2018-07-27

A Novel QoE-Centric SDN-Based Multipath Routing Approach for Multimedia Services over 5G Networks

Alcardo Alex Barakabitze

Lingfen Sun *School of Engineering, Computing and Mathematics*

Is Haka Mkwawa

Emmanuel Ifeachor *School of Engineering, Computing and Mathematics*

A Novel QoE-Centric SDN-based Multipath Routing Approach for Multimedia Services over 5G Networks

Alcardo Alex Barakabitze, Lingfen Sun, Is-Haka Mkwawa and Emmanuel Ifeachor
School of Computing, Electronics and Mathematics, Plymouth University, UK
E-mail: {alcardoalex.barakabitze, L.Sun, is-haka.mkwawa, E.Ifeachor}@plymouth.ac.uk

***Abstract*—The explosion of enhanced applications such as live video streaming, video gaming and Virtual Reality calls for efforts to optimize transport protocols to manage the increasing amount of data traffic on future 5G networks. Through bandwidth aggregation over multiple paths, the Multi-Path Transmission Control Protocol (MPTCP) can enhance the performance of network applications. MPTCP can split a large multimedia flow into subflows and apply a congestion control mechanism on each subflow. Segment Routing (SR), a promising source routing approach, has emerged to provide advanced packet forwarding over 5G networks. In this paper, we explore the utilization of MPTCP and SR in SDN-based networks to improve network resources utilization and end-user's QoE for delivering multimedia services over 5G networks. We propose a novel QoE-aware, SDN-based MPTCP/SR approach for service delivery. In order to demonstrate the feasibility of our approach, we implemented an intelligent QoE-centric Multipath Routing Algorithm (QoMRA) on an SDN source routing platform using Mininet and POX controller. We carried out experiments on Dynamic Adaptive video Steaming over HTTP (DASH) applications over various network conditions. The preliminary results show that, our QoE-aware SDN-based MPTCP/SR scheme performs better compared to the conventional TCP approach in terms of throughput, link utilization and the end-user's QoE.**

***Keywords*—SDN, Quality of Experience (QoE), Segment Routing (SR), Multiple Path TCP (MPTCP), 5G, Multimedia Applications.**

I. Introduction

Future networks (e.g., 5G) are expected to support and provide an End-to-End over-the-air latency of less than $1ms$, transmission reliability of 99.999% and approximately 100% services availability [1]. 5G is set to provide access to any service with better quality to end-users at anytime, anywhere through reliable and cost effective communications, over any medium and across multi-operator domains using different technologies, such as Software Defined Networking (SDN). Reliable transmission protocols with high transmission efficiency in wireless environments are required in order to support a multi-variety of services (e.g, live video streaming and video gaming) in 5G networks. Requirements such as high throughput, resilience and reliability, consistency and service availability, ultra low-latency have to be archived, for 5G to support these applications [1]. Efficient transfer of large data, especially multimedia traffic flows is crucial to the performance of 5G multi-rooted topology networks where multiple paths exist between pair of hosts.

In order to provide an optimal throughput for large flows and cope with the explosive growth of multimedia traffic, new transfer protocols have been designed. The aim has been to increase network reliability, availability and make a timely response to end-user's requests as well as improving end-user's Quality of Experience (QoE).

Multi-Path TCP (MPTCP) is a recent effort of the Internet Engineering Task Force (IETF) towards standardizing a transport layer protocol that improves network connectivity and provides end-users with higher data rates. MPTCP was proposed as an extension of TCP with the main idea of splitting a large flow into multiple subflows. The subflows originating from the same MPTCP connection are then sent to their destinations through different disjointed paths in the network [2]. MPTCP is a promising approach to improve performance and reliability of video streaming services in 5G networks. The only limitation of MPTCP, is the burden of maintaining large number of subflows which may cause overhead since it has no control over the network or transmission routes used by each subflow from source to destination. Controlling the number of subflows using emerging technologies such as SDN is crucial to the performance of this transport protocol [3].

However, as an implementation for route look-up, packet classification and fast packet forwarding, SDN switches are integrated with a ternary content-addressable memory (TCAM) which practically supports a limited size of 2k-20k rules [4]. SDN-based MPTCP solutions require a larger number of flow rules to be installed in each of the SDN switches along the delivery path and thereby increasing the load at the SDN switches and the controller. The requirement of more TCAM resource consumption can be eliminated by Segment Routing (SR) [5], a new approach that greatly reduces the number of forwarding rules by encoding routing information into packet header as an ordered list of labels. SR improves the scalability problem of SDN since there is no path state maintenance required in each switch or router along the service delivery path.

In this paper, we propose a novel QoE-aware, SDN-based MPTCP approach for 5G networks that dynamically controls the number of subflows and uses the available network resources efficiently by forwarding the network traffic through multiple disjointed paths. Taking the fact that, TCAM is an extremely expensive resource to build and consumes a lot of power, we make joint use of MPTCP and SR to reduce the deployment cost, increase flexibility, reliability and scalability as well as improving the end-users QoE in SDN-based virtualized IP networks.

The contributions of this paper are three-fold,

- We present promising implementation mechanisms of MPTCP and SR as a solution for Traffic Engineering (TE) management in 5G SDN-based networks.
- we propose a QoE-aware SDN-based MPTCP/SR framework to enhance the QoE for multimedia services delivery over 5G networks.
- We further propose an intelligent QoE-centric Multipath Routing Algorithm (QoMRA) that dynamically controls the number of subflows using an SDN controller and performs source routing to these subflows using SR paradigms.

The proposed QoE-aware MPTCP/SR scheme was implemented in an SDN testbed based on Mininet and POX controller. The Dynamic Adaptive Video Streaming over HTTP (DASH) service was applied to test the performance of the proposed scheme and compared with the conventional TCP approach. DASH server was based on Apache and DASH client was based on VLC-DASH plugin [6]. Video clips were segmented based on GPAC MP4Box [7] with five video representations. Preliminary results show that, our QoE-aware SDN-based MPTCP/SR scheme performs better compared to the conventional TCP approach in terms of throughput, link utilization of above 50%. Our approach also greatly reduces the number of video quality switches, startup delays and therefore improving the end-users QoE.

The rest of this paper is organized as follows: Section II provides related work. Section III presents backgrounds on MPTCP and SR and their collaborative implementations for TE management in 5G SDN-based networks. Section IV describes the proposed QoE-aware, SDN-based, MPTCP/SR controller architecture along with our intelligent QoE-centric multipath routing algorithm (QoMRA). Section V presents the implementation and performance evaluation of the proposed system. Finally, Section VI concludes the paper.

II. Related Work

To pinpoint out our motivation, we first discuss related work that implement MPTCP using SDN and then explore the applicability of SR for traffic engineering solutions in 5G SDN-based networks. In [2] a proposal that uses MPTCP in OpenFlow-enabled networks to guarantee subflows of the same MPTCP connection to be sent through disjointed paths is given. A QAMO-SDN, QoS-aware MPTCP-based solution that can provide and support service differentiation in SDN optical networks is presented in [8]. The concept of SDN is invoked in [9] to establish a large-scale measurement framework on top of GEANT and PlanetLab Europe where multiple paths can be configured dynamically, and MPTCP experiments can be orchestrated in a flexible manner. A recent MPTCP-aware SDN controller that uses packet inspection to provide deterministic subflow assignment to paths and facilitates an alternative routing mechanism for the MPTCP subflows is described in [3]. Faster download speed and improved QoE is reported in [10], where SDN is used to add or remove MPTCP paths in order to reduce large number of out-of-order packets which may cause poor performance and degrade the end-users' QoE. A responsive MPTCP approach that employs a centralized SDN controller for calculating the forwarding paths of subflows is presented in [11].

Although these SDN MPTCP-based approaches are intended to improve throughput, but they rely on installing a larger number of flow rules at the SDN switches and thereby increasing the load at the SDN switches. Such limitation can be easily eliminated using the SR technology as we show from section III A in this paper. Another drawback of the above approaches, is the lack of an efficient routing and an adaption mechanism to control the number of subflows based on the network changing conditions. In addition, some of proposals do not consider on how MPTCP can improve QoE in the aspects of future IP networks for media-rich services delivery.

While recent proposals, [12], [13] attempt to achieve the optimal quality of real-time video streaming using MPTCP, authors do not consider the global control features that SDN can offer in their implementations. Such solutions are likely to suffer from congestions when large flows are transfered from source to destinations and therefore degrading the end-user's QoE. Indeed, SDN-based SR implementations can efficiently manage resources and provide better TE solutions in multi-domain 5G networks [5]. For example, an SDN-based solution for assigning TE paths with SR, based on Multi-Protocol Label Switching (MPLS) forwarding is proposed in [14] while efficient routing mechanisms for SDN with SR that meet bandwidth requirements of routing requests are described in [15]. Motivated by previous observations of SR to reduce the amount of forwarding rules in OpenFlow switches, this paper, takes advantages of MPTCP and SR to speed up network transmission, increase network throughput, reduce overhead and further improve user's QoE for multimedia services in SDN-based networks.

III. A Multiflow SDN-based Approach

A. A MPTCP over SDN-based Networks

As shown in Fig 1, suppose that, an MPTCP connection consists of 3 subflows: *sf1:1*, *sf1:2*, *sf1:3* where *sf* is the subflow and the indexes indicate the number of a subflow of a particular MPTCP connection. In order to guarantee the end-users QoE-fairness level and performance requirements of 5G networks, the SDN controller checks the available capacity of all connected paths and selects the shortest paths to transmit the subflows of the same MPTCP connection. As shown in Fig 1, *sf1* will be transmitted via path, S1→S2→S3→S4, *sf2* via path S1→S7→S8→S4 and *sf3* will take path S1→S5→S6→S4.

Such paths can be expressed by the segment routing strategy into SR paths, using the segment label list. We explain the SR operation and mapping of MPTCP subflow paths into SR paths in the next subsections.

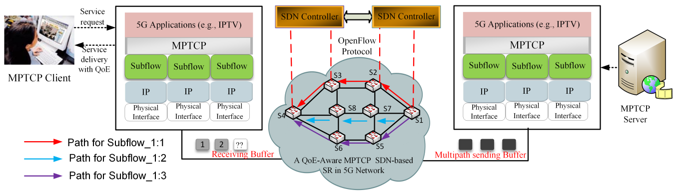

Fig. 1. A Video Streaming Scenario using MPTCP in SDN-based 5G Networks. The black boxes indicate video frames generated as data units by the application while grey boxes are the TCP packets to be received at the client side. One packet is lost and retransmitted

B. Traffic Engineering with Segment Routing in SDN

With SR, a host or an edge router is able to steer a packet through the network using an ordered list of processing/forwarding functions called segments. A segment can be a logical or a physical element such as a network node (e.g., OpenFlow Switch or router), network link or a packet filter. The SR path is formed by the chain of these elements which are identified by a list of segment identifiers (SIDs). The SIDs can be defined globally with domain wide significance such that it is recognized by all network nodes or it can be defined locally within a node processing the packet [15].

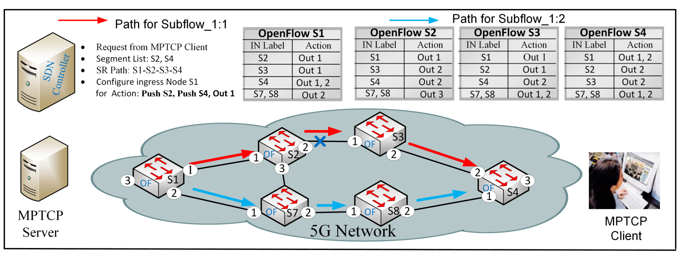

Fig. 2. Segment Routing Operation in SDN-based 5G Networks

Fig 2 illustrates an example of a SR operation in SDN-based virtualized 5G networks. In this paper, we give our attention to the centralized control plane of SR approach to implement our idea. When an MPTCP client requests a new traffic from the MPTCP server, SDN controller calculates the transmission path P_{sf1}={S1→S2→S3→S4} based on the required constraints (e.g., delay and bandwidth of a link). This path is mapped as the segment label list and returned to the ingress switch S1 which then stores it in the packet header. The segment list consists of only the label that identifies the destination node in the SR domain such that, *SL*={S4}. The forwarding table of the ingress OpenFlow switch S1 is then configured by the SDN controller with this segment list labels. The switch will modify the packet by encoding the segments as MPLS label stacks to the packet header [15].

The packets of a subflow are then forwarded from S1 to S4 through intermediate nodes without any modification to the segment list. The top label is popped when the packets arrive at the intermediate nodes. The packets are then forwarded to the destination point using the segment node that represents the next label. SR can support and provide dynamic traffic recovery in multi-domain 5G SDN-based networks during network node or link failure [5]. For example, when link S2→S3 fails, then the SDN controller which continuously monitors the network topology can redirect the traffic on the backup path S1→S2→S7→S8→S4} with the associated *SL*={S2,S4}. At node S2, the top label is popped and the packet is transmitted to S7 via its Adjacency SID 3. At S7, the packet follows a path to its destination using its Adjacency SID at interface 2.

C. Mapping of Subflows Paths to SR Paths

Consider a multimedia flow that is to be requested by the MPTCP client from the MPTCP server. Given the link capacity, l_c and the video bit rate b_r, our first task is to split a large flow into subflows and find the multiple disjointed paths for the subflows. The second task is to map these subflows paths into SR paths and perform source routing based on QoS/QoE requirements. In order to achieve the first task, a MPTCP-based flow manager as described in the next subsection is introduced as a module running in the SDN controller. The mapping of subflow paths to SR paths is performed by finding the shortest list of SIDs that allows the packet to follow a given route, the detailed implementation will be covered in the next section.

IV. QoE-aware MPTCP SDN-based SR adaptation Framework

The framework consists of the following modules: The MPTCP-flow manager, SR, QoE Management, Databases, Network information collector and the Configuration module which are integrated within the SDN controller.

A. SDN Controller

Fig 3 shows the function modules and interfaces of the proposed QoE-aware, MPTCP SDN-based SR framework. When the SDN controller receives a new request from the MPTCP client, it performs the path calculation through the MPTCP module and allocates the subflows to transmission paths. It then maps these subflow paths to SR paths as described in the previous subsection. The SR path of the new subflow is stored in the database. The SDN controller allocates and configures the SR path to the ingress switch node using the segment label list. The SDN controller communicates with switches in the data plane through the SouthBound Interface (SBI), using the OpenFlow protocol [2]. The OpenFlow protocol describes message exchanges that take place between the controller and an OpenFlow switch. When a heavy flow arrives at the switch, the Packet-In Message is sent to the SDN controller. These messages are sent during state change or flow setup to reduce the CPU utilization and control traffic in OpenFlow switches and SDN controllers [9]. The Packet-Out Message enables the controller to manage the logical state of the switch, including its configuration and details of flows.

The controller maintains the database where all SR subflow paths are stored with their QoE requirements. When a new subflow of a MPTCP connection is uploaded to the SDN controller, the module queries in the database if there exists

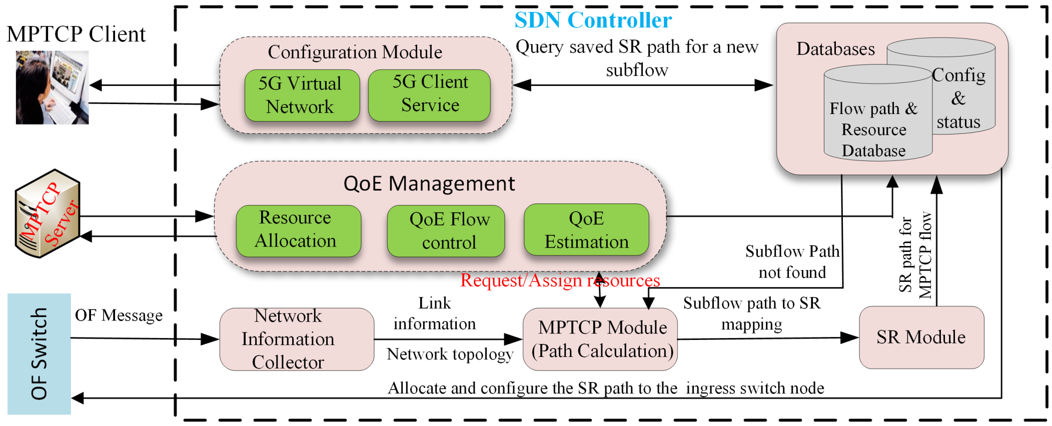

Fig. 3. A QoE-aware, MPTCP SDN-based SR Architecture

a path corresponding to this MPTCP connection. If the path exists, the subflow is allocated to a specific path of a previously assigned subflow's path of the same MPTCP connection, otherwise a new path computation for this subflow is performed using the MPTCP-flow manager/module. The new path is mapped to the SR paths and stored in the database so that it can be used later by subflows of the same MPTCP connection. The proposed SDN controller architecture can provide network programmability capabilities which allows third parties (e.g., 5G virtual network operators) and other players in the QoE provisioning chain to set-up their specific QoS/QoE control strategies.

1) MPTCP-Flow Manager: This module computes the shortest paths and then performs path allocations to the subflows (see Fig 3). In order to minimize the influences of link congestion to data transmission quality and the end-user's QoE, the MPTCP module employs admission control for media delivery, such that, the flow is admissible only if on each link, the sum of the rates of the allocated subflows does not exceed the link capacity (l_c). In that case, the MPTCP module dynamically controls the number of generated subflows at the ingress source node. Based on the collected link information, the MPTCP module communicates with the QoE management module so that resources can be assigned to the calculated paths of subflows to meet their QoE requirements.

Instead of installing these subflow paths in SDN switches as forwarding rules, we rely on SR approach where the forwarding table of the ingress switch is configured with an ordered list of segments [14]. The ingress switch then adds labels with an ordered list of segments to a packet header and forwards it to its destination point. This novel approach improves greatly the scalability, avoids link congestions and enables the deployment of effective multi-domain TE solutions in SDN-based 5G networks at a minimum cost.

2) Segment Routing Module: The computed subflow paths by the controller using the MPTCP-based flow manager have to be mapped to SR paths. We introduce the segment routing module that enhances greatly the SDN controller such that future 5G applications can exploit its features through the NorthBound Interface (NBI). We consider the SR assignment algorithm presented in [14] which finds the minimal-length of SR paths to be mapped to the subflow paths. The SR path is the shortest list of SIDs that allows the packet to follow a given route. The list of SIDs consists of network nodes (e.g., node segments) or specific interface node (e.g., adjacency node). The source codes of this algorithm can be found at [14]. For example, given a multimedia flow f requested by an MPTCP client whose ingress node is OpenFlow switch $S4$ and the egress node is switch $S1$, then the complete path for subflow $sf1$ with intermediate nodes $\{S_2,...S_{N-1}\}$ is: P_{sf1}={S1→S2→S3→S4}. With reference to Fig 2, the algorithm takes this path and the graph of the topology as inputs. It maps the specific path of the subflow and returns the segment list as an output of the assigned SR paths. In this case, the ingress switch node $S1$ of the subflow path and the egress switch $S4$ are first considered. We make an assumption that, all links may have different cost. If there exists only one shortest path and it equals to the subflow path from $S1$ to $S4$, then the engress node is used as the SID node from $S1$ to $S4$ and the mapping of this subflow path to SR path ends (see Fig 2). When a link failure occurs or it happens that the shortest path to be mapped is not equal to the subflow path or there exists more than one equal-cost shortest path, then other node segments can be considered.

3) QoE-Management Module: Based on the specific QoE requirements for a service as defined at the network level QoS parameters, the QoE management module is to manage network resources dynamically for a certain multimedia flow application. Our proposed solution can handle various sessions of multimedia applications through efficient QoE-centric control-based routing algorithms that redistribute the available network resources among the competing flows according to the end-user QoE requirements.

B. QoE-Centric Multipath Routing Algorithm (QoMRA)

The QoMRA is implemented on an SDN source routing platform using Mininet and POX controller. The aim of QoMRA is to enable service providers to route network traf-

TABLE I
REQUIRED TCP THROUGHPUT FOR QOE-CENTRIC ALGORITHM

Video resolution	Required TCP Throughput (Mbps)
1080p	5
720p	2.5
480p	1.0
360p	0.725
240p	0.325

fic through explicit multiple disjointed bandwidth-satisfying paths that meet specific service QoE guarantees. Based on clients request, QoMRA should maintain an efficient use of network resources, reduce the potential for rejecting traffic-flow demands and network congestion. We define the traffic-flow demand in this paper as the request from the source $s \in$V to destination $d \in$V with requested TCP bandwidth B_w. In order to select k multiple shortest paths for the subflows from source to destination, the algorithm considers important information for video streaming such as the available link's bandwidth, delay, packet loss rate, jitter and link criticality. The link criticality parameter $c(l)$ is used for predicting the future traffic load of subflows on the link e. We use $c(l)$ [15] to improve the routing performance based on the current load of each link e. The $c(l)$ is computed and updated by the SDN controller after every time interval t. The intent of $c(l)$ is to choose the links that can balance the loads across the network and avoid bottleneck links between pairs of communication nodes. The link criticality $c(l)$ can be defined using equation 1.

$$c(l) = \sum_{(s,d) \in TD} P_{sd}(c)/k \tag{1}$$

P_{st}(c)/k is the occurrence rate of link e in the first k shortest paths whereas TD is the traffic demand that is recorded and stored in the database in every time t. We define a link as a bottleneck when it provides a data rate that is lower than the required bitrate of a subflow to be transmitted through it. Table I shows the required TCP throughput for a specific video resolution which is selected by QoMRA when an MPTCP server receives a request from an MPTCP client. In case of buffering events due to congestion or link failure the algorithm can choose another routing path for a specific subflow as described in section III A. We mitigate the link failure or congestion by introducing a link congestion index $k(i)$, a function that increases sharply when the amount of traffic flow passing through the link e approaches its capacity. The idea behind $k(i)$ is to make use of paths with less link utilization during network congestion. When the link utilization grows beyond a certain threshold, QoMRA is able to capture the congestion state based on the information provided by the link utilization. However, QoMRA causes less overhead imposed by SR and MPTCP-flow manager modules during QoS parameters measurement that are to be used for shortest paths calculations.

Once all the subflows paths are computed based on the link constraints and are assigned to SR paths, then an MPTCP connection can be established. The video player on an MPTCP client can then start to report the QoE metrics as the video streaming continues. We give the overall steps of our proposal in Algorithm 1.

Algorithm 1: A QoE-Centric Multipath Routing Algorithm (QoMRA)

input : flow f, number of subflows *sfi*:j, *N_topology*

1 *Compute link delay, bandwidth, packet loss, jitter for all paths*;
2 *Find all subflow shortest paths* $p \in$*P in the network based on Step 1*;
3 **foreach** $p \in$*Psrc*→*dst* **do if** *p.used* +*sfi*:$j.b_r < l_c$ **then return** p ;
4 **else** Go to step 3;
5 Map all subflow shortest paths p_{sf} into SR paths
output: List of Segment labels *SL*
6 Save path p with its associated list of *SL* in DB_p ;
7 Initiate the MPTCP transmission of flows based on their QoE requirements
8 **if** f_{new} *is a new flow* **then** Query DB_p to find the paths for subflows of f_{new};
9 **if** pf_{new} *is not in* DB_p **then** Go to step 2 and issue path adding request for f_{new};
10 **end if**;
11 **end if**;
12 **end for**;
13 Continue transmission as long as the l_c is greater or equal to the required TCP bandwidth;
14 Use the link criticality and congestion index to avoid congestion

V. IMPLEMENTATION AND EVALUATION

A. Experimental Setup

In order to demonstrate the feasibility of our approach, two VMs installed with Linux (Ubuntu V16.04 LTS) and MPTCP v0.92 were used. The Mininet V2.2.2 was installed in one VM and used to model a scenario of a 5G datacenter network. As shown in Fig 4, our topology consists of three levels with 8, 4 and 2 OpenFlow switches at the edge, aggregation and core layers, respectively. One MPTCP client is attached at each access switch. The network consists of redundant links at each level in order to accommodate subflows splitted by the MPTCP module. We extended the POX controller running in the second VM with an implementation of MPTCP-flow manager and the SR module. The SR module was implemented following a customization of source codes available at [14]. For effective evaluation of our proposal, both TCP and MPTCP are supported in our testbed. We installed Apache Server on two MPTCP server machines and attached them on mininet network. The DASH client was based on VLC-DASH plugin [6]. Before starting our experiments, we make sure that client and server sides support MPTCP. Throughout our experiments, we use the video sequence "Big Buck Bunny" with a resolution of 1920x1080 pixels of 9 minutes and 56

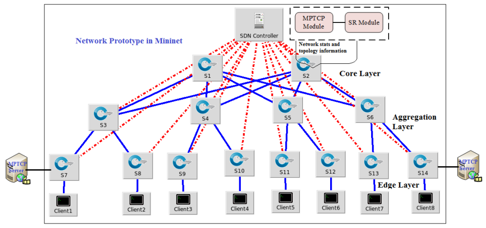

Fig. 4. Experimental Testbed

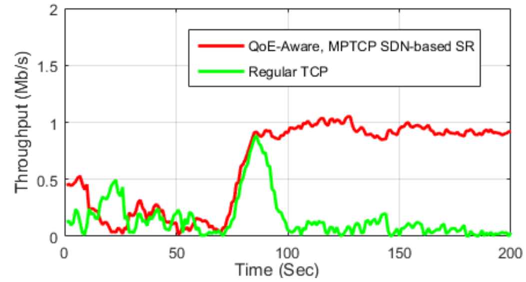

Fig. 5. Comparison of Throughput of QoE-aware MPTCP SDN-based SR and regular TCP

seconds. 5 representations of the selected video sequence with a minimum of 325Kb/s to a maximum of 5Mb/s were encoded using *ffmpeg* version 3.3.4 with the *libx265* and segmented based on GPAC MP4Box [7]. The video clips were then stored in the Apache Server. Such representations reflect the recommended and a typical video streaming rates used by YouTube. The video was further segmented into segments of sizes 2, 4, 6, 8 and 10 seconds. For a good compromise between encoding efficiency and flexibility for stream adaption to bandwidth changes, we use a segment size of 6 sec to compare the throughput, link utilization, quality switches and startup delays of the two approaches. We set the bandwidth and delay limits of (1Mb/s, 15μs), (2Mb/s, 20μs) and (3Mb/s, 30μs) for links in the edge, aggregation and core layers respectively. We considered a packet loss rate of ($<3\%$) at the core and aggregation links and set the edge links free from packet loss. When the MPTCP is enabled, we keep the default configurations of MPTCP V0.92 and configure the MPTCP path manager to a *full-mesh* that creates multiple subflows for each pair of IP addresses. This way, we limit each MPTCP connection to have only 3 MPTCP subflows. The video transmission is repeated 40 times from MPTCP client to MPTCP server. We compare our proposal with the MPTCP and regular TCP in terms of throughput, link utilization and the end-users-QoE.

B. Experimental Results

1) System Throughput: Throughput is the total size of successfully transferred data in a temporal time interval. It is calculated based on a **Payload_bits/download_time**; where a *Payload_bits* indicates the number of extracted bits of the video content per single unit time. Fig 5 shows the system throughput when an MPTCP client issues a video request from the server. We observe that, between 0-50sec, the regular TCP and our QoE-aware approach give almost similar performance. As the transmission of a video stream continues, the background traffic was introduced in the network using the *Iperf* tool. We note that, our QoE-aware MPTCP SDN-based SR approach exploits multipath and segment routing features to increase the system throughput. Conversely, the regular TCP that uses single path for transmission performs poorly. It is also clear from Fig 5 that, the throughput of a regular TCP drops from 100sec. At this point, the user experiences rebuffering events due to throughput reduction problem while our approach maintains a good viewing experience to the user.

2) Link Utilization: The link utilization is computed from the amount of used bandwidth by a video content in a particular link over the total link capacity. Fig 6 compares the link utilization for the two approaches. Our QoE-aware proposal can maintain the link utilization above 50% for the whole duration of a video streaming due to multiple paths transmission using MPTCP and SR. As the transmission continues, the link utilization of a regular TCP starts to drop at 300sec. This is so because it uses a single path for data transmission such that when the requests increases, the load of OpenFlow switches also increases making TCP unable to react quickly to congestion in the link.

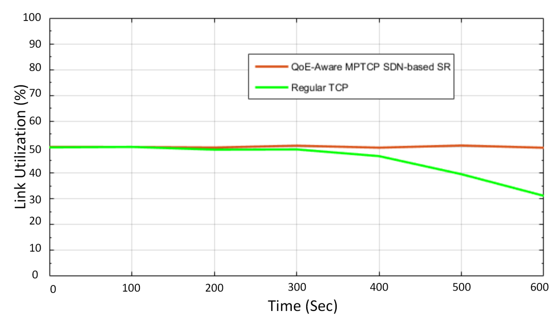

Fig. 6. Comparison of Link Utilization of QoE-aware MPTCP SDN-based SR and regular TCP

3) The Impact of Segment Length on Throughput: Fig 7 shows the impact of segment length of 2, 4, 6, 8 and 10sec on media throughput. Due to the influence of the network delay and the overhead produced by the client requests, the regular TCP performs poorly compared to our QoE-aware MPTCP SDN-based SR approach. The throughput of a segment duration with 6 seconds shows an improved performance compared to that shown in Fig 5 because there is no background traffic imposed in this experiment. Shorter segments perform better in terms of throughput. However, when longer segments are used, the MPTCP client is unable to adjust quickly and flexibly resulting into poor performance. Although the bitrate adaptation process is enhanced for shorter segments and the buffer underflows is reduced but the server load is increased due to the fact that, segments requests are issued more often by the MPTCP clients.

TABLE II
SELECTED QOE COMPARISON RESULTS FOR REGULAR TCP AND QOE-AWARE MPTCP SDN-BASED SR

	Regular TCP			QoE-Aware MPTCP SDN-based SR		
Experiment	Quality Switches	Startup Delays (sec)	Success Rate (%)	Quality Switches	Startup Delays (sec)	Success Rate (%)
1	8	6.60	85	2	2.21	96.64
2	7	6.30	87	3	1.97	97.80
3	5	5.60	84	1	1.60	97.01

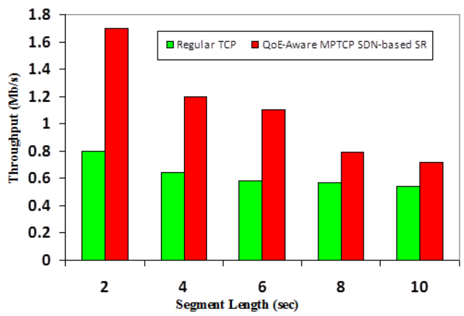

Fig. 7. The impact of segment length on throughput

4) Comparison of video quality switches and startup delays: Table II shows the selected QoE comparison results of regular TCP and our approach after performing a video request from the MPTCP server three times using a 6sec segment length. We record the startup delays, number of quality switches and the success rate of the two approaches. We define the success rate as the percentage of segments at maximum bitrate over all segments used during a video streaming session. The quality switches defines how many times the video quality changes from one bitrate to another during transmission due to change in bandwidth or delays in the networks. Low startup delays and few video quality switches provide a good user's watching experience with an improved video quality. It is clear from table II that, our QoE-aware MPTCP SDN-based SR proposal performs better compared to the regular TCP in all three experiments.

VI. CONCLUSION

Efficient transfer of large flows, especially multimedia traffic is crucial to the performance of 5G multi-rooted topology networks where multiple paths exist between pair of hosts. In an attempt to address the challenges of transmitting and delivering high-demanding multimedia applications such as UHD videos with high QoE in 5G SDN-based networks, we propose a novel QoE-centric multipath routing algorithm (QoMRA) to dynamically control the number of subflows using SDN controller and perform source routing to these subflows using SR paradigms. we make joint use of MPTCP and SR to improve the performance of SDN-based virtualized 5G networks. Compared to the conventional TCP approach, our QoE-aware MPTCP SDN-based SR shows better performance in terms of increased network throughput, link utilization and the end-users QoE.

ACKNOWLEDGMENT

The work presented in this paper is fully funded by the European Union in the context of Horizon2020 Research and Innovation Programme under Marie Sklodowska-Curie Innovative Training Networks (MSCA-ITN-2014-ETN), Grant Agreement No.643072, Network QoE-NET.

REFERENCES

[1] M. Agiwal, A. Roy, , and N. Saxena, “Next Generation 5G Wireless Networks:A Comprehensive Survey,” *IEEE Communication Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[2] M. Sandri, A. Silva, L. A. Rocha, and F. L. Verdi, “On the Benefits of Using Multipath TCP and Openflow in Shared Bottlenecks,” *IEEE 29th International Conference on Advanced Information Networking and Applications*, April 2015.

[3] S. Zannettou, S. Michael, and P. Fragkiskos, “Exploiting path diversity in datacenters using MPTCP-aware SDN,” *IEEE Symposium on Computers and Communication (ISCC)*, pp. 539–546, June 2016.

[4] T.-S. Chen, D.-Y. Lee, T.-T. Liu, and A.-Y. Wu, “Dynamic Reconfigurable Ternary Content Addressable Memory for OpenFlow-Compliant Low-Power Packet Processing,,” *IEEE Transactions on Circuits and Systems*, vol. 63, no. 10, pp. 1661–1672, October 2008.

[5] A. Giorgetti, A. Sgambelluri, F. Paolucci, F. Cugini, and P.Castoldi, “Segment Routing for Effective Recovery and Multi-domain Traffic Engineering,” *Journal of Optical Communication Networks*, vol. 9, pp. A223–A232, March 2017.

[6] ITEC, “Dynamic Adaptive Streaming over HTTP:Dash plugin for vlc,,” [Available at]:http://www-itec.uni-klu.ac.at/dash/?page_id=10, June 2012.

[7] “GPAC:Multimedia Open Source Project,,” [Available at] :https://gpac.wp.imt.fr/mp4box/.

[8] S. Tariq and M. Bassiouni, “Packet-In Message Control for Reducing CPU Load and Control Traffic in OpenFlow Switches,” *In Proceedings of IEEE CCNC*, January 2015.

[9] S. Balzs, N. Felicin, C. Levente, L. Gulys, and A. Gulys, “SDN based Testbeds for Evaluating and Promoting Multipath TCP,” *In Proceedings of IEEE ICC- Next-Generation Networking Symposium,*, pp. 3044–3050, June 2014.

[10] N. Hyunwoo, D. Calin, and H. Schulzrinne, “Towards Dynamic MPTCP Path Control Using SDN,” *IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 286–294, July 2016.

[11] J. Duan and C. W. Zhi Wangy, “Responsive multipath TCP in SDN-based datacenters”,,” *IEEE International Communication Conference (ICC)*, June 2015.

[12] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, “Streaming High-Quality Mobile Video with Multipath TCP in Heterogeneous Wireless Networks,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2345–2361, March 2016.

[13] X. Corbillon, R. Aparicio-Pardo, N. Kuhn, G. Texier, and G. Simon, “Cross-layer scheduler for video streaming over MPTCP,,” *Proceedings of the 7th International Conference on Multimedia Systems*, May 2016.

[14] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, and S. Salsano, “Traffic Engineering with Segment Routing:SDN-based Architectural Design and Open Source Implementation,” *Fourth European Workshop on Software Defined Networks (EWSDN)*, September 2015.

[15] M.-C. Lee and J.-P. Sheu, “An efficient routing algorithm based on segment routing in software-defined networking,” *Journal of Computer Networks*, vol. 103, pp. 44–55, July 2016.