Joint Admission Control and Routing Via Approximate Dynamic Programming for Streaming Video Over Software-Defined Networking

Jian Yang, *Senior Member, IEEE*, Kunjie Zhu, Yongyi Ran, Weizhe Cai, and Enzhong Yang

***Abstract*—This paper considers the optimization problem of joint admission control and routing for the video streaming service in wired software-defined networking (SDN). With the aid of the network operating system, SDN is able to support the dynamic nature of future network functions and intelligent applications. Against this changing network landscape, we rely on FlowVisor-based virtualization in the context of OpenFlow-based wired SDN to design an open optimization architecture for the joint admission control and routing, which supports flexible and agile deployment of advanced joint admission control and routing strategies. Following this architecture, we interpret the joint admission control and routing problem into the Markov decision process for maximizing the overall "revenue." In order to solve the issue of the curses of dimensionality, we invoke the function approximation technique in the context of approximate dynamic programming to conceive an online learning framework. By applying kernel-based autonomous feature extraction into the function approximation, we develop an approximate dynamic programming-based joint admission control and routing for video streaming service, which is apt to be implemented in the proposed open architecture. An emulation platform based on FlowVisor, POX, and Mininet is constructed for demonstrating the success of the proposed solution. The experimental results are presented to show the performance improvement of the proposed scheme by comparing it with the Q-learning algorithm and open shortest path first-based benchmark scheme.**

***Index Terms*—Admission control, approximate dynamic programming, routing algorithm, software-defined networking (SDN), video streaming service.**

I. Introduction

WITH the significant advances of the communication technology and the video compression technology, video services are widely provided via Internet, and the video traffic is undergoing a rapid growth. As reported in [1], consumer internet video traffic will be 80 percent of all consumer internet traffic in 2019. The video services impose tremendous challenges on the networks for the sake of satisfying stringent Quality-of-Service (QoS) requirements. In order to support both the existing and the ever-increasing new video services, the networks are expected to have the capability of customizing the advanced traffic control technologies for satisfying QoS requirements. Unfortunately, it is challenging to incorporate them in the conventional networks because the network nodes are uncontrollable and transparent for media streaming applications. Software-Defined Networking (SDN) [2] has been fast emerging as a promising network technology for building next-generation services and networks. The architecture of SDN provides the programmability of multiple network layers, e.g., management, network service, control, forwarding and transport planes, to improve the utility of networks resources, to increase network agility, as well as to encourage service innovation. Against this changing network landscape, we conceive a joint admission control and routing strategy for optimizing the video streaming services over SDN.

Generally, it is difficult to resolve the system congestion when real-time traffic is present without reducing the QoS expected by the users. Admission control is a technique used to limit the number of connections into the system in order to prevent system congestion, thus enabling the system to provide the desired QoS to newly incoming and existing streams. Application-level admission control schemes for video streaming severs have been intensively discussed for keeping traffic load at an acceptable level and guaranteeing quality for admitted sessions via resource reservation [3]–[5]. In [6], the higher priority class of requests in Video-on-Demand (VOD) system has a higher probability to get the access to the system. This method works well on popular class for decreasing the blockings when the system is nearly or fully-utilized. A threshold-based admission control for distributed VOD systems is presented in [5]. It shows that enforcing threshold restriction on downgrading blocked requests in a multirate service environment leads to improved performance and provides different QoS levels. Recently, Qadir *et al.* propose a traffic measurement based algorithm for video admission control [7], and demonstrate the proposed method having an implementation in a Quality-of-Experience (QoE) aware admission control procedure.

Due to the rapid advances in the technology of server cluster, especially the cloud computing, the performance of the server

Manuscript received March 14, 2016; revised June 25, 2016 and September 13, 2016; accepted October 30, 2016. Date of publication November 15, 2016; date of current version February 14, 2017. This work was supported in part by the State Key Program of National NSF of China under Grant 61233003, in part by the National NSF of China under Grant 61573329, in part by the Fundamental Research Funds for the Central Universities, and in part by the Youth Innovation Promotion Association CAS. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. Shiwen Mao.

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: jianyang@ustc.edu.cn; zkj4873@mail.ustc.edu.cn; yyran@ustc.edu.cn; cwz@mail.ustc.edu.cn; ezhyang@mail.ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2016.2629280

and the storage is not the bottleneck any more when deploying the video streaming services. In contrast, the network becomes the key factor in providing services with high quality.In [8], an online measurement-based admission control scheme is proposed to guarantee QoS requirements for real-time Variable Bit Rate (VBR) video traffic in wireless home networks. An approach termed *Egress Admission Control* is presented in [9], where all admission control decisions are made at egress routers alone to support the demanding QoS requirements of real-time multimedia flows in a scalable way, without sacrificing utilization or weakening the service model. In [10], by combining Call Admission Control (CAC) algorithm with Adaptive Bandwidth Allocation (ABA) algorithm, a new framework is designed for wireless cellular networks that supports real-time adaptive multimedia services. The objective of this framework is to reduce the New Call Blocking Probability (NCBP) and the Handoff Call Dropping Probability (HCDP). In [11], a brokerage-based decentralized admission control mechanism is proposed for providing scalable and flexible resource allocation. All the works mentioned above focus on the edge routers or egress nodes. For the video streaming services over networks, cooperation of different layers is necessary in order to provide the required QoS guarantees. Dynamic rate adaptation strategy combined with admission control for multimedia services is proposed to maximize the customer average QoS in [12]. In [13], the Pre-Congestion Notification (PCN) mechanism is induced to protect the video services in multimedia access network. In [14], video streaming proxy is deployed at the base station, which jointly implements the admission control strategy on the new arrivals of video users and the rate adaptation algorithm for satisfying QoE constraints of all admitted video users.

Since the service provision over network involves routing, the design of optimal joint admission control and routing is discussed in several works for the sake of maximizing the overall "revenue" while guaranteeing the QoS. In the context of an integrated services network, a method of Neuro-Dynamic Programming (NDP) is applied to construct dynamic call admission control and routing for maximizing the average value of admitted calls per unit time [15]. Considering the end-to-end delay constraint in Multiprotocol Label Switching (MPLS) networks, a routing-based admission control mechanism is proposed in [16] based on solving exactly a mathematical programming model. In [17], admission control strategy for flow-aware multi-topology adaptive routing is presented for limiting the number of flows being processed when all paths to the destination node are congested. In [18], Zhang *et al.* propose a joint admission control and routing scheme in IEEE 802.16-based mesh networks for multiple service classes in order to maximize the overall revenue from all carried connections. In [19], a rate adaptive routing on clique admission control is proposed for multimedia wireless mesh networks, which is able to provide feedbacks to the application layer whenever congestion occurs in the network. In [20], a framework for resource reservation, route selection and connection admission is discussed for provisioning end-to-end deterministic-delay service in multiservice packet networks.

As shown above, there are plenty of advanced admission control schemes for wired/wireless systems. However, applying these admission control schemes requires appropriate modification and implementation of the cross layer protocols, which means that upgrading the communication devices is necessary. Furthermore, the architecture of the conventional network is ossified, and the network devices are closed for service provider, which makes the service provider unable to deploy the customized admission control over the traditional network. This dilemma is solved in the scenario of software-defined networking where control plane is separated from the individual network nodes and into a centralized controller. Relying on a Network Operating System (NOS) in the context of SDN, the controller is able to control the SDN switches and manipulate their forwarding plane. This architecture facilitates the service provider to implement a complex service provisioning strategy, while keeping the data plane as simple as possible.

Inspired by the emerging landscape of networks, we revisit the problem of joint admission control and routing for video streaming services in the context of SDN. There are very few literatures related to admission control in SDN. In [21], flow admission control based SDN is leveraged to implement inter-technology load balancing in the 5G network architecture. In [22], trunk reservation control based call admission control with cooperative behavior of users is discussed in SDN. However, these works do not consider the combination of admission control and the routing optimization. Furthermore, they are not conceived for video streaming services. OpenFlow [23] is the first platform implementing the SDN principles. In this paper, we conceive a comprehensive joint admission control and routing for video streaming in OpenFlow-based SDN. The main contributions of this paper are enumerated as follows:

1) An open architecture for joint admission control and routing optimization is constructed based on FlowVisor [24] and OpenFlow. The FlowVisor is applied to slice an isolated logical network dedicated for video streaming service from the physical network. Upon the guest OpenFlow controller dedicated to the network slice for video streaming service, a *Policy* module is abstracted for optimizing the provisioning of the video service. This open architecture enables the service provider agile to deploy the customized admission control and routing strategy.
2) Based on the observable network state from the network view provided by OpenFlow controller, we formulate the joint admission control and routing into Markov decision problem of maximizing the overall "revenue". In order to solve the curses of dimensionality of the problem, we invoke the function approximation technique in the context of Approximate Dynamic Programming (ADP) to conceive an online learning framework. By applying kernel based autonomous feature extraction into the function approximation, we develop an approximate dynamic programming based joint admission control and routing for video streaming service, which is apt to be implemented in the proposed open architecture.
3) In order to verify the feasibility of the proposed open architecture and to evaluate the achievable performance of the proposed joint admission control and routing strategy, we develop an emulation platform by integrating

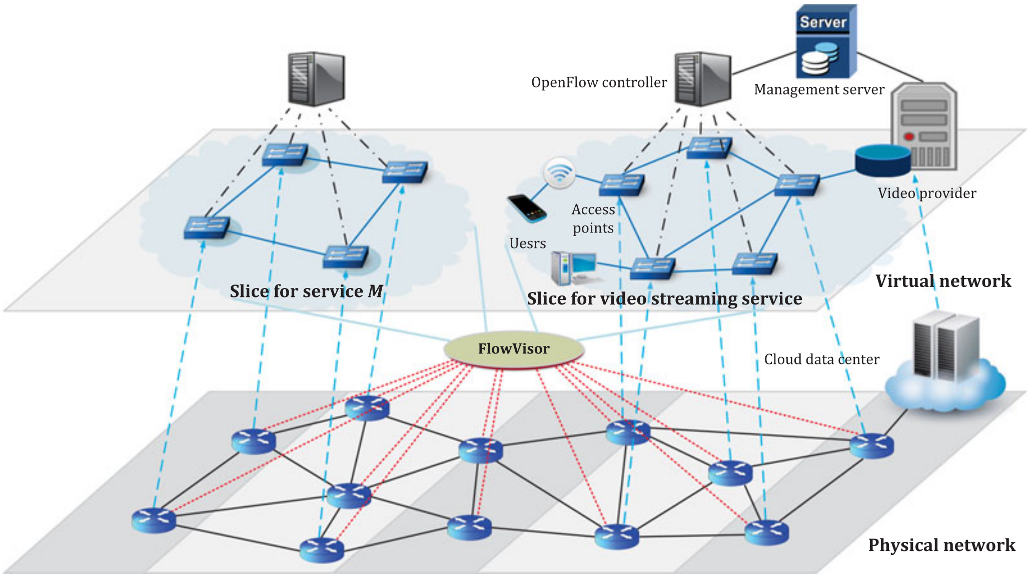


Fig. 1. Virtualized network for video streaming service.

FlowVisor, POX (OpenFlow controller), Mininet (Simulator for data plane) and a video streaming server. The experimental results show that the proposed strategy has the potential to significantly improve performance over a wide range of network loads compared with Q-learning algorithm and a commonly used heuristic Open Shortest Path First (OSPF) based method.

The rest of the paper is organized as follows. We describe the open system architecture based on FlowVisor and OpenFlow for joint admission control and routing optimization in Section II. Then the problem of joint admission control and routing in SDN is formulated as Markov decision problem of maximizing the overall “revenue” in Section III. In Section IV, we present an online learning framework based on approximate dynamic programming for the joint admission control and routing. Section V is devoted to describe the implementation of the procedures for the joint admission control and routing scheme. In Section VI, we present the experimental results from the emulation platform integrating FlowVisor, POX, Mininet and video streaming server. Finally, the conclusions are summarized in Section VII.

II. System Architecture

Network virtualization [25] enables high flexibility for promptly deploying network service, driving the innovation of networked applications. By network virtualization, multiple isolated logical networks where each runs potentially different addressing and forwarding mechanism, are able to share the same physical infrastructure. In [26], [27], a fine-grained network virtualization is proposed to slice the physical network infrastructure into several isolated subnets by designing network and switch hypervisors. The isolated subnets are referred to as network slices which are independent to each other. Hence, the network traffic from the services deployed in a network slice will not generate interference to any traffic in the other network slices. FlowVisor facilitates us to achieve network virtualization in the context of software-defined networking. In this section, we aim for utilizing FlowVisor combined with OpenFlow to construct an open and controllable architecture for flexibly deploying advanced admission control and routing strategy for video streaming services.

Fig. 1 shows the virtualized network for the video streaming service. The physical network consists of OpenFlow switches as a software-defined substrate layer. In order to achieve network virtualization and provide the network resource in the form of network slice, FlowVisor is used as an OpenFlow hypervisor that enables multiple OpenFlow controllers to share the same physical resources and to control their own slice. Multiple network slices created with the aid of FlowVisor can be used for deploying different network services as shown in Fig. 1. The isolation features of the network slices include bandwidth, topology, traffic and forwarding tables. One network slice is dedicated to the video streaming service so that applying any advanced optimization strategy for improving the video service will not interfere any other services in the other network slice. The cloud data center in Fig. 1 can also be sliced to provide the virtualized service platform for the service provider to deploy the video streaming server. A management server is designed in the network slice, and utilizes the network view provided by OpenFlow controller and the application-level information from the video streaming server to implement the admission control and routing.

Within the slice for video streaming service, we conceive an open architecture for joint admission control and routing optimization as depicted in Fig. 2. It is composed of Video Streaming Server, Management Server, OpenFlow Controller and OpenFlow Switches. The Video Streaming Server has the functions of video content storage and video streaming. Management Server acts as the heart of the proposed architecture, and consists of User Manager, Policy module for joint admission control and routing optimization as well as Video

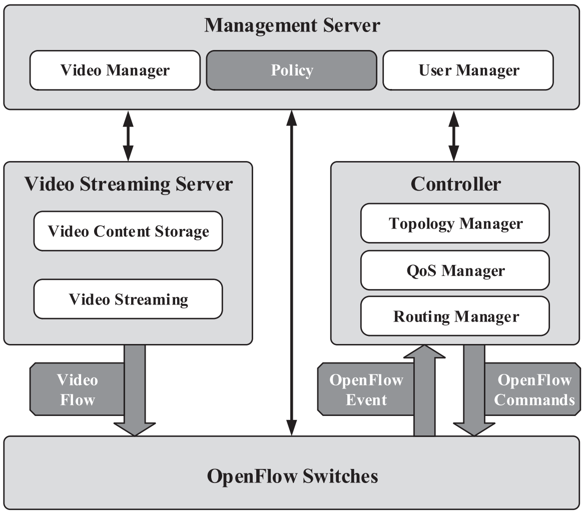


Fig. 2. Open architecture for joint admission control and routing optimization.

Manager for managing video content. The User Manager provides the function of Authentication, Authorization and Accounting (AAA). In the OpenFlow Controller, we customize QoS Manager, Topology Manager, and Routing Manager to assist us to deploy the admission control and routing policy. It should be noted that the Policy module relies on the network link status provided by QoS Manager and network view provided by Topology Manager in the OpenFlow Controller. The Policy module also maintains a data structure for the mapping between the video stream and network flow identified by $(SrcIP, SrcPort, DestIP, DestPort)$, where $SrcIP/SrcPort$ is IP/port of the video server for streaming the video, while $DestIP/Destport$ is IP/port of the terminal to receive the video stream. Via the interfaces provided by OpenFlow Controller, the management server can insert a new entry in the OpenFlow Switches' flow tables for filtering the flow specified by $(SrcIP, SrcPort, DestIP, DestPort)$, which assists the management server to control the routing of a specific video stream.

The basic serving process of the system is described as below. The terminals trigger video requests that traverse the OpenFlow Switches to the Manager Server. The User Manager in the Management Server authenticates and authorizes the access to the video service. It may be rejected if the user identity is illegal. Otherwise, the request is further forwarded to the Policy module in the Management Server to decide whether to reject it or to admit it. If accepting it, a route for the requested video flow is chosen out of a predefined list of possible paths from the access point to the video streaming server. The Policy module in the Management Server will notify the Routing Manager in the OpenFlow Controller to map the routing decision into forwarding rules and flush these rules into the flow tables in the OpenFlow switches. When the network configuration is completed, the Policy module triggers the Video Streaming Sever to stream the video content along the chosen path.

Compared to admission control and routing framework in conventional network, the proposed architecture abstracts the admission control and routing into a policy module, which provides an agile way to implement any advanced joint admission control and routing algorithm. Furthermore, by employing FlowVisor [24] to establish switch virtualization where the same hardware forwarding plane can be shared among multiple logical networks. Since the network slices are independent to each other, the implementation and operation of the joint admission control and routing algorithm in a dedicated network slice will never interfere the service traffic of the other network slices. This is the reason that the proposed architecture is open. In the next section, we will model the network according to the link status provided by OpenFlow controller and formulate the joint admission control and routing optimization as Markov decision problem.

III. Problem Formulation of Joint Admission Control and Routing

In this section, we formulate the problem of joint admission control and routing for video transmission as Markov decision process, which could be a continuous time and discrete finite-state dynamic programming problem [28]. We consider an OpenFlow network consisting of a set of nodes $\mathcal{N} = \{1, \ldots, N\}$ and a set of unidirectional wired links $\mathcal{L} = \{1, \ldots, L\}$, where each link l has a total capacity of $B(l)$ units of bandwidth which is constant over time. Let us define $b_t(l)$ to denote the traffic of the link l at the time instant t, which satisfies $b_t(l) \leq B(l)$. Then, the state $\mathbf{s}_t$ of the network at time t consists of a list of the link traffics $b_t(l)$, *i.e.*, $\mathbf{s}_t = [b_t(1), b_t(2), \ldots, b_t(L)]^T$. The set of all possible states is referred to as the state space S. Let $\mathcal{M} = \{1, \ldots, M\}$ be the set of the video identifiers, where each video m has its bandwidth requirement $r(m)$, and immediate reward $c(m)$ obtained whenever such a video session request is accepted. The bandwidth requirement $r(m)$ may characterize the average transmission rate of the video m, or its peak transmission rate, or its "effective bandwidth". Furthermore, the reward $c(m)$ is not necessary a monetary one, but may reflect the importance of different video session requests, their desired QoS as well as cost due to the video length. In the experiments of Section VI, without loss of generality, we use the monetary reward as a demonstration.

Here, the event-driven optimization framework is considered, which is to say that a decision is made only when a certain event e occurs, even though the network state evolves in continuous time. The events of interests are the arrivals of new video session requests and the departures of the existing video sessions. Note that these events naturally have the identifier of the video name, origin-destination pair (i, j), and if it corresponds to a video session termination, the bandwidth of the route ϕ occupied by the session will be released. Let $\mathcal{E}$ denotes the set of all possible arrival and departure events of the video sessions.

When an event $e \in \mathcal{E}$ take places, an action a has to be taken according to the current system state $\mathbf{s}$. If an arrival event e of the video m occurs, the action space $A(\mathbf{s}, e)$ is defined as the set of the rejection action and the possible routes which are subject to the capacity constraints and the current network state. The rejection action takes place when the bandwidth requirement does not meet the capacity constraints or we reject the transmission initiatively. When e is a departure event, re-routing all the video streams may be beneficial for achieving higher system performance. However, it will induce a considerable action space and very high computation complexity. Furthermore, reconfiguring the routes of all the video streams in a burst may cause the network instability. Hence, in this paper, we take no action when a video transmission is completed, and let $A(\mathbf{s}, e)$ be a singleton. Let $\mathbf{s}'$ denote the new network state after taking an action $a \in A(\mathbf{s}, e)$ given by the network state $\mathbf{s}$ and an event e. The instantaneous reward after taking the action a is defined as $f(\mathbf{s}, e, a)$. If e represents the arrival event of the video m and a is an action to admit along some route, we have $f(\mathbf{s}, e, a) = c(m)$, otherwise, $f(\mathbf{s}, e, a) = 0$. Naturally, admitting different videos having different length may lead to a different long-term reward.

A policy specifies the decision rule to be used at all decision epoches, and here the time of decision-making is referred to as decision epoches. The policy provides the decision maker with a prescription for action selection under any possible future system state or history. In this paper, we define a policy π to map the state space and the event space to the action space, i.e.,

$$\pi(\mathbf{s}, e) \in A(\mathbf{s}, e), \forall \mathbf{s} \in S, e \in \mathcal{E}.$$

It should be noted that given any policy π, the state $\mathbf{s}_t$ evolves as a continuous time finite state Markov process. Let t_k be the time of the kth event, and let $\mathbf{s}_{t_k}$ be the state of the system just prior to that event.

Below, we present the total long-term reward for joint admission control and routing optimization. Following a policy π, a discount total reward is defined as

$$J_\pi = \lim_{N\to\infty} E^\pi \left[\sum_{t=0}^{N} \gamma^t f_t \right] \tag{1}$$

where $E^\pi[\cdot]$ denotes the expectation operator with respect to the state transition probability controlled by the policy π, f_t is the reward at time t, and $0 < \gamma < 1$ is a discount factor that represents value reduction over time. The reason for considering discounted reward is to emphasize near-future decisions and rewards. Note that the total discounted reward is finite, since each immediate reward is bounded. The problem of maximizing the long-term total reward can be formulated as

$$\max_\pi \lim_{N\to\infty} E^\pi \left[\sum_{t=0}^{N} \gamma^t f_t \right]. \tag{2}$$

The optimal policy π^* can be estimated by

$$\pi^* = \arg\max_\pi \lim_{N\to\infty} E^\pi \left[\sum_{t=0}^{N} \gamma^t f_t \right]. \tag{3}$$

In order to solve (2), we introduce the definition of the *state-value function* and *action-value function*.

For the sake of finding an optimal policy, the decision maker needs a facility to differentiate the desirability of possible successor states, in order to decide on the best action. A common way to rank states is by computing and using a so-called state-value function which estimates the expected discounted total reward when starting in a specific state $\mathbf{s}$ and taking actions by following the policy π. Accordingly, with an initial state $\mathbf{s}$, the state-value function of the policy π is defined as

$$V^\pi(\mathbf{s}) = \lim_{N\to\infty} E^\pi \left[\sum_{t=0}^{N} \gamma^t f_t | s_0 = \mathbf{s} \right]. \tag{4}$$

where $f_t = f(s_t, e_t, \pi(s_t, e_t))$ is the immediate reward at time t. Equation (4) can be further written as

$$V^\pi(\mathbf{s}) = f(\mathbf{s}, e, \pi(\mathbf{s}, e)) + \gamma E^\pi [V^\pi(\mathbf{s}')] \tag{5}$$

where $\mathbf{s}'$ is the next state right after the event e.

Similarly, define the value of taking action a_t in the state s_t under the policy π, denoted by $Q^\pi(s_t, a_t)$, as the expected discounted reward starting from s_t, taking the action $a_t \in A(s_t, e)$, and thereafter following policy π. Accordingly, the state-action value function for the policy π is expressed as

$$Q^\pi(\mathbf{s}, a) = \lim_{N\to\infty} E^\pi \left[\sum_{t=0}^{N} \gamma^t f_t | s_0 = \mathbf{s}, a_0 = a \right]. \tag{6}$$

Equation (6) can be further written as

$$Q^\pi(\mathbf{s}, a) = f(\mathbf{s}, e, a) + \gamma E^\pi [V^\pi(\mathbf{s}')]. \tag{7}$$

A policy π is defined to be better than or equal to a policy π' if its expected discount total reward of π is higher than or equal to that of π' for all $\mathbf{s} \in \mathcal{S}$. Let π^* be the optimal policy which is better than or equal to all the other polices. Accordingly, the state-value function of the optimal policy π^* is

$$V^{\pi^*}(\mathbf{s}) = \max_{a \in A(\mathbf{s}, e)} Q^{\pi^*}(\mathbf{s}, a). \tag{8}$$

From (8), take the action with the highest expected discount reward according to $Q^{\pi^*}(\mathbf{s}, a)$. The optimal state-action value function $Q^{\pi *}(s, a)$ can be written as

$$\begin{aligned} Q^{\pi^*}(\mathbf{s}, a) &= f(\mathbf{s}, e, a) + \gamma V^{\pi^*}(\mathbf{s}') \\ &= f(\mathbf{s}, e, a) + \gamma \max_{a' \in A(\mathbf{s}', e)} Q^{\pi^*}(\mathbf{s}', a'). \end{aligned} \tag{9}$$

Correspondingly, the optimal policy π^* could be presented as

$$\pi^*(\mathbf{s}, e) = \arg\max_{a \in A(\mathbf{s}, e)} Q^{\pi^*}(\mathbf{s}, a). \tag{10}$$

This equation implies that once the state-action value function $Q^\pi(\cdot, \cdot)$ is available, the proposed dynamic programming problem of the joint admission control and routing can be solved by enumerating all the possible action due to the finite actions at each state. Hence, the estimation of $Q^\pi(\cdot, \cdot)$ is the key step in deriving the optimal admission control and routing.

IV. Online Optimization Framework for Joint Admission Control and Routing

We employ the methodology of *Policy iteration* (PI) [29], a dynamic programming algorithm, to find the optimal policy in (2). Under the conditions of finite action and state spaces as well as bounded and stationary immediate reward function, *Policy iteration* has been proven to converge to the optimal policy when $0 < \gamma < 1$ [29]. Its key idea is to use the structure of (5), (7) and (10). *Policy iteration* involves two steps, i.e., policy evaluation and policy improvement.

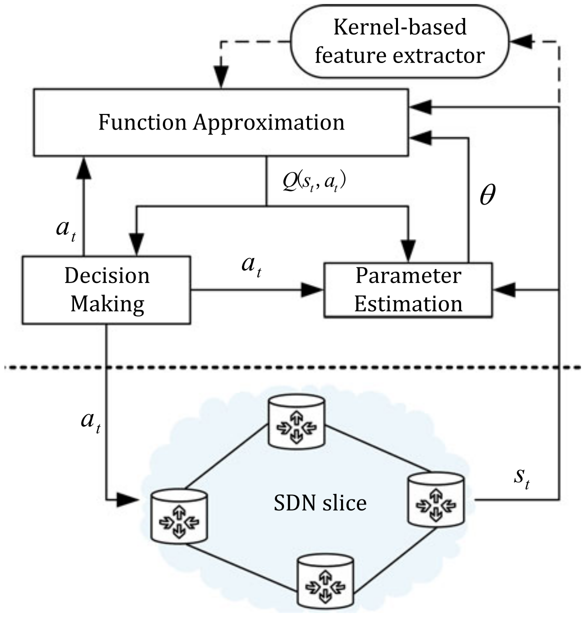


Fig. 3. Online optimization framework.

In the step of the policy evaluation, the value of a policy π is evaluated by computing the value function $V^{\pi}(\mathbf{s})$ according to (5). This step is applied to indicate how good a policy π is. In the step of the policy improvement, the policy iteration algorithm looks for a policy π that is better than the previously evaluated policy π. According to the Policy Improvement Theorem [30], $\pi' \geq \pi$ if $Q^{\pi}(\mathbf{s}, \pi'(\mathbf{s})) \geq V^{\pi}(\mathbf{s})$ for all $\mathbf{s} \in \mathcal{S}$. By applying action search to $Q^{\pi}(\mathbf{s}, a)$ at each state, the new policy π' can be obtained in the Policy improvement step. Specifically, the new policy π' is selected as follows:

$$\pi'(\mathbf{s}) = \arg \max_{a \in A(\mathbf{s}, e)} Q^{\pi}(\mathbf{s}, a). \tag{11}$$

Policy iteration operates iteratively by executing the above two steps. When the same policy is found in two consecutive iterations, we conclude that the algorithm has converged.

Following the basic idea of the policy iteration method, we conceive an online optimization framework as shown in Fig. 3. Specifically, the *Decision Making* relies on a decision function that chooses a decision a_t given the state, while the *Function Approximation* determines the contribution from a decision and computes the action-state value function $Q(s_t, a_t)$. The interaction of making decision and updating the value function is the kernel that drives the optimization of jointly admission control and routing. The *Decision Making* receives the measurement data about the network's current state s_t via the OpenFlow controller, and searches the best action a_t by the output of the *Function Approximation*. Multilayer perceptron neural networks (MLPNNs) can be applied for value function approximation since it has the capability of approximating arbitrary vector functions. However, it has the nonlinear dependence on the parameter, which induces additional challenge in parameter estimation. In this paper, we consider the linear mapper to estimate the value function by defining a parametric model as

$$Q(\mathbf{s}, a|\theta) = \Sigma_{f \in \mathcal{F}} \theta_f \phi_f(\mathbf{s}, a) \tag{12}$$

where $\phi_f(\mathbf{s}, a)$ are basis functions that reduce potentially large state variables into a small number of features, and $f \in \mathcal{F}$ is a feature. Although the features can be manually and heuristically selected based on experience, we induce a graceful approach to choose the features by utilizing structure learning and nonlinear approximation ability of sparse kernel machines. Accordingly, we conceive a *Kernel-based Feature Extractor* in the optimization architecture. The module *Parameter Estimation* is used to online estimate the parameter θ for the sake of updating the value function approximation model of (12).

V. Optimization Algorithm for Joint Admission Control and Routing

This section embodies each module of the proposed online optimization architecture in order to build an executable system for jointly optimizing admission control and routing.

A. Kernel-Based Feature Extraction

Motivated by the fact that sparse kernel machines in the context of statistical learning have better generalization capability than conventional MLPNNs with manually designed structures, we apply it to develop a kernel-based feature extraction having automatic feature representation and selection. Let us denote the state-action pair by $\sigma_t = (s_t, a_t)$. Suppose $\sigma_1, \sigma_2, \ldots, \sigma_n$ denote a sample sequence of the network states and the corresponding actions. Here, we consider a Mercer kernel where for any finite set of points $\sigma_1, \ldots, \sigma_n$, the matrix $[K]_{ij} = k(\sigma_i, \sigma_j)$ is positive definite. The kernel $k(\cdot, \cdot)$ is induced by feature mapping function $\psi : \mathcal{R}^L \rightarrow \mathcal{R}^{\mathcal{H}}$ where $L << \mathcal{H}$ and $k(\sigma_i, \sigma_j) = \psi(\sigma_i)\dot{\psi}(\sigma_j)$ represents a nonlinear similarity between two state-action pairs σ_i and σ_j. There are some well-known kernel functions such as Gaussian kernel, polynomial kernel, etc. In this paper, Gaussian kernel $\exp\{-\|\sigma - \sigma'\|^2 / 2\delta^2\}$ is chosen as the kernel function, which is applied in Section VI. The reason for choosing Gaussian kernel lies in that it has good performance and computational effectiveness to characterize nonlinear problems in higher dimensional feature spaces. We define $[\Delta b(1), \Delta b(2, ..., \Delta b(L)]^T$ as the vector of link traffic increment after taking the action a. Then, the metric [31] of state-action pairs in the Gaussian kernel is defined as

$$\|\sigma - \sigma'\|^2 = \sum_{l=1}^{L} ((b(l) - b'(l))^2 + (\Delta b(l) - \Delta b'(l))^2).$$

Suppose that at time t, after having observed $t-1$ samples $\{\sigma_i\}_{i=1}^{t-1}$, we have collected a dictionary consisting of a subset of the samples $\mathcal{D}_{t-1} = \{\tilde{\sigma}_j\}_{j=1}^{d_{t-1}}$, where, by construction, $\{\psi(\tilde{\sigma}_j)\}_{j=1}^{d_{t-1}}$ are linearly independent feature vectors. Here, d_t is the size of the dictionary at the time t. For a new sample σ_t, we test whether $\psi(\sigma_t)$ is approximately linearly dependent on the dictionary vectors. If not, we add it to the dictionary. The specific procedure is described as follows. Let $\mathbf{w} = (w_1, w_2, \ldots, w_{d_{t-1}})$

denote the linearly combining coefficients of the feature vectors in the dictionary for approximating $\psi(\sigma_t)$. The approximation error is characterized by

$$\begin{aligned}\Delta_t &= \min_{\mathbf{w}} \|\psi(\sigma_t) - \Sigma_{j=1}^{d_{t-1}} w_j \psi(\tilde{\sigma}_j)\|^2 \\ &= \min_{\mathbf{w}}\{\mathbf{w}^T \tilde{\mathbf{K}}_{t-1}\mathbf{w} - 2\mathbf{w}^T \tilde{\mathbf{k}}_{t-1}(\sigma_t) + k_{tt}\} \qquad (13)\end{aligned}$$

where $[\tilde{\mathbf{K}}_{t-1}]_{i,j} = k(\tilde{\sigma}_i, \tilde{\sigma}_j)$, $(\tilde{\mathbf{k}}_{t-1}(\sigma_t))_i = k(\tilde{\sigma}_i, \sigma_t)$, and $k_{tt} = k(\sigma_t, \sigma_t)$, with $i, j = 1, \ldots, d_{t-1}$.

Note that the problem (13) is a quadric form of $\mathbf{w}$, and has the optimal point of $\mathbf{w} = \tilde{\mathbf{K}}_{t-1}^{-1}\tilde{\mathbf{k}}_{t-1}(\sigma_t)$. Hence, we have $\Delta_t = k_{tt} - \tilde{\mathbf{k}}_{t-1}^T(\sigma_t)\mathbf{w}$. Let ε be a predefined threshold which characterizes the level of sparsity. Fewer samples will be added to the dictionary with a large ε, and there will be the deficiency of some features. In contrast, a small ε may cause the size of dictionary very large, and some of the features in the dictionary may be redundant. In the experiments of Section VI, we take $\varepsilon = 0.01$. $\Delta_t \leq \varepsilon$ implies that the feature vector $\psi(\sigma_t)$ can be approximated within a squared error ε by a linear combination of current dictionary members. In this case, we keep the dictionary unchanged, i.e., $\mathcal{D}_t = \mathcal{D}_{t-1}$. For $\Delta_t > \varepsilon$, we augment the current dictionary by $\mathcal{D}_t = \mathcal{D}_{t-1} \bigcup \sigma_t$. The spread parameter δ of the Gaussian kernel has a significant effect on the performance of kernel method. If δ is too large, nearly all the samples are regarded as one sample point, so the dictionary may have only one feature, while if δ is too small, almost all the samples are features. In [32], a traditional method, i.e., Fisher discrimination is used for determining the parameter δ. In the experiments of Section VI, we take $\delta^2 = 2$.

It has been shown in [33] that if k is a continuous Mercer kernel and the state-action space is a compact subset of a Banach space, the number of dictionary vectors is finite for any sequence $\{\sigma_i\}_{i=1}^{\infty}$ and for any $\varepsilon > 0$. In our problem, the state-action space is a finite set of a Banach space, so it is a compact subset of a Banach space [31]. This implies that after an initial period, the computational cost per step of the algorithm becomes independent of time and depends only on the dictionary size. This property enables our proposed framework feasible for online real-time learning.

B. Function Approximation and Its Parameter Estimation

By setting the basis functions to the kernel functions, the model (12) for approximating the value function is rewritten in the context of kernel approximation as

$$\tilde{Q}(\sigma|\theta) = \sum_{j=1}^{d_t} \theta_j k(\sigma, \sigma_j) \qquad (14)$$

where d_t is the size of the dictionary, and σ is the state-action pair.

Below, we derives an online estimation method for updating (14). According to (7), we have

$$E[\tilde{Q}(\sigma_t|\theta) - \gamma\tilde{Q}(\sigma_{t+1}|\theta)] = f_t \qquad (15)$$

where $E[\cdot]$ is the expectation with respect to the state transition probability following a stationary policy. Substituting (14) into (15) yields

$$E[\Sigma_{i=1}^{d_n}(k(\sigma_t, \tilde{\sigma}_i) - \gamma k(\sigma_{t+1}, \tilde{\sigma}_i))\theta_i] = f_t. \qquad (16)$$

Equation (16) can be further written as

$$E[(\tilde{\mathbf{k}}^T(\sigma_t) - \gamma\tilde{\mathbf{k}}^T(\sigma_{t+1}))\theta] = f_t \qquad (17)$$

where $[\tilde{\mathbf{k}}(\sigma)]_i = k(\sigma, \tilde{\sigma}_i)$. For a single-step observation, (17) can be expressed as

$$(\tilde{\mathbf{k}}^T(\sigma_t) - \gamma\tilde{\mathbf{k}}^T(\sigma_{t+1}))\theta = f_t + \nu_t \qquad (18)$$

where ν_t is the observation noise. At the instant $t+1$, the observations $\tilde{\mathbf{k}}^T(\sigma_i) - \gamma\tilde{\mathbf{k}}^T(\sigma_{i+1})$ and f_i are available for $1 \leq i \leq t+1$. Accordingly, we formulate an optimization problem as an exponentially weighted sum for updating θ as

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{t} \lambda^{t-i}\|f_i - (\tilde{\mathbf{k}}^T(\sigma_i) - \gamma\tilde{\mathbf{k}}^T(\sigma_{i+1}))\theta\|^2 \qquad (19)$$

where the forgetting factor λ is between 0 and 1, which could impose exponentially less weight on the older error samples. Smaller is λ, less is the contribution of the previous samples to the covariance matrix. If $\lambda = 1$, all the observations are given the same weight, and no forgetting of old data takes place. In practice, λ is usually chosen between 0.98 and 1 as recommended in [34]. It is shown in [35] that a type-II maximum likelihood estimation method can be applied to estimate λ from a set of data. In the experiments of Section VI, we take $\lambda = 0.99$.

Applying the Recursive Least Square (RLS) technique [36] to solve (19), we can derive a recursive algorithm for updating θ as

$$\begin{aligned}\mathbf{x}(t) &= \tilde{\mathbf{k}}^T(\sigma_t) - \gamma\tilde{\mathbf{k}}^T(\sigma_{t+1}) \qquad (20)\\ \mathbf{P}(t) &= \frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{P}_{t-1}\mathbf{x}(t)\mathbf{x}^T(t)}{\lambda + \mathbf{x}^T(t)\mathbf{P}_{t-1}\mathbf{x}(t)}\right)\mathbf{P}(t-1) \qquad (21)\\ \tilde{\theta}(t) &= \tilde{\theta}(t-1) + \mathbf{P}(t)\mathbf{x}(t)(f_t - \mathbf{x}^T(t)\tilde{\theta}(t-1)). \qquad (22)\end{aligned}$$

The initial values of the above algorithm can be set as $\mathbf{P}(0) = \alpha_1\mathbf{I}$, and $\tilde{\theta}(0) = \alpha_2[1, 0, \ldots, 0]^T$, where α_1 and α_2 are appropriate positive scalars, and $\mathbf{I}$ is the $d_n \times d_n$ identity matrix.

Since the feature extraction operates in the online manner, the size of the dictionary may increase due to one of the two consecutive executions of (20)–(22). Once a new feature is identified and added into the dictionary, we have to augment the dimensions of the vectors or matrix in (20)–(22). Specifically, we handle this issue as follows. Suppose Δ_{t+1} corresponding to a new sample σ_{t+1} is higher than ε. Then, $\psi(\sigma_{t+1})$ is not linearly dependent on $\mathcal{D}_t$, and σ_{t+1} is added to the dictionary by $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \sigma_{t+1}$. In order to conduct the function approximation over the augmented dictionary $\mathcal{D}_{t+1}$, we augment the parameters of (20)–(22) as

$$\begin{aligned}\tilde{\mathbf{k}}(\sigma) &= [\tilde{\mathbf{k}}^T(\sigma), k(\sigma, \tilde{\sigma}_{d_{t+1}})]^T \qquad (23)\\ \mathbf{P}(t) &= \begin{pmatrix}\mathbf{P}_t & \mathbf{0}\\ \mathbf{0}^T & 1\end{pmatrix} \qquad (24)\\ \theta(t) &= [\theta^T(t), 0]^T \qquad (25)\end{aligned}$$

where $\mathbf{0}$ is a vector of zeros having the size of d_n.

Algorithm 1: Optimization algorithm for joint admission control and routing.

```
Input:
    The Mercer kernel function: $k(\cdot,\cdot)$
    Sparsification parameter $\varepsilon$
    Initialize $\mathbf{P}(1) = [1]$, $d(1) = 1$, $\lambda$, $\theta(1)$
Output:
    Total reward: $R$
 1: Start from $t = 2$
 2: while A new event occurs do
 3:     Choose the optimal action $a_t$ according to (26);
 4:     Get reward $R = R + f_t$;
 5:     Feature extraction with sample $\sigma_t = (S_t, a_t)$
        $\mathbf{w} = \tilde{\mathbf{K}}_{t-1}^{-1}\tilde{\mathbf{k}}_{t-1}(\sigma_t)$
        $\Delta_t = k_{tt} - \tilde{\mathbf{k}}_{t-1}^{T}(\sigma_t)\mathbf{w}$
        if $\Delta_t > \varepsilon$ then
            $\mathcal{D}_t = \mathcal{D}_{t-1} \bigcup \sigma_t$;
            Augment $\tilde{\mathbf{k}}(\cdot), \mathbf{P}_{t-1}, \theta_{t-1}$ according to (23)–(25);
        end if
 6:     Update the parameter $\theta(t)$ according to (20)–(22);
 7:     Observe next state $s_{t+1}$;
 8:     $t = t + 1$;
 9: end while
10: return $R$.
```

C. Algorithm Description

Once the parameter θ_t for the function approximation (14) is estimated, we can apply the updated approximate function to develop the joint admission control and routing policy. Considering the finite discrete action set in our problem, we choose the optimal action via the action search as follows:

$$a_t = \arg \max_{a \in A(\mathbf{s},e)} [\gamma \tilde{Q}(s_t, u|\theta_t) + f_t]. \tag{26}$$

Combining the above key steps of sparsification, function approximation and decision making, we obtain the optimization algorithm for joint admission control and routing for streaming video over SDN networks, which is summarized in Algorithm 1. We choose the online feature learning process at the step 5 of Algorithm 1. At the time t, we use (13) to identify whether the state-action pair $\sigma_t = (s_t, a_t)$ is a new feature. It can be seen from (13) that the feature learning algorithm only relies on the sample σ_t and the dictionary $\mathcal{D}_{t-1}$, rather than on all the samples from time 0 to time $t-1$, thus avoiding the problem of memorizing all the samples.

The proposed algorithm is deployed in the Policy module of the open architecture as shown in Fig. 2, which acts as a decision maker. By connecting the OpenFlow Controller and switches, the Policy module is able to obtain the network topology and state information for making decision. Specifically, when a legal user requests a video transmission, a request with the video identity is sent to the Policy module of the Management Server. By obtaining the network information from the OpenFlow Controller, the Policy module executes the Algorithm 1 to make a decision of rejecting the request or admitting it. If accepting it, a route for the requested video flow is chosen, and the Policy module in the Management Server notifies the Routing Manager of the OpenFlow Controller to configure the flow tables in the OpenFlow switches.

Since we integrate FlowVisor to create a dedicated network slice for deploying video services and implementing Algorithm 1, all the video traffic is the major traffic that is controlled by our proposed algorithm. The bandwidth reservation for other minor traffic in this dedicated network slice can be executed to remove the impact of the proposed method on those minor traffic. Furthermore, due to the dedicated slice in SDN for the video transmission, the bandwidth for each video transmission is guaranteed. The rate adaptation of dynamic adaptive streaming such as Dynamic Adaptive Streaming over HTTP (DASH) is unnecessary to be triggered. Hence, the proposed algorithm is still applicable in the scenario of the co-existence of DASH.

VI. Experimental Results

This section intends to present the experimental results to show the performance of the proposed joint admission control and routing solution. Following the design in Figs. 1 and 2, we developed an emulation platform by integrating FlowVisor, POX and Mininet, where POX is a real Python-version OpenFlow controller, and Mininet is a network emulation orchestration system having a collection of end-hosts, routers, and links. The proposed ADP based joint admission control and routing strategy was implemented in the Policy module of the Management Server. For the performance comparison, we also implemented heuristic admission control with OSPF routing strategy and Q-learning algorithm [37]. OSPF is a widely applied routing algorithm which always selects the shortest path by using Dijkstra algorithm. In this benchmark algorithm, the admission control is heuristic, i.e., the video request is agreed if the available bandwidth could accommodate the video stream. Q-learning algorithm is a widely applied model-free reinforcement learning technique. It can be used to solve any given finite MDP problem by learning a state-action value function that ultimately gives the expected utility of taking a given action in a given state and following the optimal policy thereafter. When the state-action value function is learned, the optimal action with the highest value can be obtained. Below, we respectively present the results of our algorithm and OSPF algorithm in simple network topology and the results of all of three algorithms in complex network topology.

A. Simple Network Topology

We first consider a simple network with the topology depicted in Fig. 4. There are three switches and two access points (APs) in the topology. Actually, the access points are also switches in our experiment. Video server is connected to switch 1 and users' devices are attached to the APs when they request video transmissions. In the topology, solid arrow shows the unique path for the users attached to AP2, and the dotted arrows characterize the available paths for users from AP1, which are also listed in Table I. For our proposed solution, the users connected to AP1 can be served by one path selected from the two paths between

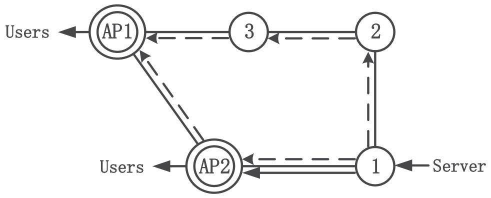


Fig. 4. Simple network topology.

TABLE I
PATHS IN SIMPLE NETWORK TOPOLOGY

Access Point	Path 1	Path 2
AP1	1 – AP2 – AP1	1 – 2 – 3 – AP1
AP2	1 – AP2	– –

TABLE II
THREE TYPES OF VIDEO

Resolution	SD	HD	Full HD
Bandwidth Demand (Mbps)	5.00	7.00	10.00
Video Duration (s)	100	100	100
Monetary Reward $c(m)$	10.00	18.00	28.00

TABLE III
PERFORMANCE OF ADP IN SIMPLE NETWORK TOPOLOGY

Video Type	Average Reward (Price Unit)	Loss Percentage
SD	1.05	3.94%
HD	1.94	1.52%
Full HD	2.92	7.35%
Total	5.91	4.90%
Access Point	Average Reward (Price Unit)	Loss Percentage
AP1	3.02	2.69%
AP2	2.89	7.12%
Total	5.91	4.90%

AP1 and the Video Streaming Server. In contrast, OSPF will preferentially choose the path 1 as long as it meets the throughput requirement for conveying the video stream. This is because each link in topology has an identical bandwidth limitation of 100 Mbps, and the cost of each port of each router is the same. Hence, path 1 which has the minimum hop is the shortest path, and OSPF will also choose path 2 when path 1 is fully loaded.

For the simplicity, three types of video clips are used in the experiment as shown in Table II. Each type has different resolution with different reward. Here, the monetary reward is used in terms of price unit. In the experiment, two hundred users randomly triggered video requests within ten minutes, where one half connected to AP1 and the other half connected to AP2. The video bitrate was also presented as shown in Table II. In order to reduce the run-time of the experiment, we used the video clips with short durations as given in Table II.

TABLE IV
PERFORMANCE OF OSPF IN SIMPLE NETWORK TOPOLOGY

Video Type	Average Reward (Price Unit)	Loss Percentage
SD	0.99	8.96%
HD	1.86	6.97%
Full HD	2.76	11.19%
Total	5.61	9.44%
Access Point	Average Reward (Price Unit)	Loss Percentage
AP1	3.09	0.00%
AP2	2.52	18.83%
Total	5.61	9.44%

Tables III and IV show the experimental results in terms of average reward and loss percentage. It can be observed that the proposed ADP based algorithm achieves higher average reward than that of OSPF based benchmark algorithm, while the ADP based algorithm reduces loss percentage by 48% compared to OSPF based benchmark algorithm. The basic reason for this is as follows. OSPF will preferentially chose the path 1 as long as it meets the throughput requirement for conveying the video stream. This may be not a wise choice because two paths in the network share the common link "1 – AP2". For our proposed solution, the users connected to AP1 can be served by one path selected from the two paths between AP1 and the Video Streaming Server, which may improve the total average reward and substantially reduce loss percentage. This fact is also verified from the second part of Tables III and IV. For the OSPF based benchmark algorithm, the results show that all the requests from AP1 were accepted, which implies that more resource of the common link "1 – AP2" was occupied by the shortest path from AP1 to the Video Streaming Server. As a result, the requests from AP2 suffer a high reject ratio. By contrast, our proposed method has a chance of choosing another path from AP1 to the video streaming server. Although AP1 incurs a slight loss percentage, AP2 substantially reduces loss percentage in comparison with the OSPF based benchmark algorithm.

In order to show more specific details, the path distributions were presented in Fig. 5. The portion marked by *No Path* in Fig. 5 denotes the percentage of request rejections caused by no available paths. For video requests from AP1, the proposed ADP based algorithm prefers path 2 because the link "1 – AP2" in the path 1 is also in the only path for users from AP2. Hence, 90% or even higher of video requests from AP1 were agreed to be served through path 2, and AP1 incurred lower loss percentage. But for OSPF algorithm, a shortest path (path 1) was preferentially chosen (more than 90% for SD and HD, nearly 80% for full HD). This means that the link "1 – AP2" is likely to be fully loaded, which results in that many requests from AP2 were rejected and rejection percentage of HD and full HD video requests from AP2 is much higher than that from AP1. The rejection percentage was verified by the percentage of *No Path* in Fig. 5. It should be noted from Fig. 5 that ADP based method has a higher loss percentage of SD video than OSPF. The reason behind this is that the proposed ADP based method prefers initiatively rejecting the SD video requests from AP2 as show in Fig. 5 in order to reserve more resource for the HD and full HD video requests, which improves the total average reward. In

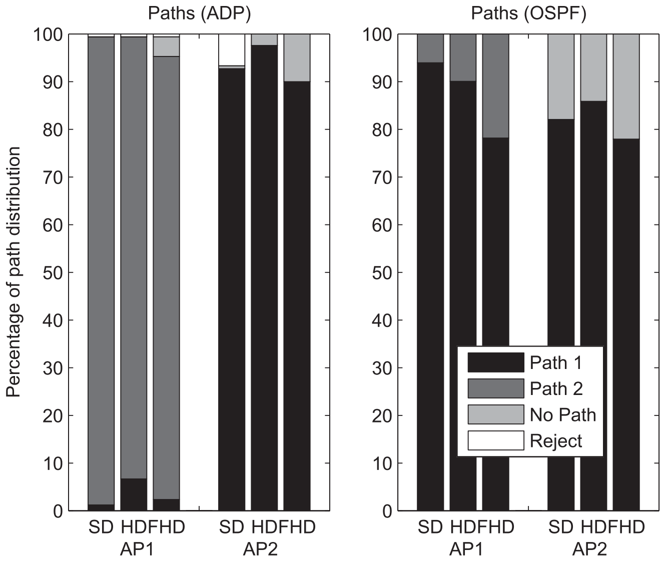


Fig. 5. Path distributions of ADP and OSPF algorithm in topology 1.

TABLE V
PATHS IN THE COMPLEX NETWORK TOPOLOGY

Paths of Access Point 1	
Path 1	1 – 2 – 7 – 12 – AP1
Path 2	1 – 2 – 8 – 12 – AP1
Path 3	1 – 4 – 8 – 12 – AP1
Path 4	1 – 2 – 3 – 6 – 11 – 18 – AP1
Paths of Access Point 2	
Path 1	1 – 9 – 13 – 16 – AP2
Path 2	1 – 2 – 7 – 12 – 16 –AP2
Path 3	1 – 4 – 8 – 12 – 16 –AP2
Path 4	1 – 4 – 8 – 13 – 16 –AP2
Path 5	1 – 5 – 9 – 13 – 16 –AP2
Paths of Access Point 3	
Path 1	1 – 9 – 14 – 17 – AP3
Path 2	1 – 5 – 9 – 14 – 17 – AP3
Path 3	1 – 4 – 8 – 13 – 17 – AP3
Paths of Access Point 4	
Path 1	1 – 9 – 15 – AP4
Path 2	1 – 5 – 10 – 15 – AP4

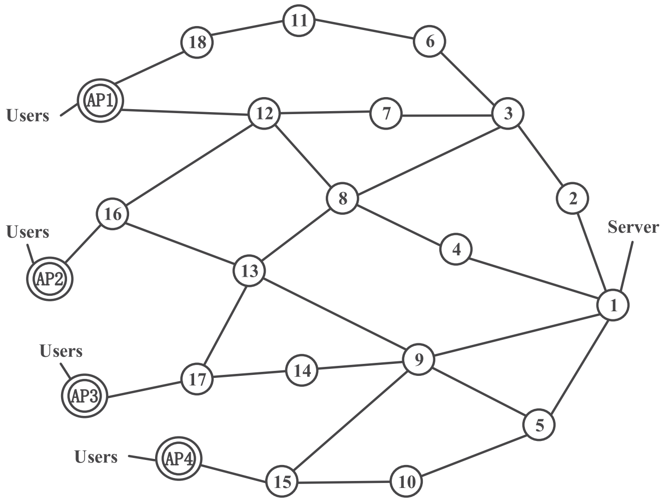


Fig. 6. Complex network topology.

TABLE VI
VIDEO INFORMATION

	Bandwidth Demand	Duration	Reward	Interval
range	5.00–10.00	100–120	10.00–30.00	6–13

TABLE VII
PERFORMANCE OF ADP, Q-LEARNING, AND OSPF ALGORITHM IN COMPLEX NETWORK TOPOLOGY

Algorithm	Average Reward (Price Unit)	Loss Percentage
ADP	8.40	9.47%
Q-Learning	7.94	14.43%
OSPF	7.73	16.69%

contrast, the OSPF based method accepted any video requests if the available bandwidth is enough for conveying the video stream.

B. Complex Network Topology

In this section, we conducted an experiment having a more complex network topology. Q-learning algorithm will also be implemented as a benchmark algorithm. The topology of the network used in the experiment is shown in Fig. 6, which consists of 22 switches including the access points and 29 links. The capacities of links are same as the previous experiment. Four access points were deployed for the users to access the network. The available paths for each access points were enumerated in Table V. In the experiment, we encoded 500 video clips whose bitrates were uniformly sampled within $[5, 10]$ Mbps as shown in Table VI. The rewards were randomly generated within $[10, 30]$ and assigned to each class of the videos. The request arrival interval was uniformly distributed within $[6, 13]$s.

Performance: Table VII characterizes the experimental results in the context of complex network topology. Each access point has 100 users to randomly request the video service. It can be seen from Table VII that the improvement of average reward by the proposed ADP based method is 0.67 in comparison with OSPF based method and 0.46 in comparison with Q-learning algorithm. In terms of loss percentage, the proposed ADP method achieves a reduction of 7.22% from 16.69% to 9.47% in contrast to OSPF as well as a reduction of 4.96% from 14.43% to 9.47% in comparison with Q-learning algorithm.

Routing: Fig. 7 characterizes the path distribution of the proposed ADP based algorithm, Q-learning algorithm and OSPF based algorithm. The difference between the proposed ADP algorithm and OSPF based algorithm is the path selection of all the APs. In the proposed ADP method, the four APs respectively prefer path 4, path 2, path 2 and path 2 for conveying the video streams, so that no competition occurs on four common links

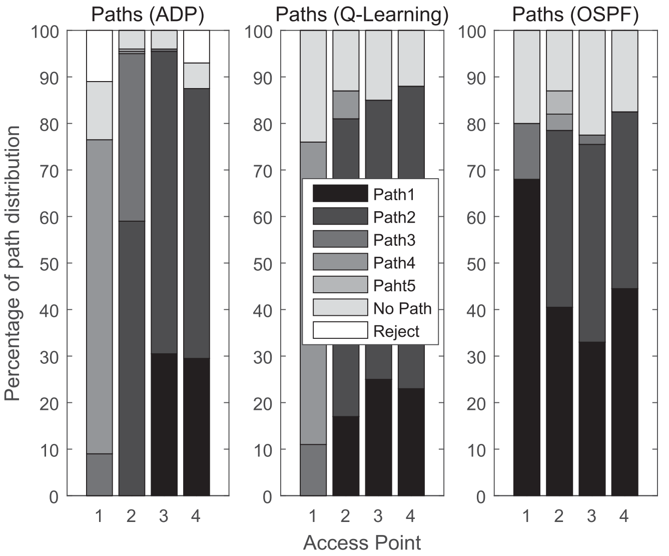


Fig. 7. Path distributions of ADP, Q-Learning, and OSPF algorithm in complex topology.

TABLE VIII
COMPARISON OF ADP, Q-LEARNING, AND OSPF ON THE REWARD AND BANDWIDTH OF REJECTIVE VIDEOS

Algorithm	ADP	Q-Learning	OSPF	All Videos
Average Bandwidth	8.19	8.15	8.08	7.65
Average Reward	18.08	18.89	19.33	20.84

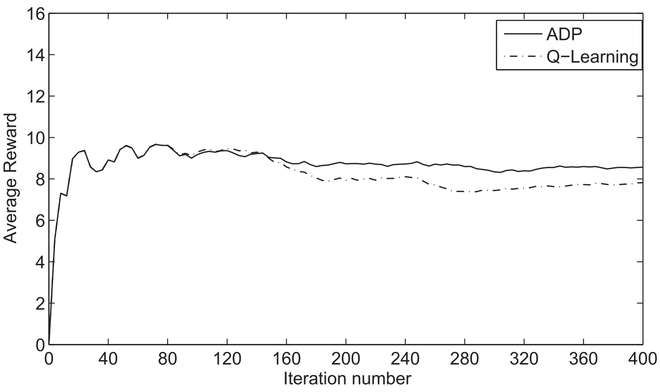


Fig. 8. Convergence of ADP algorithm and Q-learning algorithm.

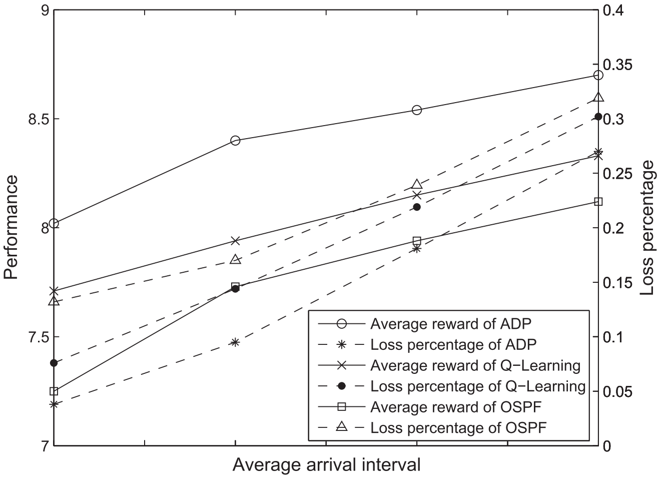


Fig. 9. Performance for different arrival intervals of video requests.

"1 – 2", "1 – 4", "1 – 9" and "1 – 5", although these paths are not the shortest ones. It can be observed that the proposed ADP method initiatively rejects a few video requests even though there are available paths for conveying the video streams. The reason for this is to reserve more resource for near-future video requests having higher reward. In contrast, OSPF based method incurs more reward loss due to admit the video requests as many as possible without considering the long-term reward. This can be further confirmed by Table VIII that characterizes reward loss due to rejecting video requests. It can be observed that the average bandwidth demand of all video requests is 7.65 and the average reward is 20.84. The rewards loss of the rejected videos corresponding to OSPF algorithm is 19.33. By contrast, the reward loss of the proposed ADP method is lower than that of the OSPF based method, which implies that our proposed method proactively rejects the video requests having low rewards, while the OSPF method does not. For the Q-learning algorithm, the four APs also respectively prefer path 4, path 2, path 2 and path 2. However, its difference to the proposed ADP method is the initiative rejection of the video requests, which causes the average reward gap as shown in Table VII.

Convergence: Fig. 8 shows the average reward of the proposed ADP algorithm and Q-learning algorithm versus the learning iterations. It can be observed that the proposed ADP algorithm converges approximately at the 160th iterations. By contrast, the average reward of the Q-learning algorithm is still fluctuating at the 280th iterations. This could be explained by their difference in the architectures and learning principles. For the proposed ADP algorithm, the core of architecture is the kernel based value function approximation. By using the sparse kernel machines, a set of basis vectors are selected from the data samples. By constructing the basis functions with these basis vectors, we could obtain the approximate value function as show in (14). This kernel based feature learning method is to implement data-driven feature representation and learning so that a better learning efficiency can be achieved. However, Q-learning algorithm updates the state-action value by visiting the state-action pair, rather than constructs an approximate value function. This makes the convergence of the Q-learning algorithm depending on the size of the state-action space. Although the state-action space in this paper is finite, it is extremely huge, which reduces the convergence rate as well as the achievable average reward corresponding to the Q-learning algorithm.

Robustness: We further carried out an experiment for illustrating the robustness of the proposed method against the video request arrivals. The time ranges for generating the request arrival interval covers $\{10.5\text{ s}, 9.5\text{ s}, 8.5\text{ s}, 7.5\text{ s}\}$, which represents the different arrival rate. Other parameters were set to the same as those in the experiment of the complex network topology. The experimental results were plotted in Fig. 9. It can be observed that our proposed algorithm outperforms both Q-learning algorithm and OSPF based algorithm. This result illustrates that the proposed method is robust against different arrival rates.

VII. Conclusion

In this paper, we considered the joint admission control and routing optimization for video streaming service in the context of software-defined networking. We conceived an open architecture based on FlowVisor and OpenFlow for promptly deploying advanced optimization strategies. Based on the observed network state provided by OpenFlow Controller, we formulated the joint admission control and routing problem into Markov decision problem of maximizing the overall "revenue". Online learning framework was further proposed to solve the optimization problem via approximation function technique in the context of approximate dynamic programming. The kernel based feature extraction was invoked for constructing the approximation function, which further assisted us to derive the ADP based joint admission control and routing optimization for video streaming service. The advantage of this method is enabling the system capable of self-optimization as well as feasible of handling the curse of dimensionality in the context of MDP. In order to verify the success of the proposed solution, we developed an emulation platform by integrating FlowVisor, POX and Mininet. The experimental results in the simple network topology and the complex network topology show that the proposed method has higher achievable performance in comparison with Q-learning algorithm and the OSPF based heuristic scheme.

Acknowledgment

The authors sincerely appreciate the anonymous reviewers' valuable time and comments to assist us in improving this paper.

References

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2014–2019," Cisco Systems, Inc., San Jose, CA, USA, Jun. 1, 2016. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-n gn-ip-next-generation-network/white_paper_c11-481360.html

[2] S. Sezer *et al.*, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[3] R. Zimmermann *et al.*, "Comprehensive statistical admission control for streaming media servers," in *Proc. ACM Multimedia Conf.*, 2003, pp. 75–78.

[4] S. Bakiras and V. O. K. Li, "Maximizing the number of users in an interactive video-on-demand system," *IEEE Trans. Broadcast.*, vol. 48, no. 4, pp. 281–292, Dec. 2002.

[5] P. Mundur, A. K. Sood, and R. Simon, "Class-based access control for distributed video-on-demand systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 844–853, Jul. 2005.

[6] S. Alwakeel and A. Prasetijo, "Probability admission control in class-based Video-on-Demand system," in *Proc. IEEE Int. Conf. Multimedia Comput. Syst.*, Apr. 2011, pp. 1–6.

[7] Q. M. Qadir, A. A. Kist, and Z. Zhang, "A novel traffic rate measurement algorithm for quality of experience-aware video admission control," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 711–721, May 2015.

[8] Y.-H. Tseng, E. H.-K. Wu, and G.-H. Chen, "An admission control scheme based on online measurement for VBR video streams over wireless home networks," *IEEE Trans. Multimedia*, vol. 10, no. 3, pp. 470–479, Apr. 2008.

[9] C. Cetinkaya, V. Kanodia, and E. W. Knightly, "Scalable services via egress admission control," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 69–81, Mar. 2001.

[10] N. Nasser, "Service adaptability in multimedia wireless networks," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 786–792, Jun. 2009.

[11] H. Park and M. Van der Schaar, "Quality-based resource brokerage for autonomous networked multimedia applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 12, pp. 1781–1792, Dec. 2009.

[12] S. Weber and G. de Veciana, "Rate adaptive multimedia streams: Optimization and admission control," *IEEE/ACM Trans. Netw.*, vol. 13, no. 6, pp. 1275–1288, Dec. 2005.

[13] S. Laté, K. Roobroeck, T. Wauters, and F. D. Turck, "Protecting video service quality in multimedia access networks through PCN," *IEEE Commun. Mag.*, vol. 49, no. 12, pp. 94–101, Dec. 2011.

[14] C. Chen, X. Zhu, G. de Veciana, A. C. Bovik, and R. W. Heath, "Rate adaptation and admission control for video transmission with subjective quality constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 1, pp. 22–36, Feb. 2015.

[15] P. Marbach, O. Mihatsch, and J. N. Tsitsiklis, "Call admission control and routing in integrated services networks using neuro-dynamic programming," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 2, pp. 197–208, Feb. 2000.

[16] D. Oulai, S. Chamberland, and S. Pierre, "A new routing-based admission control for MPLS networks," *IEEE Commun. Lett.*, vol. 11, no. 2, pp. 216–218, Feb. 2007.

[17] J. Domzal *et al.*, "Admission control in flow-aware multi-topology adaptive routing," in *Proc. IEEE Int. Conf. Comput., Netw. Commun.*, Feb. 2015, pp. 265–269.

[18] S. Zhang, F. R. Yu, and V. C. M. Leung, "Joint connection admission control and routing in IEEE 802.16-based mesh networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1370–1379, Apr. 2010.

[19] S. K. Dhurandher *et al.*, "A distributed adaptive admission control scheme for multimedia wireless mesh networks," *IEEE Syst. J.*, vol. 9, no. 2, pp. 595–604, Jun. 2015.

[20] K. M. Elsayed, "A framework for end-to-end deterministic-delay service provisioning in multiservice packet networks," *IEEE Trans. Multimedia*, vol. 7, no. 3, pp. 563–571, Jun. 2005.

[21] S. Namal, I. Ahmad, A. Gurtov, and M. Ylianttila, "SDN based inter-technology load balancing leveraged by flow admission control," in *Proc. 2013 IEEE SDN Future Netw. Serv.*, 2013, pp. 1–5.

[22] S. Miyata, K. Yamaoka, and H. Kinoshita, "Optimal threshold characteristics of call admission control considering cooperative behavior of users," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process.*, Aug. 2013, pp. 165–170.

[23] N. Mckeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM Sigcomm Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[24] "FlowVisor: A network virtualization layer," accessed on: Oct. 2009. [Online]. Available: http://OpenFlowSwitc.org/downloads/technicalreports/openflow-tr-2009-1-flowvisor.pdf

[25] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: A survey," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 24–31, Nov. 2013.

[26] S.-C. Lin, P. Wang, and M. Luo, "Jointly optimized QoS-aware virtualization and routing in software defined networks," *Comput. Netw.*, vol. 96, pp. 69–78, Feb. 2016.

[27] A. X. Porxas, S.-C. Lin, and M. Luo, "QoS-aware virtualization-enabled routing in software defined networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 5771–5776.

[28] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. vol. 1, Belmont, MA, USA: Athena Scientific, 1995.

[29] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley-Intersicence, 2005.

[30] R. S. Sutton and A. Barto, *Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.

[31] "Topics in real and functional analysis," accessed on: Oct. 2004. [Online]. Available: http://www.mat.univie.ac.at/∼gerald/ftp/book-nlfa/nlfa.pdf

[32] W. Wang, Z. Xu, W. Lu, and X. Zhang, "Determination of the spread parameter in the Gaussian kernel for classification and regression," *Neurocomputing*, vol. 55, no. 3, pp. 643–663, 2003.

[33] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2275–2285, Aug. 2004.

[34] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing: A Practical Approach*. Upper Saddle River, NJ, USA: Pearson Education, 2002.

[35] S. Van Vaerenbergh, I. Santamaría, and M. Lázaro-Gredilla, “Estimation of the forgetting factor in kernel recursive least squares,” in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.*, Sep. 2012, pp. 1–6.

[36] S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ, USA: Pearson Education, 1996.

[37] P. Blasco, D. Gunduz, and M. Dohler, “A learning theoretic approach to energy harvesting communication system optimization,” *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1872–1882, Apr. 2013.

[38] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming* (Optimization and Neural Computation Series 3), vol. 7, Belmont, MA, USA: Athena Scientific, pp. 15–23, 1996.

[39] A. Testolin *et al.*, “A machine learning approach to QoE-based video admission control and resource allocation in wireless systems,” in *Proc. IEEE 13th Annu. Mediterranean Ad Hoc Netw. Workshop*, Jun. 2014, pp. 31–38.

[40] M.-H. Lu, P. Steenkiste, and T. Chen, “Video transmission over wireless multihop networks using opportunistic routing,” in *Proc. Packet Video 2007*, 2007, pp. 52–61.

[41] S. Laga *et al.*, “Optimizing scalable video delivery through OpenFlow layer-based routing,” in *Proc. IEEE Netw. Oper. Manage. Symp.*, May 2014, pp. 1–4.

[42] D. Wu, S. Ci, H. Wang, and A. K. Katsaggelos, “Application-centric routing for video streaming over multihop wireless networks,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1721–1734, Dec. 2010.

[43] S. Latré *et al.*, “An autonomic PCN based admission control mechanism for video services in access networks,” in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage.-Workshops*, 2009, pp. 161–168.

[44] L. Chen, S. H. Low, and J. C. Doyle, “Joint congestion control and media access control design for ad hoc wireless networks,” in *Proc. 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Mar. 2005, vol. 3, pp. 2212–2222.

[45] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic power allocation and routing for time-varying wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[46] D. P. Bertsekas and S. Ioffe, “Temporal differences-based policy iteration and applications in neuro-dynamic programming,” Lab. Inform. Decision Syst., MIT, Cambridge, MA, USA, Tech. Rep. LIDS-P-2349, 1996.

[47] R. Yu, Z. Sun, and S. Mei, “Packet scheduling in broadband wireless networks using neuro-dynamic programming,” in *Proc. IEEE 65th Veh. Technol. Conf.*, Apr. 2007, pp. 2776–2780.

[48] D. Huang, W. Chen, P. Mehta, S. Meyn, and A. Surana, “Feature selection for neuro-dynamic programming,” in *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, Hoboken, NJ, USA: Wiley, 2011, pp. 535–559.

[49] A. Mellouk, S. Hoceini, and Y. Amirat, “Adaptive quality of service-based routing approaches: Development of neuro-dynamic state-dependent reinforcement learning algorithms,” *Int. J. Commun. Syst.*, vol. 20, no. 10, pp. 1113–1130, 2007.

[50] C. H. Liu, K. K. Leung, and A. Gkelias, “A generic admission-control methodology for packet networks,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 604–617, Feb. 2014.

[51] S.-L. Hew and L. B. White, “Interference-based dynamic pricing for WCDMA networks using neurodynamic programming,” *IEEE Trans. Autom. Control*, vol. 52, no. 8, pp. 1442–1448, Aug. 2007.

Jian Yang (M’08–SM’15) received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2001 and 2006, respectively.

From 2006 to 2008, he was a Postdoctoral Scholar with the Department of Electronic Engineering and Information Science, USTC. Since 2008, he has been an Associate Professor with the Department of Automation, USTC. His research interests include distributed system design, modeling and optimization, multimedia over wired and wireless, and future network.

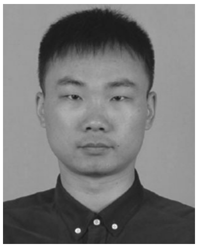
Dr. Yang was the recipient of the Lu Jia-Xi Young Talent Award from the Chinese Academy of Sciences in 2009.

Kunjie Zhu received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2013, where he is currently working toward the Ph.D. degree in the School of Information Science and Technology.

His research interests include multimedia communications, neural network, and stochastic optimization of multimedia transmissions in future networks.

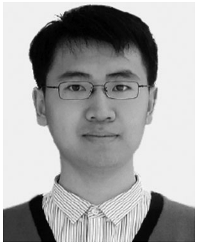
Yongyi Ran received the B.S. and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2008 and 2014, respectively.

He is currently a Postdoctoral Researcher with USTC. His research interests include cloud computing, service management, future network, and stochastic optimization.



Weizhe Cai received the B.S. degree from the Northwestern Polytechnical University, Xi’an, China, in 2012, and is currently working toward the Ph.D. degree at the School of Information Science and Technology, University of Science and Technology of China, Hefei, China.

His research interests include video encoding and decoding, energy harvesting wireless system, and stochastic optimization.



Enzhong Yang received the B.S. degree from the University of Science and Technology of China, Hefei, China, and is currently working toward the Ph.D. degree at the School of Information Science and Technology, University of Science and Technology of China.

His research interests include cloud computing, data center network, and the exploration of new multimedia applications in future networks.