Received May 3, 2020, accepted May 10, 2020, date of publication May 14, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2994346

A Multipath Transport Scheme for Real-Time Multimedia Services Based on Software-Defined Networking and Segment Routing

WEI ZHANG, WEIMIN LEI, (Member, IEEE), AND SONGYANG ZHANG
Department of Communication and Electronic Engineering, School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China

Corresponding author: Wei Zhang (zhangwei1@mail.neu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1702000, in part by the Fundamental Research Funds for the Central Universities of China under Grant N2016014, in part by the Liaoning Provincial Natural Science Foundation of China under Grant 20180551007, and in part by the National Natural Science Foundation of China under Grant 61671141.

ABSTRACT It is still a challenging task to provide satisfactory quality of experience (QoE) to users by traditional routing mechanism in current public Internet for real-time multimedia services with high bandwidth and strict end-to-end latency constraint. Multipath transmission is a promising way to improve the performance of data delivery, due to its advantage in bandwidth aggregation capacity over multiple paths to speed up high volume transfers between endpoints. However, existing multipath transmission has many disadvantages such as the requirements for end-host support (i.e. the necessary multi-homed network setting, and operating system or upper applications updates of user devices), and blindingly pursuing maximum of transmission throughput. Path allocation and end-to-end transmission control are two important factors affecting the quality of multipath transmission. As an emerging network architecture, software-defined networking (SDN) is flexible, manageable, and responsive to rapid changes in traffic requirements. In this paper, we propose a multipath transport scheme based on SDN with segment routing (SR) which can meet the bandwidth and low end-to-end transmission latency requirements of real-time interactive multimedia services. The centralized SDN controller is extended to allocate multiple disjoint paths meeting bandwidth requirement meanwhile balancing the network load. The proprietary multipath media server (MMS) dynamically controls the assignment between subflows and routing paths to reduce the end-to-end transmission latency and thus improve QoE of end-users. To show the feasibility of our approach, we construct an SDN platform using Mininet. Simulation results show that our approach performs better compared to the conventional approaches in terms of traffic balancing and end-to-end transmission delay.

INDEX TERMS Multipath transport, path allocation, real-time multimedia service, segment routing, software-defined networking, transmission control.

I. INTRODUCTION

The explosive growth of multimedia services, such as live video streaming, video gaming and virtual reality, calls for efforts to optimize network transmission which can manage the increasing amount of data traffic on future network. Current Internet provides the best-effort packet delivery service by using the simple shortest path first (SPF) algorithm in an autonomous system (AS). This algorithm selects the shortest path that has the least number of links and thus uses the smallest amount of network resources. However, this approach does not consider the available capacity of all feasible candidate links. Thus, it may cause some links to become congested earlier, which can lead to poor resource utilization. Providing satisfactory quality of experience (QoE) for high-bandwidth video services in current Internet is still highly challenging [1], [2], especially for the real-time video communication services which have more stringent requirement on delay and are allergic to throughout fluctuation.

In order to provide sufficient throughput for large flows, new transfer protocols have been designed. Multipath TCP (MPTCP) [3], which has been standardized by the Internet

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

Engineering Task Force (IETF), can enhance the performance of network applications through bandwidth aggregation over multiple paths. As an extension of TCP, MPTCP provides a completely reliable delivery for network applications. But it is unsuitable for real-time interactive multimedia services that are tolerant to a small amount of packet loss. On the other hand, the premises for MPTCP is that user devices need to be equipped with multi-homed interfaces and system kernels need to be updated to support MPTCP. Due to these requirements, MPTCP is currently limited to datacenters scenarios. Building overlay network on existing Internet infrastructure is another alternative [4]. In our earlier work, we proposed a multipath transport framework based on application-level relay (MPTS-AR) [5], [6], in which both the control plane and the data plane work in application layer and user devices are responsible for multipath management.

These improvement mechanisms mentioned above have no control over the network or need to update system kernels or applications in user devices. For the former issue, as Software Defined networking (SDN) [7] is flexible, manageable and responsive to rapid changes in traffic requirements, it is reasonable and wise that the assignment of transmission paths used by each subflow in multipath transport is performed by the centralized SDN controller with the global view of the whole network topology and status. However, in multipath scenario, a multimedia flow is divided into several subflows. Each subflow is considered as an independent data flow and individually delivered in the data plane, which will significantly increase the number of forwarding entries stored in SDN switches and the load at the SDN controller and switches. This problem can be eliminated by Segment Routing (SR) [8] which is essentially a source routing approach that greatly reduces the number of forwarding rules in network nodes by encoding routing information into packet header as an ordered list of labels.

For the latter issue, multipath transport function can be migrated to inside the network. Reference [9] proposed an endpoint-transparent multipath transport solution with SDN in which the use of multipath is enabled in layer-2 or layer-3. However, edge switch can only implement simple packet scheduling policy. In this paper, we suggest that high-powered relay nodes can be specially deployed to replace user devices for multipath management. In fact, the idea of deploying relay nodes is not novel and has been adopted in many fields. For example, for video on demand services, the video content is divided into many video chunks which are cached at Content Delivery Network (CDN) nodes close to consumers ahead of time. For real-time communication services, Traversal using Relays around NAT (TURN) servers are necessary to relay data traffic for real time multimedia communication when direct path is inaccessible due to NAT problems. Relay servers are found to be used to improve video communication quality when the default path falls into poor condition by analyzing a dataset of 430 million calls from Skype [10].

In this paper, we explore the utilization of multipath transmission and SR in SDN-based networks for delivering real-time multimedia services. From the perspective of traffic engineering (TE), the centralized SDN controller is employed for intelligently allocating multiple disjoint paths for increasing real-time multimedia services according to the available network resources to balance traffic load and improve network resource utilization. From the perspective of service quality, Multipath Media Server (MMS), which are the external traffic sources and destinations, dynamically controls the assignment between subflows and routing paths to minimize the packet disordering and thus reduce the end-to-end transmission latency and improve QoE of end-users. And in this way multipath transport is plugged in without any modification to the user devices. The contributions of this paper are as follows:

1) We present a promising software defined multipath transmission mechanism (SD-MPT) with segment routing for real-time interactive multimedia service, comprising an SDN controller with better knowledge and control of available paths, leveraging SR to eliminate the scalability problem faced by SDN network especially with multipath support.
2) We further propose an adaptive packet scheduling method in MMS to reduce end-to-end service delay, enabling finer packet scheduling across available paths to adapt to heterogeneous path capacities and network dynamics.

The rest of this paper is organized as follows. In Section 2, we review the background and related work about multipath transport and protocols. In Section 3, we present the proposed SDN-based multipath transmission framework. In Section 4, we describe the packet scheduling method in MMS. The performance studies are presented in Section 5. Finally, Section 6 concludes the paper.

II. RELATED WORK

To pinpoint our motivation, we discuss the related work in the field of integration of SDN, segment routing, and multipath transmission in this section.

Usually, a traditional switch stores the entire network topology and forwards packets based on computed path according to this network topology; an SDN switch maintains forwarding rules at the flow granularity. Either way, switch has to keep a large amount of forwarding information. As a new approach, SR can be used to greatly reduce the number of forwarding rules by encoding routing information into packet header as an ordered list of labels. The scalability problem faced by large network especially SDN can be estimated by SR since there are much fewer path state maintenances required in each switch or router along the path. Reference [11] provides a comprehensive survey of SR in a wide range of network applications, such as traffic engineering, network resiliency, network monitoring, and service function chaining.

Indeed, SDN-based SR implementations can efficiently manage network resources and provide better TE solution in future networks. For example, an SDN-based solution for assigning TE paths with SR based on Multi-Protocol Label Switching (MPLS) forwarding is proposed in [12]. SR and multicast are combined in [13], in which SDN controller computes an explicit multicast tree. An SDN based architecture for managing MPLS traffic engineering based on SR-MPLS principles is proposed in [14]. An efficient routing algorithm based on SR in SDN that meets bandwidth requirements of routing requests is described in [15].

With multipath transport, network resource utilization can be improved, and the increased reliability and throughput can also enhance the user experience. Multipath transport has been applied in various settings for data delivery. Equal Cost Multi-Path (ECMP) routing [16], which is widely supported in many routing protocols including OSPF and Routing Information Protocol (RIP), routes packets along multiple paths of equal cost in large-scale network. ECMP uses random hashing to uniformly split flows over different shortest paths and is widely deployed in datacenters nowadays. However, ECMP performs poorly in asymmetric topologies which are common in today's networks due to link failure and heterogeneous network components [17].

At the transport layer, a couple of multipath transport protocols such as multipath TCP (MPTCP) [18] and CMT-SCTP [19] have been proposed as alternative transport approaches. In particular, MPTCP provides the ability to split a flow into multiple paths thus provides better performance and resilience to failures than regular TCP. However, the burden of maintaining a large number of subflows may cause nonnegligible overhead to user devices. Moreover, user side has no control over the network and transmission routes used by each subflow. Thus, MPTCP performance is greatly affected by the routing mechanism of the subflows. Especially when MPTCP is combined with flow-based ECMP, MPTCP performance is critically degraded as distinct subflows may end up using the same path after random hashing.

Many efforts have been made in the field of SDN-based MPTCP for different network scenarios, especially datacenter network [20]–[23]. In [20], packet inspection is used to provide deterministic subflow assignment to paths and an alternative routing mechanism is facilitated for MPTCP subflows. In [21], the controller decides the number of subflows for each MPTCP session in responding to the current traffic condition. In [22], two MPTCP-compatible solutions that are subflow-oriented and connection-oriented respectively are proposed to balance the load of pools of servers in SDN-based datacenter. In [23], an SDN architecture in datacenters is presented to improve throughput of large flows by initiating new MPTCP subflows. In [24], SDN is used to add or remove MPTCP paths in order to reduce a large number of out-of-order packets which may cause poor performance and degrade the end-user's QoE. These 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 increase the load at the SDN switches and the controller. It is worth mentioning that flow tables in SDN switches are implemented by TCAM (Ternary Content-Addressable Memory) [25], which is small, costly and energy-hungry. Reference [26] and [27] make use of MPTCP and SR to maximize the throughput of the traffic flows in SDN-based data center and 5G networks respectively. In all these solutions, the SDN controller is MPTCP-aware and has close interaction with MPTCP mechanism, which affects severely the performance and scalability of SDN controller in larger networks.

In terms of latency and interactivity, video services are divided into video streaming service and real-time interactive video service. For video streaming service, a second-level waiting delay is often allowed before playing video streaming. Extensive solutions have been proposed to improve quality of experience for video streaming service. Among them, Dynamic adaptive Streaming over HTTP (DASH) [28] is widely applied in VoD system. In DASH system, a video program is divided into many video chunks which are encoded with different quality. These video chunks are cached in advance at CDN nodes close to consumers, and clients download different video chunks from server with instruction of the deployed adaptive bitrate (ABR) selection algorithm. In some efforts, MPTCP is used to improve performance and reliability of video streaming services [29].

For interactive real time communication applications, they have more stringent requirement on end-to-end delay and are allergic to throughout fluctuation. ITU-T G.114 recommends a less than 150 millisecond one-way delay is excellent for media quality although delays between 150 and 400 milliseconds are still acceptable. Unfortunately, current proposed multipath transmission protocols CMP-SCTP and MPTCP are mainly designed for bulk data transfer. They strictly guarantee in-order delivery and reliable transmission, which are not appropriate for delay sensitive real-time video services.

For multipath transport of real time applications, some studies proposed the multipath versions of real-time transport protocol (RTP). In [30], multiflow RTP (MRTP) was proposed for ad hoc networks. An obsoleted IETF draft [31] proposed multipath RTP and described protocol extension details. Both of them only focused on protocol extension and implementation for multipath and did not discuss important issues such as path management and packet scheduling. In our earlier work, we proposed a multipath transport framework based on application-level relay (MPTS-AR) [5], [6], in which both the control plane and the data plane work in application layer. An independent overlay network needs to be deployed to provide relay-based multipath transport service; user devices need to update upper applications to manage multiple paths and subflows.

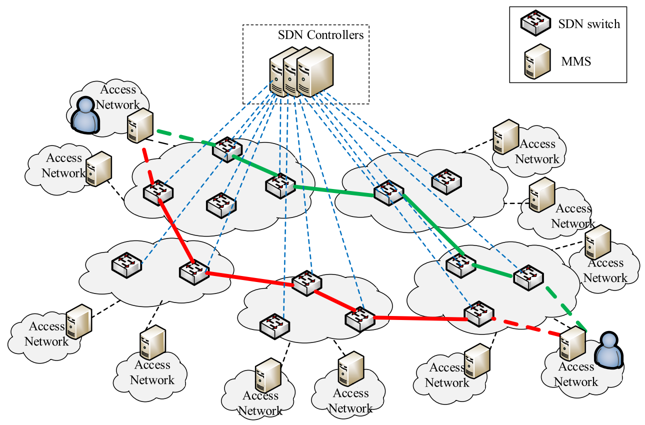


FIGURE 1. Overview of the multipath transmission scheme in SDN for real time multimedia service.

III. DESIGN OF SOFTWARE DEFINED MULTIPATH TRANSMISSION MECHANISM FOR REAL-TIME MULTIMEDIA SERVICE

A. OVERVIEW

As shown in Fig. 1, we illustrate an overview of software defined multipath transmission (SD-MPT) mechanism for real time multimedia service. Multipath Media Server (MMS) which acts as external traffic source and destination, provides multipath transport service for end-to-end multimedia flows. MMSs can be high-performance servers specifically deployed at the edge of SDN network, or existing relay/TURN servers in communication system extended with multipath transport functions. A multimedia flow will be delivered concurrently via multiple paths between MMS pairs. The part of a flow on an individual path is called a subflow. Instead of user devices, MMSs near user side are responsible for multipath transport functions including path management, flow partitioning and scheduling, subflow recombination, transport feedback, and so on. The introduction of the MMS component has several advantages: (1) Multipath transport process is completely transparent to user devices that do not need to make any changes to transport layer protocols and upper applications. (2) Compared to millions of user devices, a much smaller set of MMSs can be challenged to provide authentication and authorization information to establish secure channels with centralized SDN controller. The security of the controller can be improved through limiting communication only with the trusted entities. (3) It is an increasingly common practice for the endpoints with high access bandwidth through well-connected access network. The bandwidth constraints of an end-to-end session are gradually migrating from user access network to inside the network. And there are usually no disjoint paths between user devices and adjacent MMSs.

The centralized controller provides an ideal environment for implementing more efficient routing mechanisms for massive flows with a global view of the network. The controller can calculate various sets of paths between nodes thus better balancing the traffic and exploiting the available path diversity in a given topology. The SDN controller is responsible for the calculation of paths, the installation of the appropriate OpenFlow rules to the switches, and the response to path allocation requests from MMSs.

Source MMS first estimates the required throughput for a being-established multimedia flow according to the multimedia service type, video resolution, and frame rate, and then requests multiple available paths from the SDN controller by sending a PATH-REQ message. This message contains the address 4-tuple, the estimated throughput, and so on. After receiving a PATH-REQ message, SDN controller estimates the required throughput for the specific flow and allocates paths according to the algorithm described in the next subsection. Therefore, the main problem to be solved in SDN controller is to decide a routing plan that is responsible for computing multiple paths between pairs of communication nodes under given QoS requirements.

MMSs maintain a multipath session for each multimedia flow. A SD-MPT session is responsible to deliver a flow over multiple paths and can be defined by 4-tuple (source and destination IP addresses and port pairs). Each SD-MPT session has a unique identifier, called token, which is used for the association of multiple subflows. MMS may concurrently use multiple paths to obtain higher throughput, or may send all traffic on a specific path while other paths are kept as alternatives to enhance resilience. If no packets are received on a flow within a given time, SDN switch will withdraw the corresponding flow entry from forwarding table. Therefore, MMS should keep all paths alive by actively sending keep-alive packets periodically. To maximize quality of experience for real-time multimedia service, MMS monitors delivery quality of all active paths using transport feedback information and adaptively adjusts load shares among multiple paths.

User devices are unaware of the presence of MMSs. The address information of MMSs is usually negotiated from the out-of-band signaling (e.g., session initiation protocol (SIP), real-time streaming protocol (RTSP)) used for establishing multimedia flows, which is beyond the scope of this paper.

B. SDN WITH MULTIPATH AND SR

1) CONTROL PLANE

The control plane is implemented via extensions to a centralized SDN controller. All SD-MPT flows are controlled by the controller. Here we only discuss the control procedures of SD-MPT traffic which use multipath transmission and segment routing. Fig. 2 shows the function modules and interfaces of the extended multipath-aware SDN controller.

We enhance an SDN controller with multipath allocation (or MP) and segment routing (or SR) modules. The SDN controller is requested to allocate multiple paths for a multimedia flow with a specified bit rate, knowing the link capacity. The MP module will first allocate multiple hop-by-hop paths for the flow, solving *a multipath assignment problem*. Then, for each hop-by-hop path, SR module will compute a corresponding SR path for instructing the subflow

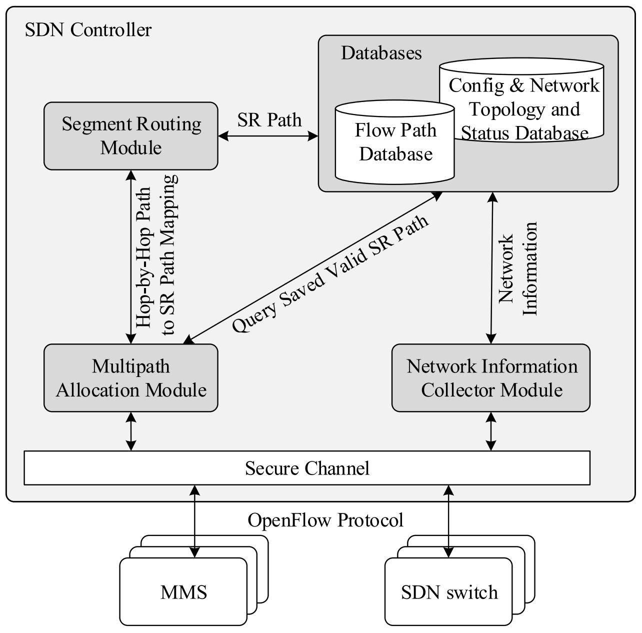

SDN Controller
Databases
Config & Network Topology and Status Database
Flow Path Database
Segment Routing Module
SR Path
Hop-by-Hop Path to SR Path Mapping
Query Saved Valid SR Path
Network Information
Multipath Allocation Module
Network Information Collector Module
Secure Channel
OpenFlow Protocol
MMS
SDN switch

FIGURE 2. The function modules and interfaces of the extended multipath-aware SDN controller.

packets through the assigned hop-by-hop path, solving *a SR assignment problem*. With SR, a host or an edge router/switch is able to steer a packet through the network using an ordered list of processing/forwarding functions called segments.

The multipath assignment algorithm will be described in the following subsection. We can use the SR assignment algorithm proposed in [12] to find the minimal-length SR paths corresponding to each hop-by-hop path, according to the default IP routing tables of the nodes. The SDN controller then calculates the corresponding segment list for each SR path, generates segment routing entry, and distributes routing rules down to the switch node in the beginning of SR path. SDN controller does not need to add forwarding rules to other switches along the SR path. In other words, the intermediate nodes do not need to maintain any per-flow state. This approach reduces the burden of the links between controller and switches and the size of forwarding tables in switches.

The set of SR paths allocated for a flow are stored in the database so that they can be used directly later by other subsequent being-established flows of the same MMS pair. This can further reduce CPU utilization of the SDN controller. The entries in database can be set to expire in a preset interval such as one minute, so that paths can get refreshed with the change of traffic distribution and link failure. When a new request arrives, SDN controller first queries in the database if a set of valid SR paths corresponding to the same MMS pair exists. If no valid SR paths exist, a new path allocation process for this MMS pair will be performed.

2) DATA PLANE

Switches in the data plane are divided into two types: loading switch and forwarding switch. Loading switch, i.e. the ingress switch in SR path, has the responsibility of loading segment

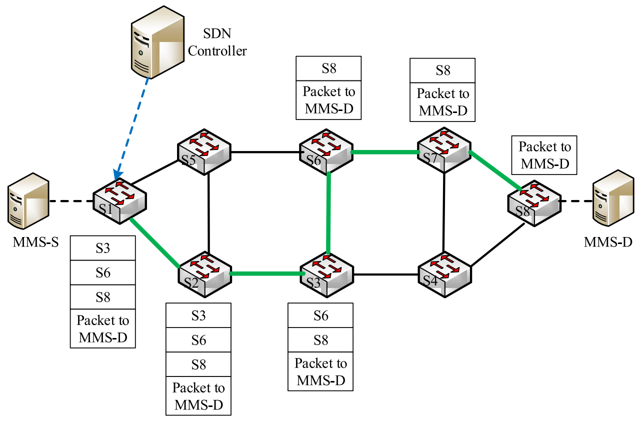


FIGURE 3. An example of SR-based routing process in SD-MPT.

list into packets. Forwarding switches just forward packets according to the SR header fields in packets. It is obvious that every switch can be loading one and forwarding one at the same time from the perspective of different flows.

In SDN switch, the rules tell the switch where to forward a packet. Each entry in the flow table called flow entry specifies an action for flows that match the fields in the entry. Each entry in the group table called group entry can specify more than one action. A flow entry can refer to a group entry in order to apply multiple actions to its matching flows. One way to implement segment routing in SDN network is to leverage the Group tables. In Open vSwitch, the OpenFlow protocol has been modified to push all the required labels with a single action as a single flow entry.

We illustrate an example of SR-based routing process in SD-MPT in Fig. 3. The mark on each switch is the node segment. The SDN controller computes the SR paths by the SR module, and configures the forwarding table of the ingress switch with the corresponding segment list. Assume that one of the hop-by-hop paths is {S1, S2, S3, S6, S7, S8}. It is mapped into the SR path {S3, S6, S8} because the default paths between nodes S1 and S3 and nodes S6 and S8 are {S1, S2, S3} and {S6, S7, S8} respectively, same as part of the hop-by-hop path. MMS-S assigns part of the multimedia flow to this path. First, as a loading switch, switch S1 executes actions in a group entry for the matching flow from MMS-S, i.e., encodes the segment list into the packet header, processes the top label, and then forwards the packet along the shortest path toward the network device with the node segment S3. Then, switch S3 pops the top label, and the following top label is S6. Therefore, switch S3 forwards the packet toward switch S6. Switch S6 forwards the packet to S8 along the shortest path and so on. Intermediate nodes S2 and S7 forward the packet without any modification to the segment list in packet header.

C. MULTIPATH ASSIGNMENT ALGORITHM

One aim of our research is to endow Internet service providers with the ability to provide traffic with feasible paths that meet specific service guarantees. The centralized SDN controller is a good place to implement traffic engineering that works by explicitly and dynamically directing the traffic through a finite and competitive network for upgrading network performance. In current traditional routing scheme, it is common that some links in the network are overloaded while other links are being underutilized. To increase the network throughput, our proposed routing algorithm needs to reduce the potential for network congestion by obtaining and utilizing a state of network resource usage from a long-term perspective. To improve user satisfaction, the algorithm also needs to combine with service-related constraints, such as a bounded end-to-end delay or a guaranteed bandwidth requirement. In addition, we need to pay attention to the path hop count. A longer path length usually means a longer delay time and more consumption of link bandwidth resources than a smaller path length. As described above, SR can eliminate the complexity of maintaining large numbers of forwarding rules and bring several orders of scaling gains. However, it causes extra wastage of network resources due to the bandwidth occupied by the segment lists in the packet header. Therefore, our algorithm also needs to limit the maximum path length between communication nodes.

In [15], a heuristic routing algorithm with a bandwidth guarantee and an extra-hop limitation is proposed for SDN network. Based on this algorithm, we propose a heuristic multipath assignment algorithm that allocates multiple disjoint paths.

We define the notations used in this paper. An SDN network topology can be represented as a weighted graph $G = (U\&V, E)$, where U and V are the set of nodes, and E is the set of edges. Each vertex in U and V represents an MMS node and an SDN switch node respectively, and each edge in E represents a link between nodes. For each edge $e \in E$, let $w(e)$ denote the weight of the corresponding link. The problem is converted into calculating the minimum weight path with the limitation of the maximum path length between a pair of communication nodes.

The weight function $w(e)$ is crucial to the performance of routing algorithm. Its intent is to balance the load across the network and avoid bottleneck links between pairs of communication nodes. It should reflect both current loading status and future traffic load expectation of a link. As in [15], the current loading status of link e is reflected by the congestion index $s(e) = f(e)/b(e)$. $f(e)$ denotes the total amount of traffic flows carried by the link e at present, and $b(e)$ denotes the remaining bandwidth of the link e. Obviously, $s(e)$ is an increasing and convex function which increases quickly when the amount of traffic passing through the link e approaches its capacity.

The authors in [32], [33] observed the related property of traffic predictability. In particular, network traffic exhibits a strong temporal and spatial locality phenomenon. That is, the traffic distribution observed in the recent past will hold in the future. Therefore, the statistical traffic matrix (TM) which records the traffic demands in the just past time interval P can be used to predict the volume of future traffic demands between sets of source and destination pairs. The traffic demand is represented as all communication requests from a source s to a destination d with the total requested bandwidth B_{sd}. With a global view of the network, the centralized SDN controller can easily estimate a precise and timely TM using OpenFlow. In [15], the concept of link criticality is presented to express the future traffic load of a link. But the value of link criticality $c(e)$ is calculated only with the occurrence rate of link e in the first k shortest paths of any pair (s, d), not taking into account the different bandwidth demands on the links. We update the equation of link criticality $c(e)$ for the link e as shown in (1). K_{sd} and B_{sd} denote a set of the first k shortest routes and the total requested bandwidth between a pair of source and destination nodes (s, d) respectively. $K_{sd}(e)$ denotes the number of times that the first k shortest routes pass through the link $e \in E$. At the beginning of the algorithm, all pairs of nodes are assumed to have an equal possibility for communication, and the measure of link criticality is mainly derived from the network topology. Then the $c(e)$ is computed and updated by the SDN controller in time interval P based on the estimated communicating pairs in TM rather than all of the pairs of nodes in the network.

$$c(e) = \sum_{\forall(s,d)\in TM} (K_{sd}(e) * B_{sd}/k) \tag{1}$$

As in [15], the weight of link e is designated by the link criticality value $c(e)$ and the link congestion index $s(e)$. The link weight $w(e)$ is calculated as (2). Obviously, $w(e)$ tends to refer to the link criticality $c(e)$ when the network load is light, and the link congestion index $s(e)$ when the network load is heavy. The weight of a path is the sum of the weight of all links on this path. After evaluating the link weight $w(e)$ for each link periodically, the SDN controller can use the newest weighted graph $G = (U\&V, E)$ to carry out the multipath allocation task.

$$w(e) = \alpha * c(e) + (1 - \alpha) * s(e) \quad 0 < \alpha < 1 \tag{2}$$

Whenever receiving a path allocation request for a pair of (s, d), our algorithm first deletes the links whose remaining bandwidth are less than the predetermined threshold $MIN - B_{sd}$ from the weighted graph G. Considering the bandwidth aggregation capability of multipath transmission, $MIN - B_{sd}$ can be obtained as shown in (3). $REQ - B_{sd}$ is the requested bandwidth in the path allocation request.

$$MIN - B_{sd} = \beta * REQ - B_{sd} \quad 1/M < \beta < 1 \tag{3}$$

As mentioned above, a path with a higher hop count consumes more network resource and has a longer segment list in packet header. Thus, our problem is converted into selecting the first M disjoint and minimum weight paths from a given source s to a destination d with the constraint that the path length is no longer than a threshold $MAX - H_{sd}$. $MAX - H_{sd}$ is the tolerable maximum hop count for the node pair (s, d), and can be computed according to (4). $SP - H_{sd}$ denotes the hop count of the shortest path between s and d, which is calculated and stored in advance.

$$MAX-H_{sd}=(1+\gamma)^{*}SP-H_{sd} \quad 0<\gamma<1 \tag{4}$$

In order to ensure that the selected paths are disjoint, our problem can be converted into solving the optimal path selection problem with a path length constraint M times. Bellman-Ford algorithm [34] can be used to solve the minimum weight path problem with a hop count constraint in polynomial time. Based on the idea, the steps of multipath assignment algorithm are shown in algorithm 1. The 4-8 lines delete edges with $b(e)$ less than $MIN-B_{sd}$ from G. The Bellman-Ford algorithm with a hop count constraint is executed to select an optimal path with minimum weight in 12-23 lines. The 25-32 lines generate the optimal path P with the aid of S array which stores precursor of each vertex in the path, and meanwhile all links on the path are assumed to have an infinite weight value. The for-loop structure in 10-33 lines generates M optimal disjoint paths.

Algorithm 1 Allocate M Paths After Receiving a Path Allocation Request PATH-REQ

```
Input: $G=(U\&V,E)$, PATH-REQ, $M$
Output: $M$ hop-by-hop paths
1: Extract $(s,d)$ from PATH-REQ
2: Extract $REQ-B_{sd}$ from PATH-REQ
3: Compute $MIN-B_{sd}$ according to (3)
4: for each edge $e(u,v)\in E$ do
5:    if $b(e(u,v))<MIN-B_{sd}$ then
6:       $w(e(u,v))\leftarrow\infty$
7:    end if
8: end for
9: Compute $MAX-H_{sd}$ according to (4)
10: for $m\leftarrow 1$ to $M$ do
11:    /* Bellman-Ford algorithm */
12:    for each vertex $v\in U\&V$ do
13:       $D(v)\leftarrow\infty$
14:    end for
15:    $D(s)\leftarrow 0$ for source vertex $s$
16:    for $i\leftarrow 1$ to $MAX-H_{sd}$ do
17:       for each edge $e(u,v)\in E$ do
18:          if $D(u)+w(e(u,v))<D(v)$ then
19:             $D(v)\leftarrow D(u)+w(e(u,v))$
20:             $S(v)\leftarrow u$
21:          end if
22:       end for
23:    end for
24:    /*generate the path $P$ with $S(d)$*/
25:    $cur\_vertex\leftarrow d$
26:    add $cur\_vertex$ to $P[m]$
27:    while $cur\_vertex\neq s$ do
28:       $pre\_vertex\leftarrow S(cur\_vertex)$
29:       add pre_$vertex$ to $P[m]$
30:       $w(e(pre\_vertex,cur\_vertex))\leftarrow\infty$
31:       $cur\_vertex\leftarrow pre\_vertex$
32:    end while
33: end for
34: return $P$
```

D. CONSIDERATIONS FOR ACTUAL DEPLOYMENT

The explosive growth of multimedia traffic motivates both application service providers (ASP) and internet service providers (ISP) to optimize network transmission to manage the increasing amount of data traffic on network. According to our solution, ASPs can deploy a large number of proprietary servers with high network bandwidth and computing performance as MMSs near user-side network, or update existing relay/TURN servers with multipath transport function.

Providing clear advantages over traditional network architecture, SDN has attracted growing attention from many ISPs and its availability has been extended to routers in WANS in recent years. An ISP can operate its own SDN platform in its domain and share information with other ISPs. In addition, we note that the load at an SDN controller is an inherent issue in the SDN paradigm, and it is suggested that for very large scale SDNs multiple controllers should co-exist to alleviate performance bottlenecks [35].

SDN and SR can work both in a single AS and across multiple ASs. For inter-AS scenarios, BGP Peering segments are defined in SR [8] for the expression of source-routed inter-domain paths. As stated in an IETF draft in reference [36], the SDN controller can collect the internal topology and the related IGP segment identifiers (SIDs) by collecting the IGP link state database (LSDB) of each area or running a BGP-LS [37] session with a node in each IGP area. Furthermore, the SDN controller is able to obtain an accurate description of the egress topology and associate BGP Peering SIDs to the various components of the external topology by the collected BGP-LS routes. A hierarchical multi-domain control plane for SDN networks based on SR has been demonstrated in reference [38]. The control plane is composed by an orchestrator application which runs on top of multiple SDN controllers and leverages their northbound APIs to create multi-domain SR based services. Orchestrator interacts the SDN controllers and builds a global network view that will be used to perform the path computation and to instantiate SR services.

Real network is composed of many kinds of traffic. For short-lived flows with light traffic, they are relatively small compared with elephant flows and have limited influence on network. It is wise to direct these flows following the traditional routing rule, for the purpose of minimizing the burden to SDN controller and the number of forwarding rules in SDN switches. So, when SD-MPT is deployed in the actual network, hybrid SDN [39], [40] seems to be one better choice. From OpenFlow protocol v1.3.0, a hybrid switch mode called OpenFlow-hybrid is proposed, which supports both OpenFlow operation and normal switching operation such as traditional L2 Ethernet switching and L3 routing. Under this circumstance, normal traffic and SD-MPT traffic co-exist in the network. For every new packet, switch node first checks whether the packet is traditional or SD-MPT. If the packet is classified to belong to a traditional flow, it will be forwarded by normal switching operation. In particular, the packet is identified by its destination and forwarded to corresponding outgoing link along the shortest path that is calculated via OSPF protocol. For SD-MPT packets, the switch in the beginning of segment list has the responsibility of loading segment list into packet, and other switches just forward packet following the instructions indicated by packet headers.

IV. ADAPTIVE PACKET SCHEDULING IN MMS

A. OVERVIEW

Various user applications have different QoS requirements. For example, for over-the-top video streaming such as YouTube and Netflix, throughput is considered important to provide high video resolution and avoid video pause events. Real-time traffic applications such as VoIP and online gaming are strictly latency-sensitive. It is worth noting that in real time services, video frames are captured at fixed intervals and the output bitrate of video encoder is stable in a short time span. In this paper, our packet scheduling method is designed to minimize end-to-end service delay under the constraint of bandwidth guarantee over multiple paths.

It seems to be tempting to have more subflows per SD-MPT session to better exploit bandwidth aggregation. However, the maintenance of a large number of subflows imposes extra overheads to the MMSs. In source MMS, higher CPU utilization is required for packet scheduling. Moreover, multipath transport may cause a large number of out-of-order packets, especially when the paths have different transmission capacities such as bandwidths and delays. Destination MMS has to temporarily hold the packets received ahead-of-time in a reordering buffer until their sequence numbers are in the desired order, which may cause a nonnegligible delay before forwarding them. More subflows per session requires a larger buffer to cope with packet reordering, which causes a longer extra delay. It is thus desirable to have as few subflows as possible with satisfying performance.

In our proposed method, source MMS assigns multimedia packets to multiple active paths and monitors current transmission qualities of connected paths. The available bandwidth information of allocated paths estimated by SDN controller provides decent decision reference for the load distribution procedure in MMS at the beginning of the flow. However, transmission delay performance of a path remains unknown until there are data packets passing through this path. And due to network dynamics, the optimal path at present may become suboptimal after a short time if the path falls into congestion. Thus, source MMS can only obtain accurate path qualities through latest packet transmission information combined with destination MMS.

Source MMS identifies poor path that mainly increases the reordering buffer size with the aid of feedback mechanism. It suspends the path that causes a relatively higher loss or

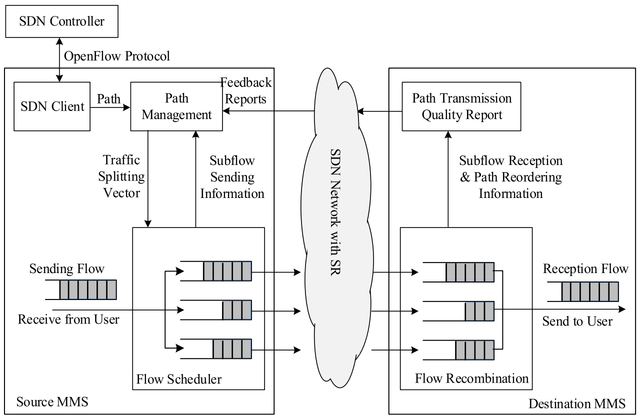


FIGURE 4. The functional structure of MMS.

discard rate due to late forwarding to destination user compared to other available paths, and attaches the path again when the transmission performance is acceptable. The load shares of active paths are dynamically adjusted to gradually match path transmission capacities.

The functional structure of MMS is illustrated in Fig. 4. The SDN client module interacts with SDN controller to request multiple paths. MMS communicates with SDN controller over a connection which may be encrypted using transport layer security (TLS) or run directly over TCP. MMS can get the IP address of SDN controller in a number of ways, such as manual configuration and DNS queries. When a multimedia flow needs to be established, MMS sends a PATH-REQ message to SDN controller to request routing paths. Then SDN controller allocates multiple available paths according to the path allocation algorithm described in Section III-C and returns them to the corresponding MMS. The path management module is responsible for maintaining and updating states of all paths and accordingly adjusts the traffic splitting vector according to the feedback reports from destination MMS, which are described in the next subsection. The flow scheduler module schedules multimedia packets to multiple paths based on the traffic splitting vector. The flow recombination module receives and restores subflow packets, meanwhile records the receiving statistics information of each path such as packet loss, packet discards, and packet disordering. The path transmission quality report module generates feedback reports according to the recombination information provided by the flow recombination module and sends them back to the source MMS.

B. PATH STATE

Multipath transport control faces a lot of challenges produced by path diversity. Concurrent delivery over multiple paths with significant performance difference may cause worse performance than using a single path. In addition, in reality the path transmission performance, such as available bandwidth and transmission delay, is variable due to network dynamics. Therefore, MMS should be able to adapt to the performance changes of diverse paths by adjusting the load shares as far as possible.

Source MMS groups all available paths based on the path delay. The paths in the same group have similar transmission delay and the sum of their bandwidths meets capacity requirements of the flow. The chosen paths, marked as *active*, are used for current flow delivery, and the rest of the paths, marked as *suspended*, may be enabled in case the chosen paths degrade or fail.

As mentioned above, real-time services are strictly latency-sensitive. So, we introduce a threshold $MAX - DR$ at MMS in order to limit the size of the buffering delay caused by reordering buffer. That is, when the buffering delay of a packet reaches the threshold, destination MMS will release and forward the packet immediately, ignoring missing packets. $MAX - DR$ can be dynamically set to the tolerative one-way delay of real-time service minus the maximum of the estimated transmission delays of all active paths.

Destination MMS periodically generates feedback reports including the packet losses and discards for each subflow as well as information to calculate per path RTT. Packet discard here refers to an event that occurs when a packet arrives too late causing the buffering delay of another packet received ahead-of-time exceeding $MAX - DR$. A packet that triggers a packet discard event in destination MMS is very likely to become useless and be discarded by end user side. Packet loss here refers to an event that a packet is discarded by intermediate routers due to link failure or congestion.

We define some rules for marking paths as different states including *suspended, congested*, *lossy*, *discarded* and *non-congested*. If packet discards in a single interval is reported, a path is marked as *discarded*. If this behavior is observed over 3 successive intervals, we can conclude that this path has much higher transmission delay than other active paths, thus this path is marked as *suspended*. If packet losses in a single interval is reported, a path is marked as *lossy*. If this behavior is observed over 3 successive intervals, we can conclude that this path is very likely to be congested, thus it is marked as *congested*. The rest of paths are marked as *non-congested*. Source MMS adjusts the respective traffic share of active paths according to path states.

C. PACKET REORDERING

Since the buffering delay in source MMS is negligible, the latency D caused between an MMS pair is composed by the transmission delay and the buffering delay in destination MMS. The transmission delay D_t can be reasonably estimated as one half of the smoothed roundtrip time $SRTT$. When a sent packet is acknowledged, a roundtrip time (RTT) sample can be obtained, and the way to update SRTT is shown in (5) where δ is usually set as 0.85.

$$SRTT = (1 - \delta) * SRTT + \delta * RTT \tag{5}$$

The buffering delay D_r grows with the size of reordering buffer in destination MMS. To identify poor paths that mainly increase the reordering queue size, we try to quantify the disordering degree caused by each path. For packet p_i, st_i is the sending time at source and rt_i is the reception time at destination. For packets p_i and p_j, if $st_i < st_j$ and $rt_i > rt_j$, the two packets form a *disordering pair* at the destination. The magnitude of a disordering pair, referred to *disordering step*, can better reflect the degree of packet out-of-ordering. The disordering step of a disordering pair p_i and p_j, is denoted as pr_{ij} and calculated as follow:

$$pr_{ij} = FSN_j - FSN_i + NUM_{ij} \tag{6}$$

FSN_i and FSN_j are the flow sequence number (FSN) of packet p_i and p_j in the flow. NUM_{ij} is the number of packets that arrive at the destination MMS between p_j and p_i and whose FSN is greater than FSN_j.

The disordering step of a path pair P_l and P_k, denoted as PR_{lk} $(1 \le l, k \le M)$, is calculated as the maximum of disordering steps of the disordering pairs caused by packets arrived at the destination MMS through path P_l and P_k:

$$PR_{lk} = \max_{p_i \in p_l, p_j \in p_k} (pr_{ij}, 0) \tag{7}$$

The larger value of PR_{lk} means that path P_l with larger transmission delay causes longer reordering buffer compared to path P_k. A *disordering vector* is calculated for each path. The disordering vector of path P_l is denoted as $PR_l = (PR_{l1}, PR_{l2}, \ldots, PR_{lM})$, $(1 \le l \le M)$. Destination MMS calculates disordering vectors for all paths periodically such as after receiving NUM_{rcvd} packets, and feeds the disordering matrix $A = (PR_1, PR_2, \ldots, PR_M)^T$ back to source MMS. NUM_{rcvd} should be set based on the data rate or frame rate of the flow and the default value is 100. For a path with small transmission delay, most values in its disordering vector are zero, and others are usually small positive values.

D. TRAFFIC SPLITTING VECTOR

Source MMS adaptively adjusts the load shares of all active paths according to the path state and disordering information mentioned above. Let $F = (F_1, F_2, \ldots, F_M)$ be the traffic splitting vector. F_k $(1 \le k \le M)$ is the load share of the path P_k. It is obvious that F meets the following condition:

$$\begin{cases} \sum_{k=1}^{M} F_k = 1 \\ 0 \le F_k \le 1 \qquad (1 \le k \le M) \end{cases} \tag{8}$$

At the beginning of the flow, source MMS initializes the traffic splitting vector according to the estimated available bandwidth of allocated paths from SDN controller as follow:

$$F_k^{(0)} = \frac{b(P_k)}{\sum_{l=1}^{M} b(P_l)} \quad (1 \le k \le M) \tag{9}$$

When the state or transmission performance of a path changes during transmission process, source MMS recalculates the traffic splitting vector. If a path is marked as suspended after reporting packet discard events in successive report intervals, source MMS suspends this path and sets its load share to zero. Let P' be the set of paths that are in the suspended state. From the perspective of network resources,

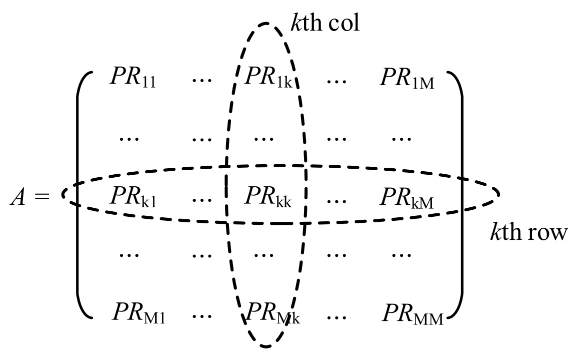

kth col
$A =$
PR_{11} … PR_{1k} … PR_{1M}
PR_{k1} … PR_{kk} … PR_{kM}
PR_{M1} … PR_{Mk} … PR_{MM}
kth row

FIGURE 5. The structure of disordering matrix.

we should reduce the traffic load of the congested paths so that they can recover from congestion state as fast as possible. For real-time services, packet missing, including packet loss caused by intermediate routers and packet discard caused by too-late-arriving in destination, affects the quality of user experience. Therefore, source MMS should reduce the load share on congested, lossy, and discarded path to θ_1, θ_2 and θ_3 of the previous value respectively, where $0 \leq \theta_1 \leq \theta_2 \leq \theta_3 \leq 1$. Let P'' be the set of paths that are in congested, lossy, and discarded states.

$$F_k^{(t+1)} = \lambda * F_k^{(t)} \quad \lambda = \theta_1, \theta_2 or \theta_3, \quad for\ P_k \in P' \quad (10)$$

For other paths not belong to P' and P'', they are in appropriate or light network state. Let $\tilde{P} = P - P' - P''$. In disordering matrix as shown in Fig. 5, the element PR_{kl} $(1 \leq l \leq M)$ in kth row means the maximum disordering step caused by late-arrived packets on path P_k compared to path P_l, representing the disadvantages of path P_k. Similarly, the element PR_{lk} $(1 \leq l \leq M)$ in kth column means the maximum disordering step caused by early-arrived packets on path P_k compared to path P_l, representing the advantages of path P_k. Therefore, source MMS should switch a fraction of load share from disadvantaged paths to advantaged paths to minimize the number of out-of-order packets in destination MMS and thus achieve low buffering delay. The traffic splitting vector is updated as follow:

$$\begin{cases} F_k^{(t+1)} = \left(1 - \sum_{P_i \in P'} F_i^{(t+1)}\right) \dfrac{F'^{(t)}_k}{\sum_{P_i \in \left(P - P'\right)} F'^{(t)}_i} \\ for\ P_k \in \tilde{P} \\ F'^{(t)}_i = \max\left(F_i^{(t)} + \dfrac{\sum_{j=1}^{M}\left(PR_{ji} - PR_{ij}\right)}{NUM_{rcvd}}, 0\right) \\ for\ P_i \in \tilde{P} \end{cases} \quad (11)$$

Algorithm 2 shows the abstract pseudocode of our adaptive packet scheduling method.

V. IMPLEMENTATION AND EVALUATION

We conducted simulation experiments to evaluate the performance of the proposed multipath transmission scheme in two factors: path allocation and end-to-end transmission control.

Algorithm 2 Adaptive Packet Scheduling Method

```
Input: $(s, d), M$
1: Obtain $M$ paths from SDN controller
2: Initialize the traffic splitting vector $F$ based on (9)
3: while the flow is not over do
4:   if receiving a feedback report do
5:      Update path states and $P'$ based on the feedback
        reports
6:      for each $P_k \in P'$ do
7:         $F_k \leftarrow 0$
8:      for each $P_k \in P''$ do
9:         Update $F_k$ for $P_k$ based on (10)
10:     end for
11:     for each $P_k \in \tilde{P}$ do
12:        Update $F_k$ for $P_k$ based on (11)
13:     end for
14:  end if
15:  Schedule flow packets based on $F$
16: end while
```

A. EXPERIMENTAL SETUP

We created an SDN testbed by using Mininet and POX SDN controller on a computer with Intel(R) Core i5-7500, 3.40GHZ CPU, 15.6 GB RAM over Ubuntu 16.04 system. We extended the SDN controller with an implementation of the multipath allocation and the segment routing modules. We did not implement segment routing by modifying OpenFlow protocol in virtual SDN testbed. The SR path was calculated based on the overlap between the allocated path and the corresponding shortest path, which was proposed in reference [12]. Mininet is used to model a scenario of virtual network. We built our experimental topology on the basis of Cernet network provided by the Internet Topology Zoo[1]. We deleted five external links associated with Beijing node and one redundant link between Beijing and Xi'an nodes. To ensure that there are multiple paths between any two nodes, we added new links connecting a node associated with only one link initially in Cernet to one of its geographical neighbor nodes. As shown in Fig. 6, our experimental topology consists of 35 OpenFlow switches and 57 links between switches. Four new added links are labeled as green lines. Each switch node is attached with one MMS. We set the bandwidth and delay limits of (10Gbps, 2ms) for links between MMSs and their associated switches, (1Gbps, 5ms) for links between switches. This ensured that the links between MMSs and ingress switches were not bottlenecks in the paths. SDN switches reported various networking information to the SDN controller every second. The messages exchanged between the controller and MMS were implemented by extending OpenFlow protocols, which imposed negligible overhead compared to data flows.

The maximum subflow number M was set to 4 based on the following considerations. (1) The characteristics of the

[1]www.topology-zoo.org

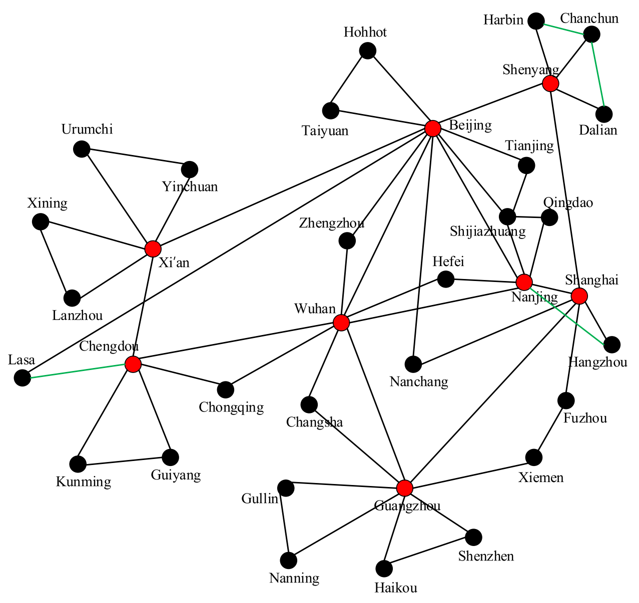


FIGURE 6. Network topology of simulation experiments built on basis of Cernet network.

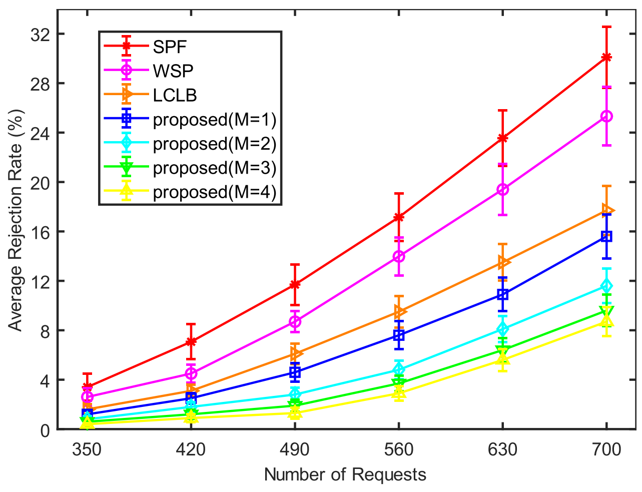


FIGURE 7. Comparison of average rejection rate versus number of requests.

real-time multimedia service are that its duration is on the order of minutes or even hours, and the output bitrate is stable in a short time span due to the fixed video framerate and spatial resolution. (2) The aim of multipath transport for this kind service is not maximum of transmission throughput but minimum of end-to-end service delay as well as improvement of network resource utilization. More subflows does not help much for this aim. (3) In addition, the time complexity of our proposed path allocation algorithm is proportional to M. Therefore, M should not be too large.

The first part of experiments was conducted to evaluate the performance of path allocation algorithm. MMS pairs randomly requested to transmit a multimedia flow. In the actual network, real-time multimedia services can have different video resolutions and framerates due to various hardware performances of user devices and service types. Therefore, in our experiments, the bandwidth demands of flows were set to follow a uniform distribution within the range of 0.5Mb/s to 12Mb/s. In addition, this part of experiments was to evaluate network resource scheduling capability of SDN controller with multipath. Thus, MMSs were assumed to equally distribute flow load among all allocated paths for simplicity.

The second part of experiments was conducted to evaluate the performance of packet scheduling method. We selected an MMS pair (one for source and the other for destination) and transmitted a pre-encoded H.264 stream of the foreman video clip (2098 frames; YUV 4:2:0) with a resolution of 1080P (1920∗1080) at 30 fps in a continuous loop. The bandwidth demand of this flow was about 8.5Mbps. This flow was encapsulated and end-to-end transmitted by RTP and the associated RTP Control Protocol (RTCP). The RTCP receiver reports were extended to feedback receiving information in destination MMS. In our experiments, the sending interval of RTCP receiver reports was set to one second, and the network overhead incurred by feedback mechanism was less than 5% of session bandwidth.

B. EXPERIMENTAL RESULTS OF PATH ALLOCATION

In this section, we investigate the impact of the number of the allocated paths M for a flow on network resource utilization. In addition, we also compare the performance of our path allocation algorithm with other algorithms. We chose two traditional routing algorithms including the SPF [41] and widest shortest path (WSP) [42] algorithms, and the algorithm (we refer to it as LCLB) proposed in reference [15]. The width of a path refers to the available bandwidth, and the length commonly represents the number of path hops. The WSP algorithm aims to select the shortest path that has the most residual bandwidth. In particular, if there are multiple paths with the same shortest hops, the WSP selects the widest. LCLB algorithm takes the link criticality and the link residual bandwidth into consideration. In our experiments, we use the following metrics to benchmark the performance of different approaches: a) the rejection rate, b) the network throughput, c) the link utilization, and d) the path length. All the results given below were obtained by averaging the results in 50 runs.

Each MMS as a source randomly picked one of other MMSs as the destination and requested M paths to transmit a flow between each pair. The controller assigned up to M paths for each flow. The flows logically continued until the end of experiments. Each MMS acted as source MMS of D flows. Thus, the total number of the requested flows was 35^*D, and the total number of the allocated paths was up to 35^*D^*M. The parameter D can be changed to simulate different load levels. The parameter α, β, and γ in equation (2)-(4) were set to 0.5, 0.6, and 0.6 respectively. The parameter k, standing for the first k shortest paths to compute the link criticality, was set to 3. The time interval P updating the traffic matrix TM was set to 1s.

1) REJECTION RATE

The rejection rate refers to the rate of the number of flows that are rejected to the total number of requested flows that

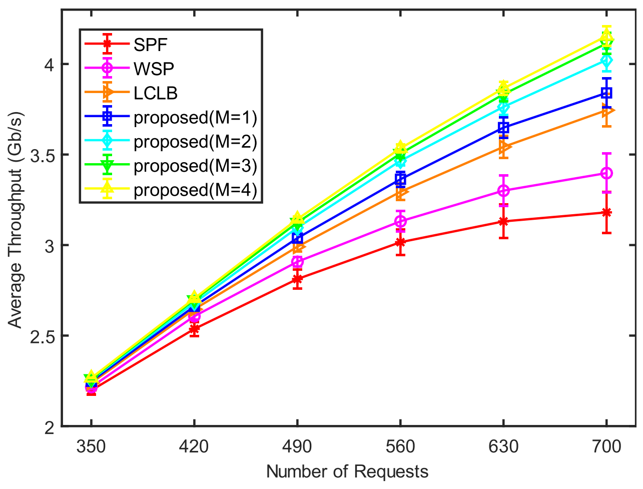


FIGURE 8. Comparison of average network throughput versus number of requests.

have arrived at the network. A flow will be rejected when the network cannot provide an appropriate path with sufficient capacity. Fig. 7 shows the average rejection rate achieved with our algorithms under four different setting of M, as well as those achieved with SPF, WSP, and LCLB. The 95% confidence interval is obtained from 50 runs. It is obvious that SPF has the worst performance indicated by greater values of rejection rate. SPF starts rejecting flows under a light load and grows rapidly as the number of requests increases. The fundamental cause is that SPF only takes the hop count into account regardless of current link utilization. This leads to the resources of some critical links to be used up rapidly. As an extension of SPF, WSP performs better than SPF by choosing the widest path when there are multiple shortest paths exist. However, WSP's capacity of load balancing is limited because its principle is still to take the shortest path.

Our proposed algorithm and LCLB take both current loading status and future traffic distribution expectation of links into consideration to reduce network congestion. And our algorithm also limits the maximum length of the allocated path using that of the shortest path as reference. As shown in Fig. 7, our algorithm under all settings achieves better performance than other three algorithms. And with the increase of the number of paths (M) assigned for a flow, the rejection rate reduces. This is expected for the intelligent routing management and multipath transport. On the one hand, with the leverage of the centralized SDN controller, our algorithm and LCLB can track the available capacity of network links and allocate appropriate paths for more multimedia flows depending on varying network status. On the other hand, our algorithm achieves better performance than LCLB by the load balancing capacity of multipath transport and further considering the effect of flow size on link load. Note that the gap between the $M = 4$ case and the $M = 3$ case is small because there are less than four disjoint paths under the maximum length limit between a considerable number of MMS pairs in our experimental topology.

2) NETWORK THROUGHPUT

The network throughput is the amount of bandwidth of the satisfied flows whose bandwidth requirement is met and is successfully transmitted by the network. Fig. 8 shows the average network throughput of various algorithms. It is easy to understand that the network throughputs of all algorithms increase as the amount of the requested bandwidth. With the similar result in rejection rate, SPF achieves the lowest network throughput overall. This is because SPF tends to exhaust the network resources of critical links earlier. The WSP performs better than SPF in an overall perspective. Our algorithm achieves better network throughout than others by serving more data flows. The improved performance ratio declines slightly as M grows larger than 3.

3) LINK UTILIZATION RATE

The link utilization rate refers to the rate of the consumed bandwidth of a link to the total bandwidth of this link. Since the links between MMSs and their associated switches have much higher bandwidth capacity, we only investigate the links between switches in our experiments. Fig. 9 shows the CDFs of link utilization rate achieved with different algorithms under light and heavy network loads. It is obviously that SPF results higher link utilization rate of some critical links even under relatively light network load ($D = 10$). On the other hand, SPF has more links with low link utilization rate due to higher rejection rate and lower network throughput under heavy network load ($D = 20$). That is, SPF contributes to unbalanced network resource use. The situation is alleviated to a small extent for WSP by balancing the load among multiple shortest paths if exist. Compared to the other three algorithms, our algorithm effectively avoids high link utilization rate of critical links in light network load, and improves the link utilization rate of more links in the case of heavy network load. This confirms our path allocation algorithm has a strong network load balancing capacity with the leverage of the centralized SDN controller and multipath transport. For example, the load on the link <Beijing, Wuhan> can be unloaded partially by the cascade of <Beijing, Zhengzhou> and <Zhengzhou, Wuhan>.

4) PATH LENGTH

Fig. 10 shows the average path length achieved with different algorithms. Besides hop count, we investigate the SR count for our algorithm. The SR count refers to the length of SR path in terms of segments. The SPF and WSP algorithms have the lowest hop counts because of the shortest path strategy they take. LCLB and our algorithm consider the factors of current and future expected network bandwidth consumption. Meanwhile, in order to avoid taking a lengthy path, our algorithm limits the path length by using the hop count constraint $MAX - H_{sd}$ whose value is proportional to the corresponding shortest path length; LCLB uses the extra-hop E_{sd} as a restriction, which was set to two in our experiments. The average hop count of our algorithm with $\gamma = 0.6$ is about 0.8 larger

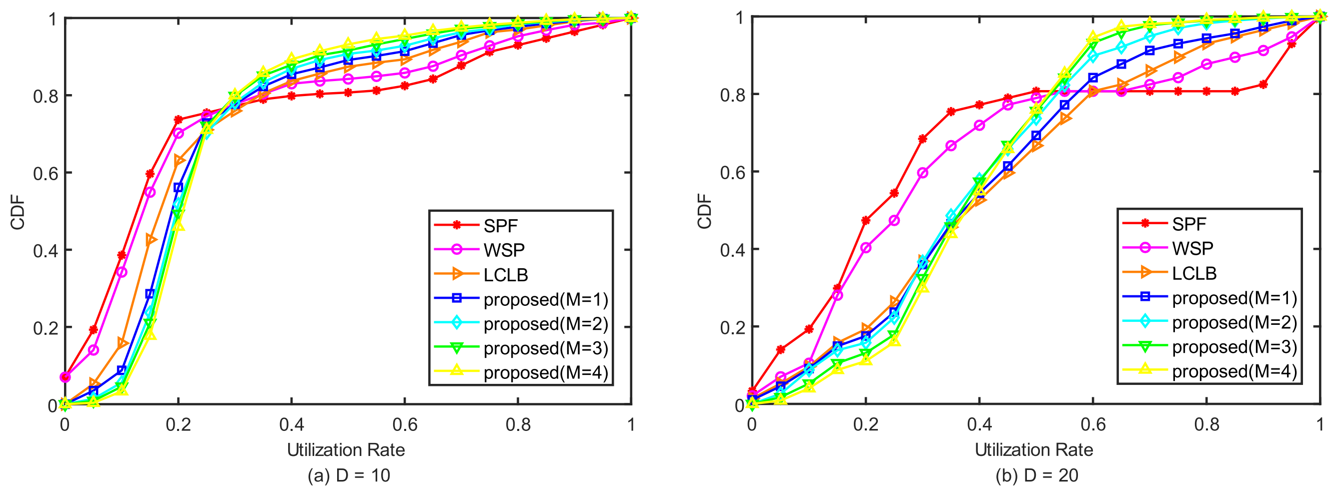


FIGURE 9. Comparison of CDF of link utilization rate. (a) under light network load ($D = 10$). (b) under heavy network load ($D = 20$).

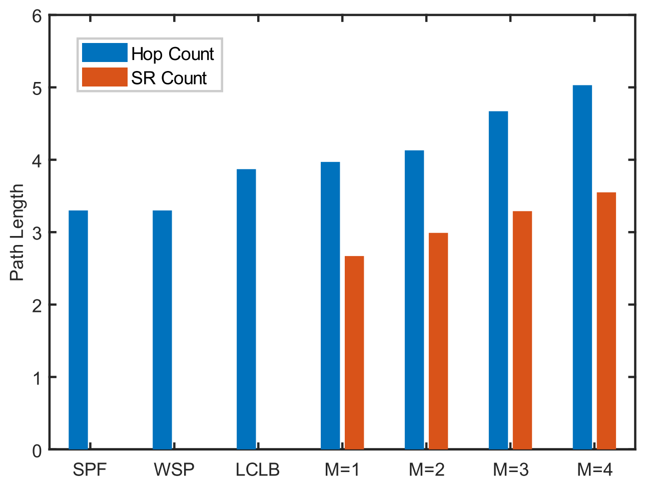


FIGURE 10. Comparison of path length ($\gamma = 0.6$).

than SPF and WSP when $M = 2$, and it increases slowly as M increases. As shown in Fig. 10, the SR count is relatively low but grows definitely as M increases. This is because the path allocated by our algorithm has less overlaps with default path as M increases. SR needs extra packet headers to express routing information, and larger SR count consumes larger amount of network resources. Therefore, there is a tradeoff between network throughput and network overhead.

C. EXPERIMENTAL RESULTS OF PACKET SCHEDULING IN MMS

In this section, we evaluate the performance of our packet scheduling algorithm over paths that exhibit bandwidth and transmission delay differentials. In our experiments, four paths ($M = 4$) have been allocated by SDN controller in advance. We also make comparison of our algorithm with three typical approaches: single-path transmission (SP), equal distribution scheduling (EQUAL), and Weighted Round-Robin (WRR) scheduler. The path chosen for SP is set to the path that has the best capacity among four paths. In equal distribution, the flow is equally allocated to multiple paths and the load shares do not change with path characteristics. The WRR scheduler sends bursts of packets along the paths weighted according to their capacity as $w_j = c_j / \sum_{i=1}^{n} c_i$, where w_j is the weight associated with path j and c_j its estimated capacity. In our experiments, c_j was set to the available bandwidth of allocated path j, which was estimated by SDN controller. We use the following metrics to benchmark the performance of different approaches: a) the packet loss rate, and b) the buffering delay. The buffering delay refers to the interval that the time of a packet forwarded to end user minus the time of the packet arriving to the destination MMS, i.e. the interval that a packet is cached in the reordering buffer.

1) EFFECT OF AVAILABLE BANDWIDTH DIFFERENTIAL

For the better description of bandwidth capabilities of and differentials among multiple paths, let $x_j = \mu_1 S / j^{\mu_2}$ denote the available bandwidth of the jth path. S denotes the bandwidth demand of the video flow, μ_1 is the ratio of the first path bandwidth x_1 over the bandwidth demand S, and μ_2 ($0 \leq \mu_2 \leq 1$) is referred as the skew factor representing the degree of bandwidth differentials. As μ_1 increases, the paths have more available bandwidth resources. When $\mu_2 = 0$, all paths have the same available bandwidth. As μ_2 increases, the bandwidth differentials among paths become more skewed. Except for available bandwidth, other metrics were the same among all paths.

Fig. 11 gives the total packet loss rate of received flow at the destination MMS versus μ_1 and μ_2. We obtain the following observations: (1) When the available bandwidth of a single path cannot meet the bandwidth demand of the flow, the SP algorithm achieves very high total packet loss rate.

Three multipath algorithms reduce significantly the total packet loss rate. This is expected for multipath transport which has been proved effective for bandwidth aggregation. (2) When the single path has enough bandwidth resource, the SP algorithm can achieve even better performance than

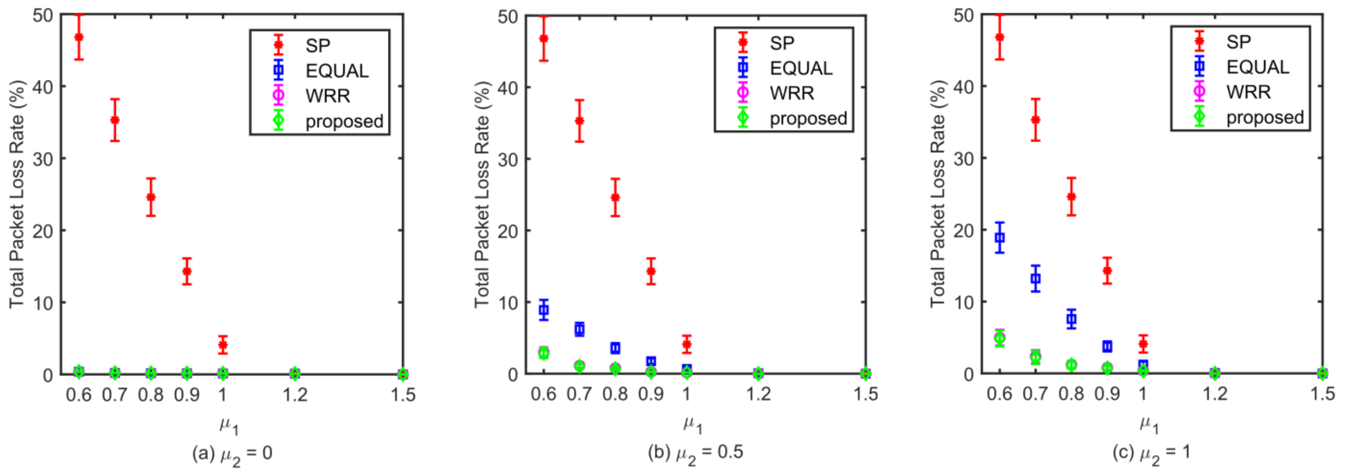


FIGURE 11. Comparison of the total packet loss rate versus μ_1 and μ_2.

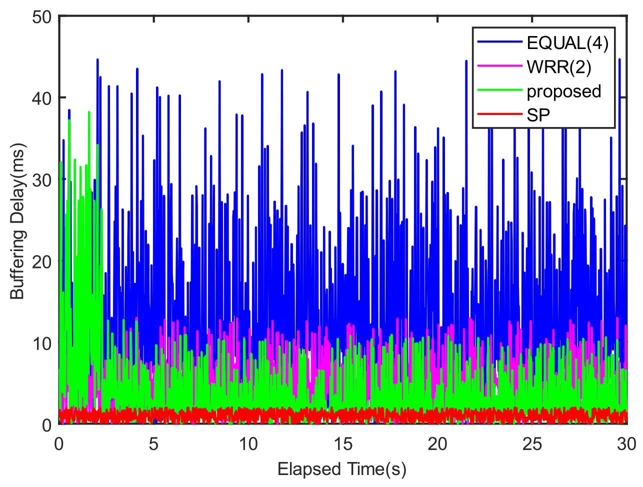


FIGURE 12. Comparison of the dynamic variation of buffering delay with the elapse of time.

three multipath algorithms. This indicates that the effect of bandwidth aggregation brought by multipath transport is significant in bandwidth-constrained scenarios but not obvious when bandwidth is not limited. (3) When the available bandwidths of multiple paths are heterogeneous ($\mu_2 > 0$), our algorithm and WRR perform better than EQUAL which equally allocates packets to four paths regardless of path differentials. Our algorithm and WRR achieved similar performance in case of stable bandwidth capacity.

2) EFFECT OF TRANSMISSION DELAY DIFFERENTIAL

The delay jitters of four paths were set to 2ms; the mean transmission delays of four paths were set to 20ms, 30ms, 50ms, and 60ms respectively to simulate the effect of transmission delay differentials on buffering delay. Each path had enough available bandwidth. Other network metrics were the same among all paths.

Fig. 12 illustrates the dynamic variation of buffering delays of packets in a specific flow with the elapse of time. And the

TABLE 1. The average load shares of all paths under various buffering delay threshold ***MAX − DR***.

$MAX\text{-}DR$	P_1	P_2	P_3	P_4
45ms	3.8Mbps	3.4Mbps	1.3Mbps	0
35ms	4.6Mbps	3.9Mbps	0	0
25ms	5.6Mbps	2.9Mbps	0	0
15ms	8.5Mbps	0	0	0

threshold $MAX - DR$ was set to 35 ms for our algorithm. It is easy to understand that SP algorithm achieves the smallest buffering delay as all packets of the flow are transmitted over the same path. For EQUAL, WRR, and our proposed algorithms, multiple packets belonging to the same encoded frame were distributed to different paths, which inevitably leaded to out-of-order packets and thus larger buffering delays. Our algorithm initially distributed the packets to all paths equally. After receiving three successive feedback reports indicating that packet discard events happened on the third and fourth paths at about the third second, our algorithm suspended these two paths and only kept the first and second paths as active. In our experiments, EQUAL and WRR achieved similar performance in case of same bandwidth capacity, so Fig. 12 only shows the results of the EQUAL algorithm with four paths and WRR algorithm with the first and second paths.

We observe that our algorithm performed significantly better than the EQUAL and WRR, and the buffering delay of our algorithm was noticeably decreased as the load shares of active paths gradually converged based on the feedback information. Fig. 13 is the corresponding box plot that shows the distribution of buffering delays. For our algorithm, Fig. 13 uses the buffering delay samples after the load shares of active paths converging to a relatively steady state.

Table 1 shows the average load shares of all paths after the traffic splitting vector in our algorithm converging to a relatively stable state. When $MAX - DR$ is relatively large, more paths are kept active for concurrent multipath transfer.

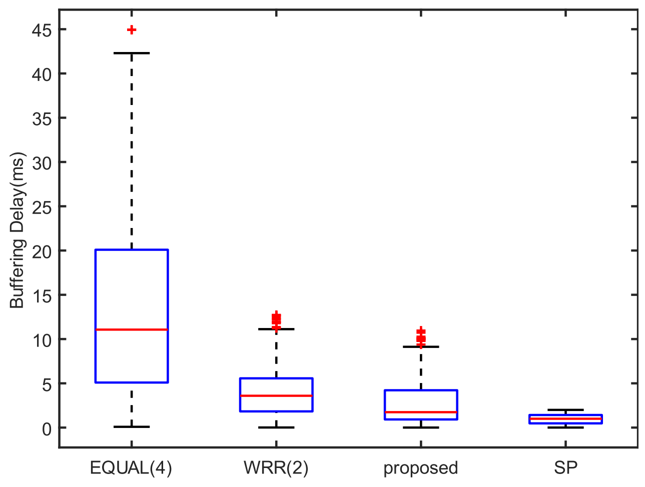


FIGURE 13. **Buffering delays of various algorithms in a specific flow.**

As $MAX - DR$ decreases, more load share is switched to the paths with low transmission delay. In practice, MMS can adaptively set $MAX - DR$ to the tolerative one-way delay of real-time service minus the maximum of the estimated transmission delays of all active paths. Consequently, multipath transport can be exploited as far as possible for balancing network load under end-to-end latency constraint.

VI. CONCLUSION

The explosive growth of novel multimedia services calls for the efforts from both internet service providers and application service providers to improve efficiency of data delivery. In this paper, we have presented a novel multipath transmission scheme in SDN with segment routing for real-time interactive multimedia services with the aim of balancing network traffic as well as improving the quality of end-user's service experience. Taking advantage of the global view of SDN, the path allocation algorithm in the controller takes future traffic distribution expectation and current loading status of links into consideration to reduce network congestion. In addition, segment routing is used to eliminate the scalability problem faced by SDN especially with multipath transmission. High-performance multipath media servers specifically deployed at the edge of SDN network provide multipath transport service, which are unperceived by end users. And multipath media servers adaptively adjust load shares among multiple paths to minimize the performance impact due to out-of-order packets. The simulation results showed that our method outperforms other algorithms in terms of optimizing network resource utilization and end-to-end service delay.

REFERENCES

[1] C. Yu, Y. Xu, B. Liu, and Y. Liu, "'Can you SEE me now?' A measurement study of mobile video calls," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2014, pp. 1456–1464.

[2] A. Zhou, H. Zhang, G. Su, L. Wu, R. Ma, Z. Meng, X. Zhang, X. Xie, H. Ma, and X. Chen, "Learning to coordinate video codec with transport protocol for mobile video telephony," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2019, p. 29.

[3] A. Ford, C. Raiciu, M. J. Handley, and O. Bonaventure, *TCP Extensions for Multipath Operation With Multiple Addresses*, document IETF RFC 6824, Jan. 2013. [Online]. Available: https://datatracker.ietf.org/doc/rfc6824/

[4] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient overlay networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 1, p. 66, Jan. 2002.

[5] W. Zhang, W. Lei, S. Liu, and G. Li, "A general framework of multipath transport system based on application-level relay," *Comput. Commun.*, vol. 51, pp. 70–80, Sep. 2014.

[6] W. Zhang, W. Lei, Y. Guan, G. Li, and L. Yang, "Considerations for application-layer multipath transport control," *Int. J. Commun. Syst.*, vol. 30, no. 17, Nov. 2017, Art. no. e3343, doi: 10.1002/dac.3343.

[7] ONF Market Education Committee, "Software defined networking: The new norm for networks," Open Netw. Found., Palo Alto, CA, USA, USAONF White Paper, Apr. 2012.

[8] C. Filsfils, S. Previdi, and L. Ginsberg, *Segment Routing Architecture*, document IETF RFC 8402, Jul. 2018. [Online]. Available: https://datatracker.ietf.org/doc/rfc8402/

[9] D. Banfi, O. Mehani, G. Jourjon, L. Schwaighofer, and R. Holz, "Endpoint-transparent multipath transport with software-defined networks," in *Proc. IEEE 41st Conf. Local Comput. Netw. (LCN)*, Nov. 2016, pp. 307–314.

[10] J. Jiang, R. Vafin, H. Zhang, R. Das, G. Ananthanarayanan, P. A. Chou, V. Padmanabhan, V. Sekar, E. Dominique, M. Goliszewski, and D. Kukoleca, "Via: Improving Internet telephony call quality using predictive relay selection," in *Proc. Conf. ACM SIGCOMM Conf. SIGCOMM*, 2016, pp. 286–299.

[11] Z. N. Abdullah, I. Ahmad, and I. Hussain, "Segment routing in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 464–486, 1st Quart., 2019.

[12] 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," in *Proc. 4th Eur. Workshop Softw. Defined Netw.*, Sep. 2015, pp. 111–112, doi: 10.1109/EWSDN.2015.73.

[13] J.-P. Sheu and Y.-C. Chen, "A scalable and bandwidth-efficient multicast algorithm based on segment routing in software-defined networking," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6, doi: 10.1109/ICC.2017.7997197.

[14] A. Bahnasse, F. E. Louhab, H. Ait Oulahyane, M. Talea, and A. Bakali, "Novel SDN architecture for smart MPLS traffic engineering-DiffServ aware management," *Future Gener. Comput. Syst.*, vol. 87, pp. 115–126, Oct. 2018.

[15] M.-C. Lee and J.-P. Sheu, "An efficient routing algorithm based on segment routing in software-defined networking," *Comput. Netw.*, vol. 103, pp. 44–55, Jul. 2016.

[16] C. E. Hopps, *Analysis of an Equal-Cost Multi-Path Algorithm*, document RFC 2992, Nov. 2000. [Online]. Available: https://datatracker.ietf.org/doc/rfc2992/

[17] J. Zhou, M. Tewari, M. Zhu, A. Kabbani, L. Poutievski, A. Singh, and A. Vahdat, "WCMP: Weighted cost multipathing for improved fairness in data centers," in *Proc. 9th Eur. Conf. Comput. Syst. EuroSys*, Apr. 2014, pp. 1–14, doi: 10.1145/2592798.2592803.

[18] A. Ford, C. Raiciu, and M. Handley, *Architectural Guidelines for Multipath TCP Development*, document RFC 6182, Mar. 2011. [Online]. Available: https://datatracker.ietf.org/doc/rfc6182/

[19] J. R. Iyengar, P. D. Amer, and R. Stewart, "Concurrent multipath transfer using SCTP multihoming over independent End-to-End paths," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 951–964, Oct. 2006.

[20] S. Zannettou, M. Sirivianos, and F. Papadopoulos, "Exploiting path diversity in datacenters using MPTCP-aware SDN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 539–546.

[21] J. Duan, Z. Wang, and C. Wu, "Responsive multipath TCP in SDN-based datacenters," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2015, pp. 5296–5301, doi: 10.1109/ICC.2015.7249165.

[22] P. Manzanares-Lopez, J. P. Munoz-Gea, and J. Malgosa-Sanahuja, "An MPTCP-compatible load balancing solution for pools of servers in OpenFlow SDN networks," in *Proc. 6th Int. Conf. Softw. Defined Syst. (SDS)*, Jun. 2019, pp. 39–46.

[23] F. Alharbi and Z. Fei, "An SDN architecture for improving throughput of large flows using multipath TCP," in *Proc. 5th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/ 4th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom)*, Jun. 2018, pp. 111–116.

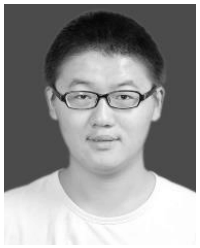
[24] H. Nam, D. Calin, and H. Schulzrinne, "Towards dynamic MPTCP path control using SDN," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Jun. 2016, pp. 286–294, doi: 10.1109/NETSOFT.2016.7502424.
[25] T.-S. Chen, D.-Y. Lee, T.-T. Liu, and A.-Y. Wu, "Dynamic reconfigurable ternary content addressable memory for OpenFlow-compliant low-power packet processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 10, pp. 1661–1672, Oct. 2016.
[26] J. Pang, G. Xu, and X. Fu, "SDN-based data center networking with collaboration of multipath TCP and segment routing," *IEEE Access*, vol. 5, pp. 9764–9773, 2017.
[27] A. A. Barakabitze, L. Sun, I.-H. Mkwawa, and E. Ifeachor, "A novel QoE-centric SDN-based multipath routing approach for multimedia services over 5G networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7, doi: 10.1109/ICC.2018.8422617.
[28] T. Stockhammer, "Dynamic adaptive streaming over HTTP: Standards and design principles," in *Proc. 2nd Annu. ACM Conf. Multimedia Syst. MMSys*, 2011, pp. 133–144.
[29] Y.-C. Chen, D. Towsley, and R. Khalili, "MSPlayer: Multi-source and multi-path video streaming," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 8, pp. 2198–2206, Aug. 2016.
[30] S. Mao, D. Bushmitch, S. Narayanan, and S. S. Panwar, "MRTP: A multiflow real-time transport protocol for ad hoc networks," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 356–369, Apr. 2006.
[31] V. Singh, T. Karkkainen, J. Ott, S. Ahsan, and L. Eggert, (Nov. 2014). *Multipath RTP (MPRTP) Draft-Singh-Avtcore-Mprtp-10, IETF Draft*. [Online]. Available: https://tools.ietf.org/html/draft-singh-avtcore-mprtp-10
[32] S. Deng and H. Balakrishnan, "Traffic-aware techniques to reduce 3G/LTE wireless energy consumption," in *Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol. CoNEXT*, 2012, pp. 181–192.
[33] R. Jain and S. Routhier, "Packet trains-measurements and a new model for computer network traffic," *IEEE J. Sel. Areas Commun.*, vol. SAC-4, no. 6, pp. 986–995, Sep. 1986.
[34] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 1990.
[35] F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN control: Survey, taxonomy, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 333–354, 1st Quart., 2018.
[36] C. Filsfils, S. Previdi, and G. Dawra, (Dec. 2017). *Segment Routing Centralized BGP Egress Peer Engineering*. Draft-IETF-Spring-Segment-Routing-Central-EPE-10. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-spring-segment-routing-central-epe/
[37] H. Gredler, J. Medved, and S. Previdi, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*, document RFC 7752, Mar. 2016. [Online]. Available: https://datatracker.ietf.org/doc/rfc7752/
[38] N. Kukreja, R. Alvizu, A. Kos, G. Maier, R. Morro, A. Capello, and C. Cavazzoni, "Demonstration of SDN-based orchestration for multi-domain segment routing networks," in *Proc. 18th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2016, pp. 1–4.
[39] S. Vissicchio, L. Vanbever, and O. Bonaventure, "Opportunities and research challenges of hybrid software defined networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 70–75, Apr. 2014.
[40] H. Xu, X.-Y. Li, L. Huang, H. Deng, H. Huang, and H. Wang, "Incremental deployment and throughput maximization routing for a hybrid SDN," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1861–1875, Jun. 2017.
[41] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 118–124, Oct. 2002.
[42] R. Guerin, D. Williams, and A. Orda, *QoS Routing Mechanisms and OSPF Extensions*, document IETF RFC 2676, 1999.

WEI ZHANG received the B.S. degree in computer science and technology from the Ocean University of China, Qingdao, in 2002, the M.S. degree in computer science and technology from the Institute of Computing Technology, Chinese Academy of Sciences, Shenyang, China, in 2005, and the Ph.D. degree in communication engineering from Northeastern University, Shenyang, China, in 2014.

From 2005 to 2009, she was a Research Assistant with the Institute of Computing Technology, Chinese Academy of Sciences. Since 2010, she has been a Lecturer with the Department of Communication and Electronic Engineering, College of Computer Science and Engineering, Northeastern University. She has published more than ten research articles in IEEE journals and international conferences. She is also an Inventor of four Chinese patents. Her research interests include multimedia communication, protocols and services in next generation networks, software-defined networking, and network architecture of cloud computing.

WEIMIN LEI (Member, IEEE) received the B.S. degree in computer science and technology from Nankai University, Tianjin, China, in 1992, the M.S. degree in computer science and technology from the Institute of Computing Technology, Chinese Academy of Sciences, in 1995, and the Ph.D. degree from the Dalian University of Technology, China, in 1995.

He is currently a Full Professor with the Department of Communication and Electronic Engineering, College of Information Science and Engineering, Northeastern University, Shenyang, China. He has published more than seventy research articles in IEEE journals and international conferences. He is also an Inventor of ten Chinese patents and pending applications. His research interests include protocols and services in IP multimedia systems, multipath transmission, overlay networks, and ad-hoc networks.



SONGYANG ZHANG received the B.S. degree in communication engineering from Northeastern University, Shenyang, China, in 2013, where he is currently pursuing the Ph.D. degree in communication engineering with the School of Computer Science and Engineering. His current research interests include multipath transmission for real-time traffic, network congestion control, and overlay networks.