electronics

Article

Video Streaming Adaptive QoS Routing with Resource Reservation (VQoSRR) Model for SDN Networks

Majda Omer Elbasheer [1], Abdulaziz Aldegheishem [2], Nabil Alrajeh [3] and Jaime Lloret [4,*]

1 College of Computer Science and Information Technology, Sudan University of Science and Technology, P.O Box 407, Khartoum 11111, Sudan; majda.omer@gmail.com
2 Urban Planning Department, College of Architecture and Planning, King Saud University, P.O. Box 57448, Riyadh 11574, Saudi Arabia; aldeghei@ksu.edu.sa
3 Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, P.O. Box 10219, Riyadh 1433, Saudi Arabia; nabil@ksu.edu.sa
4 Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universitat Politécnica de Valencia, Camino Vera s/n, 46022 Valencia, Spain
* Correspondence: jlloret@dcom.upv.es

Abstract: Video streaming has become extremely widespread, especially with the growing number of users and the spread of mobile devices, along with the increase in the availability and diversity of multimedia applications and communication technologies. Real-time video communication requires awareness of the quality of experience (QoE) to provide customers with a satisfactory service, for example, in smart cities that use video surveillance systems. The quality of service (QoS) is dependent on network performance, which directly affects the QoE. However, reliance on traditional network infrastructure and routing protocols cannot assure QoS. The emergence of software defined networks (SDN) may eliminate current network limitations. Due to SDN's global view and programmability characteristics, such capabilities could help in providing an automated QoS control and management. This paper introduces video streaming adaptive QoS-based routing and resource reservation (VQoSRR), which gives SDN networks the ability to meet video demands and enhance user experience over best effort networks, such as the ones required for video surveillance in smart cities. In order to implement QoS-based routing (QBR), we developed algorithms for calculating routing, installing routing paths in the forwarding devices, and shifting traffic to an alternative path when QoE is violated. As well, we used queuing mechanisms to allocate resources based on the QoE requirements of video streaming. Our results indicate that resource reservation mechanisms combined with QoS-based routing enable effective control over routes and resources. Our framework guarantees the video quality as well. This technique of using video streaming would improve the tools and applications used for smart cities such as surveillance systems for hospitals and civil defense organizations.

Keywords: video streaming; software defined networks (SDN); QoS; QoE; QoS-based routing (QBR); resource reservation; video surveillance; smart cities

check for updates

Citation: Elbasheer, M.O.; Aldegheishem, A.; Alrajeh, N.; Lloret, J. Video Streaming Adaptive QoS Routing with Resource Reservation (VQoSRR) Model for SDN Networks. *Electronics* **2022**, *11*, 1252. https://doi.org/10.3390/electronics11081252

Academic Editors: Dimitris Kanellopoulos, Miguel García-Pineda and Daniel G. Costa

Received: 7 March 2022
Accepted: 12 April 2022
Published: 15 April 2022

1. Introduction

With the explosive growth of the Internet, video streaming driven by providers of streaming media services is already a large portion of Internet traffic today and is predicted to continue growing. Another ongoing trend that feeds this growth is the increasing number of smartphone devices, social media users, and the advancement of networks technologies, such as Wi-Fi and 5G connections. According to the Cisco Global Forecast Highlights report [1], video traffic will account for 80% of global Internet traffic by 2021.

This increasing demand for high-quality online video requires network operators and media service providers to adopt new strategies and technologies. In this regard, QoS-based routing has emerged to enable the routing layer to enhance traffic performance and overall quality of service (QoS) by meeting multiple users' QoS requirements. However, its routing decisions depend on multiple metrics, and finding a path that meets the composed multiple metrics of delay, cost, packet loss probability, and jitter is an NP-complete problem [2].

Due to the special characteristics of video streaming and its applications, the networks must be managed to provide a convenient and guaranteed level of QoS and QoE. However, the QoS model in the traditional Internet faced unresolved problems such as bounded visibility of the global topology, per-hop decisions, and the difficulty of providing different levels of QoS for different applications flows. Furthermore, some of the existing IP short path routings algorithms cannot verify the link QoS required to adapt to the requested requirements of QoS for the flow, since it is unaware of the available QoS over the path [3]. In contrast, IP link-state protocols provide QBR by flooding updates to exchanging routing information and reflecting an up-to-date view of the network in order to calculate new routes. However, the frequent flooding process can impose significant communication overhead on forwarding devices, and repeatedly changing the routing paths can increase the delay experienced by the end-users [3,4]. Another aspect of the shortest path routing is that it allows the using of multiple equal-cost paths as alternatives to redirect traffic from one path to another when a better one is available. The router will redirect the traffic even if the existing route has an acceptable but non-optimal cost that can meet the service requirements of the already-existing traffic. Along with the short path routing issues described above, the Internet best effort service led to an insufficient QoS implementation.

In addition, Internet video streams with higher resolution definitions have become more popular recently. The Cisco study also estimates that the standard-definition (SD) video traffic will decrease to 24.5% in 2021 compared to 61.4% in 2016. In contrast, high-definition (HD) video traffic will rise to 56.3% in 2021, up from 36.4% in 2016; and ultra-high-definition (UHD) or 4K will be 19.2% of Internet video traffic in 2021, up from 2.2% in 2016 [1]. The video resolution is one of the video content characteristics that impact the video quality of experience (QoE) because it indicates the level of detail in a video frame. Displaying videos at a higher definition will maximize user-perceived quality; however, increasing the video resolution will make more visible those problems that may occur during the video delivery, causing different levels of degradation for the user QoE. As a result, it requires a good level of network QoS [5] because QoS parameters, such as jitter, delay, lost packets, etc., influence video QoE [6].

Additionally, video traffic is resource-intensive and consumes a lot of network bandwidth, especially high-resolution videos, and videos with a high bit rate; therefore, any planned model for videos streaming must provide resource reservation capabilities. Additionally, it must balance between these resource allocation techniques and QoS-based routing methods to avoid affecting network QoS, in turn avoiding affecting video QoE.

In recent years, the emergence of software defined networking architecture has allowed for innovative approaches in networking. With SDN, the control of the network is decoupled from the forwarding devices, enabling more flexible network management and programmability. SDN controllers use the OpenFlow protocols to gather information about the network topology and its states in order to support the controlling functionality. In contrast to link-state protocols, using a high specification controller to collect the updates of the data plane state can reduce the communication overhead on all forwarding devices due to the load of the flooding process used in the exchange of the link-state. Furthermore, the controller awareness of states can facilitate controlling the frequency of the emission of updates. In addition, the global view of resource availability enables traffic flows to be rerouted dynamically to ensure efficient resource utilization.

The main contribution of this paper is to propose a video streaming adaptive QoS-based routing and resource reservation (VQoSRR) model for SDN networks. It investigates the QoS-based routing and resource reservation and shows how their combination could achieve overall enhancement in the QoS and QoE of video streaming. The model differentiates video traffic based on the QoE parameters of the video resolution, and this will directly reflect the impact of the QoS improvement on the user experience.

Many approaches have already been proposed to achieve video streaming QoS in SDN, some of which use QoS-based routing, while others focus on resource reservation, we discuss their relationship to our proposed VQoSRR in Section 2. We designed VQoSRR to perform QoS-based routing within a single administrative domain (intra-domain) but not across inter-domain networks. In addition, we focused on using the QoS metrics that can solve the QoE problems at the network level, not at the video application level. Our integration of resource reservation mechanisms with QBR can provide adequate control over the route and resources, but this comes at the cost of extra setup time, even though, generally, the QoS will be more enhanced. It would be interesting to consider new parameters like scalability to address the issues of an increased number of users and videos. However, this parameter is not within the scope of this paper. We argue that our approach provides several advantages:

- Combining per-flow QoS routing with resource reservation is minimizing the packet loss and latency for QoS flows and non-QoS flows, firstly, by guarantees that each flow type uses different routing paths, and secondly, by guaranteeing effective bandwidth allocation for QoS flows.
- The differentiation of traffic based on QoE parameters (as video resolution); this implies the effect of QoS directly reflected to the end-user impressions.
- The proposed framework reduces the overhead of obtaining network status by pre-computing of alternative paths.

In this work, we extend the work presented in [7], which proposed a QoS-based routing algorithm (two lowest loss widest paths algorithm (TwoLLWPs)). This paper modified the QoS-based routing model to incorporate resource reservations and add new routing functionalities.

This paper describes the experience and results of identifying an SDN network architecture that can provide QoE for video end-users by selecting routing paths based on several individual QoS metrics and reserving network resources for those routing paths. The key contributions of this study are summarized below:

- We design a management system that monitors and collects performance information.
- We develop QBR algorithms for path selection, rerouting traffic to an alternative path, and installing routing paths based on video streaming QoE requirements.
- We define a higher-level reservation control strategy to enable administrating of allocating bandwidth for the different flows according to their requirements. Furthermore, we couple it with a method to utilize the per-class queuing system to reserve bandwidth for the transmitted video to optimize QoS/QoE and enhance the overall resource usage.
- We apply our methodology by streaming videos of different resolutions and evaluating their quality performance under a network topology experiencing packet loss and congestion.

The rest of this paper is organized as follows. Section 2 includes the related work. The proposed video VQoSRR framework is described in Section 3. The experiments and results are presented in Section 4. Finally, the conclusion and future work are shown in Section 5.

2. Related Work

Many solutions have been proposed for enabling video QoS in SDN networks. Ghalwash and Huang [8] proposed a framework for applying QoS in an SDN-based network. For QoS, they select the traffic route based on the shortest end-to-end delay metric, using the Dijkstra algorithm. The framework monitored the port utilization to reduce congestion and packet loss. Three types of applications, namely TCP, UDP, and VoIP, were evaluated based on reduced delay, jitter, and packet loss.

Egilmez et al. [9–11] designed an OpenQoS multimedia controller to provide a dynamic end-to-end QoS routing, according to the network state. Their solution enhanced the video QoS for end-users by offering two paths: one is a QoS route specified for the multimedia, and the second is the shortest path for the other data. Using the LARAC algorithm, OpenQoS calculated the QoS paths that depend on the packet loss rate as routing metrics. To support dynamic rerouting of QoS packets, the controller monitored the network links, so in case of congestion, it invoked the routing algorithm to calculate a new path. However, the OpenQoS framework queried the network status each second. This would result in additional overhead for the controller to compute routing per network state, especially if the state had not changed significantly.

To enhance network quality of service, Sendra et al. [12] proposed SDN routing optimization based on reinforcement learning (RL). The proposed routing protocol used this artificial intelligence (AI) method to select the optimal paths with the least cost according to the network status.

Yu and Ke [13] presented a genetic algorithm-based routing method to provide efficient video delivery over SDN, called GA-SDN. It used chromosome fitness as a metric to choose the best path from multiple possible solutions. In case of congestion, the link weight will be increased to decrease fitness. Since their approach defined fitness as the reverse of path cost, when there was a higher fitness, it indicated a higher QoS. Similarly, Parsaei et al. [14] proposed a model for critical delay-sensitive telesurgery applications based on SDN networks. The model used a type-2 fuzzy system (T2FS) and cuckoo optimization algorithm (COA) to generate reliable QoS routes. The model computed two paths, with the first set as the primary path and the other an alternative in case of failures; the delay metric was used as a link constraint to determine which routes are optimal between the remote surgeon and the operating robots at the patient's side.

Henni et al. [15] developed a framework for QoS routing in SDN to enhance video streaming quality and best effort flows throughput. The authors leveraged SDN properties to create a consistent view of the network, consistent decisions, and a consistent enforcement strategy of rules. Their approach minimized the concentration of video streams on links to reduce packet loss and maximize QoS.

Volpato et al. [16] introduced an architecture that integrated autonomic and proactive QoS management into SDN environments. This architecture enabled QoS configuration on data plane devices. In addition, it monitored, predicted, and analyzed the network performance to achieve resources' optimizations and avoid degradations in the QoS. However, it focused on optimizing resource utilization without making any guarantees of meeting the service thresholds.

Sharma et al. [17] proposed a framework to enable QoS for business customer traffic and provide on-demand prioritization based on flow differentiation and resource reservations. Flows are classified based on the type of service (TOS) field and the destination IP to differentiate best effort traffic from business customer traffic. The authors applied the rate shaping technique to reserve queues, and they configured each router with a high priority queue and a low priority queue. In addition, the FlowQoS [18] system utilized SDN to provide per-flow QoS for broadband access networks. FlowQoS depended on classification and rate shaping based on the policies defined by the user. This system created a virtual switch topology inside the router and configured each switch according to a user-defined rate. Similarly, Khater and Hashemi [19] implemented differentiated services on SDN networks to enhance the quality of service. They distinguished the flow within the network by changing the differentiated service code point (DSCP) value in the ToS field and then assigning different queues to each flow based on the DSCP value. When necessary, their proposed method shifted the flows between switch queues to prevent increasing delays, while utilizing available capacity within other switch queues to accommodate new flows.

Rego et al. [20] described an architecture for monitoring urban traffic in emergencies based on SDN. Their approach combined SDNs and Internet of Things (IoT) networks for more effective management of emergency resources. Their architecture enabled the modification of vehicle routes dynamically by changing traffic lights to facilitate the movement of emergency service units.

Canovas et al. [21] proposed a multimedia traffic management system based on the QoE estimation scheme and traffic pattern classification for SDN networks. They implemented two models. The first one is a QoE model based on Bayesian regularized neural networks (BRNN) for multimedia traffic classification based on the objective QoE. A second model determines which video characteristics should be changed to improve QoE in difficult situations. These characteristics are selected based on QoS parameters.

Xu et al. [22] proposed a QoS-enabled management framework to support the transmission of video streaming and multimedia applications over SDN networks. They classified traffic as a QoS flow or a best effort flow. The framework used either an algorithm that optimized routes or a queueing mechanism to guarantee the service requirements. The routing algorithm dynamically rerouted the high priority flow when network congestion occurs; if there is no feasible path to transmit the QoS flow, the framework enabled a queue reservation instead. Owens and Durresi [23] designed a video over software-defined networking (VSDN) architecture and protocol for optimizing QoS routing and queuing for video transmutation. They implemented a signaling QoS framework like integrated services (IntServ) to guarantee end-to-end QoS for video applications. With the VSDN protocol, video applications can request video service from the network by providing a QoS API used by sender and receiver. Despite that, the architecture may not be scalable in large networks due to the potential signaling overhead between video senders and receivers.

Finally, Yan et al. [24] presented the HiQoS SDN framework to guarantee QoS. It provided services differentiation and multipath routing. HiQoS computed multiple paths between the source and destination using a modified Dijkstra algorithm; the optimal ones were chosen based on the lowest bandwidth consumption. As a result, it was resilient against link failures through rerouting flow to another route. However, HiQoS had only used minimal bandwidth utilization as a metric, ignoring other video streaming quality metrics. Our proposed VQoSRR, instead, employs queue mechanisms to meet bandwidth guarantees for video traffic in addition to providing two routing paths between the source and destination to satisfy multiple QoE constraints.

According to the previously mentioned related works, many of them are focused on per-flow QoS routing, for example, [8–13]. In contrast, others focused on resource reservation schemes only, such as [16–19]. Other approaches used both methods interchangeably [22] or together [23,24]. Using QoS-based routing only determines the path with the best chance of acquiring the requested QoS. However, it does not involve a mechanism to reserve the required resources [3]. It also imposes an overhead because it gathers traffic status information actively. On the other hand, using resource reservation alone provides a mechanism for reserving network resources. Although, it does not provide a method for determining which network path has sufficient resources for the requested QoS [3]. In addition, most studies focused on enhancing network performance without considering video streaming thresholds to ensure QoS/QoE requirements are maintained.

Moreover, there have been multiple mechanisms in the traditional Internet for providing QoS in intra-domain and inter-domain networks, such as integrated services (IntServ) [25], differentiated services (DiffServ) [26], and multiprotocol label switching MPLS [27]. Nevertheless, they have restricted deployment due to the absence of global network views inherited from existing best effort networks. In addition, these techniques cannot adapt to changing conditions on the network topology. Furthermore, the lack of centralized and automated network configurations could cause significant administrative overhead in large networks, leading to network failures and difficulty applying policy enforcement.

In our approach, instead, video flows are dynamically placed on QoS guaranteed routes that meet multiple individual QoE metrics, while reducing the overhead of obtaining network status by pre-computing alternative paths. Additionally, we reserve resources for network flows according to their importance to avoid degradation of video stream quality, especially during high network loads. A comparison of the proposed approach and other SDN QoS architectures is shown in Table 1.

Table 1. A comparison of some SDN QOS ARCHITECTURES related research with the proposed VQoSRR.

Techniques	QoS Solution	Video Threshold Metric	Video QoS/QoE Parameters	Flow Resource Management Model
Ghalwash and Huang [8]	SP	Not used	Not used	Not used
Egilmez et al. [9–11]	QR	Jitter: Guaranteed	Bitrate	Not used
Volpato et al. [16]	DRE	Bandwidth, Loss, Latency: Optimized	Not used	Differentiation by Transport Port Address and protocol. Queues provided according to the Knowledge Base context.
Sharma et al. [17], Seddiki et al. [18]	DRE	Not used	Not used	Queue reservation with differentiating service based on IP header.
Khater and Hashemi [19]	DRE	Not used	Not used	Queue reservation with differentiating service based on DSCP.
Xu et al. [22]	QR or DRE	Delay: Guaranteed	N/D	Queue reservation with differentiating service based on different level of priority.
Owens and Durresi [23]	QR+RE	N/D	Video Resolution	Queue Reservation similar to IntServ.
Yan et al. [24]	SR+DRE	No metric used	N/D	Queue Reservation with differentiating service based on source IP Address
VQoSRR (proposed)	QR+DRE	Bandwidth and Packet Loss Rate: Guaranteed	Video Resolution	Queue reservation with differentiating service based on DSCP.

SR: Shortest Path Routing. QR: QoS based Routing based Solution. RE: Reservation based Solution. DRE: QoS Differentiation and Reservation based Solution. N/D: Not Defined.

3. The Proposed VQoSRR Framework

In this part, we describe the proposed VQoSRR framework. Section 3.1 explains the network model architecture, Section 3.2 describes the QoS routing mechanism, and Section 3.3 describes the VQoSRR reservation method.

3.1. Network Model Architecture

Figure 1 illustrates the general proposed SDN network model and design components. Three layers make up the model: the application plane, control plane, and data plane; we will discuss each layer briefly.

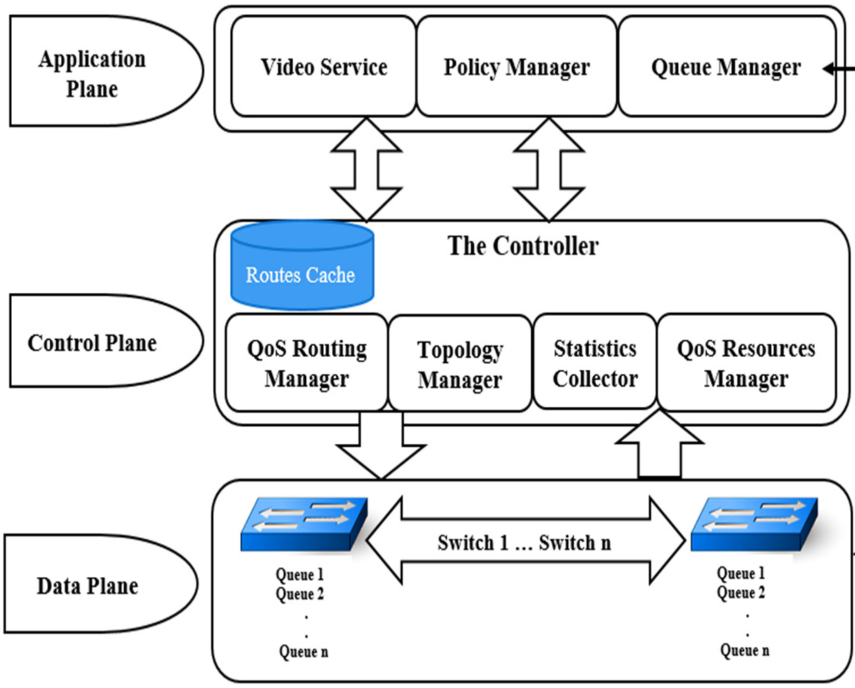


Figure 1. The VQoSRR SDN architecture.

3.1.1. Application Plane

- **Video Service**: This work evaluated the video performance using two different video formats, standard definition and high definition (HD ready falls into this category), all encoded with H.264/AVC. We developed video service applications (both client and server) in Python. The TCP protocol was used to exchange messages between client and server, while the RTSP and RTP over UDP protocols were used to transmit video data. A client application could request a video and specify its resolution and bitrate. However, the server streamed videos over the network using the GStreamer RTSP server. GStreamer is a GNU LGPL-licensed software library and application that allows the reading, converting, recording, editing, and streaming of audio and video files [28].
- **Policy Manager** is responsible for defining and reflecting policy rules to the controller and the queue manager. This paper defined these policy rules:
    a. As thresholds for HD and SD video streams, packet loss rate and bandwidth were used as QoS parameters.
    b. For different traffic types, this work developed three types of service categorization: the first group needs quality of service requirements to be met (called group A); the second group can accept acceptable performance guarantees (called group B); the third group does not require any QoS guarantees (called Best effort).
- **Queue Manager** allows the configuration of queues and ports, in addition to other characteristics.

3.1.2. Control Plane

- **The Topology Manager** keeps track of the network topology graph, requesting and receiving information from the data plane about the connected forwarding devices, new attached elements, or failed links.
- **The Statistics Collector** collects information from OpenFlow switches and periodically polls it so the controller can get an idea of the network's state, such as the availability of resources and whether the network is congested or not.

- **The QoS Routing Manager** module is responsible for QoS-based route calculation; it applies the routing algorithms to obtain a path for the flows based on its requirements. Further, it is responsible for flow admission control, determining whether the specific route is available to maintain the QoS guarantee of ongoing traffic, and informing the controller of this information. For storing calculated paths, this module uses a route cache structure. Further, the module keeps track of which resources can be admitted and which cannot by storing certain flags.
- **The QoS Resource Manager's** primary role is to reserve the resources for video flow, classify the traffic, and manage flow classes and queues. In addition, the module task is to set up and install flow rules for new incoming flows or update existing ones in forwarding devices.

3.1.3. Data Plane

The data plane is the network topology that enables video transmission between end devices.

3.2. The VQoSRR QoS Routing Mechanism

We proposed a QoS-aware adaptive routing approach for video streaming. It calculates the forwarding paths between any two nodes and enables QoS configuration on data plane devices. In this work, we focused on enhancing the QoE of two types of video streaming resolutions (HD and SD), so we investigated their QoS thresholds and applied them as criteria for quality decisions. The QoS parameters thresholds that we guaranteed in our QBR method are packet loss and bandwidth, and we used them as metrics to choose the feasible paths, more details in [7]. Described below is the workflow of our routing mechanism functionalities:

- Initially, the controller uses the topology manager and statistics collector to discover the network topology and collect network status information from the forwarding elements. Then, we generate a weighted graph where each link is associated with packet loss rate and available bandwidth values. These two steps run periodically for a specified configured time.
- Secondly, when the server initiates a new video stream flow, the switch sends a copy of the first packet of the flow to the controller QoS routing manager to find the routing path.
- Next, we find QoS-based routing feasible paths by QoS routing manager algorithms, where the VQoSRR determines two routing paths to balance between frequent dynamic updating of network state and reduces routing computation overheads. The idea is to use one route for the current flow routing and store the other as an alternative path for rerouting purposes.
- After that, the QoS resource manager sends back the routing rules of the flow routing path to the switches by using the OpenFlow protocol.
- Finally, a dynamic routing modification happens whenever the state changes. As metrics need to be updated frequently, flow path procedures should minimize computation overhead associated with routing. Therefore, when the VQoSRR controller receives a new network status, it does not calculate a new path directly for running flows; it instead uses an algorithm and predefined flags to determine whether to use the alternative routing paths or generate a new one. These flags facilitate the admission control process under policies. Table 2 shows an example of the routing path and flags' storage structure; for example, the first row indicates there is ongoing flow with ID 1. In addition, we find two paths that meet their thresholds: the first path is the current flow path, and the second path is an alternative in case the first path violates the flow thresholds. The path count field indicates the number of paths attached to this flow (there are two paths available: one primary and one alternative). Finally, the admission field tells whether this flow has admitted its QoS requirement or been rejected (it admitted for Flow 1).

Table 2. Example of the flow path cache and flags.

Flow id	First Path	Alternative Path	Path Type	Path Count	Flow Admission
Flow 1	S1-S3-S5	S1-S2-S4-S5	meet two metrics	2	Admitted
Flow 2	-	-	-	-	Rejected
Flow 3	S1-S2-S4-S7	null	meet one metrics	1	Admitted

This work integrates three algorithms in order to provide QoS-aware video routing. First algorithm (TwoLLWPs) [7] determines two feasible paths between the server and client based on packet loss rate and available bandwidth; these metrics meet the video QoE request constraints. A detailed investigation of TwoLLWPs and its implementation has already been published [7]. Second algorithm is for adaptive rerouting. Third algorithm is for setting up and updating flow tables by paths' rules. Below are the descriptions of the second and third algorithms, Sections 3.2.1 and 3.2.2, respectively.

3.2.1. The Dynamic Traffic Rerouting Algorithm (DR-RA)

Most QoS routing methods use on-demand path computation, but this has two disadvantages. First, it delays the process of forwarding traffic. Second, it involves the execution of a path computation algorithm for each flow request, adding further overhead to the routers (controllers in case of SDN), notably when the frequency of path calculation is high. In addition, if the QoS metrics change frequently, this will lead to frequent routing updates, which means more computation overhead. Thus, the collecting link metrics should not be reliant on too much dynamism. Furthermore, the complexity of the routing algorithm increased processing overhead [3,4]. Thus, this study proposes a dynamic traffic rerouting algorithm to address these mentioned issues.

The dynamic traffic re-routing algorithm (DR-RA) is responsible for updating the route's cache and rerouting traffic by using the alternative path or generating a new one. Briefly, it performs in this manner: it runs periodically, starting with reading the network statistics based on a predefined interval time. Next, it checks if the current path violates the flow QoS requirements, then deletes its flow entries from the switch. Afterward, it examines the alternative route to see if it satisfies the flow quality requirement. If not, it calculates another path. Using the alternative path presents the following advantages: (1) decreasing the time spent recalculating the routing path by utilizing the alternative route rather than using the routing algorithm again; (2) increasing response times for installing flow; and (3) providing two paths could increase resilience when a path fails. Figure 2 illustrates the flowchart for DR-RA algorithm.

3.2.2. Installing or Updating Flow Path (IUFP) Algorithm

This algorithm is in charge of receiving the switch packet in request, and pushing the path rules into the flow tables for new incoming flow, or updating tables if the configuration rules of already proceeding flow changes. In addition, the IUFP is responsible for admitting or rejecting flow requests because the VQoSRR must route the video stream along a path that can accommodate its QoS requirements, such as meeting packet loss and bandwidth thresholds. Otherwise, it indicates that the QoS currently requested cannot be admitted. Algorithm 1 illustrates this scenario.

Algorithm 1: Installing or Updating Flow Path (IUFP)

Input: Packet_In request
Output: Packet_Out response
Step 1: //Identify if request for new or for proceeding flow by searching in the HashMap paths cache using flow Id:
if *flow Id not found, then*
  isNewFlow go to Step 2.
else
  isOldFlow go to Step 3.
Step 2: Call Algorithm 1 (TwoLLWPs),
if: *Flow Admission Status* == *R*, then
    delete its flow entry from HashMap, reject this flow, end procedure.
else
    pick first path from HashMap, send Flow path response to the switches, end procedure.
Step 3: Call Algorithm 2 (DR-RA),
if: *Flow Admission Status* == *A*, then
    pick first path from HashMap, send Flow path response to the switches, end procedure
else
    delete its flow entry from HashMap, reject this flow, end procedure.

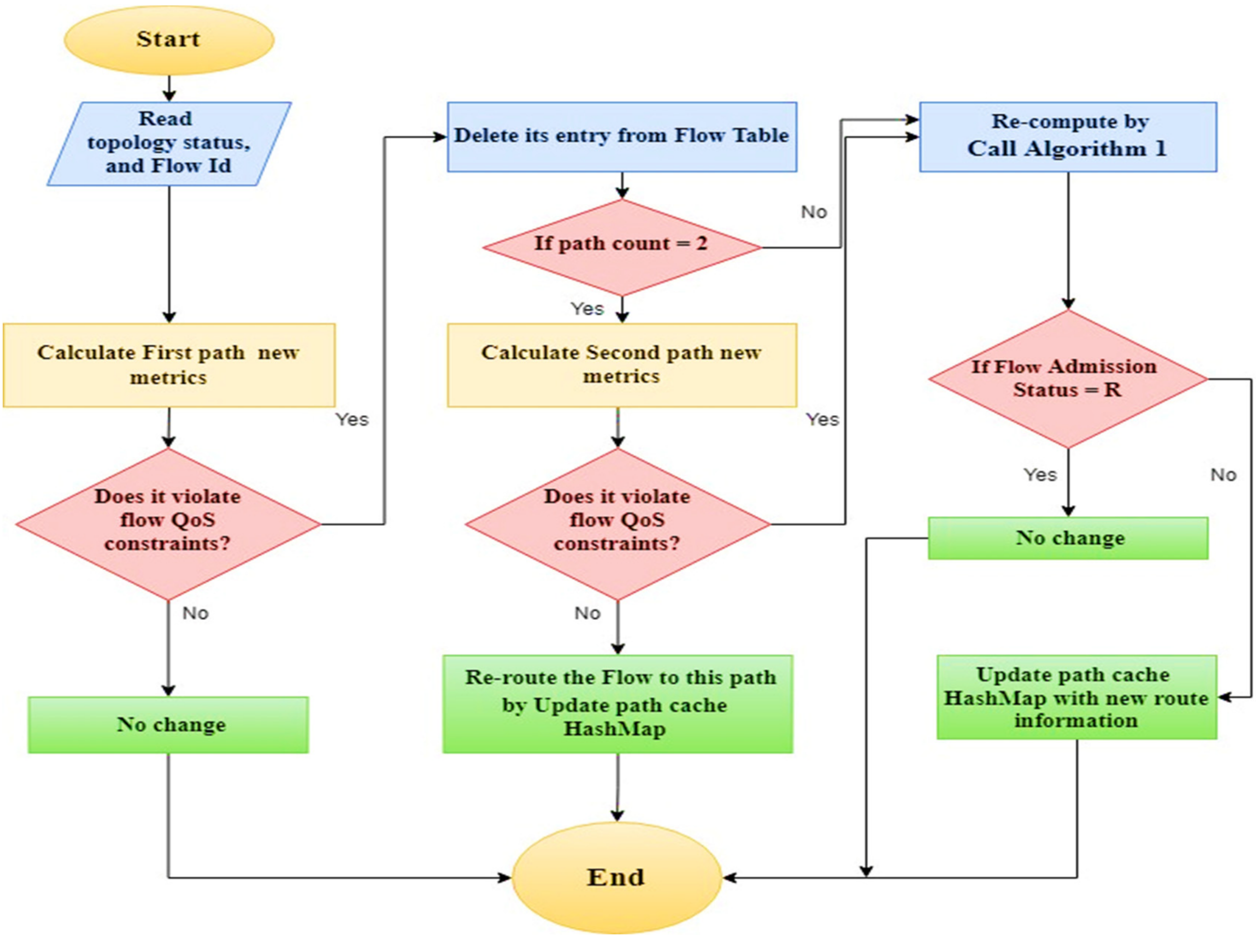


Figure 2. Dynamic re-routing algorithm (DR-RA) flowchart.

3.3. The VQoSRR Reservation Method

Frequently changing traffic metrics may cause QoS-based routing to consume network resources and overburden the controller as a result of frequent routing updates. Hence, the proposed solution relied not just on QBR, but also on queuing mechanisms to achieve a balance between routing overhead and quality assurance and controlling the data flow.

Incorporating resource reservations with QoS routing leads to fine control over the route and resources; this allows better congestion management, thus reducing route updating and providing more stability for QoS routes. Additionally, this method also contributes to reducing latency, a factor that can impact video streaming QoE.

The VQoSRR suggested defining a higher-level reservation control strategy with suitable administrative mechanisms to enable fairness to flow according to their requirements, coupled with a method to configure the underlying resources. The idea is to utilize the rate and per-class queueing reservation mechanism to prioritize flows. The task is to map application traffic to different QoS levels, with each assigned a rated weight then partitioned queues based on the level-specified rate weight. Afterward, each packet is marked to a class with the differentiated service code point (DSCP), and according to that, the network forwarders handle packets. Figure 3 shows the main components of the approach followed by this study to allocate resources for QoS flows and non-QoS flows. Further explanation is provided in the following sections.

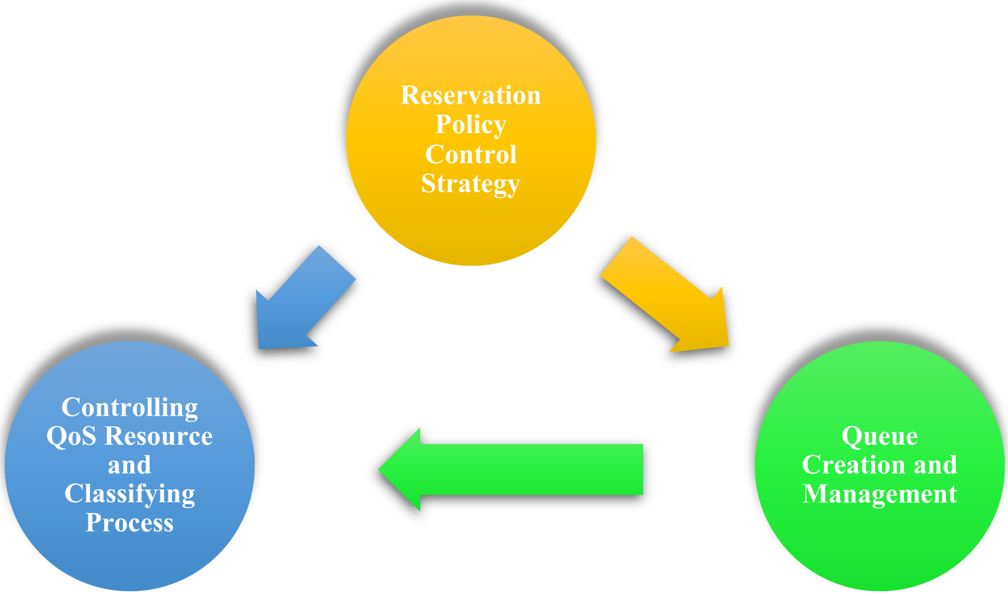


Figure 3. The main components of resource reservation approach.

3.3.1. Reservation Policy Control Strategy

The proposed policy mechanism calculates a weighted rate for each expected flow, then a portion of the total link rate is assigned based on this weight. The study considered employing the QoS thresholds and application service category (strict or soft constraints) as parameters to estimate the weight. The proposed policy defines QoS thresholds as ranges at the policy manager, then different applications' flows are grouped to one of these ranges by the administrator. Consequently, let FG_x be the aggregated flow group under a specific threshold range ($\mathcal{TR}p_j$), where $1 \leq j \leq n$, and p indicate the precedence of the $\mathcal{TR}$ according to its FG quality requirements, whether they are high or low, so that the highest requirements take the first precedence, followed by the next highest, and so on.

Moreover, assume FG_x belongs to one of three service categories: strict or hard QoS constraints, soft or tolerated QoS constraints, and the best effort category without any guarantees. So, let $\mathcal{S} \in \{k, l, m\}$, $k > l > m$, where $\mathcal{S}$ is a parameter denoted to the service category. Additionally, assume that each threshold range and flow group take a degree of importance:

- $\mathcal{Y}$ importance factor of $\mathcal{TR}$ of FG_x.
- $\mathcal{Z}$ importance factor of FG_x.

Based on the above assumptions, the objective is to calculate a weighted bandwidth rate (wr) for FG_x from the total link rate $\mathcal{R}$ in according to the importance of their QoS constraints. Firstly, if the threshold range precedence is smaller than other threshold ranges, then the important factor of it ($\mathcal{Y}$) will be higher, which is formulated as follows:

$$\left\{\begin{array}{c} if \min\left[\mathcal{TR}p_j{}^{FG_1}, \mathcal{TR}p_j{}^{FG_2}, \mathcal{TR}p_j{}^{FG_3}, \ldots, \mathcal{TR}p_j{}^{FG_n}\right] \\ = \mathcal{TR}p_j{}^{FG_x} \, \forall\, x \, \in i,\, 1 \le i \, \le n \\ then\ let \\ \mathcal{Y}^{FG_x} = N \\ where\ f_x(\mathcal{N}) > f_{i-x}(\mathcal{N}) \end{array}\right\} \tag{1}$$

where $\mathcal{N}$ is a number representing the higher weight.

Secondly, from (1), the importance of the flow group is determined from the threshold range importance and service category as,

$$\mathcal{Z}^{\,FG_x} = \mathcal{Y}^{FG_x} + \mathcal{S}_i \tag{2}$$

Consequently, multiple service classes are created, i.e., one class for each flow group $(FG_x, \mathcal{C}_x)$. Then, each class of service $\mathcal{C}_x$ assigned queue with a given wr from $\mathcal{R}$ where wr is found as:

$$wr_{\mathcal{C}_x} = \frac{\mathcal{R} * \mathcal{Z}^{\,FG_x}}{100} \tag{3}$$

where,

$$\sum_{i=1}^{n} wr_{\mathcal{C}_i} \le \mathcal{R}(4) \tag{4}$$

3.3.2. Queue Creation and Management

Ref. [29] states that multimedia streaming services need to use a rate queueing system where schedulers set a minimum, a maximum, or both rates. Therefore, this study allocates the bandwidth for each flow group using rate limiting through using queue manager. It operates as a standalone application, developed in a shell script combined with Python to accomplish queue configurations at network switches; this includes the creation of queues, prioritizing them, assigning bandwidth rates, and destroying the configuration settings. The application communicates with the policy manager to obtain the set-up parameters, including the number of classes and the weighted rate for each class.

3.3.3. Controlling QoS Resource and Classifying Process

Whenever a video server initiates a new stream, the SDN controller QoS resource manager receives the first packet as a packet-in request; after that, it creates a flow entry on the edge switch connected to the server, enabling the high-priority flow with the DSCP field. The alteration of packet fields must be completed at the border of the network, as forwarding decisions are made based on the DSCP value. Then, regular DiffServ forwarding can take place inside the core network. The controller classifies any arriving flow by application traffic transmission protocol, source port, and destination port. Traffic classification is performed corresponding to the QoS policy settings. After that, the controller creates match and action rules at switches to direct the flow to its specified queue based on its DSCP. Figure 4 gives an exemplified explanation of how the resource reservation method works.

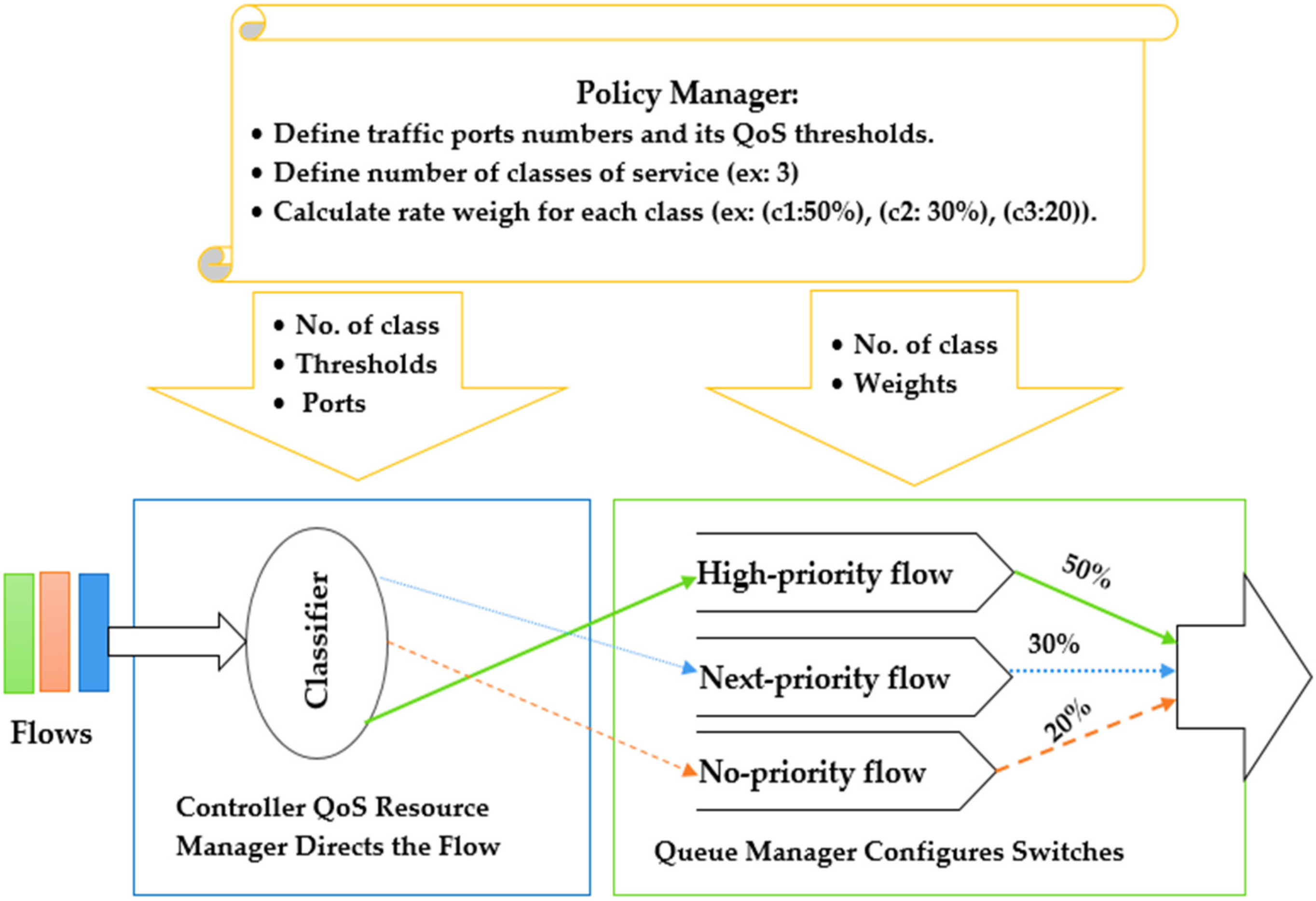


Figure 4. Illustrative example of the queues allocation process.

4. Experiments and Results

This section presents the configuration of the evaluation environment and experiment results.

4.1. Experimental Setup

We tested the performance of the proposed modules using the open-source Floodlight controller, which is an open-source controller for SDN created by the developer community. It is a Java software with a multithreaded interface that supports OpenFlow protocols 1.0–1.5 [30].

A Mininet emulator is used to simulate the network topology [31]. It creates networks, switches, controllers, and network performance parameters using Linux network software, and it supports OpenFlow switches and SDN. We deployed the data plane using Open vSwitch (OVS) as the network forwarding device [32]. It is an open-source and multilayer virtual SDN switch that supports many protocols and standard management interfaces, such as OpenFlow, NetFlow, and IPV6. The tested network consists of nine OVSs and six hosts. Figure 5 illustrates the network topology used in this work.

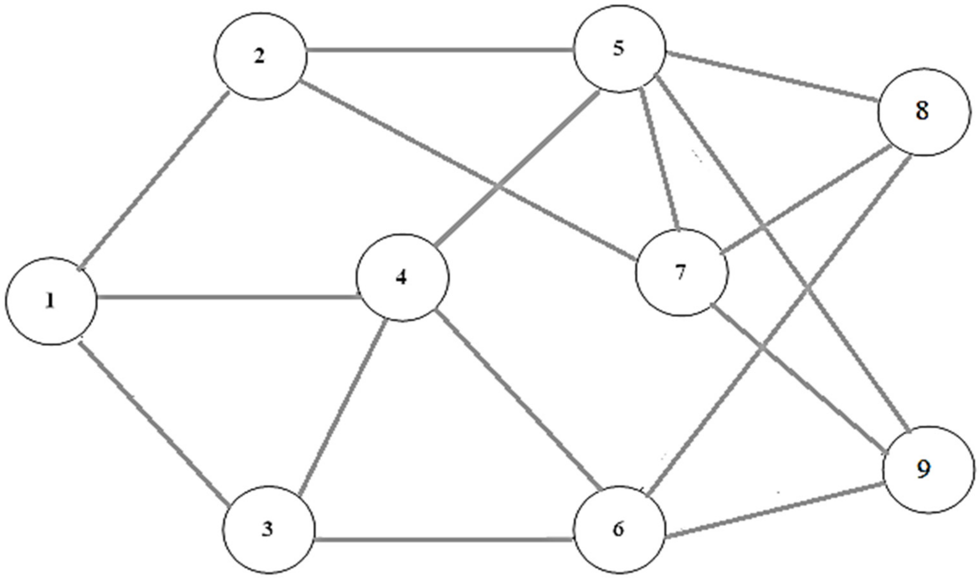


Figure 5. The main topology graph.

Additionally, the testing was conducted on two computers, with Intel core™ i7-7500U, 2.70 GHz CPU, 2 Cores, and 16 GB RAM. We used two operating systems, one was the Microsoft Windows 10 Home, x64-based PC System. The second was the Linux Ubuntu 18.04.5 LTS (Bionic Beaver) OS, x64-based PC System. The default Ethernet network link bandwidth was 100Mb/s. Moreover, the setup included two virtual machines with two processors, one of which contained the Floodlight controller master version and the other contains Mininet Simulator version 2.3.0.

We generated background traffic between hosts using Iperf software [33]. In addition, we used Caminandes Llamigos videos [34] with characteristics shown in Table 3, and the video service was used to transport video traffic between the server and clients.

Table 3. Videos' simulated QoE parameters.

Video Name	Type	Bitrate kbps	Size MB	Duration	Frame/s
caminandes_llamigos_480p	SD (854 × 480)	847	17.5	1:30	24
caminandes_llamigos_720p	HD (1280 × 720)	1660	32.1	2:30	24

4.2. QoE Measurement Metrics

In order to measure the QoE, the delivered video on each client was recorded and matched with the original video sequence using the MSU video quality measurement tool [35]. The proposed model was validated using both objective and subjective metrics of QoE. There are two types of objective metrics used in this study: the structural similarity index metric (SSIM) [36] and the video multimethod assessment fusion (VMAF) [37]. While we used the mean opinion score (MOS) [38] as the subjective metric, a total of 15 users assessed the quality test cases. The following section describes the performed experiments and presents the obtained results.

4.3. Dynamic Rerouting Algorithm Experimental Results

This section aims to assess the performance of the dynamic traffic rerouting algorithm (DR-RA). The simulation was carried out with the following parameters. The link capacity was set to 50 Mb/s. The majority of links had a loss of 1%, while the links on the longest paths had no loss. There were two hosts with best effort (UDP traffic with 5 Mb/s). For SD and HD, bandwidth constraint metrics were 3 and 10 Mbps, respectively, while packet loss metrics were 0.5% and 0.005%. The controller collected the statistics every 4, 6, or 9 s, whereas the DR-RA execution occurred at t = 5, 7, or 10 s.

We designed four test scenarios, firstly, videos were sent using the best effort network without using the proposed VQoSRR. Secondly, the VQoSRR had been tested without using the traffic rerouting algorithm (DR-RA), instead only using the TwoLLWPs. The last two scenarios analyzed the effect of the DR-RA. In the first case, the video flow threshold was assigned to the first service group (A), which had hard QoS constraints and implied the flow must route over the lowest loss path. In the second case, the flow belonged to the second service group (B), which had soft constraints; in this situation, when no lowest loss path exists, the flow takes the widest path.

Figure 6 illustrates a comparison of the four scenarios; it clarifies the effect on video quality by these cases. The results indicate that the SSIM average values of DR-RA provided high quality; the average SSIM of group A was around 0.97 for HD and about 0.96 for SD, and the average SSIM values for group B were close to 0.94 for HD and 0.93 for SD. In comparison, when DR-RA was not applied, the video quality decreased, where the SSIM value of HD's average = 0.92 and SD's average = 0.93. Further, the best effort results (without using the proposed VQoSRR) highly correlated with the degradation of video quality, where SSIM averaged around 0.90 for HD and 0.91 for SD.

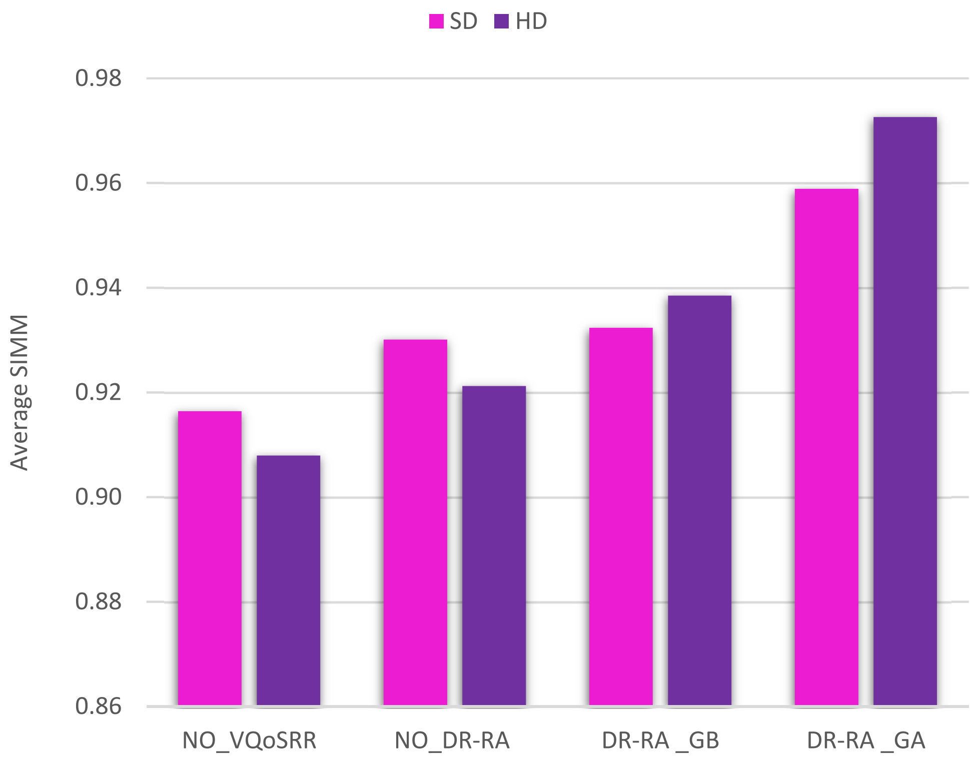
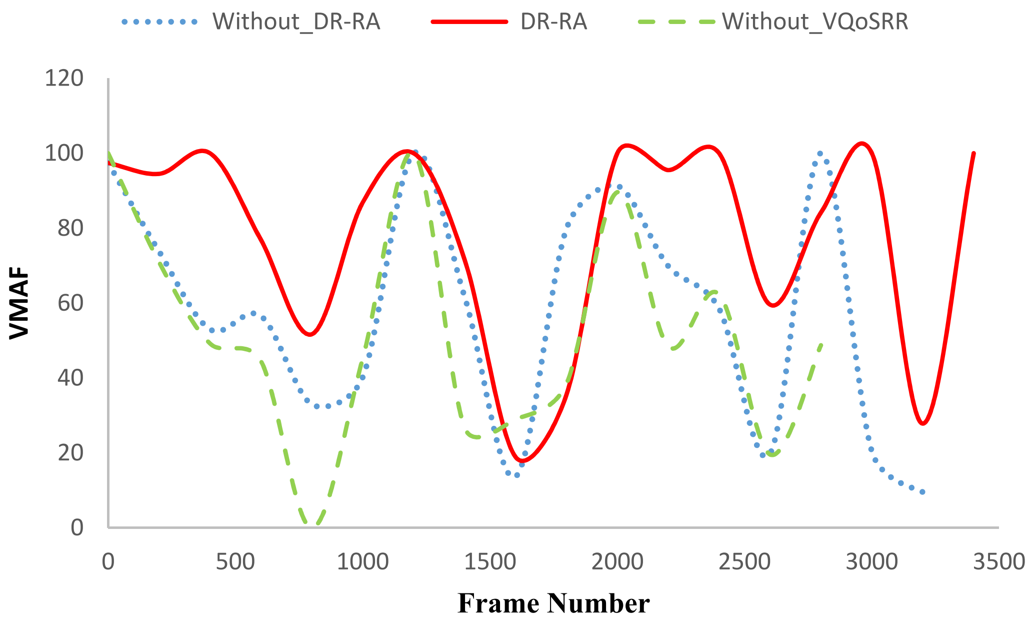
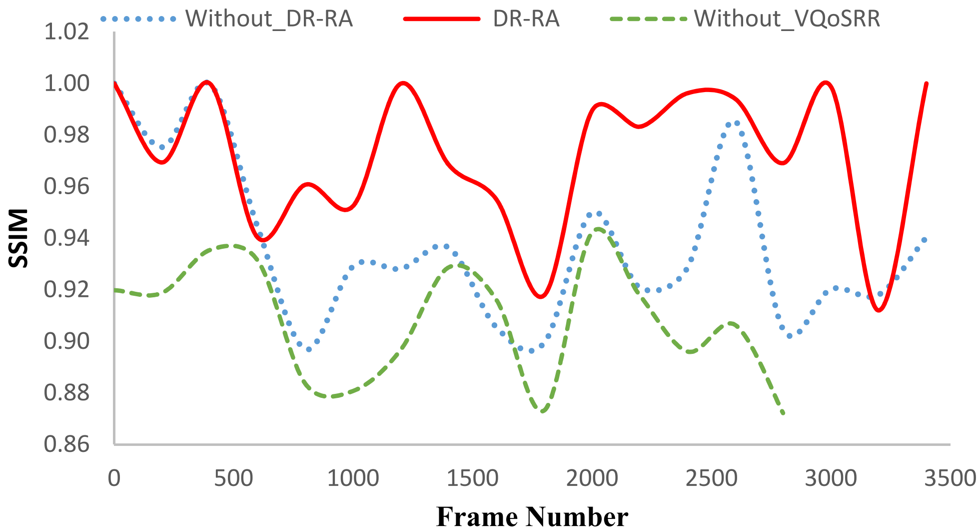


Figure 6. Average SSIM of the four testing scenarios.

Figure 7a,b present VMAF and SSIM measured during video delivery using the real-time transport protocol (RTP), with DR-RA, without DR-RA, and without the VQoSRR separately. A video delivered without DR-RA had higher VMAF and SSIM than a video delivered without VQoSRR. Additionally, when delivered using DR-RA, it had a higher quality than both of them. The reason is that DR-RA can reroute video traffic based on network conditions by re-estimating packet loss probability and bandwidth utilization in order to comply with video QoS requirements, which in turn results in better video quality.



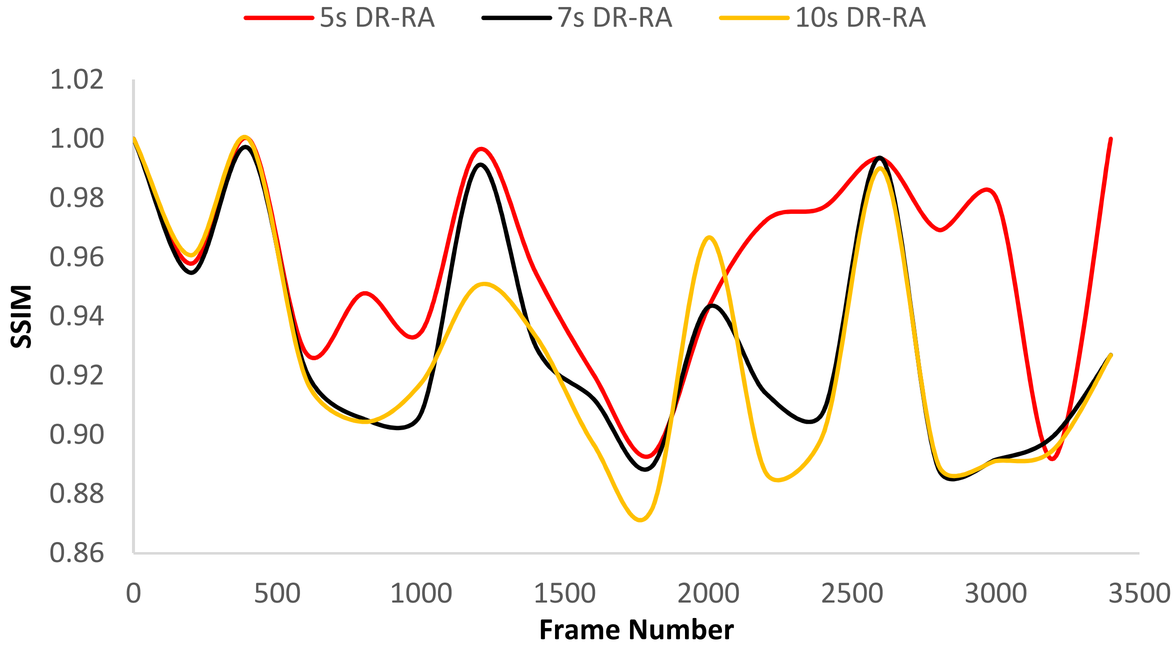
(**a**)



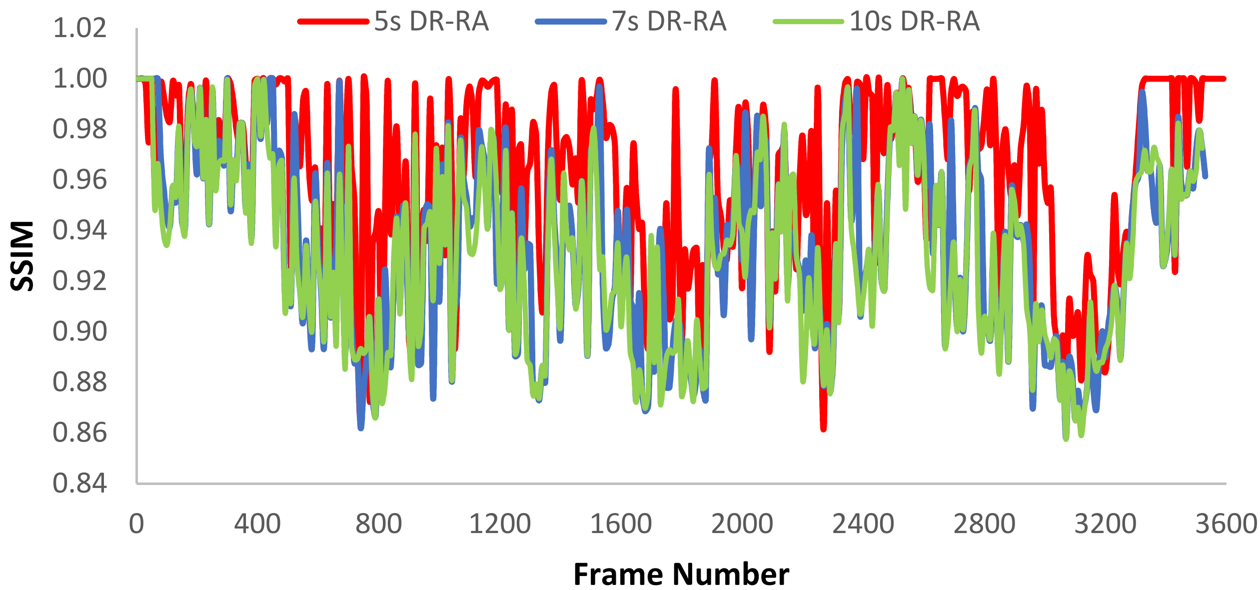
(**b**)

Figure 7. Comparison when using DR-RA, without DR-RA, and without the proposed VQoSRR. (**a**) Comparison of VMAF; (**b**) comparison of SSIM metrics.

DR-RA was executed periodically at predetermined intervals. Hence, this section studies the effect of the execution interval time on ongoing video traffic in order to analyze whether shifting traffic between one path and another affects video quality and which is the most optimal execution interval. In this study, interval times were (5, 7, or 10 s). Figure 8a,b present the achieved results for the received SD video, respectively. The SSIM values are measured every 10 frames in Figure 8a and every 200 frames in Figure 8b. It can be observed from the figures that SSIM was higher when the interval (t = 5 s) compared to (t = 7 s) and (t = 10 s); whereas, there was no significant difference between t = 7 s and t = 10 s in terms of SSIM measurements.



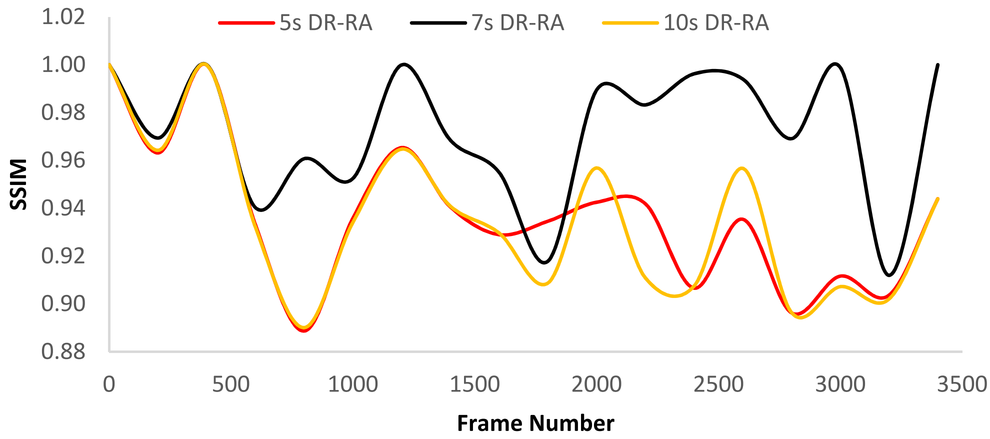
(**a**)



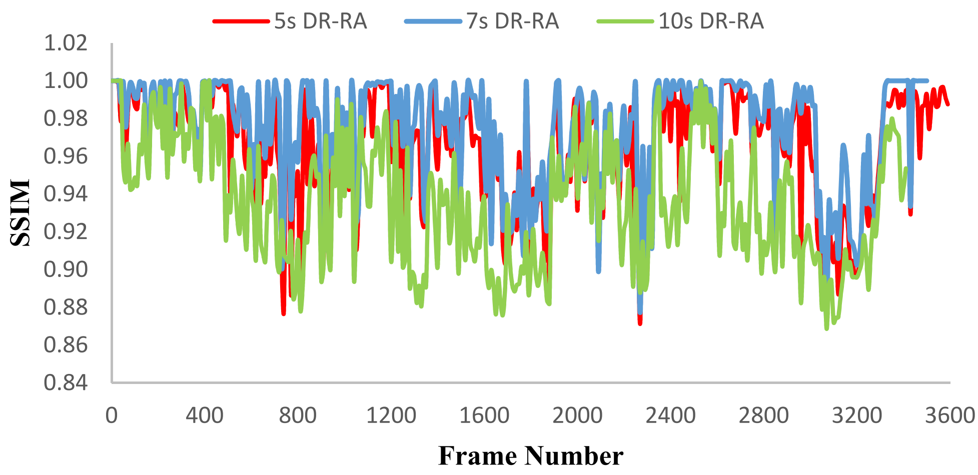
(**b**)

Figure 8. Effects of executing dynamic rerouting algorithm at different times in SD video resolution. (**a**) SSIM values are measured every 10 frames; (**b**) SSIM values are measured every 200 frames.

Furthermore, Figure 9a,b also show the effects of executing the dynamic rerouting algorithm at different time intervals on HD video quality measured by SSIM every 10 frames in Figure 9a and every 200 frames in Figure 9b. As can be seen from the figures, the DR-RA improved the SSIM when interval (t = 7s).



(a)



(b)

Figure 9. Effects of executing dynamic rerouting algorithm at different times on HD video resolution. (**a**) SSIM values are measured every 10 frames; (**b**) SSIM values are measured every 200 frames.

4.4. QBR with Reservation Experimental Results

To validate the proposed reservation methodology, an investigation has been performed to observe how reservation is affecting video QoE. The following simulations parameters helped to measure the performance of the suggested technique. This experiment used three video streams' traffic in MP4 format; VQoSRR classified them into an HD video stream, SD video stream, and best effort video stream. Both HD and SD have specific bandwidth and packet loss rate metrics. They transmitted and decoded simultaneously in real-time through the network; their QoE parameters specifications have been described previously in Table 3. Additionally, on each OVS switch port connected to the network, three queues have been added. One queue was reserved for each of the VQoSRR flow types. The maximum capacity supported on every link was 50 Mbps.

To obtain the results, we conducted three test cases. The first case used the controller default setting without considering QoS and traded all videos as best effort (No_VQoSRR). The second case applied only the QoS-based routing proposed algorithm with the dynamic rerouting algorithm (VQoS-R). The third case combined the dynamic route computation with the per-class queue reservation scheme (VQoS-RR). Simulation parameters that were common for all the test cases are presented in Table 4.

Table 4. Simulation parameters.

Video Type	No_VQoSRR	VQoS-R		VQoS-RR		
		Loss Metric	Bandwidth Metric	Loss Metric	Bandwidth Metric	Queues Rates
Video 1, SD	-	0.5%	3 Mb/s	0.5%	3 Mb/s	Estimated around 30%
Video 2, HD	-	0.05%	10 Mb/s	0.05%	10 Mb/s	Estimated around 45%
Video 3, HD	-	Best Effort	Best Effort	Best Effort	Best Effort	Estimated around 25%

Figure 10a,b illustrate the SSIM calculated in the test cases for SD and HD, respectively. In the figures, it can be seen that the proposed methods (VQoS-R and VQoS-RR) resulted in an improvement in the SSIM scores, and significantly, they were more efficient than the traditional controller shortest path method (No_VQoSRR) where the quality degradation was mostly observed. Furthermore, VQoS-RR offered a better quality video stream than VQoS-R.

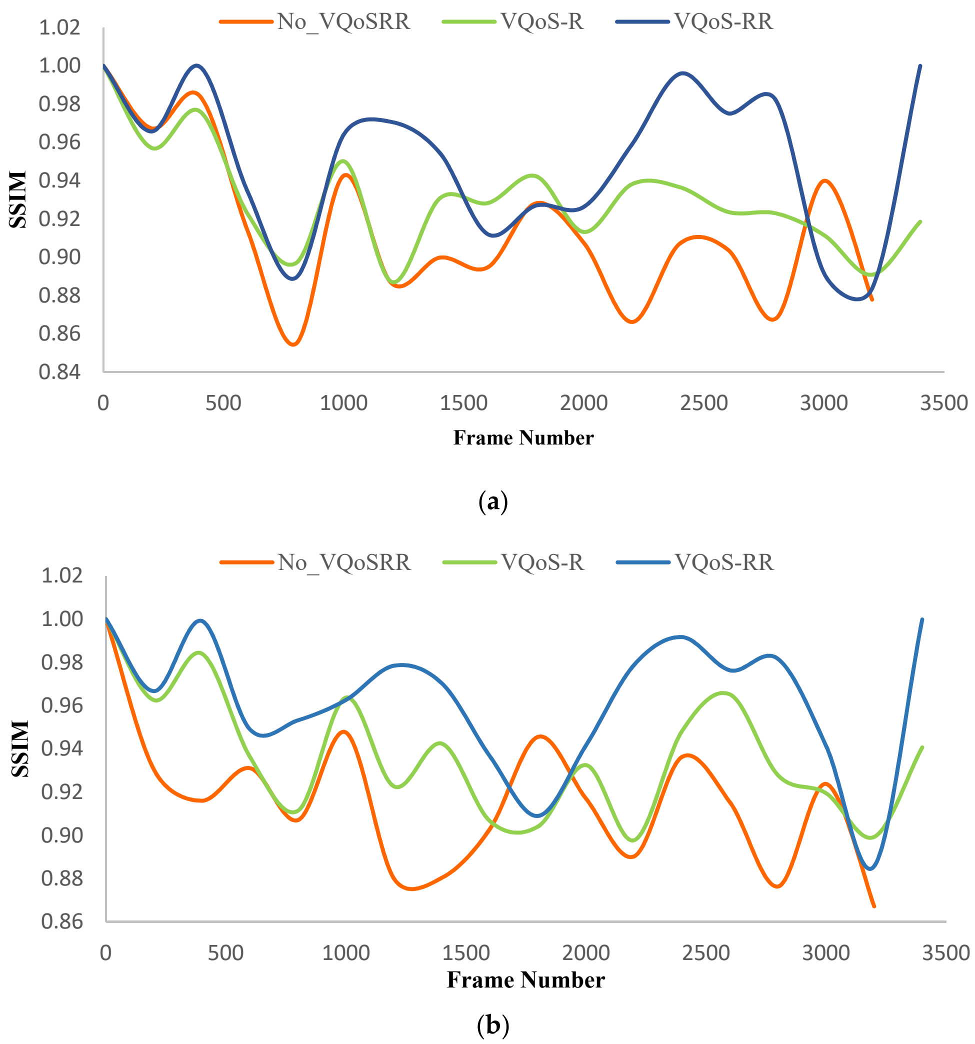


Figure 10. (**a**) Influence of test cases on SD video resolution; (**b**) influence of test cases on the HD video.

Moreover, Figure 11 shows the computed SSIM average values for the received videos streams. As observed from the graph, VQoS-R and No_VQoSRR yielded lower SSIM scores compared to using VQoS-RR. As we have seen, the VQoS-RR provided good average values for all transmitted videos, including the best effort videos. From the figure, we can observe that VQoS-RR improved video quality; for instance, in SD, the SSIM value measured in the case of No_VQoSRR was 0.91, and when measured in the case of VQoS-RR increased to 0.95. Accordingly, the SSIM value for HD was 0.92 for No_VQoSRR, whereas it increased to 0.96 for VQoS-RR. The reason is that the VQoS-RR method combined the reservation mechanism with QoS-based routing, which enhanced end-user perception of quality.

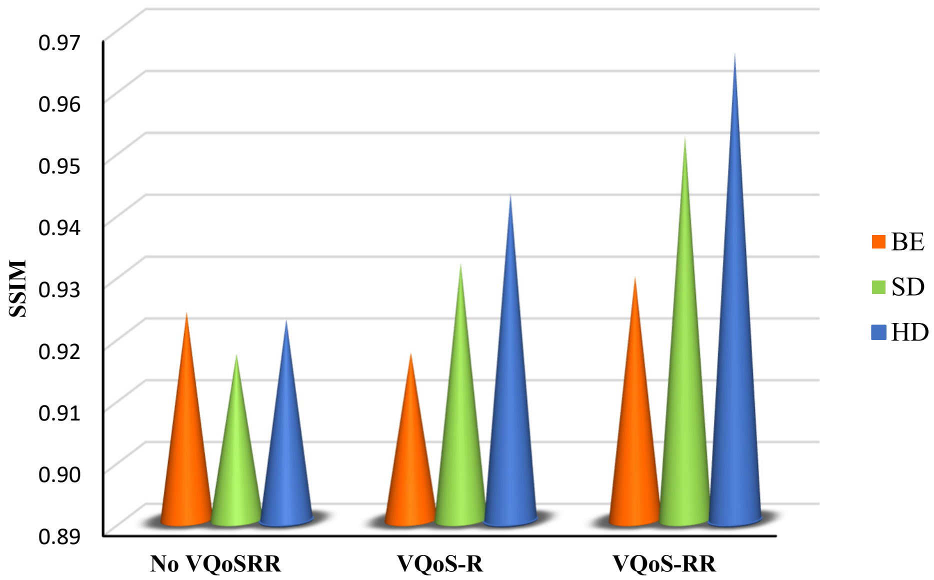
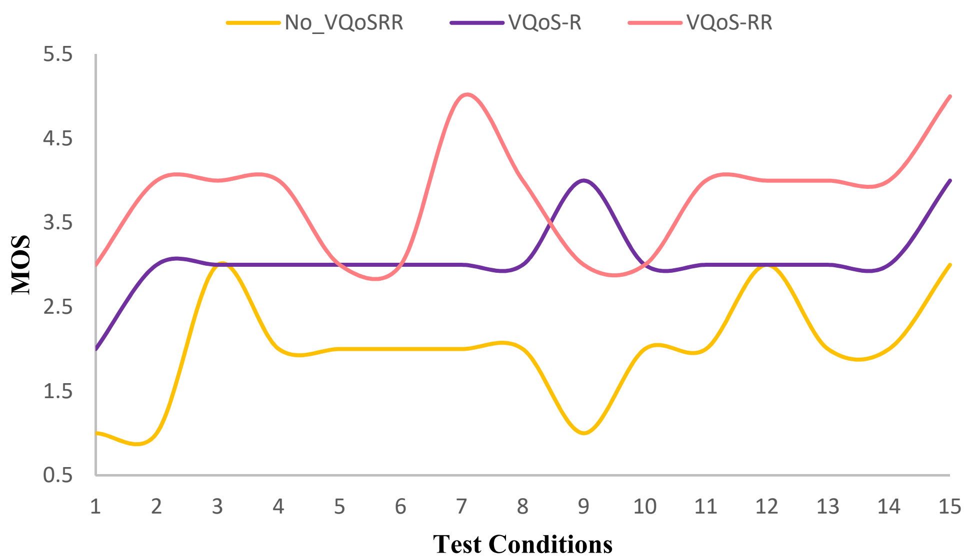


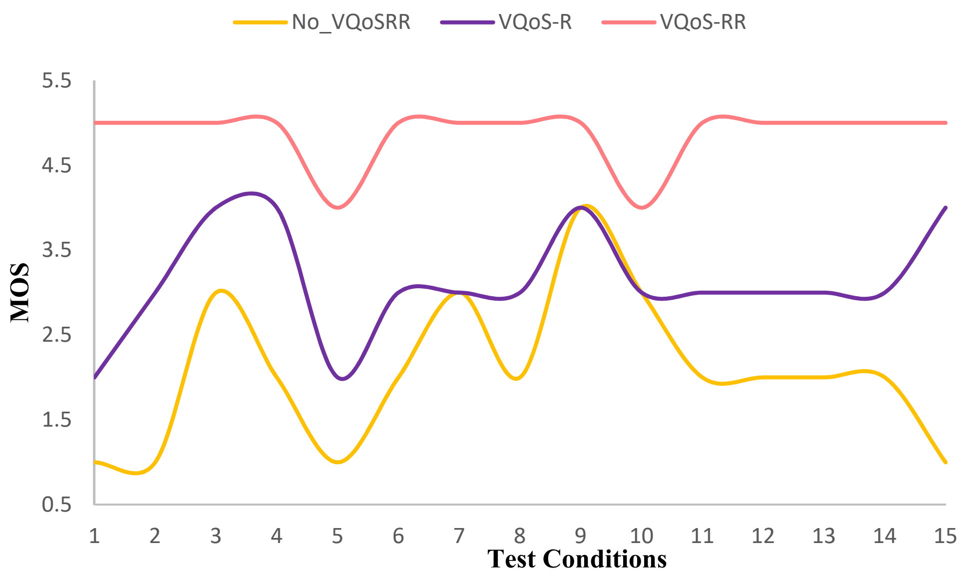
Figure 11. The mean SSIM value of the tested videos.

Below are the results obtained from the subjective QoE tests, where Figures 12 and 13 represent the MOS scores rated by users of 15 test conditions for each test case; these measurements were based on MOS scores (1–5).



(**a**)

Figure 12. *Cont.*



(**b**)

Figure 12. The MOS values of the tested videos. (**a**) SD resolution; (**b**) HD resolution.

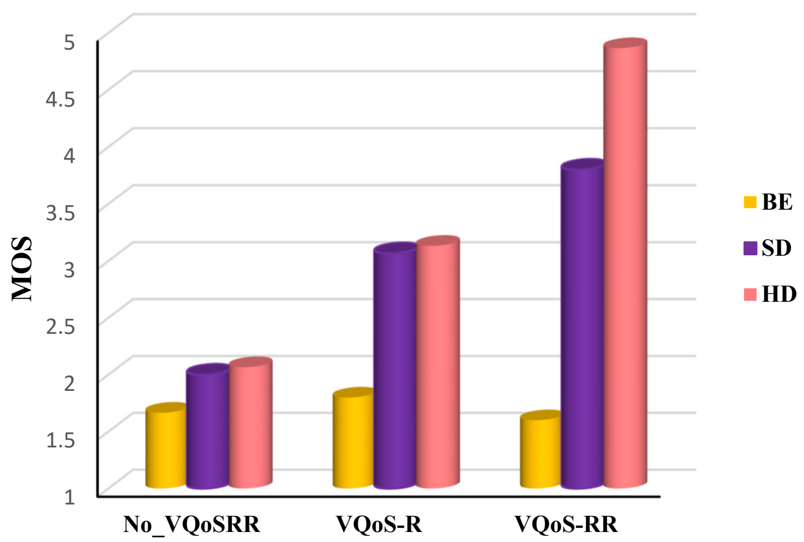
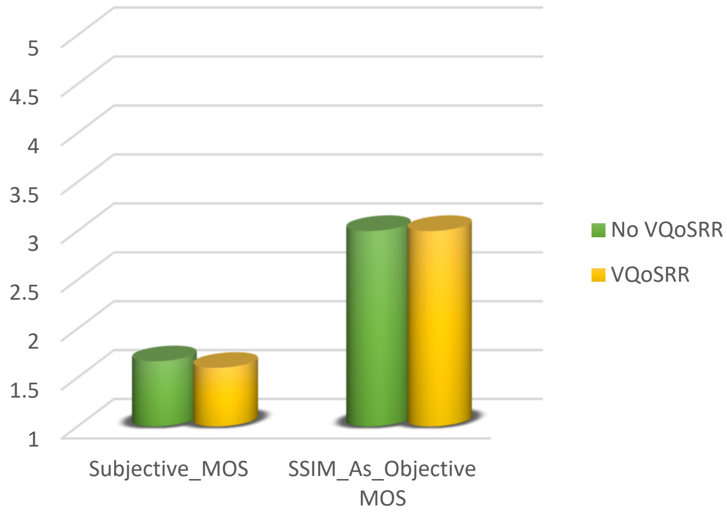


Figure 13. The average MOS value of the tested videos.

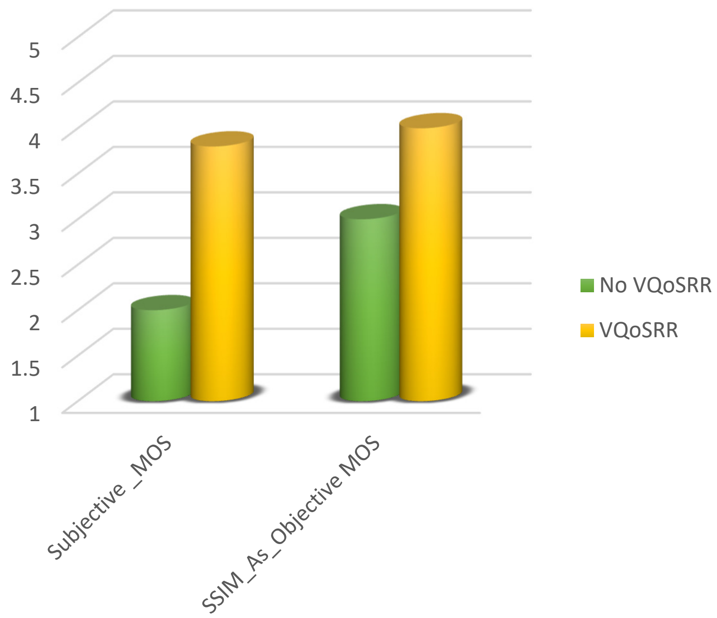
Figure 12a,b present a comparison for SD and HD videos with the MOS scores, respectively. As observed from the figures, the distortion caused by the packet loss affected the measured MOS, where the values achieved without the VQoSRR were significantly lower than those measured compared to the VQoS-R and VQoS-RR. Take, for instance, Figure 12a: on average, MOS without the VQoSRR was about 40% of 5 compared to 61.3% for VQoS-R, and 76% for VQoS-RR, respectively. In addition, Figure 12b presents a similar pattern for HD, where VQoS-RR scored 97.3%, and VQoS-R was 62.7, compared to No_VQoSRR, which achieved 41.3%.

Figure 13 shows the MOS average values calculated to measure the quality difference between the three test cases. The MOS measured for No_VQoSRR reached 1.6 for the best effort video, and 2 for SD and 2.1 for HD, which all these values indicate a poor quality video. In terms of VQoS-R, BE MOS was 1.8 (poor quality), and SD MOS increased from 2 (poor quality) to 3 (fair quality), as well as an HD improvement from 2.1 (poor quality) to 3.1 (fair quality). While in VQoS-RR, the MOS remained at 1.6 (poor quality) for BE, 3.8 (fair to good quality) for SD, and 4.8 (good to excellent) for HD. As we can see from the figure, the VQoS-RR improved the video quality; for example, in SD, the measured MOS increased from 40.0% for No_VQoSRR to 76.0% for VQoS-RR, which implies a 36% increase. While in HD, the measured MOS increased from 41.3% for No_VQoSRR to 97.3% for VQoS-RR, which implies a 56% increase. Therefore, not using the VQoSRR will result in low MOS, which means severe video streaming quality degradation. Additionally, this result indicates that the proposed system succeeded in achieving good user perception.

Figure 14a–c represent the correlation of the results obtained from the subjective tests against those obtained from the objective tests for the best effort (BE) video, SD video, and HD video, respectively. In this comparison, we mapped the values of objective SSIM metrics to an equivalent MOS scale (objective MOS).

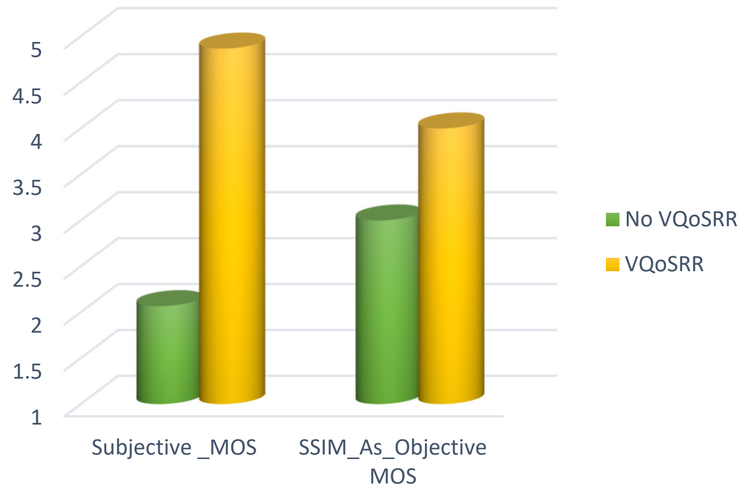


(**a**)



(**b**)

Figure 14. *Cont.*



(**c**)

Figure 14. Comparison between subjective and objective metrics QoE. (**a**) For BE class service; (**b**) for SD class service; (**c**) for HD class of service.

In Figure 14a, the subjective MOS and objective MOS for BE traffic with and without VQoSRR were slightly correlated, with subjective MOS not exceeding 1.7 and objective MOS reaching around 3. Figure 14b shows that SD in the case of VQoSRR had highly correlated MOS and SSIM results, whereas MOS without VQoSRR had a low quality compared to the objective MOS with a fair quality. Finally, the results of the HD video in Figure 14c demonstrated that there was a poor approximation between SSIM and subjective MOS in both test cases. With VQoSRR, subjective MOS scored higher than objective MOS, while without it, the opposite happened. Further, when comparing this result of HD to SD in Figure 14a, we can see that subjective QoE tests had higher MOS ratings in HD, which reflects that most users prefer watching videos at a high resolution.

Statistical Analysis

We assessed the MOS results data of the HD video from three conducted test cases using a hypothesis analysis to determine if there was any significant difference in the case of using the proposed VQoSRR versus not using it. The statistical analysis was performed using one-factor analysis of variance (ANOVA), and its results were calculated in Microsoft Excel. This test evaluates whether there are significant differences in the means of three or more independent groups. The Hypotheses for all test cases were defined as follows: hypothesis H_0: means MOS values are equal from all 3 test cases where $\mu 1 = \mu 2 = \mu 3$; hypothesis H_1: MOS means are not all equal, and the significance level p-value was set to $\alpha = 0.05$. Table 5 presents the ANOVA results. According to the p-values in Table 5, it can be seen that there was a significant difference when comparing the proposed VQoSRR to N_ VQoSRR because the p-value from the hypothesis test H was less than 0.0001, which is lower than 0.05.

Table 5. Statistical Analysis Results for MOS Metric of HD Video.

Source of Variation	Sums of Squares (SS)	Degrees of Freedom (df)	Mean Squares (MS)	F	p-Value	Remark
Between Treatments	60.6	2	30.3	75.8	<0.0001	Significant
Error (or Residual)	18.6	42	0.44			
Total	79.2	44				

4.5. Results Comparison

The video quality has been studied in two experiments (testing the DR-RA algorithm and the reservation method) using both objective and subjective metrics. These experiments showed a better perception of video quality when using the VQoSRR method than the default settings for SDN controllers when delivering video based on resolution (SD and HD) QoS thresholds. These thresholds were packet loss and bandwidth. Furthermore, the following conclusions can be drawn: (1) generally, the number of video frames received by VQoSRR is higher than the number of frames received without using it, since the proposed QBR avoids routes that have the highest loss rate; (2) a lower received quality video results in degradation in the video viewing experience. For example, the perceived video quality achieved is highest in the case of VQoSRR for all transmitted videos, even the best effort videos. However, all of them experience more degradation in video viewing in the case of No_VQoSRR. (3) Tests of DR-RA cover only short video lengths; an analysis of long-duration videos may be necessary to determine the most appropriate interval of the algorithm. In addition, changing routes frequently by DR-RA can cause other QoS problems, such as increasing the delay variation experienced by end-users. Therefore, rerouting should not be subject to frequent changes. (4) The subjective MOS metric reflects the differences in results more clearly than the objective SSIM metric; (5) MOS test results indicate that VQoSRR improves video QoE by up to 97.3% and 76% for HD and SD video resolutions, respectively, compared to not using it. (6) We recommend evaluating the effectiveness of VQoSRR's QoS scalability when there is a large amount of high-priority videos.

5. Conclusions and Future Work

The VQoSRR is a QoS and QoE-aware SDN framework for video streaming that adjusts network resources and service characteristics based on the user's needs. Because the proposed method combines the reservation mechanism with QoS-based routing, this allows the determining of which network path has adequate resources to support the requested QoS, as well as providing reserving and requesting network resources at the same time. Due to this, it is suitable for use with various applications, such as IPTV services and video surveillance in smart city environments.

We used SSIM, VMAF, and MOS QoE measurements for SD and HD videos to evaluate the proposed framework's performance. The experiments proved that VQoSRR enhanced the user perception of quality and allowed fine control over routes and resources.

From the results of our proposal, the video quality improved. For instance, for SSIM measurements, when we used VQoS-RR, the SSIM quality value increased from 0.91 to 0.95 for SD videos, whereas for HD videos, it increased from 0.92 to 0.96 (Figure 11). Likewise, in the case of MOS measurements, the VQoSRR increased the average MOS by 36.0% for SD and 56% for HD (Figure 13). We also observed that subjective MOS reflected the differences in results more clearly than the objective SSIM for comparing results obtained with the proposed model.

Additionally, we noticed from the testing results of the DR-RA algorithm that the HD achieved the highest SSIM at 7 s, while the SD had the best result at 5 s, which indicates that DR-RA should be executed at longer intervals for HD than SD in order to shift the flow to a new path.

Our experiment did not explore all possible video durations; therefore, long-duration videos may require further analyses to identify the most appropriate execution interval of the algorithm for the different video resolutions. Further, we will investigate the scalability issues in a large-scale network environment in future work. Additionally, we will integrate the proposed model with the multicast transmission. This technique of using video streaming would improve the tools and applications used for smart cities such as surveillance systems for hospitals and civil defense organizations.

Author Contributions: Methodology, M.O.E.; validation, M.O.E., and J.L.; investigation, M.O.E., A.A., N.A., J.L.; writing—original draft preparation, M.O.E.; writing—review and editing, A.A., N.A.

and J.L.; supervision, J.L.; funding acquisition, A.A. and N.A. All authors have read and agreed to the published version of the manuscript.

Funding: Researchers Supporting Project number (RSP-2021/295), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cisco. Visual Network Index (VNI) Complete Forecast Highlights. 2016, p. 3. Available online: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecasthighlights/pdf/Global_2021_Forecast_Highlights.pdf (accessed on 12 October 2021).
2. Wang, Z.; Crowcroft, J. Quality-of-service routing for supporting multimedia applications. *IEEE J. Sel. Areas Commun.* **1996**, *14*, 1228–1234. [CrossRef]
3. Crawley, E.; Nair, R.; Rajagopalan, B.; Sandick, H. *A Framework for QoS-Based Routing in the Internet*; Network Working Group: Washington, DC, USA, 1998. [CrossRef]
4. Masip-Bruin, X.; Yannuzzi, M.; Domingo-Pascual, J.; Fonte, A.; Curado, M.; Monteiro, E.; Kuipers, F.; Van Mieghem, P.; Avallone, S.; Ventre, G.; et al. Research challenges in QoS routing. *Comput. Commun.* **2006**, *29*, 563–581. [CrossRef]
5. Pokhrel, J. *Intelligent Quality of Experience (Qoe) Analysis of Network Served Multimedia and Web Contents*; Institut National des Télécommunications: Paris, France, 2015.
6. Lloret, J.; Garcia, M.; Atenas, M.; Canovas, A. A QoE management system to improve the IPTV network. *Int. J. Commun. Syst.* **2011**, *24*, 118–138. [CrossRef]
7. Elbasheer, M.O.; Aldegheishem, A.; Lloret, J.; Alrajeh, N. A QoS-Based routing algorithm over software defined networks. *J. Netw. Comput. Appl.* **2021**, *194*, 103215. [CrossRef]
8. Ghalwash, H.; Huang, C.-H. A QoS Framework for SDN-Based Networks. In Proceedings of the 4th IEEE International Conference on Collaboration and Internet Computing, CIC, Philadelphia, PA, USA, 18–20 October 2018. [CrossRef]
9. Egilmez, H.E.; Civanlar, S.; Tekalp, A.M. An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks. *IEEE Trans. Multimed.* **2012**, *15*, 710–715. [CrossRef]
10. Egilmez, E.H.; Dane, T.S.; Bagci, T.K.; Tekalp, M.A. OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks. In Proceedings of the 2012 Conference Handbook—Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC, Hollywood, CA, USA, 3–6 December 2012; IEEE: Red Hook, NY, USA, 2012.
11. Egilmez, H.E.; Gorkemli, B.; Tekalp, A.M.; Civanlar, S. Scalable video streaming over OpenFlow networks: An optimization framework for QoS routing. In Proceedings of the IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011. [CrossRef]
12. Sendra, S.; Rego, A.; Lloret, J.; Jimenez, J.M.; Romero, O. Including artificial intelligence in a routing protocol using Software Defined Networks. In Proceedings of the IEEE International Conference on Communications Workshops, Paris, France, 21–25 May 2017; pp. 670–674.
13. Yu, Y.-S.; Ke, C.-H. Genetic algorithm-based routing method for enhanced video delivery over software defined networks. *Int. J. Commun. Syst.* **2017**, *31*, e3391. [CrossRef]
14. Parsaei, M.R.; Boveiri, H.R.; Javidan, R.; Khayami, R. Telesurgery QoS improvement over SDN based on a Type-2 fuzzy system and enhanced cuckoo optimization algorithm. *Int. J. Commun. Syst.* **2020**, *33*, e4426. [CrossRef]
15. Henni, D.E.; Ghomari, A.; Hadjadj-Aoul, Y. A consistent QoS routing strategy for video streaming services in SDN networks. *Int. J. Commun. Syst.* **2019**, *33*, e4177. [CrossRef]
16. Volpato, F.; Da Silva, M.P.; Goncalves, A.L.; Dantas, M.A.R. An Autonomic QoS management architecture for Software-Defined Networking environments. In Proceedings of the IEEE Symposium on Computers and Communications, Heraklion, Greece, 3–6 July 2017. [CrossRef]
17. Sharma, S.; Staessens, D.; Colle, D.; Palma, D.; Goncalves, J.; Figueiredo, R.; Morris, D.; Pickavet, M.; Demeester, P. Implementing quality of service for the software defined networking enabled future internet. In Proceedings of the 2014 3rd European Workshop on Software-Defined Networks, EWSDN 2014, Budapest, Hungary, 1–3 September 2014. [CrossRef]
18. Seddiki, M.S.; Shahbaz, M.; Donovan, S.; Grover, S.; Park, M.; Feamster, N.; Song, Y.-Q. FlowQoS. In Proceedings of the Third Workshop on Hot Topics in Software Defined Networking, Chicago, IL, USA, 4 August 2014. [CrossRef]
19. Khater, A.; Hashemi, M.R. Dynamic Flow Management Based on DiffServ in SDN Networks. In Proceedings of the 6th Iranian Conference on Electrical Engineering (ICEE), Mashhad, Iran, 8–10 May 2018. [CrossRef]
20. Rego, A.; Garcia, L.; Sendra, S.; Lloret, J. Software Defined Network-based control system for an efficient traffic management for emergency situations in smart cities. *Future Gener. Comput. Syst.* **2018**, *88*, 243–253. [CrossRef]
21. Canovas, A.; Rego, A.; Romero, O.; Lloret, J. A robust multimedia traffic SDN-Based management system using patterns and models of QoE estimation with BRNN. *J. Netw. Comput. Appl.* **2019**, *150*, 102498. [CrossRef]
22. Xu, C.; Chen, B.; Qian, H. Quality of Service Guaranteed Resource Management Dynamically in Software Defined Network. *J. Commun.* **2015**, *10*, 843–850. [CrossRef]

23. Ii, H.O.; Durresi, A. Video over Software-Defined Networking (VSDN). *Comput. Netw.* **2015**, *92*, 341–356. [CrossRef]
24. Yan, J.; Zhang, H.; Shuai, Q.; Liu, B.; Guo, X. HiQoS: An SDN-based multipath QoS solution. *China Commun.* **2015**, *12*, 123–133. [CrossRef]
25. Braden, R.; Clark, D.; Shenker, S. RFC1633: Integrated Services in the Internet Architecture: An Overview. Request for Comments (RFC) 1633 (IETF). 1994. Available online: www.rfc-archieve.org (accessed on 10 February 2020).
26. Blake, S.; Black, D.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W. An Architecture for Differentiated Services. RFC 2475. 1998. Available online: https://www.hjp.at/doc/rfc/rfc2475.html (accessed on 3 March 2020).
27. Rosen, E.; Rekhter, Y. RFC2547: BGP/MPLS VPNs. RFC 2547. March 1999. Available online: https://www.rfc-archive.org/getrfc?rfc=2547 (accessed on 20 February 2020).
28. Gstreamer. 2020. Available online: https://gstreamer.freedesktop.org/ (accessed on 15 June 2020).
29. Babiarz, J.; Chan, K.; Baker, F. *Configuration Guidelines for DiffServ Service Classes Status*; Network Working Group: Washington, DC, USA, 2006.
30. Floodlight. 2020. Available online: https://floodlight.atlassian.net/wiki/spaces/floodlightcontroller/overview (accessed on 8 June 2021).
31. Lantz, B.; Heller, B.; McKeown, N. A network in a laptop: Rapid prototyping for software-defined networks. In Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets-9, Monterey, CA, USA, 20–21 October 2010. [CrossRef]
32. OpenVSwitch. Available online: https://www.openvswitch.org/ (accessed on 27 January 2021).
33. Iperf. TCP/UDP Bandwidth Measurement Tool. 2021. Available online: https://iperf.fr/ (accessed on 6 November 2021).
34. Caminandes. 2021. Available online: http://www.caminandes.com/ (accessed on 19 May 2021).
35. MSU Video Quality Measurement Tool. 2021. Available online: https://www.compression.ru/video/ (accessed on 2 September 2021).
36. Wang, Z.; Lu, L.; Bovik, A.C. Video quality assessment based on structural distortion measurement. *Signal Process. Image Commun.* **2004**, *19*, 121–132. [CrossRef]
37. Li, Z.; Aaron, A.; Katsavounidis, I.; Moorthy, A.; Manohara, M. Toward a Practical Perceptual Video Quality Metric. *Netflix Blog*. 2006, 6. Available online: http://techblog.netflix.com/2016/06/towardpractical-perceptual-video.html (accessed on 25 August 2020).
38. ITU-T RECOMMENDATION. Subjective Video Quality Assessment Methods for Multimedia Applications. 2008. Available online: https://www.itu.int/rec/T-REC-P.910 (accessed on 15 July 2021).