IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 21, 2019, accepted September 20, 2019, date of publication September 24, 2019,
date of current version October 21, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943518

Centralized QoS Routing Using Network Calculus for SDN-Based Streaming Media Networks

SHENG ZHU[1,2], ZHEN SUN[1,2], YONG LU[1,2], LIANMING ZHANG[1,2], YEHUA WEI[1,2], AND GEYONG MIN[3], (Senior Member, IEEE)

[1]Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha 410081, China
[2]Key Laboratory of Internet of Things Technology and Application, College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China
[3]College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter EX4 4QF, U.K.

Corresponding author: Lianming Zhang (zlm@hunnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572191, and in part by the Key Research Foundation of Education Bureau of Hunan, China, under Grant 17A130.

ABSTRACT Streaming media transmission requires strict quality of service (QoS) parameters such as maximum delay and delay jitter. An effective streaming media routing algorithm is a key factor in ensuring QoS. The existing solution only considers a single parameter indicator in the performance parameters such as bandwidth, delay, and utilization of the link, and fails to comprehensively measure the data flow in the network. It is not possible to comprehensively measure the relationship between the business attributes and the QoS parameters. Firstly, the deterministic upper bounds of QoS parameters in streaming media networks are solved by using network calculus theory, and the QoS parameters are normalized, and a multi-constrained QoS resource allocation model is established; the separation of control and forwarding planes is defined by using software-defined networking (SDN) to deploy the multi-constrained QoS resource allocation model in the control plane; the QoS routing system of streaming media network based on the SDN is designed and implemented, including flow table scheduling model, routing function, measurement and forwarding modules. In the routing function module of the SDN controller, a multi-constrained QoS routing algorithm based on network calculus is implemented. Experimental results show that the proposed multi-constrained QoS resource allocation model based on network calculus and the multi-constrained centralized QoS routing algorithm based on the SDN have good performance.

INDEX TERMS SDN, network calculus, streaming media, multi-constrained QoS, centralized QoS routing.

I. INTRODUCTION

A. MOTIVATION: QOS OF STREAMING MEDIA

As the household use and penetration rates have increased, Over-The-Top (OTT) video streaming has become a mainstream behavior [1]. Recently, the streaming service OTT report 2018 released by ComScore, a global Internet information service provider in the United States [2], shows that the market of OTT is taking a stable growth trend; OTT devices are had by about 59.5 million homes in the United States. It grew by 17% a year, accounting for two-thirds of all internet-connected homes. Those homes spent 54 hours watching OTT video content in April 2018, increasing 28% in viewing time year-on-year. The use of streaming data increased by 73%, almost as much as that of computers.

Streaming transmission is the key technology to realize streaming media. In the case of streaming transmission, streaming media data has the basic characteristics of real-time and isochronous. The client terminal shall guarantee the synchronous relationship among various media in the streaming service period. Therefore, streaming media transmission has strict requirements on parameters of QoS such as maximum delay and delay jitter.

In the TCP/IP reference model, TCP protocol adopts a sliding window control mechanism. Data transmission is started and closed dynamically with the flow control window, which is difficult to meet the real-time and isochronous transmission requirements of streaming media. The connectionless feature of UDP protocol can improve the rate of transmission. Although it can meet the real-time requirements of streaming media to some extent, it cannot meet the need for streaming media transmission due to its unreliability.

The associate editor coordinating the review of this manuscript and approving it for publication was Mubashir Husain Rehmani.

To realize real-time transmission and broadcast of streaming media on IP networks, a communication control layer should be added between the transmission layer and application layer when designing a streaming media server, and corresponding real-time transmission protocols should be adopted, such as RTP, RTCP, and RTSP to realize the real-time transmission of streaming media data. However, the solution of IP-based streaming media mentioned above has a large amount of work to maintain and lacks flexible control functions and extension mechanism, which makes it difficult to meet the requirements of QoS, such as maximum delay and delay jitter of rapidly increasing streaming media data [3], [4].

B. BASICS: QOS ROUTING BASED ON THE SDN

QoS routing is a mechanism to select paths based on the available resources of networks and the QoS requirements of business flows. To realize QoS strategy and traffic awareness routing, an important prerequisite is to collect the traffic status information of the global network in real-time, such as transmission delay, bandwidth and packet loss rate of each link. The performance of any QoS routing algorithm is related to the accuracy of network state information obtained. That is, the more accurate the traffic state information is, the more accurate the QoS routing algorithm is in implementing network traffic scheduling.

The SDN is a new open and programmable network architecture by using the coupling idea of network control and forwarding [5]. The network administrator can flexibly choose the corresponding QoS strategy according to the users' demand for transmission flow, realizing the bandwidth allocation more efficiently, and making full use of the bandwidth resources of the whole network. The OpenFlow protocol is the first standard communication interface defined between the control layer and the forward layer in the SDN architecture [6]. The OpenFlow switch provides support for QoS by implementing a simple queue mechanism.

Under the SDN architecture, the centralized controller can obtain the traffic state information of the global network. The SDN uses the global network view to provide the configuration engines of OpenFlow for QoS control of each flow, to realize QoS routing optimization. There are existing researches about mainstream media modeling, analysis, and scheduling [7], [8], and routing in the SDN [9]–[18]. However, these QoS routes are imprecise for streaming media networks.

Network calculus is a network performance analysis tool [19] based on the min-plus algebra and the max-plus algebra, which can accurately solve the parameters of network QoS. Azodolmolky *et al.* [20] first used network calculus to model the upper bound of delay and queue length in SDN switch, and analyzed the buffer length of the SDN controller and switch. Guck *et al.* [21], [22] has studied the separation between fast routing and resource allocation in industrial Internet based on the separation of SDN control and forwarding by using network calculus theory. There are some existing studies on QoS of SDN and its routing based on network calculus [23]–[29]. However, these studies do not look at the impact of different business types on resource allocation and routing performance. Our previous works focused on a routing algorithm in a large-scale SDN [30] and a software-defined congestion control algorithm for IP networks [31].

In this paper, we study the resource allocation routing of multi-constraint QoS based on network calculus. Especially we study the deterministic multi-constraint centralized QoS routing framework in a software-defined streaming network environment. The main contribution of this paper is as follows:

- We propose a deterministic multi-constrained QoS resource allocation model based on network calculus. Based on the single-node QoS model and the end-to-end delay model, it normalizes the QoS parameters and comprehensively considers the impact of network bandwidth, delay, and switch buffer on streaming media transmission. While ensuring the QoS, link utilization is better improved.
- To guarantee the QoS of streaming media stream, we design and implement a deterministic multi-constrained centralized QoS routing (DMCQR) algorithm based on network calculus. The algorithm divides the service flow into a streaming media stream and other data flows. The routing of the streaming media stream depends on the QoS resource configuration model mentioned above for modeling analysis to select an optimal forwarding path.
- We extended the basic functions of the controller and built a DMCQR system based on software-defined streaming media network, including flow table scheduling model, functions in the routing system, measurement and forwarding modules, and DMCQR. The algorithm embeds a functional module.

The remainder of this paper is organized as follows. In Section II, related work is introduced. In Section III, we introduce the basic theory of network calculus and deduce the deterministic upper bound of QoS parameters, and establish the multi-constraint QoS resource model. We design the flow table scheduling model of the SDN switch and the architecture of the QoS routing system in Section IV. In Section V, we implement the routing system and routing algorithm and conduct experiments. A conclusion is made in Section VI.

II. RELATED WORK

A. QOS ROUTING BASED ON THE SDN

To guarantee the transmission of multimedia video streams with QoS, Egilmez *et al.* [9] proposed the OpenQoS which classifies transmission traffic into multimedia video streams and other data streams by matching rules for OpenFlow. For the multimedia video stream, OpenQoS adds a service layer on the OpenFlow controller in combination with the characteristics of delay and packet loss for the measurement on the transmission path, and selects a transmission path satisfying QoS parameters. Other data streams remain on the original shortest path. Given the large-scale SDN network deployment, Egilmez *et al.* further proposed a distributed control-plane framework based on OpenQoS to support QoS service demands of multi-operator in multimedia business flow. At the same time, an optimization framework for implementing end-to-end QoS services across multiple domains is proposed, and a message mechanism for QoS routing information interaction between controllers and different domains is designed. Ishimori *et al.* [10] made some improvement like extending the OpenFlow network data channel, adding QoS module, controlling multiple package scheduler of Linux kernel, providing QoS message to abstract the complexity of queue configuration and improving the flexibility of QoS control. Sonkoly *et al.* [11] extended the OCF and proposed QoS formalization. The implemented extension modules of the OCF include OCF Expedient, opt-in Manager, FlowVisor, and OF datapath. Fine-grained QoS control is realized on an experimental bed of the OCF. Ongar *et al.* [12] proposed a centralized management and orchestration framework in the SDN, to implement the real-time multimedia applications to distinguish the network services to achieve SLA. The framework defines an extended QoS architecture to seamless integrate the standard form of the SDN with other solutions, using SDN ability in the integration of wired and wireless environment for multimedia applications to provide network QoS. Jeong *et al.* [13] proposed a QNOX for the general OpenFlow/SDN, including SE, CE, MJE and CKE. The SE is responsible for the acceptance of user service requests, such as computing and storage capacity, requirements of QoS parameters, performance requirements and security levels. The ME is responsible for network resource discovery, multilayer/multi-domain QoS sensing virtual coverage network configuration, virtual network topology management and performance monitoring. Cui *et al.* [14] introduced a network classification technology applied in the SDN, which enabled networks to sense application, distinguish various application flows and know the needs of different flows. The technology takes advantage of new intelligent features to forward traffic based on bandwidth and delay requirements to improve QoS of applications and optimizing network resource allocation. Sharma *et al.* [15] combined SDN control with traditional network management functions and proposed an i-NMCS framework. The framework transforms the specific strategy of network operators into SDN control function, and makes routing selection and QoS configuration according to network state and flow demand. Kotronis *et al.* [16] proposed a CXP model, which uses orchestration based on SDN to receive requests for end-to-end paths guaranteed by QoS embeds paths in inter-domain virtual topologies and supervises QoS guarantees provided by supervision. Based on the SDN and NaaS paradigms, Bueno *et al.* [17] proposed a NCL, which provides a dynamic, on-demand end-to-end network resource supply mechanism and supports the management of different types of flows according to the QoS parameters of different types of flows, and allows them to be served by dynamically configuring networks. Liu *et al.* [18] proposed a scheme to optimize the quality of HTTP video using the SDN. Under this scheme, users can obtain video resources with higher QoS and QoE from adjacent routers. Fu and Wu [32] proposed the DMRA in SDN by using the residual disjoint paths as a network flow from the source node to the terminal node. This scheme gives a method of computing the shortest path from a single shortest path to multiple disjoint shortest paths.

B. QOS FRAMEWORK OF THE SDN BASED ON NETWORK CALCULUS

Azodolmolky *et al.* [20] first used the network calculus framework to analyze the behavior of SDN switches, including the delay and queue length boundary of the SDN switch, the buffer length of SDN controllers and SDN switches. An analysis model of the SDN based on network calculus theory was proposed. Qin *et al.* [23] designed an SDN architecture suitable for the Internet of Things (IoT) environment, in which the IoT SDN controller utilizes network calculus and genetic algorithm to optimize the IoT application. Guck and Kellerer [24] established communication services with end-to-end real-time service quality based on the SDN, and proposed a deterministic network model, and used network calculus to calculate the optimal path of each stream through the priority queue. Duan [25] proposed a network as a service framework based on the SDN, making it possible to support end-to-end network service orchestration. An abstract model of network service capability based on network calculus was proposed, and the bandwidth allocation of network service providing end-to-end QoS guarantee was studied. Heise *et al.* [26] and Koohanestani *et al.* [27] studied SDN network performance respectively based on network calculus. Chen *et al.* [28] proposed the deterministic delay guarantee of the dynamic service chain in the SDN based on network calculus. Nguyen *et al.* [29] modeled the software-defined wireless access network. Guck *et al.* established a network model for real-time QoS in an industrial environment based on an SDN using network calculus theory in [33], studied the separation between industrial Internet fast routing and resource allocation based on the separation thought of the SDN control plane and forward plane by using network calculus theory in [21], [22].

However, the above work only considers the single parameter index in the performance parameters such as bandwidth, delay and utilization of the link, and fails to comprehensively measure the relationship between the business attributes of the data flow in the network and the QoS parameters.

III. QOS MODEL FOR STREAMING MEDIA BASED ON NETWORK CALCULUS

A. NETWORK CALCULUS

Network calculus is a set of mathematical results obtained by an in-depth study of communication networks. It is a network performance analysis tool based on min-plus algebra and max-plus algebra [19], [34]–[36]. Network calculus calculates network performance parameters by the arrival curve and service curve, providing a theoretical framework for the QoS guarantee of streaming media in the SDN. This section mainly introduces the definitions and theorems of network calculus [34].

Definition 1: (Wide-Sense Increasing). If a function f is continuous and has a first derivative, then the set of wide-sense increasing functions is defined as

$$F = \{f(t)|f(0) \geq 0, \forall u \leq t, f(u) \leq f(t), t \in [0, +\infty]\} \quad (1)$$

where $F_0 = \{f(t)|f(t) \in F, f(0) = 0\}$.

Definition 2: (Infimum). The two functions f and g are non-negative wide-sense increasing functions, and the infimum of the functions f and g is

$$f \wedge g = \min\{f, g\}, f, g \in F \quad (2)$$

Definition 3: (Min-Plus Convolution). Let f and g be two non-negative wide-sense increasing functions and their values may be infinite, then the min-puls convolution of f and g is

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\} \quad (3)$$

for $t < 0$, we have $(f \otimes g)(t) = 0$.

Min-plus convolution has the associativity and commutativity.

$$(f \otimes g)(t) = (g \otimes f)(t) \quad (4)$$

and

$$(f \otimes g) \otimes h = f \otimes (g \otimes h) \quad (5)$$

In addition, if f and g are concave functions and $f(0) = g(0)$, we have $f \oplus g \leq \min\{f, g\}$.

Definition 4: (Arrival Curve). Given a non-negative wide-sense increasing function α, if and only if $I(t) - I(s) \leq \alpha(t-s), \forall s \leq t$, α is called the arrival curve of I, or I is restricted to the arrival curve α. It can also be expressed as

$$I(t) \leq (I \otimes \alpha)(t) \quad (6)$$

In particular, if the arrival curve is

$$\alpha(t) = \min\{pt + M, rt + b\} \quad (7)$$

where I is limited by $T-SPEC(M, p, r, b)$. In IntServ networks, (M, p, r, b) is an IETF traffic specification $T-SPEC$, which is widely used to illustrate the traffic characteristics of IntServ networks.

It is especially noted that if Eqs. (6) and (7) are meaningful, the functions are sub-additive.

Definition 5: (Sub-Additivity). When a function α has subadditivity, that is

$$\alpha(s+t) \leq \alpha(s) + \alpha(t), \forall s, t \geq 0 \quad (8)$$

Definition 6: (Service Curve). Assuming a traffic flow passes through a network system S, and its input and output functions are R and R^*, respectively, if and only if the wide-sense increasing function β satisfies

$$\beta(t) = \begin{cases} \beta(t) = 0, & t \leq 0 \\ R^* \geq \inf_{s \leq t}\{R(s) + \beta(t-s)\}, & t > 0 \end{cases} \quad (9)$$

where β is called the service curve provided by the system S for the business flow.

Theorem 1: (Nodes in Series). If a traffic flow passes through a series of nodes: S_1, S_2 and the service curve provided by each node S_i is β_i, where $i = 1, 2$, then the service curve β provided by the system with two nodes in series is

$$\beta = \beta_1 \otimes \beta_2 \quad (10)$$

Theorem 1 fully reflects the situation of burst adding a delay in the Internet. Considering that multiple nodes are connected in series, Theorem 2 can be obtained.

Theorem 2: (End-to-End Service Curve). When the service curve provided by node $i(i = 1, 2, N)$ is expressed as β_i, the end-to-end service curve β_{e2e} of N nodes in series is

$$\beta_{e2e} = \beta_1 \otimes \beta_2 \otimes \cdots \otimes \beta_i \otimes \cdots \otimes \beta_N \quad (11)$$

Theorem 3: (Backlog Bound). Assuming that a traffic flow with an arrival curve α passes through a network system S with a service curve β, at any time t, the backlog $R(t)-R^*(t)$ satisfies

$$R(t) - R^*(t) \leq \sup_{s \geq 0}\{\alpha(s) - \beta(s)\} \quad (12)$$

Backlog refers to the number of packets waiting for the service in system S at t time. In the first case, the system S is a complex network, so the backlog is represented as the amount of data transmitted by the system S. In the second case, the system S is a simple buffer and the backlog represents the queue length waiting for the service in the buffer.

Lemma 1: Assuming that the service curve of a network node satisfies the rate-latency function, the relationship between the rate R and the delay T is

$$T = \frac{C}{R} + \frac{L}{c} \quad (13)$$

where C is the maximum packet of traffic flow, L is the maximum packet of all traffic flows through the network node and c is the total rate of the scheduler.

B. QOS MODEL FOR SINGLE NODE

Assuming a streaming media flow limited by the arrival curve $T-SPEC(M, p, r, b)$ passes through system S shown in Fig. 1, and the service curve β of system S is the rate-latency function $\beta_{R,T}$, we have $\beta = \beta_{R,T}$. In this case, the service rate R is greater than or equal to the average rate r of traffic flow.

The maximum distance between α and β from the vertical direction can be measured in Fig. 1. According to Equ. 12, the upper bound of backlog can be obtained.

$$B_{max} = \max\{\alpha(T), \alpha(s) - \beta(s)\} \quad (14)$$

where $s = \frac{b-M}{p-r}$.

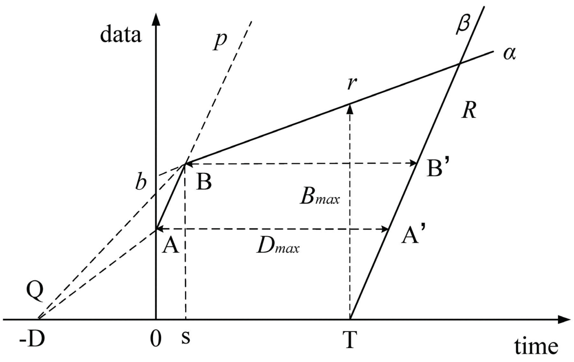


FIGURE 1. QoS parameters of streaming media.

Therefore, the upper bound D_{max} of delay is

$$D_{max} = \max\{\frac{\alpha(t)}{R} + T - s, \frac{M}{R} + T\} \tag{15}$$

Using min-plus algebra, we can obtain the required buffer size for streaming media, this is

$$B_{max} = r \times T + b + (\frac{b-M}{p-r} - T)^+ \times ((p-M)^+ - p + r) \tag{16}$$

The upper bound of the delay of streaming media is

$$D_{max} = \frac{M + \frac{(b-M)\times(p-R)^+}{p-r}}{R} + T \tag{17}$$

The required effective bandwidth is the maximum value in the slope of r, QA and QB, and we have

$$E_D = \max\{\frac{M}{D}, r, p \times (1 - \frac{D - \frac{M}{p}}{s+D})\} \tag{18}$$

where D is the maximum delay of QoS guarantee, and $(x)^+$ represents $\max\{0, x\}$.

C. END-TO-END DELAY MODEL

Resource reservation is accomplished by establishing connections between ports. The node i on the path assigns the connection rate R_i for streaming media, where $R_i \geq r$, and the service curve is $\beta_{R,T}$. According to Lemma 1, the rate R and the delay T in the rate-latency function of the entire path to the streaming media stream are

$$R = \min_{i=1,2,\cdots,N}\{R_i\} \tag{19}$$

and

$$T = \Sigma_{i=1}^{N}(\frac{C_i}{R_i} + D_i) \tag{20}$$

where C_i and D_i indicate respectively the rate-dependent and rate-independent delay deviations of routing for streaming media, so the values of C_i and D_i are determined by routing policies. After passing through N nodes, the rate-dependent and rate-independent delay deviations C_{tot} and D_{tot} of streaming media are respectively

$$C_{tot} = \Sigma_{i=1}^{N} C_i \tag{21}$$

and

$$D_{tot} = \Sigma_{i=1}^{N} D_i \tag{22}$$

According to Eqs. (20), (21) and (23), we have

$$T = \frac{C_{tot}}{R_n} + D_{tot} - d \tag{23}$$

where $d = \Sigma_{i=1}^{N} C_i(\frac{1}{R} - \frac{1}{R_i})$. By Equ. (17), if so $R \geq r$, we obtain the upper bound of end-to-end delay, this is

$$D_{max} = \frac{M}{R} + \frac{(b-M)\times(p-R)^+}{R(p-r)} + \frac{C_{tot}}{R} + D_{tot} - d \tag{24}$$

The fixed delay of the transmission process, such as the delays of routing information uploading controllers and controllers pushing flow table, is expressed by the delay function $\delta_{d_{tot}}$, and the service curve can be represented by $\beta_{R,T} \otimes \delta_{d_{tot}}$. We obtain the upper bound of end-to-end delay as follows

$$D_{max} = \frac{M}{R} + \frac{(b-M)\times(p-R)^+}{R(p-r)} + \frac{C_{tot}}{R} + D_{tot} + \delta_{d_{tot}} - d \tag{25}$$

D. MULTI-CONSTRAINED QOS RESOURCE ALLOCATION MODEL

By considering the performance indicators of network links, such as remaining bandwidth, delay, and packet loss rate, we can obtain the most suitable path. Because of the different meanings of these network performances, it is necessary to standardize the performance indicators by making them comparable, that is, we need to normalize the performance indicators. In this paper, we adopt min-max normalization as shown in Equ. (26).

$$x^{\Delta} = \frac{x - min\{\}}{max\{\} - min\{\}} \tag{26}$$

where $max\{\}$ represents the maximum value, and $min\{\}$ means the minimum. We have

$$E(x^{\Delta}) = \frac{E(p) - min\{E(path)\}}{max\{E(path)\} - min\{E(path)\}} \tag{27}$$

and

$$D(x^{\Delta}) = \frac{D(p) - min\{D(path)\}}{max\{D(path)\} - min\{D(path)\}} \tag{28}$$

and

$$B(x^{\Delta}) = \frac{B(p) - min\{B(path)\}}{max\{B(path)\} - min\{B(path)\}} \tag{29}$$

where the remaining bandwidth of path p is $E(p)$, the total delay $D(p)$ of path p is the sum of the delay of all links in the path and the buffer size of nodes is expressed by $B(p)$.

When there are many paths to choose, the Packet-in message first determines which type of service the packet belongs to, and then calculates the QoS resource allocation on the

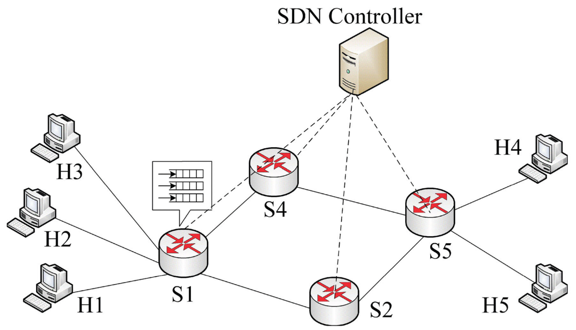


FIGURE 2. An implementation scenario.

corresponding path. Using Equ. (30), we can solve the problem of the multi-constrained QoS resource allocation $W(p)$ on path p, this is

$$W(p) = \alpha \times E(x^{\triangle}) - \beta \times D(x^{\triangle}) - \gamma \times B(x^{\triangle}), p \in path \tag{30}$$

The remaining bandwidth in Equ. (30) is positively related to W, so it $E(x^{\triangle})$ is positive. $D(x^{\triangle})$ and $B(x^{\triangle})$ is negatively related to W, so they are negative. Because streaming media has different QoS requirements for remaining bandwidth, latency, and buffer size, the values of α, β and γ are also different in W. When streaming media is routing, it is necessary to find a path that satisfies both low delay and high remaining bandwidth. The larger the W value indicates that the path is more suitable for streaming media. So, we have $\alpha \geq \gamma \geq \beta$.

IV. MULTI-CONSTRAINED QOS ROUTING SYSTEM DESIGN

The deterministic multi-constrained QoS routing mechanism based on the SDSMN is as follows. First, the SDSMN controller selects a path conforming to the streaming media QoS requirement by using the function module according to the QoS requirement of the streaming media data; the SDSMN controller pushes the flow table and rules to configure the route in nodes; it forwards the streaming media data according to the rules. The related model and architecture design are described as shown.

A. FLOW TABLE SCHEDULING MODEL IN SWITCH

In the SDSMN, the packets of different applications come into SDSMN switches, and they form multiple queues. The flow table is pushed by programming the controller so that these queues are properly scheduled to implement the forwarding of different types of traffic. An implementation scenario is shown in Fig. 2.

To meet different performance requirements, we distinguish the data flows and divide the data flows into streaming media flows and other flows. Streaming media flows have a higher priority than other flows, while the same type of flows adopts a fair queuing principle. In this paper, we propose a hybrid scheduling model with aggregate priority, as shown in Fig. 3.

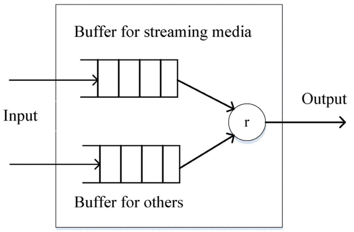


FIGURE 3. Hierarchical scheduling model.

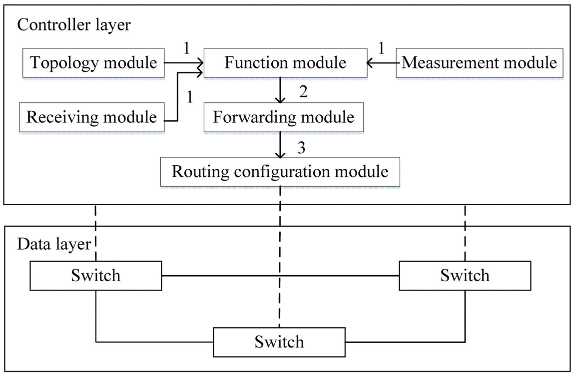


FIGURE 4. QoS routing architecture.

B. QOS ROUTING ARCHITECTURE

The QoS routing architecture consists of a controller layer and an underlying device layer of the SDSMN, as shown in Fig. 4. The workflow of the QoS routing system is as follows: (1) If the streaming media flows with QoS requirements arrive, the packet information of flows is routed to the controller. The receiving module of the controller receives the packet information and sends it to the function module; (2) the network resource status path is Obtained from the measurement module, and the network topology is obtained from the topology module; (3) the routing configuration module is responsible for configuring the switches on the path; (4) after the routing configuration module configures the route, the forwarding module is invoked, and the forwarding module generates a new flow entry and sends it to the switch.

In the QoS routing architecture, the topology of the network is recorded in the topology module. The main function of the topology module is to monitor the change of the network topology and update the topology in time. The receiving module is configured to receive data packets. The routing configuration module is responsible for the configuration of the switch. In this paper, we focus on the function, measurement and forwarding modules.

1) FUNCTION MODULE

The function module is the core of the QoS routing architecture. When the data packet arrives at the routing node,

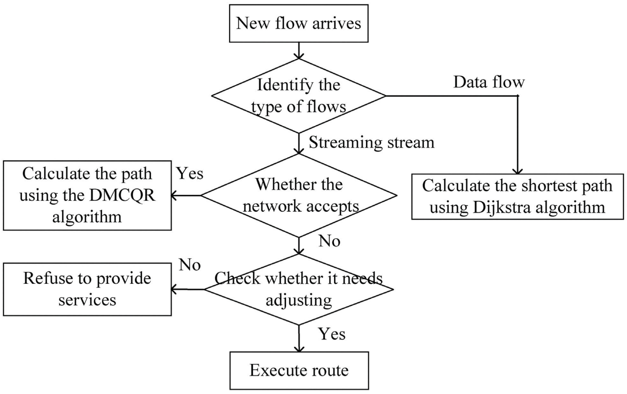


FIGURE 5. The process of function module.

the route reports to the SDSMN controller, and the SDSMN controller recognizes the type of the data flow. If it is not a streaming stream, the shortest path is calculated directly using the Dijsktra algorithm. If it is a streaming media stream, the network accomplishes acceptance judgment. If it is accepted, we use the proposed DMCQR routing algorithm to calculate the path. If it is not accepted, we check whether it needs adjusting, and then we execute the route based on the result. The process of the function module is shown in Fig. 5.

2) MEASUREMENT MODULE

When the function module selects the optimal path, in addition to obtaining the underlying network topology from the topology module, it should also obtain the allocation of network resources, so that when selecting a path for the streaming media, the link with relatively less load can be preferentially selected, thereby reducing the impact of the application flow.

The OpenFlow protocol allows administrators and programmers to obtain information such as flow entries, ports, and queues on the underlying route. They can also obtain information such as the number of packets received and sent in the route, the number of bytes, the corresponding time, and the number of lost packets. The principle of collecting these routing information is that the SDSMN controller sends a Flow Statistics Request message to the route and then queries the routing information at regular intervals. The shorter the interval, the more accurate the obtained routing information. The SDSMN controller can obtain the underlying network resource status and network performance indicators by querying these routing information. It is also because of the features of the SDSMN that the SDSMN has strong network monitoring capabilities.

The measurement module includes link discovery, packet loss rate measurement, and delay measurement. Where the delay is the more important network performance parameter in QoS, but there is no function to measure the network delay in the measurement module of the SDN controller, such as Floodlight. The tester sends a sniffing packet with a timestamp to the node through the SDSMN controller, passes the sniffing packet to another node, and finally passes it back to the SDSMN controller. The delay between the two nodes is calculated using the time difference between the back and forth.

3) FORWARDING MODULE

The forwarding module of the SDSMN controller is mainly to operate on the flow table and forward the flow table to the router. In the forwarding module, the Forwarding class extends the Forwarding Base which is abstract class, and implements the IOFMessageListener. The processing procedure of the forwarding module is as follows: receiving a message, checking and determining the message type, invoking a routing decision, delivering a flow table, and installing a flow table.

V. DMCQR ALGORITHM

The function module of the QoS routing system classifies the flows. If the data is a non-streaming media, the Dijkstra algorithm is used to calculate the shortest routing path, otherwise, the proposed DMCQR algorithm is used to calculate the optimal routing path. The process of the DMCQR algorithm is as follows: When the SDSMN controller obtains the Packet-in packet, the DMCQR algorithm is first used to perform statistics on all paths from the source route to the destination route and judge the number N of paths obtained by the algorithm. If $N < 1$, it returns null. If $N = 1$, this path is returned. If $N > 1$, the corresponding QoS resource allocation $W(p)$ for all paths is calculated by Equ. 30, and the path corresponding to the maximum QoS resource allocation $W(p)$ is returned. The pseudocode of the DMCQR algorithm is shown in Algorithm 1.

VI. PERFORMANCE EVALUATION

A. EXPERIMENTAL ENVIRONMENT

The experimental environment was run under Ubuntu 16.04 STL i386. We tested the performance of DMCQR using the RYU controller and Mininet [37]. RYU is a lightweight SDN controller that can be customized to integrate python scripts. RYU is a lightweight SDN controller that can be customized to integrate python scripts. Mininet is an open-source software developed by Stanford University, OpenFlow-based network controllers prototyped in Mininet can usually be transferred to hardware with minimal changes for full line-rate execution.

First, we start the RYU controller. The RYU controller loads basic modules based on its profile information, including the topology module and link discovery module. The topology is created by Mininet connected to the RYU controller. The link discovery module and the topology module construct the topology of the entire network by processing Packet-in messages. If you need to collect the underlying network information, you can configure the network awareness module in the software-defined streaming media network controller. By configuring it with OpenvSwitch, a switch that supports the OpenFlow protocol, real-time underlying

Algorithm 1 DMCQR Algorithm

```
Input: Network topology $G(V, E)$, where $V$ and $E$ are respectively a set of routing nodes and a set of paths in a SDSMN; Packet-in packets.
Output: $PS$.
Obtaining a set $Paths$ of paths in $G(V, E)$ by using a network topology module.
if $Paths.size() > 1$ then
    Selecting a set $PS$ of paths which satisfy the QoS parameters of the flows.
    if $PS.size() > 1$ then
        for $all\ p \in PS$ do
            According to the value of $(M, p, r, b)$ of the flows, calculating QoS resource allocation $W(p)$ using $x^{\triangle}$ in Equ. 26.
            Storing the path $p$ and its $W(p)$ in $PS$.
        end for
        Path $\rightarrow$ Selecting a path with the largest QoS resource allocation $W(p)$ in $PS$.
    else
        if $PS.size() == 1$ then
            Path $\rightarrow$ $PS$.
        else
            Path $\rightarrow$ Selecting a path from the paths that satisfies the minimum number of hops and the larger available bandwidth.
        end if
    end if
else
    if $Paths.size() == 1$ then
        Path $\rightarrow$ $Paths$.
    else
        Path $\rightarrow$ NULL.
    end if
end if
Return $PS$.
```

network information can be collected. The network awareness module can use OpenFlow to send HTTP requests in order to obtain real-time link bandwidth usage.

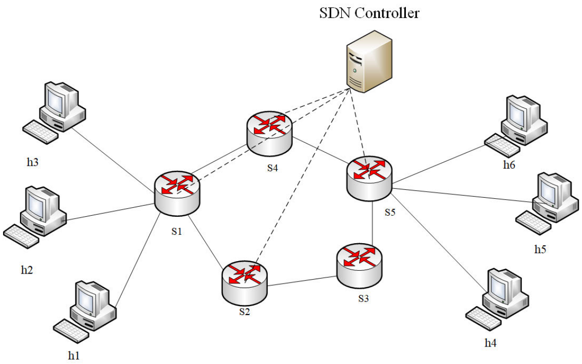


FIGURE 6. An example of experimental topology.

TABLE 1. Link parameters.

link	Delay	Bandwidth
P1	3 ms	10 Mbits
P2	2 ms	9 Mbits

TABLE 2. Parameters of QoS resource allocation.

α	β	γ
0.4	0.2	0.2

A physical host and a virtual machine are used in our experiment. The RYU controller is installed as a device of the control layer on the physical host and its address is 192.168.253.3. The Virtualbox whose address is 192.168.56.101 is installed on the physical host and the Mininet is installed on the Virtualbox to build the network topology. The underlying network is built on the Mininet, including six hosts (h1-h6) and five switches S1, S2, S3, S4 and S5. The experimental topology is shown in Fig. 6. Where path P1 represents path S1 - S4 - S5, and path P2 represents path S1 - S2 - S3 - S5. The bandwidth and delay are shown in Tab. 1. The parameters of the QoS resource allocation in Equ. 30 is shown in Tab. 2. We tested a number of α, β and γ values and found that the parameters did not affect the experimental results when the $\alpha > \beta > \gamma$ rule was followed. The link parameters used in the simulation experiment are compared with the data center to reduce and simplify the analogy, so the network is still in line with the real network.

B. EXPERIMENT RESULTS AND ANALYSES

In this section, we compare the DMCQR algorithm with DMRA and Dijkstra. The experimental topology is shown in Fig. 6, which considers total throughput, latency, and link utilization.

1) COMPARISON OF TOTAL THROUGHPUT OF DIFFERENT ALGORITHMS

In this experiment, we use Iperf network performance software to send the UDP packet test, the transmission bandwidth and time are shown in Fig. 7, the experimental comparison diagram is shown in Fig. 8. The h1-h4 transmission bandwidth is 3 Mbps, the h2-h5 transmission bandwidth is 6 Mbps, and the h3-h6 transmission bandwidth is 8 Mbps, and transmission starts at $t = 0$, $t = 5$, and $t = 10$ seconds, respectively. Taking h1-h4 as an example, the host h1 sends a message to the host h4 (ip address is 10.0.0.4) by the following command.

```
iperf -c 10.0.0.4 -5 0x10 -u -i 5 -t 30 -b 3M
```

For the Dijkstra algorithm, the source, and destination addresses of the three traffic flows are the same, so it path is the same, but since the capacity of P1 is only 10 M, the maximum throughput should not exceed 10 M, which is consistent

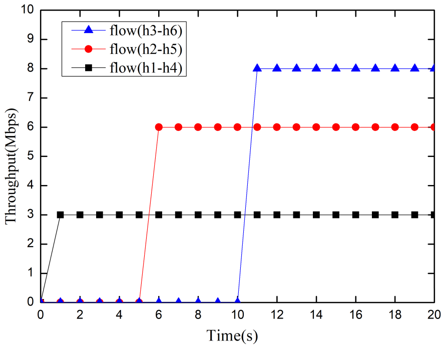


FIGURE 7. The client sends traffic at different times.

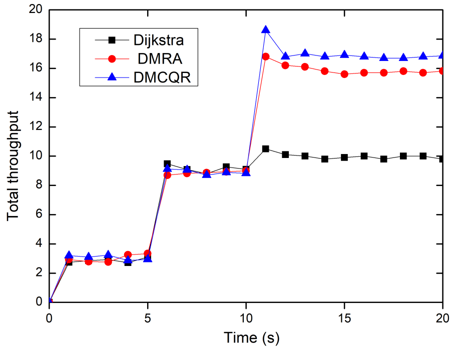


FIGURE 8. Throughput of different algorithms.

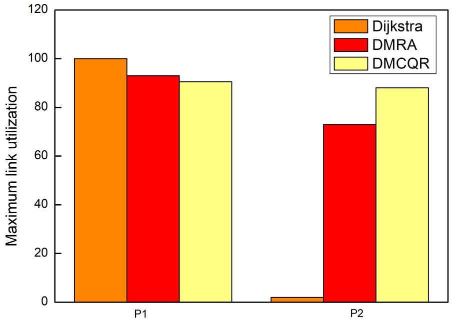


FIGURE 9. Link utilization of P1 and P2.

with the experimental results; for the DMRA algorithm The routable path has two paths, P1 and P2, so the final total throughput maybe 13 Mbps, 16 Mbps and 17 Mbps. From the experimental results, the total throughput is roughly stable at 16 Mbps; for the DMCQR algorithm, two are compared. At the same time of the remaining bandwidth in the link, the delay and the buffer are also counted. The routing decision is not simply made because of the remaining bandwidth of the link and the total throughput is about 17 Mbps.

2) COMPARISON OF MAXIMUM LINK UTILIZATION

We sent three different service flows in this experiment. The bandwidth size and time are shown in Fig. 7. And when the network traffic is based on stability, we compare the link utilization of different algorithms. The utilization size and path are shown in Fig. 9.

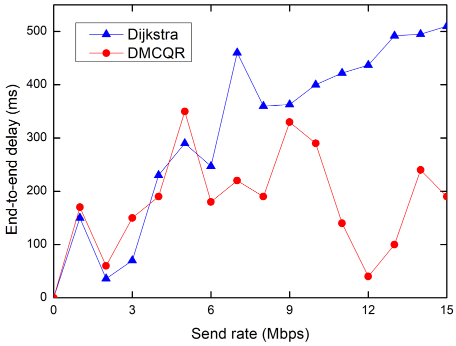


FIGURE 10. End-to-end delay for Dijkstra vs DMCQR.

As can be seen from Fig. 8, the Dijkstra algorithm has a utilization rate of 100% on the P1 path and 0% on the P2 path, which is related to the fact that Dijkstra always selects the least hop path. The utilization rate of the DMRA algorithm on the P1 and P2 paths are 93% and 73%, respectively. Generally, when the link utilization is above 95%, congestion is considered, and the higher link utilization also represents more. Link congestion is prone to occur, so pay attention to the possibility of congestion in the DMRA algorithm occurring on the P1 link. The utilization rate of the DMCQR algorithm on the links is 90% and 88% respectively. In comparison, the algorithm makes better use of network resources under the premise of avoiding congestion.

3) COMPARISON OF DELAY AND JITTER

This experiment compares the end-to-end delays of the Dijkstra, DMRA, and DMCQR algorithms at different transmission rates, as shown in Figs. 10 and 11. The delay of the Dijkstra algorithm is increasing. The main reason is that as the transmission rate increases, the load in the path increases. When the load of the path approaches or exceeds the bearer value of the path, the service data packet waits in the queue. Increase, and then the corresponding delay increases. For DMRA and DMCQR algorithms, the delays increase first, then decrease and then increase. The main reason is that with the decrease of available resources of the network, the algorithm selects a more suitable path for the data packets, effectively slowing down the link. Congestion is possible, and the delay is better protected.

As the transmission rate increases gradually, the DMCQR algorithm shows better delay guarantee performance, but in the first half of the experiment, the delays of the DMRA and DMCQR algorithms are relatively close, so we compare the jitter levels of the two algorithms, such as Fig. 12 shows. The DMRA algorithm has a large fluctuation range, which can reach 210 ms at the highest, and the jitter of the DMCQR algorithm is roughly 150 ms.

In summary, the above results show that the deterministic multi-constrained QoS routing framework proposed in this paper can quickly make routing decisions while providing accurate QoS guarantee. In terms of bandwidth utilization,

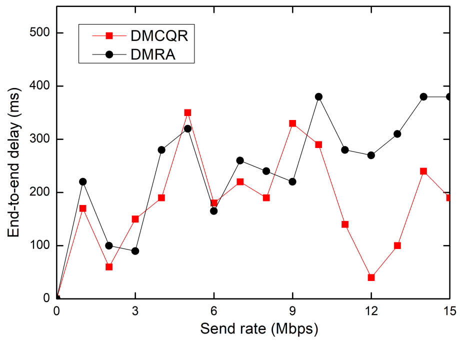


FIGURE 11. End-to-end delay for DMRA vs DMCQR.

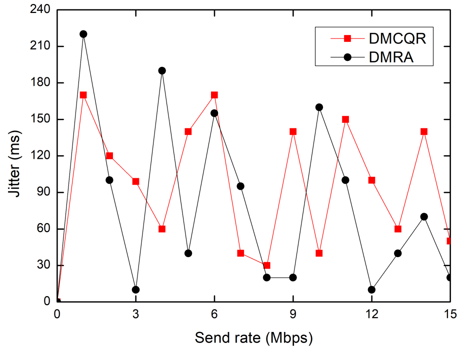


FIGURE 12. Load rate vs. time.

packet loss rate, main path load rate and end-to-end delay, the proposed DMCQR algorithm has better performance.

VII. CONCLUSION

In this paper, we investigated a framework of the QoS of streaming media, which can provide accurate QoS guarantee and fast routing. Based on the network calculus theory, we established the model of QoS parameters of deterministic buffer size, effective bandwidth, delay, etc. On this basis, we have established the multi-constraint QoS resource allocation model. We used the separation thought of network control and forwarding and deployed the deterministic multi-constraint QoS resource allocation model in the SDSMN controller, to realize the fast QoS routing of streaming media.

We designed the deterministic multi-constraint QoS routing system based on the SDSMN, and explored and developed the modules of the routing system such as functions, measurement and forwarding to realize the DMCQR algorithm. Experiment results show that the proposed DMCQR algorithm had better performance in terms of effective bandwidth utilization rate, packet loss rate, path load rate and end-to-end delay compared with the performance of the Dijsktra algorithm.

REFERENCES

[1] J. Nogueira, L. Guardalben, B. Cardoso, and S. Sargento, "Catch-up TV forecasting: Enabling next-generation over-the-top multimedia TV services," *Multimedia Tools Appl.*, vol. 77, no. 12, pp. 14527–14555, Jun. 2018.

[2] S. Engleson. *State of OTT*. Accessed: Aug. 22, 2019. [Online]. Available: https://www.comscore.com/Insights/Pres-entations-and-Whitepapers/2018/State-of-OTT

[3] S. Keshav, "Paradoxes of Internet architecture," *IEEE Internet Comput.*, vol. 22, no. 1, pp. 96–102, Jan./Feb. 2018.

[4] C. Yang and Y. Liu, "On achieving short channel switching delay and playback lag in IP-based TV systems," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 1096–1106, Jul. 2015.

[5] N. Mckeown, "*Software-defined networking*," INFOCOM Keynote Talk, 2009.

[6] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[7] W. Miao, G. Min, Y. Wu, H. Wang, and J. Hu, "Performance modelling and analysis of software-defined networking under bursty multimedia traffic," *ACM Trans. Multimedia Computing, Commun., Appl.*, vol. 12, no. 5, Dec. 2016, Art. no. 77.

[8] J. Huang, L. Xu, L. Xu, C.-C. Xing, J. Lu, and S. Yu, "Modeling and performance analysis for multimedia data flows scheduling in software defined networks," *J. Netw. Comput. Appl.*, vol. 83, pp. 89–100, Apr. 2017.

[9] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "Open QoS: An OpenFlow controller design for multimedia delivery with end-to-end quality of service over software-defined networks," in *Proc. Asia Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Dec. 2012, pp. 1–8.

[10] A. Ishimori, F. Farias, E. Cerqueira, and A. Abelém, "Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking," in *Proc. Eur. Workshop Softw. Defined Netw.*, Oct. 2013, pp. 81–86.

[11] B. Sonkoly, A. Gulyas, F. Németh, J. Czentye, K. Kurucz, B. Novák, and G. Vaszkun, "On QoS support to Ofelia and OpenFlow," in *Proc. Eur. Workshop Softw. Defined Netw.*, Oct. 2012, pp. 109–113.

[12] F. Ongaro, E. Cerqueira, L. Foschini, A. Corradi, and M. Gerla, "Enhancing the quality level support for real-time multimedia applications in software-defined networks," in *Proc. Int. Conf. Comput., Netw. Commun.*, Feb. 2015, pp. 505–509.

[13] K. Jeong, J. Kim, and Y.-T. Kim, "QoS-aware network operating system for software defined networking with generalized OpenFlows," in *Proc. Netw. Oper. Manage. Symp.*, Apr. 2012, pp. 1167–1174.

[14] H. Cui, Y. Zhu, Y. Yao, Y. Li, and Y. Liu, "Design of intelligent capabilities in SDN," in *Proc. Int. Conf. Wireless Commun., Veh. Technol., Inf. Theory Aerosp. Electron. Syst.*, May 2014, pp. 1–5.

[15] P. Sharma, S. Banerjee, S. Tandel, R. Aguiar, R. Amorim, and D. Pinheiro, "Enhancing network management frameworks with SDN-like control," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, May 2013, pp. 688–691.

[16] V. Kotronis, X. A. Dimitropoulos, R. Kloti, B. Ager, P. Georgopoulos, and S. Schmid, "Control exchange points: Providing QoS-enabled end-to-end services via SDN-based inter-domain routing orchestration," in *Proc. Res. Track 3rd Open Netw. Summit*, 2014. Accessed: Aug. 22, 2019. [Online]. Available: https://www.usenix.org/system/files/conference/ons2014/ons2014-paper-kotronis.pdf

[17] I. Bueno, J. I. Aznar, E. Escalona, J. Ferrer, and J. A. Garcia-Espin, "An OpenNaaS based SDN framework for dynamic QoS control," in *Proc. IEEE SDN for Future Netw. Services*, Nov. 2013, pp. 1–7.

[18] H. Liu, Y. Hu, G. Shou, and Z. Guo, "Software defined networking for HTTP video quality optimization," in *Proc. 15th IEEE Int. Conf. Commun. Technol.*, Nov. 2014, pp. 413–417.

[19] L. Zhang, J. Liu, and K. Yang, "Quality of service modelling of virtualized wireless networks: A network calculus approach," *Mobile Netw. Appl.*, vol. 19, no. 4, pp. 572–582, Aug. 2014.

[20] S. Azodolmolky, R. Nejabati, M. Pazouki, P. Wieder, R. Yahyapour, and D. Simeonidou, "An analytical model for software defined networking: A network calculus-based approach," in *Proc. IEEE Global Commun. Conf.*, Dec. 2013, pp. 1397–1402.

[21] J. W. Guck, M. Reisslein, and W. Kellerer, "Model-based control plane for fast routing in industrial QoS network," in *Proc. IEEE 23rd Int. Symp. Qual. Service (IWQoS)*, Jun. 2015, pp. 65–66.

[22] J. W. Guck, M. Reisslein, and W. Kellerer, "Function split between delay-constrained routing and resource allocation for centrally managed QoS in industrial networks," *IEEE Trans. Ind. Informat.*, vol. 12, no. 6, pp. 2050–2061, Dec. 2016.

[23] Z. Qin, G. Denker, C. Giannelli, and P. Bellavista, "A software defined networking architecture for the Internet-of-Things," in *Proc. IEEE Netw. Oper. Manage. Symp.*, May 2014, pp. 1–9.

[24] J. W. Guck and W. Kellerer, "Achieving end-to-end real-time quality of service with software defined networking," in *Proc. IEEE 3rd Int. Conf. Cloud Netw.*, Oct. 2014, pp. 70–76.

[25] Q. Duan, "Network-as-a-service in software-defined networks for end-to-end QoS provisioning," in *Proc. 23rd Wireless Opt. Commun. Conf.*, May 2014, pp. 1–5.

[26] P. Heise, F. Geyer, and R. Obermaisser, "Deterministic OpenFlow: Performance evaluation of SDN hardware for avionic networks," in *Proc. Int. Conf. Netw. Service Manage.*, Nov. 2015, pp. 372–377.

[27] A. K. Koohanestani, A. G. Osgouei, H. Saidi, and A. Fanian, "An analytical model for delay bound of OpenFlow based SDN using network calculus," *J. Netw. Comput. Appl.*, vol. 96, pp. 31–38, Oct. 2017.

[28] Y.-J. Chen, L.-C. Wang, F.-Y. Lin, and B.-S. P. Lin, "Deterministic quality of service guarantee for dynamic service chaining in software defined networking," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 991–1002, Dec. 2017.

[29] K. Nguyen, K. Ishizu, and F. Kojima, "An evolvable, scalable, and resilient control channel for software defined wireless access networks," *Comput. Electr. Eng.*, vol. 57, pp. 104–117, Jan. 2017.

[30] L. Zhang, Q. Deng, Y. Su, and Y. Hu, "A box-covering-based routing algorithm for large-scale SDNs," *IEEE Access*, vol. 5, pp. 4048–4056, 2017.

[31] Y. Hu, T. Peng, and L. Zhang, "Software-defined congestion control algorithm for IP networks," *Sci. Program.*, vol. 2017, Dec. 2017, Art. no. 3579540.

[32] M. Fu and F. Wu, "Investigation of multipath routing algorithms in software defined networking," in *Proc. Int. Conf. Green Inform.*, Aug. 2017, pp. 269–273.

[33] J. W. Guck, A. Van Bemten, and W. Kellerer, "DetServ: Network models for real-time QoS provisioning in SDN-based industrial environments," *IEEE Trans. Netw. Service Manag.*, vol. 57, no. 4, pp. 1003–1017, Dec. 2017.

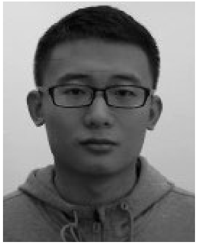
[34] J. Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Berlin, Germany: Springer-Verlag, 2004.

[35] L. Zhang, J. Yu, and X. Deng, "Modelling the guaranteed QoS for wireless sensor networks: A network calculus approach," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, Aug. 2011, Art. no. 82.

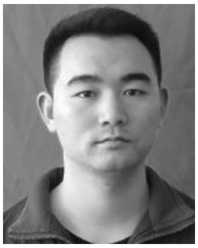
[36] Y. Long, Z. Lu, and H. Shen, "Composable worst-case delay bound analysis using network calculus," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 3, pp. 705–709, Mar. 2018.

[37] *Mininet*. Accessed: Aug. 22, 2019. [Online]. Available: http://mininet.org

SHENG ZHU was born in Changde, Hunan, China, in 1993. She received the B.S. degree in computer science from the Hunan University of Humanities, Science and Technology, Loudi, China, in 2015. She is currently pursuing the master's degree with Hunan Normal University. Her research interests include computer networks and software-defined networking.



ZHEN SUN was born in Wenshang, Shandong, China, in 1994. He received the B.S. degree in materials science and engineering from Jishou University, Changsha, China, in 2016. He is currently pursuing the master's degree with Hunan Normal University. His research interests include computer networks, software-defined networking, and network calculus.



YONG LU was born in Xiangyang, Hubei, China, in 1994. He received the B.S. degree in computer science and technology from Wuhan Technology and Business University, Changsha, China, in 2016. He is currently pursuing the master's degree with Hunan Normal University. His research interests include computer networks and software-defined networking.

LIANMING ZHANG received the B.S. and M.S. degrees from Hunan Normal University, China, and the Ph.D. degree from Central South University, China. He completed a two-year Postdoctoral Fellowship in complex networks with the South China University of Technology. He manages research projects funded by various sources, such as the National Natural Science Foundation of China and industries. He is currently a Professor with the College of Information Science and Engineering, Hunan Normal University, where he leads the Key Laboratory of the Internet of Things Technology and Application. He has authored more than 100 articles. His research interests include the Internet of Things, edge computing, and software-defined networking.

YEHUA WEI received the Ph.D. degree in computer science and engineering from Hunan University, Changsha, China, in 2009. He is currently an Associate Professor with the College of Information Science and Engineering, Hunan Normal University, Changsha. His research interests include embedded systems, cyber-physical systems, and the Internet of Things. He is a member of the China Computer Federation.

GEYONG MIN received the B.Sc. degree in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1995, and the Ph.D. degree in computing science from the University of Glasgow, Glasgow, U.K., in 2003. He is currently a Professor of high-performance computing and networking with the Department of Mathematics and Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter, U.K. His research interests include the next-generation Internet, wireless communications, multimedia systems, information security, high-performance computing, ubiquitous computing, modeling, and performance engineering.

• • •