *mutation*(P_i, $minthr_{br}^{h}$)

Data: μ: Mutation probability
mutated = 0
Input: n: $minthr_{br}^{h}$: Minimum guaranteed bitrate for the new user

1 **while** !*mutated* **do**
2   MP_i = Mutated (P_i) with probability μ
3   **if** *total_bandwidth*(MP_i) $\geq minthr_{br}^{h}$ **then**
4   *mutated* = 1;
5     $P_i = MP_i$

However, ensuring guaranteed bandwidth for new clients alone is not enough to maintain QoS for all clients in the system. The path selections for these new clients should not compromise the bandwidth guarantee for existing clients. In order to ensure that the clients who already have flows will maintain the guaranteed service quality, an additional module for the SDN controller was designed to perform traffic management. Limitations were applied to manage bandwidth on specific network interfaces associated with switches of the paths assigned to the clients. With these changes made in the SDN controller modules, the proposed architecture has been developed to provide services according to the classes to which the users connected to the system are affiliated. The results of the tests performed on this system are presented in the performance evaluation section.

C. PROVIDING QoS AND RESILIENCY JOINTLY WITH MPTCP

Another advantage of MPTCP is that it provides resilience against problems such as link failure and disconnection when multiple subflows are used and if these subflows transmit packets over different paths. In such a case, although communication cannot be made over subflows with errors, communication can continue over other paths. Considering the probability of link failures when selecting paths for MPTCP subflows can ensure that clients are minimally affected by errors that may occur on the network.

$$\Delta_p = [1 - (1 - \delta_1)^{hopcount}] \tag{4}$$

$$\sum_{r=1}^{|P_c|} (1 - \Delta_p).\frac{abw_p}{|H_p| + 1} \tag{5}$$

$$\sum_{p=1}^{|P_c|-1} \sum_{r=p+1}^{|P_c|} |(1 - \Delta_p).d_p - (1 - \Delta_r).d_r| \tag{6}$$

The developed GA has been modified to provide a higher level of resilience with a revision made to the fitness function. The probability of the failure of link l on the network is given as δ_l.

This probability can either be a fixed value defined by the network operator or computed based on a statistical evaluation of historical failure events. It can be updated dynamically each time a new client joins the system, i.e., whenever the GA is rerun. In this study, we assume that the probability δ_l is predefined and determined by the network operator. Let Δ_p denote the failure probability of path p. Accordingly, the failure probability for this path can be computed using equation (4), where *hopcount* denotes the number of hops along path p. In order to assign paths in a way that will create a more resilient infrastructure, the $\zeta(.)$ and $\xi(.)$ functions in the fitness function are redefined as given in functions (5) and (6), respectively, using the Δ_p value. The $(5)^{th}$ equation calculates the aggregated available bandwidth of the paths assigned to the new client. It multiplies the bandwidth portion allocated to the client—accounting for other flows on the same path—by the probability that the path does not fail, i.e., (1 - Δ_p). This function thus provides a probabilistic, weighted estimate of the available bandwidth in a potentially faulty environment, where link failures may occur. Similarly, the $(6)^{th}$ equation incorporates path failure probabilities into the evaluation of delay differences between the assigned paths. By weighting the path delays with their respective probabilities of not failing, the function favors chromosomes that include more reliable paths while still considering their delay values.

Notably, our system is designed to support the execution of the GA in both event-driven and periodic modes, enabling adaptive path selection under dynamic network conditions. While this capability was not fully tested due to hardware limitations—all experiments were conducted on a laptop—the observed runtime performance suggests that the system can operate effectively in real-time when deployed on a dedicated server. All codes developed for the proposed approach are available in the project's GitHub repository.

In the next section, we provide a comparative performance evaluation of the proposed approaches presented in this section.

IV. PERFORMANCE EVALUATION

In order to evaluate the performance of the proposed GA based approach, we conducted a series of experiments within the Mininet environment. Mininet is an emulation platform that provides the infrastructure to emulate topologies and physical network components and to implement SDN based approaches. As the SDN controller, we implemented the Java based, open-source Floodlight controller. DASH[1] was used as the video streaming application to evaluate the performance. In this study, the MPTCP implementation is based on MPTCPv0, integrated into the Linux kernel following the legacy version provided by MPTCP Linux project.[2]

As described earlier, the GA based path selection is implemented as a module within the SDN controller, which manages forwarding rules without modifying the MPTCP kernel. This approach avoids kernel-level complexity and helps maintain system stability. It also minimizes computational overhead and supports real-time responsiveness by leveraging the controller's global network view. Big Buck Bunny video files, available in four different representations with average bitrate of 1500 Kbps, 2000 Kbps, 2500 Kbps, and 2600 Kbps, were placed on the servers. Two distinct topologies were implemented to conduct the tests. The first topology is the CompuServe network topology, obtained from the Internet Topology Zoo[3], a repository of real-world network topologies. The CompuServe topology, mainly used by ISPs, consists of 11 switches and 14 links to connect them. The second topology is a well-known data-center topology, Jellyfish [37]. This topology consists of 20 four-degree switches, interconnected randomly. All clients and servers in these experiments are connected to the topologies via three network interfaces.

Given that both topologies provide a large number of paths between the clients and servers, we employed a preprocessing methodology to establish a feasible path scale for the respective topologies. Paths with high delay values were eliminated from the initial pool. This preprocessing resulted in population sizes of 150 and 300 for the CompuServe and Jellyfish topologies, respectively.

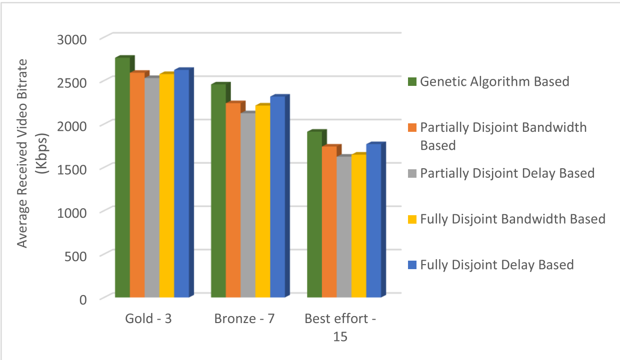
To evaluate the performance, we observed key QoE parameters in experiments conducted under heterogeneous network conditions, characterized by links with limited bandwidth and a high standard deviation in the distribution of path bandwidths. The delay values of the paths were set between 100 ms and 700 ms, while the mean bandwidth of the paths was set to approximately 23 Mbps. All bandwidth and delay values were distributed with Poisson. Paths with varying delay values were intentionally configured to create a test environment in which packet reordering could occur if path selection does not account for delay differences. This setup allows for evaluating the algorithm's effectiveness in mitigating the impact of packet reordering. Furthermore, we conducted tests with 25 clients. At this scale, the dynamic behavior of DASH applications introduces highly variable network conditions, enabling performance evaluation under realistic and challenging scenarios. Specifically, clients request video segments at irregular intervals, resulting in dynamically fluctuating available bandwidth as perceived by each client.

In order to provide comparable performance evaluation, four path selection approaches were also implemented and tested over the same network setting. The path selection approaches that are used for the comparison consider a combination of bandwidth, delay difference, and disjointness as criteria for selecting paths.

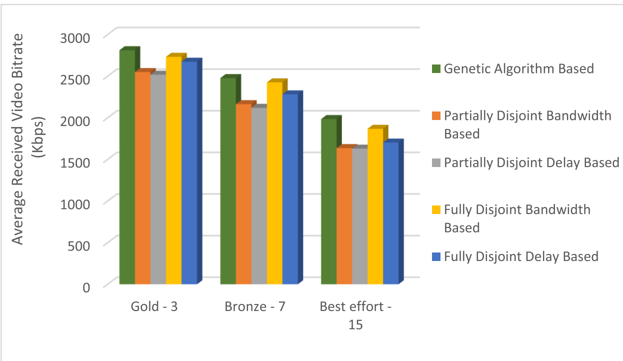
- Partially Disjoint Bandwidth based: In this approach [38], the maximum available bandwidth of paths is considered when assigning paths to clients. As a path is selected for a subflow, the other subflow will not use the links of the assigned path unless a link's bandwidth capacity exceeds a certain threshold.
- Partially Disjoint Delay based: In this approach [39], the lowest latency of the paths is considered in assigning paths to the clients. As a path is assigned to a subflow, the links of the current subflow can not be used in another subflow of the client unless the latency is lower than a certain threshold.
- Fully Disjoint Bandwidth based: In this study [28], a fully disjoint strategy is used. As a client joins the network, the algorithm assigns paths based on the maximum available bandwidth. It is worth mentioning that once a link is assigned to a subflow, the same links can not be used in another client subflow.
- Fully Disjoint Delay based: This study uses a fully disjoint strategy [40]. In addition, subflows with the lowest delay difference are assigned to the clients. In order to avoid choosing paths with high end-to-end delays, an upper bound is considered to eliminate paths with higher latency.

The next section provides comparative results for all approaches.

[1]https://github.com/Dash-Industry-Forum/dash.js
[2]https://multipath-tcp.org
[3]http://www.topology-zoo.org/dataset.html

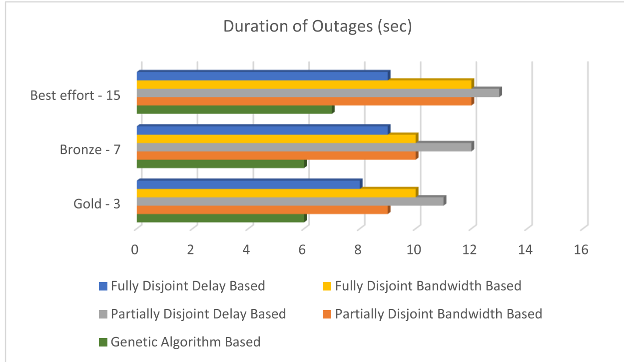


(a) CompuServe Topology

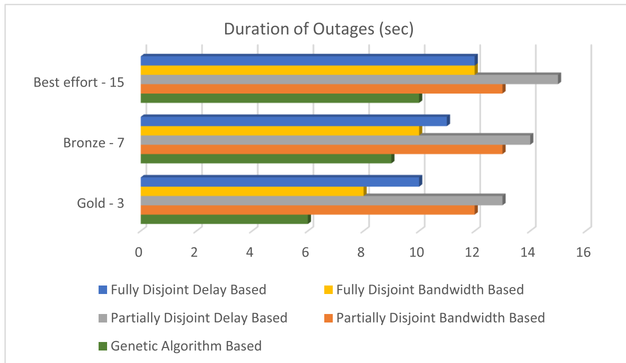


(b) JellyFish Topology

FIGURE 2. **Average received video bitrate, QoS experiments.**



(a) CompuServe Topology



(b) JellyFish Topology

FIGURE 3. **Average duration of outages, QoS experiments.**

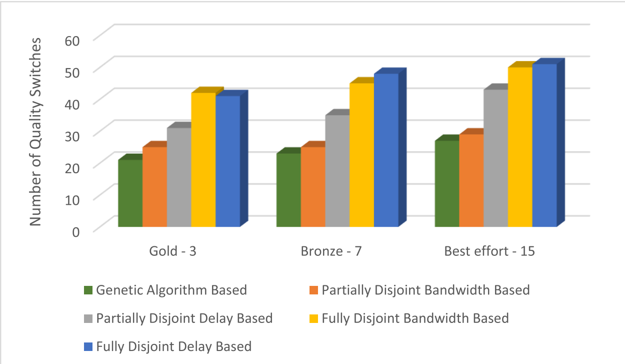
A. EXPERIMENTS TO EVALUATE THE PERFORMANCE REGARDING QUALITY OF SERVICE

The GA's performance evaluation providing different service classes was performed with three user types: Gold, Silver, and Best Effort. Gold users were guaranteed a bitrate of at least 2500 Kbps, while Silver users were guaranteed a bitrate of at least 2000 Kbps. No specific guaranteed bitrate was determined for the third group of users. In the experiments, the received video bitrate, the number of quality changes, and the average duration time were observed with 25 users. Three clients are in the Gold class, 7 in the Bronze, and the other 15 clients are in the Best Effort class. To provide a fair comparison environment, in the comparison studies used in the tests, if the bandwidth values of the paths assigned to the clients do not comply with the restriction given in the user class, they are not used.

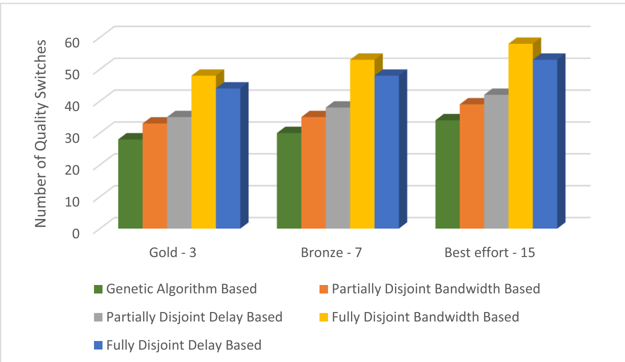
Fig. 2 shows the average bitrate graphs obtained in the comparative tests conducted with 25 clients. For all studies, the clients obtained the guaranteed bitrates in the classes to which they belong. However, it was observed that the clients with the GA based approach played videos with higher bitrates than other comparison studies. Among other studies, some approaches performed better than others regarding user type, topology, and bandwidth distribution.

The duration of outages is presented in Fig. 3, where this QoE parameter represents the duration of the average of the video stalls experienced by the clients. As the graphs illustrate, higher outages are observed in the studies with partially disjoint paths compared to others. This situation caused users to play lower-quality videos on the same network. Nevertheless, this behavior was not observed in the GA based approach using partially disjoint paths. In addition, the outage duration results indicate that packet reordering significantly impacts service continuity, and approaches that consider delay differences among paths perform better. In the CompuServe topology, the Fully Disjoint Delay based algorithm—which also considers delay differences—yields the second-lowest outage duration after our proposed method. However, this trend is not maintained in the Jellyfish topology, as factors other than packet reordering also contribute to outage duration. Nonetheless, our approach consistently achieves the lowest outage durations across both topologies, owing to its comprehensive consideration of delay differences, bandwidth, disjointness, and service differentiation. In the experiments, the outage time remained below 8 seconds with the GA, while it could reach up to 14 seconds in other studies.

Fig. 4 demonstrates the average number of video quality changes for 25 clients. The GA shows good performance improvements compared to other path selection algorithms. The results indicate that the GA optimizes network paths

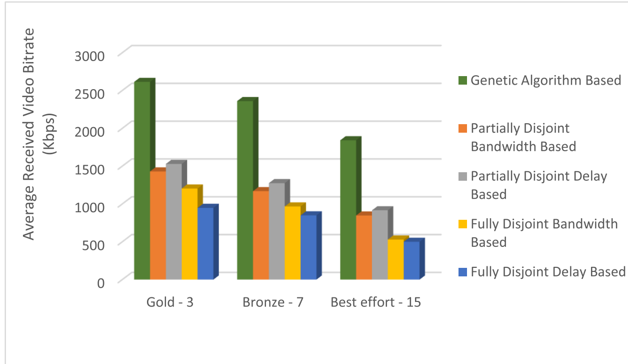


(a) CompuServe Topology

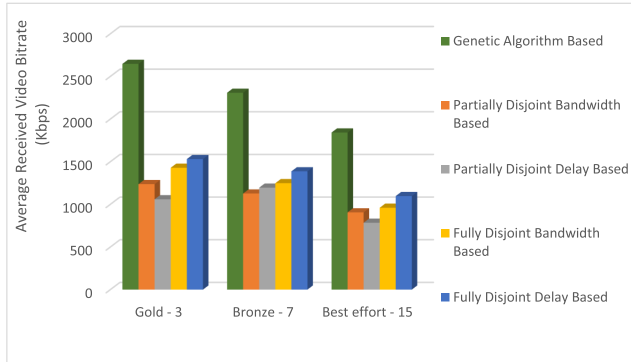


(b) JellyFish Topology

FIGURE 4. **Average number of quality switches, QoS experiments.**



(a) CompuServe Topology



(b) JellyFish Topology

FIGURE 5. **Average received video bitrate, QoS and resiliency experiments.**

more effectively than fully disjoint selection methods. This algorithm offers greater adaptability and efficiency, particularly useful in dynamic network environments where conditions frequently change.

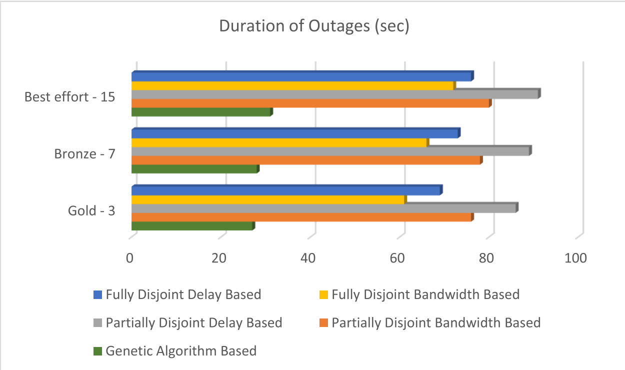
In our testbed, all DASH clients utilize an 8-second startup buffer and commence playback after receiving four consecutive 2-second segments. At startup, the Adaptive Bit Rate (ABR) logic consistently selects the lowest bitrate (1500 kbps), and given that the available network bandwidth reliably supports this rate in all scenarios, the initial buffering time remains stable across clients. Empirical measurements indicate that the initial delay is consistently between 2 and 2.5 seconds. Due to this uniformity, we consider initial buffering time to have negligible impact on comparative QoE evaluation and therefore exclude it from detailed discussion in the results section.

In these experiments, clients download the full video without engaging in user behaviors such as fast-forwarding or seeking to different parts of the content. If such behaviors were introduced, outage durations would likely increase across all approaches to a similar extent, as these actions accelerate buffer depletion. This effect arises from the nature of DASH based systems, where any content navigation results in a new segment request. Therefore, while such interactions were not part of this study, their impact would likely be consistent across all tested methods.

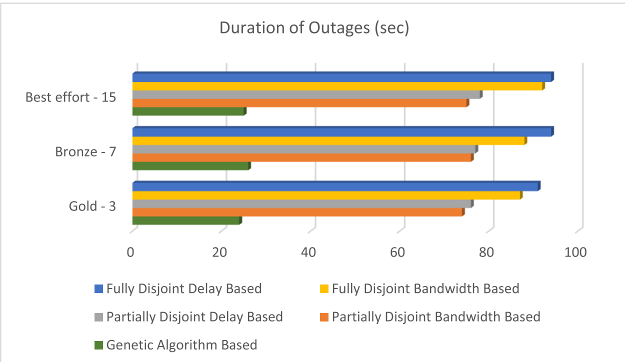
B. EXPERIMENTS FOR EVALUATING THE PERFORMANCE REGARDING RESILIENCY

This section gives the performance of the GA developed to increase the stability of the MPTCP connection and minimize the impact of errors such as link loss/access problems on the network. In the performance measurements performed with 25 clients and two topologies with two different bandwidth distributions, errors were created randomly in 15% of the links on the network.

In the version of the GA, which considers providing only QoS, the number of subflows selected by the algorithm was typically two. However, with the involvement of the resiliency consideration, our approach introduces a new level of adaptability, selecting three subflows for all clients. This adaptability is a crucial strength of our algorithm, allowing it to adjust to different scenarios. When resiliency is not a factor, the algorithm minimizes the number of subflows to address the packet ordering problem. However, when resiliency is a concern, the GA reacts to this condition, increasing the number of subflows by the number of client interfaces to maintain service. The average bitrate graphs in Fig. 5 demonstrate the adaptability of this approach, guaranteeing service to Gold and Bronze users. In contrast, other studies that do not consider resiliency struggle to provide guaranteed service consistently.

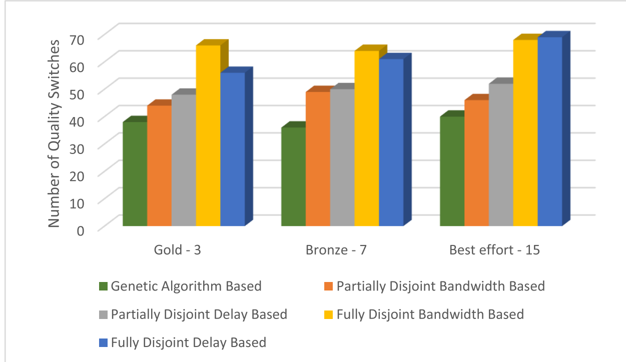


(a) CompuServe Topology

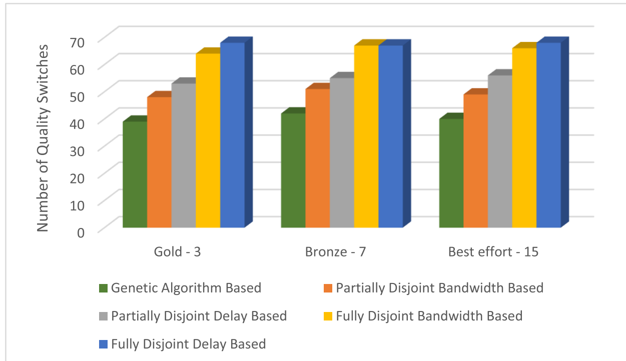


(b) JellyFish Topology

FIGURE 6. **Average duration of outages, QoS and resiliency experiments.**



(a) CompuServe Topology



(b) JellyFish Topology

FIGURE 7. **Average number of quality switches, QoS and resiliency experiments.**

Fig. 6 shows the duration of outages for this set of experiments. In this test group, more outages were observed than the outage values observed in the previous set of experiments. One of the general reasons for this is that the bandwidth values were kept lower in this section. However, another important reason for the increased outage duration is the difficulty in accurately estimating bandwidth on the client side. This occurs due to path errors or insufficient bandwidth. The GA based approach selected paths for a more significant number of subflows in order to increase the durability at this stage. While the algorithm, which generally determines two subflows for clients in other approaches, determines three subflow paths for all clients with this approach. A disadvantage of this situation is its negative effect on packet ordering. This effect was another factor that increased the outage of duration times. While the duration of outage varies between 20-30 seconds with the GA based approach, it can be observed between 50-90 seconds in other approaches.

Fig. 7 indicates that the GA offers superior performance in terms of resiliency compared to the other four algorithms. This suggests that the GA is more effective in managing network variations and maintaining stable video quality. This effectiveness is likely due to the algorithm's ability to dynamically adapt to changing network conditions and better optimize the selection of efficient paths that mitigate quality degradation.

Fig. 8 shows the video bitrate observed over time in a sample client included in the Gold class in the CompuServe topology. The sample client selected for all approaches is the client with the worst results in the test. An error occurred in 15% of the links on the network 200 seconds after the start of the video streaming session. The bandwidth of the paths used by the clients assigned with the ga remained above 2500 Kbps even after the outage. On the other hand, it is observed that the video bitrate obtained in other approaches that do not consider resilience decreases. When the GA based approach is used, the outage duration observed by this client is measured as 22 seconds. The outage duration for the sample clients given in the graph is 70 seconds in the Partially Disjoined Bandwidth based approach, 71 seconds in the Partially Disjoined Delay based approach, 54 seconds in the Fully Discrete Bandwidth based approach, and 58 seconds in the Fully Discrete Delay based path selection approach.

To highlight the proposed GA's ability to provide uninterrupted service after modifying the fitness function, we conducted a test using a Partially Disjoint Bandwidth and Delay based approach using the GA that we formerly proposed in [3] which does not consider resiliency while selecting paths. As the graph clearly shows, the Partially

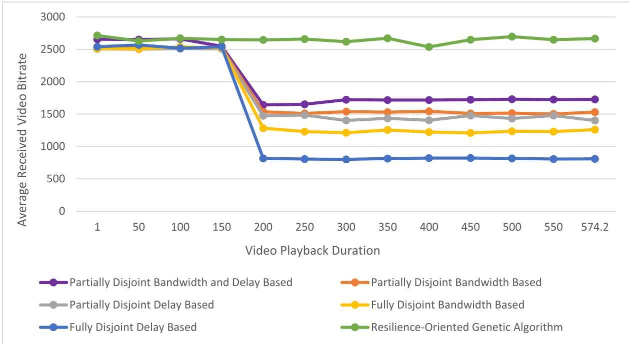


FIGURE 8. Average video bitrate as a function of time.

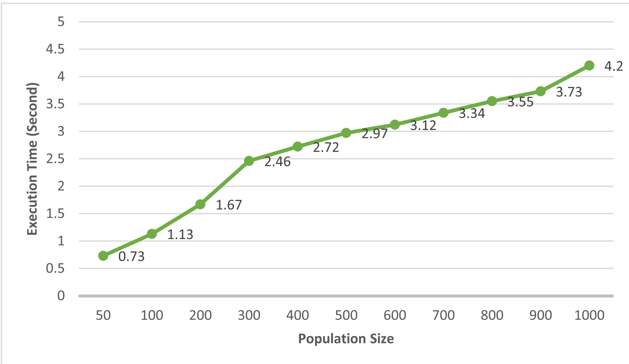


FIGURE 9. The genetic algorithm run-time performance with various population sizes.

Disjoint Bandwidth and Delay based GA was unable to maintain the guaranteed bandwidth for the client.

This research involved configuring a complete network setup, which included the SDN network with Floodlight Controller, Industry DASH video client, Mininet, and the GA along with the comparison approaches, on a computer equipped with an Intel® Core^{TM} i7-10750H processor and 16.0 GB of RAM. The number of paths differed in the network topologies utilized in this configuration, resulting in varying running times for the GA depending on the topology used. The population size was set at 150 for the CompuServe topology and increased to 300 for the Jellyfish topology. Moreover, we evaluated how different population sizes might influence the execution time by executing the developed GA with varying population sizes and logging the run-time. The results of this experiment are presented in Fig. 9. As illustrated in the graph, The running time of the algorithm ranged between 0.5 seconds and 2.46 seconds, depending on the network topologies implemented in the experiments. According to the results, the proposed GA based approach can deliver results even in time-sensitive scenarios.

V. CONCLUSION

In conclusion, this paper has demonstrated the efficacy of a GA based service differentiation approach with MPTCP transmission within an SDN architecture. Our approach ensures optimal path selection for different service classes and provides higher resiliency against network failures by dynamically adapting to changing network conditions. The proposed method effectively guarantees bitrate for different user classes, ensuring that the QoE aligns with the respective service class requirements. This achievement is attributed not only to the guaranteed bitrate levels but also to the underlying path selection mechanism, which accounts for delay differences across paths. These combined features enhance the overall performance and reliability of the system, demonstrating its suitability for providing differentiated services in dynamic network environments. The experimental results in various network topologies and conditions indicate that the GA outperforms traditional methods in maintaining connectivity and delivering consistent video quality. This research underscores the potential of integrating advanced algorithms into SDN-controlled MPTCP systems to meet the growing demands of high-quality media streaming over the Internet. These findings are particularly relevant for next-generation video streaming applications, such as cloud gaming, high quality streaming, and IoT based video services, where network stability and differentiated QoS are critical. It is worth noting that the experimental scale was determined based on the capabilities of the computing system, which featured an Intel i7-10750H processor and 16 GB of RAM.

In this study, the fitness function in the GA is designed to select path combinations that satisfy the bandwidth requirements of the given service class while minimizing delay differences. The function employs fixed weights to balance these two objectives. As a potential improvement, dynamically adapting these weights based on service class sensitivity or real-time network conditions could further enhance performance. This could be achieved through the integration of a machine learning model capable of learning optimal weight configurations under varying scenarios. Exploring this adaptive approach remains a promising direction for future work.

In future work, we will address scalability challenges in large-scale environments and design experiments that incorporate dynamic bandwidth conditions, bursty traffic patterns, heterogeneous network interfaces—including 5G, WiFi, and satellite—and evolving network topologies to more accurately reflect real-world deployment scenarios. We also plan to further refine the algorithm and explore its applicability to other latency-sensitive and bandwidth-intensive applications beyond video streaming, such as IIoT. Additionally, we aim to leverage emerging in-network computing and caching techniques to improve QoS. These directions are intended to enhance the adaptability and robustness of our approach under more dynamic and heterogeneous network conditions.

APPENDIX

ABBREVIATIONS

MPTCP Multipath Transmission Control Protocol.
TCP Transmission Control Protocol.

SDN	Software-Defined Networking.
QoS	Quality of Service.
QoE	Quality of Experience.
GA	Genetic Algorithm.
DASH	Dynamic Adaptive Streaming over HTTP.
IP	Internet Protocol.
HoL	Head-of-Line.
KPI	Key Performance Indicator.
IIoT	Industrial Internet of Things.
OSPF	Open Shortest Path First.
VC	Virtual Communication (Channel).
MANET	Mobile Ad Hoc Network.
SD-WAN	Software-Defined Wide Area Network.
WAN	Wide Area Network.

ACKNOWLEGMENT

The authors thank TÜBİTAK for their support.

REFERENCES

[1] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, "How hard can it be? Designing and implementing a deployable multipath TCP," in *Proc. 9th USENIX Symp. Networked Syst. Design Implement. (NSDI 12)*, Apr. 2012, pp. 399–412.

[2] M. Morawski and P. Ignaciuk, "Influence of congestion control algorithms on head-of-line blocking in MPTCP-based communication," in *Proc. 27th Telecommun. Forum (TELFOR)*, Nov. 2019, pp. 1–4, doi: 10.1109/telfor48224.2019.8971059.

[3] S. Bikas and M. Sayıt, "Improving QoE with genetic algorithm-based path selection for MPTCP," *IEEE Trans. Netw. Service Manage.*, vol. 21, no. 4, pp. 3874–3888, Aug. 2024, doi: 10.1109/TNSM.2024.3411104.

[4] T. V. Mathew, "Genetic algorithm," Tech. Rep., 2012, vol. 53.

[5] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm—A literature review," in *Proc. Int. Conf. Mach. learningbig datacloud parallel Comput. (COMITCon)*, Jan. 2019, pp. 380–384, doi: 10.1109/COMITCON.2019.8862451.

[6] H. Li, Y. Wang, R. Sun, S. Guo, and H. Wang, "Delay-based congestion control for multipath tcp in heterogeneous wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshop (WCNCW)*, 2019, pp. 1–6, doi: 10.1109/WCNCW.2019.8902835.

[7] L. H. G. FerrazD, M. F. Mattosand, and O. C. M. B. Duarte, "A two-phase multipathing scheme based on genetic algorithm for data center networking," in *Proc. IEEE Global Commun. Conf.*, Apr. 2014, pp. 2270–2275, doi: 10.1109/GLOCOM.2014.7037197.

[8] P. Tomar, G. Kumar, L. P. Verma, V. K. Sharma, D. Kanellopoulos, S. S. Rawat, and Y. Alotaibi, "CMT-SCTP and MPTCP multipath transport protocols: A comprehensive review," *Electronics*, vol. 11, no. 15, p. 2384, Jul. 2022, doi: 10.3390/electronics11152384.

[9] W. Wang, X. Wang, and D. Wang, "Energy efficient congestion control for multipath TCP in heterogeneous networks," *IEEE Access*, vol. 6, pp. 2889–2898, 2017, doi: 10.1109/ACCESS.2017.2785849.

[10] V. P. Hristov and A. V. Hristov, "Optimizing packet transmission mechanism with multipath technologies," in *Proc. 10th Int. Sci. Conf. Comput. Sci. (COMSCI)*, 2022, pp. 1–4, doi: 10.1109/COMSCI55378.2022.9912609.

[11] A. Bhardwaj and H. El-Ocla, "Multipath routing protocol using genetic algorithm in mobile ad hoc networks," *IEEE Access*, vol. 8, pp. 177344–177482, 2020, doi: 10.1109/ACCESS.2020.3027043.

[12] S. Qi, L. Yang, L. Ma, S. Jiang, Y. Zhou, and G. Cheng, "MOMTA-HN: A secure and reliable multi-objective optimized multipath transmission algorithm for heterogeneous networks," *Electronics*, vol. 13, no. 14, p. 2697, Jul. 2024, doi: 10.3390/electronics13142697.

[13] R. Agarwal and R. Bhatia, "Multipath routing using genetic algorithm in elastic optical network," *J. Opt.*, vol. 53, no. 3, pp. 2316–2321, Jul. 2024, doi: 10.1007/s12596-023-01395-4.

[14] A. Chai, L. Wang, C. Guo, M. Li, W. Yin, and Z. Fang, "AS-MPCA: An adaptive scheduling algorithm for industrial Internet of Things based on multi-population co-evolution," *Internet Things*, vol. 31, Jul. 2025, Art. no. 101596, doi: 10.1016/j.iot.2025.101596.

[15] S. Benzerogue, A. Sahraoui, M. Derdour, and A. Kouzou, "Optimizing emergency video delivery in connected vehicular networks using a genetic approach," in *Sebha Univ. Conf. Proc.*, vol. 4, no. 1, 2025, pp. 141–146.

[16] D. Kreutz, F. M. V. Ramos, P. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[17] R. Majdabadi, M. Wang, and L. Rakai, "SODA-stream: SDN optimization for enhancing QoE in DASH streaming," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, May 2022, pp. 1–5, doi: 10.1109/NOMS54207.2022.9789779.

[18] H. Nam, K.-H. Kim, J.-Y. Kim, and H. Schulzrinne, "Towards QoE-aware video streaming using SDN," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 1317–1322, doi: 10.1109/GLOCOM.2014.7036990.

[19] A. A. Barakabitze and R. Walshe, "SDN and NFV for QoE-driven multimedia services delivery: The road towards 6G and beyond networks," *Comput. Netw.*, vol. 214, Jul. 2022, Art. no. 109133, doi: 10.1016/j.comnet.2022.109133.

[20] L. Wang, X. Wuand, and D. T. Delaney, "QoE-oriented routing mixing application KPIs and link metrics through machine learning," *IEEE Access*, vol. 12, pp. 10166–10180, 2024, doi: 10.1109/ACCESS.2024.3390577.

[21] M. Beshley, N. Kryvinska, H. Beshley, O. Panchenko, and M. Medvetskyi, "Traffic engineering and QoS/QoE supporting techniques for emerging service-oriented software-defined network," *J. Commun. Netw.*, vol. 26, no. 1, pp. 99–114, Feb. 2024, doi: 10.23919/JCN.2023.000065.

[22] M. Taha, "An efficient software defined network controller based routing adaptation for enhancing QoE of multimedia streaming service," *Multimedia Tools Appl.*, vol. 82, no. 22, pp. 33865–33888, Mar. 2023, doi: 10.1007/s11042-023-14938-5.

[23] G. Manfredi, L. De Cicco, and S. Mascolo, "Optimal QoE-fair resource allocation in multipath video delivery network," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 3487–3500, Sep. 2022, doi: 10.1109/TNSM.2022.3162750.

[24] N. Al-Imareen and G. Lencse, "Enhancing real-time video streaming quality via MPT-GRE multipath network," *Electronics*, vol. 14, no. 3, p. 497, Mar. 2025, doi: 10.3390/electronics14030497.

[25] D. He, X. Yu, C. Lin, C. Westphal, Z. Ming, L. Cui, X. Zhou, J. J. Garcia-Luna-Aceves, and Y. Li, "Enhancing video conference applications with VCApather: A network as a service perspective," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 21, no. 2, May 2025, Art. no. Article 1, doi: 10.1145/3732780.

[26] Y. A. Pimpalkar, S. Ravindran, J. Bapat, and D. Das, "A novel E2E path selection algorithm for superior QoS and QoE for 6G services," *IEEE Trans. Netw. Service Manage.*, vol. 22, no. 2, pp. 1174–1187, Jun. 2025, doi: 10.1109/TNSM.2024.3519707.

[27] L. P. Aguirre Sanchez, Y. Shen, and M. Guo, "MDQ: A QoS-congestion aware deep reinforcement learning approach for multi-path routing in SDN," *J. Netw. Comput. Appl.*, vol. 235, Mar. 2025, Art. no. 104082, doi: 10.1016/j.jnca.2024.104082.

[28] K. Gao, C. Xu, J. Qin, S. Yang, L. Zhong, and G.-M. Muntean, "Qos-driven path selection for mptcp: A scalable sdn-assisted approach," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6, doi: 10.1109/WCNC.2019.8885932.

[29] K. Herguner, R. S. Kalan, C. Cetinkaya, and M. Sayit, "Towards QoS-aware routing for DASH utilizing MPTCP over SDN," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–6, doi: 10.1109/NFV-SDN.2017.8169827.

[30] Y. Chen, F. Lin, S.-Y. Hsu, T.-L. Sun, Y. Huang, and C.-H. Hsiao, "Adaptive traffic control: Openflow-based prioritization strategies for achieving high quality of service in software-defined networking," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 3, pp. 1–12, Jul. 2025.

[31] Z. Zeng, M. Qin, and D. Liang, "A cybertwin-enabled multipath transmission scheme in cloud native networks," *IEEE Trans. Mobile Comput.*, vol. 24, no. 7, pp. 1–16, Jul. 2025, doi: 10.1109/TMC.2025.3550129.

[32] M. J. Alenazi, "Evaluating multipath TCP resilience against link failures," *ISeCure*, vol. 11, no. 3, p. 15, 2019, doi: 10.22042/isecure.2019.11.0.15.

[33] T. Tsuru, M. Hasegawa, Y. Shoji, K. Nguyen, and H. Sekiya, "An implementation and evaluation of MPTCP-based IoT router," *Multimedia Tools Appl.*, vol. 82, no. 18, pp. 28389–28404, Jul. 2023, doi: 10.1007/s11042-023-14781-8.

[34] Y. Zhang, J. Tourrilhes, Z.-L. Zhang, and P. Sharma, "Improving SD-WAN resilience: From vertical handoff to WAN-aware MPTCP," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 1, pp. 347–361, 2021, doi: 10.1109/TNSM.2021.3052471.

[35] S. González, A. De la Oliva, C. J. Bernardos, and L. M. Contreras, "Towards a resilient openflow channel through mptcp," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, 2018, pp. 1–5, doi: 10.1109/BMSB.2018.8436865.

[36] N. Vuppalapati and T. G. Venkatesh, "Candidate selection algorithms for hybrid IP/SDN networks with multi-link failures," *IEEE Trans. Netw. Service Manage.*, vol. 22, no. 2, pp. 1219–1231, Jun. 2025, doi: 10.1109/TNSM.2024.3504534.

[37] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking data centers randomly," in *Proc. 9th USENIX Symp. Networked Syst. Design Implement. (NSDI)*, vol. 9, Apr. 2012, pp. 225–238.

[38] Z. Jiang, Q. Wu, H. Li, and J. Wu, "Scmptcp: SDN cooperated multipath transfer for satellite network with load awareness," *IEEE Access*, vol. 6, pp. 19323–19332, 2018, doi: 10.1109/ACCESS.2018.2820719.

[39] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifeachor, "A novel QoE-centric SDN-based multipath routing approach for multimedia services over 5G networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7, doi: 10.1109/ICC.2018.8422160.

[40] R. Alvizu, G. Maier, M. Tornatore, and M. Pióro, "Differential delay constrained multipath routing for SDN and optical networks," *Electron. Notes Discrete Math.*, vol. 52, pp. 277–284, Jun. 2016, doi: 10.1016/j.endm.2016.09.042.

SHADI BIKAS received the B.S. degree in computer science from Orumiyeh Payame Noor University, in 2013, and the M.S. and Ph.D. degrees in information technology from Ege University, in 2019 and 2024, respectively. She is currently a Faculty Member with the Department of Computer Engineering, TOBB University of Economics and Technology, Ankara, Türkiye. Her research interests include optimization algorithms for software-defined networking and multipath TCP, as well as reinforcement learning and deep learning.

MÜGE SAYIT (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in information technologies from the International Computer Institute, Ege University, Türkiye, in 2005 and 2011, respectively. She was an Assistant Professor with the International Computer Institute, where she was an Associate Professor, in 2017. She has been working as the Principal Investigator or a Researcher in various research and development projects. She is currently with the School of Computer Science and Electronic Engineering, University of Essex, U.K. Her research interests include software defined networking, network function virtualization, future networks, video streaming, and video codecs.