Learning-Based Transmission Protocol Customization for VoD Streaming in Cybertwin-Enabled Next Generation Core Networks

Si Yan, Qiang Ye, *Member, IEEE,* and Weihua Zhuang, *Fellow, IEEE*

***Abstract*—Next generation core networks are expected to achieve service-oriented traffic management for diversified quality-of-service (QoS) provisioning based on software-defined networking (SDN) and network function virtualization (NFV). In this paper, a learning-based transmission protocol customized for video-on-demand (VoD) streaming services is proposed for a Cybertwin-enabled next generation core network, which provides caching-based congestion control and throughput enhancement functionalities at the edge of the core network based on traffic prediction. The per-slot traffic load of a VoD streaming service at an ingress edge node is predicted based on the autoregressive integrated moving average (ARIMA) model. To balance the tradeoff between network congestion and throughput enhancement, a multi-armed bandit (MAB) problem is formulated to maximize the expected overall network performance in a long run, by capturing the relationship between transmission control actions and QoS provisioning. A comprehensive transmission protocol operation framework is also presented with in-network congestion control and throughput enhancement modules. Simulation results are presented to validate the efficacy of the proposed protocol in terms of packet delay, goodput ratio, throughput, and resource utilization.**

***Index Terms*—Cybertwin-enabled next generation core networks, SDN, NFV, network slicing, transmission protocol customization, congestion control, throughput enhancement, MAB, VoD streaming services.**

I. Introduction

To satisfy stringent and differentiated quality-of-service (QoS) demands from diversified applications (e.g., human-centric services, massive ultra-reliable low-latency communication) [1], the next generation core networks are expected to achieve the performance increase by a factor of 10 to 100 times [2]. A driving force for the networking paradigm shift from fifth-generation (5G) to Beyond 5G (B5G) or even sixth-generation (6G) is to enable more efficient control on different protocol and network functions to realize more fine-grained network operation and service customization for better QoS guarantee. Relying on the software-defined networking (SDN), the control intelligence is decoupled from the data plane to centrally manage data traffic over a core network. Traffic flows of the same service type from different end hosts is aggregated at the ingress edge node of a core network and traverse a sequence of network functions (e.g., firewall, intrusion detection system) in the core network for packet-level processing to fulfil the service requirements [3]. With the emergence of network function virtualization (NFV) [4], network/service functions are softwarized and implemented in virtual machines, which are also referred to as virtual network/service functions (VNFs/VSFs) installed in generalized commodity servers [5]. The NFV enables flexible function instantiation on different softwarized platforms, also called NFV nodes, with reduced capital and operational expenditure (CapEx and OpEx). However, with an SDN/NFV-enabled open programmable physical substrate, a fundamental issue is how to efficiently instantiate different layers of virtualized functions, e.g., network functions, transmission control functions, and service functions, to achieve different granularities of QoS provisioning.

Existing works mainly focus on determining the routing path of each traffic flow traversing a sequence of VNFs embedded on different NFV nodes with properly allocated processing and transmission resources to achieve high end-to-end (E2E) performance [6], [7]. However, the VNF placement and routing path configuration with associated resource allocation are performed in a large timescale (e.g. minutes or hours), which do not capture the small-timescale traffic burstiness. To better accommodate traffic variations from different services, more fine-grained transmission control is required to reduce the level of in-network traffic congestion and, at the same time, maintain high E2E throughput and low packet transmission delay. Loss-based transmission protocols, such as TCP Tahoe [8], TCP Reno [8], and TCP NewReno [9], are widely adopted in modern communication systems [10], which use feedback-based observation methods (e.g., transmission timeout, and duplicate ACKs) to detect packet loss and adjust the size of a congestion window (CWND) for controlling the source sending rate. TCP Tahoe aggressively reduces the CWND once a packet loss event is detected, which throttles the E2E throughput [8]. Although TCP Reno and TCP NewReno intend to mitigate throughput reduction, the study in [10] shows that they are less efficient when the bandwidth-delay product (BDP) of a transmission path becomes large. Binary increase control (BIC) [11] and its enhanced version CUBIC [12] are two effective mechanisms to control the congestion window size for networks with large BDP, and the CUBIC can balance the tradeoff between network performance and fairness among

Si Yan and Weihua Zhuang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, N2L 3G1 (email: {s52yan, wzhuang}@uwaterloo.ca).

Qiang Ye (Corresponding author) is with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN 56001 USA (email: qiang.ye@mnsu.edu).

flows. However, the well-known bufferbloat issue for loss-based protocols, i.e., the in-network nodes with large buffers taking a long time to get overflowed, still exacerbates the network congestion and degrades the QoS performance.

Therefore, how to enhance the transmission protocol performance by properly balancing between congestion control and service-oriented QoS provisioning remains an important but challenging issue. A potential solution to make a loss-based protocol react fast to congestion events in the core network is to enhance in-network control capability. With SDN and NFV, some protocol functionalities for in-network congestion detection and reaction can be activated to obtain fast congestion response and E2E performance improvement. Moreover, the in-network control needs to be customized for service-oriented QoS provisioning. Cybertwin is a promising architecture to enable different levels of network control at the edge of core networks [2], [13]. In a Cybertwin-enabled E2E network architecture, edge nodes, connecting end users to a core network, are augmented with higher protocol-layer functionalities, e.g., transmission control, user data logger, and mobility management, to enable more delicate control for network slicing. Specifically, instead of directly connecting to servers in the core network for requesting services, end users first make a connection with a Cybertwin-enabled edge node, which further categories and aggregates user requests into different service groups by interpreting user quality-of-experience (QoE) to service-level requirements based on the application-layer information retrieval (e.g., user identity and location). The edge node then sends the service requests with quantitative QoS requirements to network operators/service providers, on behalf of end users, for creating network slices with properly allocated resources and customized protocols for different services. With Cybertwin, fine-grained transmission control functionalities can be realized in network to enforce more efficient protocol operation and achieve service-oriented protocol customization. In this paper, we present a transmission customization protocol operating at Cybertwin-enabled edge nodes (i.e., ingress and egress nodes) of a core network, where in-network selective caching and enhanced transmission functionalities are enabled for supporting VoD streaming services. Specifically, to mitigate the network congestion level, the ingress node caches a certain number of video packets through selective caching functionality to reduce the E2E delay by taking into consideration the video traffic load and the available resources along the E2E transmission path. The prediction of the number of video packet arrivals in each time slot at the ingress node is based on the ARIMA model for making proactive packet caching decisions. To improve E2E throughput without further incurring new congestion events, the enhanced transmission functionality is activated by re-sending some of the cached video packets from the ingress node to the video clients. To capture the implicit relation between enhanced transmission actions with packet caching and QoS performance with unknown traffic arrival statistics, an action selection module based on the multi-armed bandit (MAB) framework is employed to select proper transmission control actions at the ingress node via balancing exploration and exploitation. The action-selection strategy is updated by observing the feedback reward at the end of each time slot. The *cold-start* problem exists in the considered scenario when the protocol operates under new network conditions [14]. By taking into consideration the cold-start issue, we formulate the control action selection problem as a contextual bandit problem [14], [15]. The LinUCB algorithm is adopted to solve the formulated problem (i.e., determine the control action in each time slot), which has been theoretically proved to have strong regret bound [15]. The contributions of this paper are summarized as follows:

1) A customized transmission protocol for video services (SDP-VS) is presented for a Cybertwin-enabled next generation core network, where in-network selective caching and enhanced transmission functionalities are enabled to balance the tradeoff between network congestion and E2E throughput. Per-slot video traffic load is predicted based on the ARIMA model to make proactive transmission control actions;
2) An MAB-based action selection module is employed at the ingress node to capture the implicit relationship between the congestion control and QoS performance with unknown video traffic arrival statistics. The enhanced transmission function is activated when the network condition improves, which further increases the network resource utilization and the E2E throughput.

The remainder of this paper is organized as follows. In Section II, the system model under consideration is described. Section III presents the proposed SDP-VS protocol, including a detailed description of the protocol operation, the traffic prediction algorithm, and the in-network protocol functionality activation mechanism via the MAB learning. Simulation results are discussed in Section IV, which demonstrate the effectiveness of the proposed protocol. Finally, Section V concludes this work. Main symbols used in this paper are summarized in Table I.

II. System Model

In this section, we present the network model, the VoD streaming system, the protocol functionalities, and the performance metrics.

A. Network Model

We consider a Cybertwin-enabled core network where traffic of one service type from different end source nodes is aggregated as one traffic flow at a core network ingress node. As shown in Fig. 1, multiple traffic flows traverse the core network. Each traffic flow is required to be processed by a chain of VNFs which are implemented on a set of NFV nodes. Between consecutive NFV nodes, there are a number of in-network switches connected by physical links to forward the traffic. The transmission path of each traffic flow in the core network is determined by the SDN controller [16]. To improve resource utilization, more than one traffic flow often passes a common set of network elements (in-network switches, physical links, or NFV nodes) and share the same pool of physical resources [7]. Two types of resources are considered, i.e., 1) computing resources at NFV nodes, and 2) transmission

TABLE I: List of main mathematical symbols

Symbols	Definition
Δ_s	Length of a video segment
$\lambda_j^{(l)}(t_k)$	Traffic load of the j-th cross-traffic flow at V_l
$a(k)$	Control action for the k-th time slot
$a_1(k)$	Action for selective caching functionality
$a_2(k)$	Action for enhanced transmission functionality
C_l	Total capacity of V_l
d	Degree of differencing
$\mathbf{D}_a$	Matrix of observations for arm a
$d_a(k)$	Average E2E delay of the k-th time slot
M_l	Number of cross-traffic flows at node V_l
N_e	Number of enhancement layers of a video segment
p	Order of the autoregressive model
q	Order of the moving-average model
$r(k)$	E2E available resources of VoD streaming slice for the k-th time slot
$\mathbf{R}_a$	Response vector of action a
$R_{a(k)}(k)$	Reward of executing action $a(k)$ in the k-th time slot
$r_l(t_k)$	Available resources of V_l at time t_k
$\hat{t}(k)$	Predicted video traffic load of the k-th time slot
$t_i(k)$	Actual Video traffic load of layer i for the k-th time slot
$\hat{t}_i(k)$	Predicted video traffic load of layer i for the k-th time slot
T_r	E2E delay requirement
T_s	Duration of a time slot
V_l	The l-th node in the VoD streaming slice
$\mathbf{x}_k$	Context information of the k-th time slot

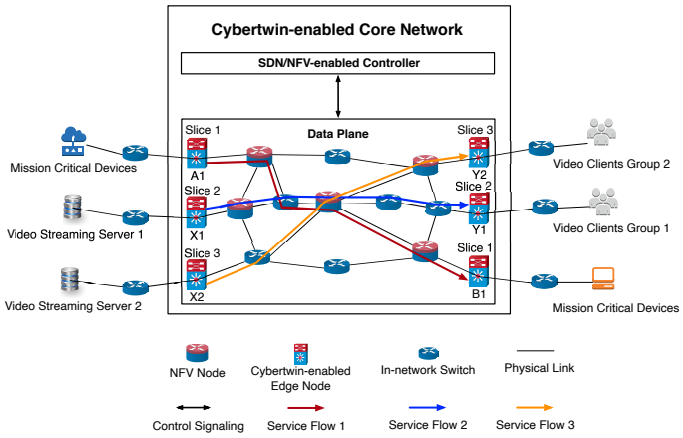


Fig. 1: Multiple services network topology.

resources over physical links [3]. Given the transmission path and the allocated resources of a traffic flow, a customized transmission protocol is employed to achieve service-oriented control. To ensure QoS isolation among different services, a *network slice* is created to support the packet transmission of each flow, which consists of an E2E transmission path with properly allocated resources and customized transmission protocol.

A unicast VoD streaming slice has a linear topology between a pair of edge nodes (e.g., Slice 2 in Fig. 1). The set of nodes in the slice is denoted by $\mathcal{V} = \{V_1, V_2, \ldots, V_L\}$, where L is the total number of nodes in the slice. A node is either an in-network switch or an NFV node which has a first-in-first-out (FIFO) buffer to queue arrived packets. We assume that the buffer always has sufficient space to queue a newly arrived packet. The bottleneck resource type of an in-network switch and NFV node is transmission resources and computing resources respectively. Here, the resource type of a node in a VoD streaming slice refers to its bottleneck resource type. At each node, the video traffic flow shares the resources with multiple cross-traffic flows. The number of cross-traffic flows traversing node V_l is denoted by M_l ($l = 1, 2, \ldots, L$). Time is partitioned into slots of constant duration T_s [17]. Denote t_k as the time instant when the k-th time slot starts. At t_k, the average traffic rate of the j-th cross-traffic flow at V_l is calculated as [18]

$$\lambda_j^{(l)}(t_k) = \left\lfloor \frac{n_j(t_k) - n_j(t_k - T_s)}{T_s} \right\rfloor \tag{1}$$

where $n_j(t_k)$ represents the number of packets of the j-th flow that have arrived at V_l by t_k, $\lfloor \cdot \rfloor$ is the floor function. Denote by C_l the total capacity (in packet/s) of V_l. The available resources (in packet/s) on V_l at time t_k, denoted by $r_l(t_k)$, is given by [18]

$$r_l(t_k) = C_l - \sum_{j=1}^{M_l} \lambda_j^{(l)}(t_k). \tag{2}$$

The E2E available resources, $r(k)$, at the k-th time slot for a VoD streaming slice is determined as

$$r(k) = \min\{r_1(t_k), r_2(t_k), \ldots, r_L(t_k)\}. \tag{3}$$

The server-side edge node (client-side edge node) of the VoD streaming slice is the ingress node (egress node) which is assumed to have enough caching resources to buffer the

packets chosen by the selective caching functionality. For example, nodes X_1 and Y_1 in Fig. 1 are the ingress and egress nodes respectively of Slice 2. For backward compatibility on end hosts, the ingress (egress) node is an in-network proxy server which maintains the TCP connections with the video server (clients) [19]. The ingress node replies an ACK packet to the video server for every received video packet. All the video packets received by the egress node are converted to TCP packets, which are copied and cached at the egress node, and are then forwarded to the corresponding video clients. Each video client replies an ACK packet of each received video packet for acknowledgement. When the egress node receives an ACK packet from a video client, it removes the corresponding video packet from the egress node caching buffer. However, if a video packet is lost between the egress node and the video client, the egress node either receives duplicate ACKs or experiences retransmission timeout. In this case, the egress node retransmits the lost packet and activates the TCP congestion control mechanism.

B. VoD Streaming System

The scalable video coding (SVC) technique is used to encode video files in the server [20]. Each video is divided into a series of video segments. Denote by Δ_s the length of a segment. Each segment is further encoded into several layers, including one base layer and N_e enhancement layers. Different layers of a video segment can be stored and streamed independently in form of small video chunks. The base-layer chunks are required to decode segments at video clients. An enhancement-layer chunk can be decoded only if all the lower enhancement-layer chunks and the base-layer chunk from the same video segment are received by the client. The more enhancement-layer chunks are received, the higher video quality will be. Before sending the chunks into the network, each chunk is fragmented and encapsulated into multiple video packets. The quality of the streamed video segments, indicated by the number of SVC layers, is controlled by the video clients [21], [22]. When all the base-layer packets of the requested segments are received by a client, the client needs to determine the requested quality for the following several segments based on the current buffer level, i.e., the number of playable video segments in the client buffer. The desired quality information is transmitted to the video server by the HTTP GET message [23].

C. Protocol Functionalities

To achieve in-network control for a VoD streaming slice, SDP-VS incorporates the following functionalities: header conversion functionality, selective caching functionality, and enhanced transmission functionality. When a congestion event occurs in the VoD streaming slice, the ingress node selectively caches incoming packets into the caching buffer. Once the network condition improves, the packets which can enhance video quality are retrieved from the caching buffer for enhanced transmission. At the beginning of each time slot, the ingress node of VoD streaming slice selects appropriate functionality

	1 - 8 bits	9 - 16 bits	17 - 24 bits	25 – 32 bits
1	Protocol	Total Length		Data Offset
2	Checksum		Flag	
3	Ingress Node Address			
4	Egress Node Address			
5	Ingress Node Port Number		Egress Node Port Number	
6-8	Client ID			
9	Segment Number			Layer Number

Fig. 2: The SDP-VS header.

based on the network condition. The protocol functionalities of SDP-VS is described in the following:

1) **Header conversion functionality** - It is deployed at the ingress node to add an SDP-VS header over all the video packets passing through [23]. The header format is shown in Fig. 2. Between the edge nodes of a VoD streaming slice, the source (destination) IP address of the video packet is indicated by the *Ingress (Egress) Node Address* field. The sending (receiving) port number at the ingress (egress) node is included in the *Ingress (Egress) Node Port Number* field. The fields enclosed by the red dashed rectangular box is referred to as *slice ID* for slice differentiation. The *Protocol* field indicates the applied transmission protocol for the video traffic flows in the core network, i.e., SDP-VS. The fields of *Total Length*, *Data Offset* and *Checksum* are necessary to packets traversing the network. The *Flag* field is used to differentiate the types of packets in the VoD streaming slice. The *Client ID* contains the IP addresses and port numbers of the server and clients. The *Segment Number* and *Layer Number* of a video packet are extracted from the application layer payload at the ingress node. Note that the layer number of base-layer packets and i-th enhancement-layer packets is 0 and i respectively;
2) **Selective caching functionality** - An SVC codec enables flexible video decoding, and video contents can be successfully decoded even in the absence of enhancement-layer packets. Hence, higher layer packets can be selectively cached in the network, without significant degradation of user experience. By exploiting the caching resources, instead of dropping packets when network is congested, we design a selective in-network caching policy to temporarily store certain packets at the ingress node for a fast response to network dynamics;
3) **Enhanced transmission functionality** - To increase the video quality once the network condition improves, we design an enhanced transmission functionality for SDP-VS. At each time slot when the enhanced transmission is activated, the ingress node determines how many cached packets should be transmitted in the slot, and the cached packets are pushed from the caching buffer to the VoD streaming slice.

D. Performance Metrics

To verify the performance of the proposed SDP-VS, we evaluate the following four QoS metrics in VoD streaming systems with and without SDP-VS:

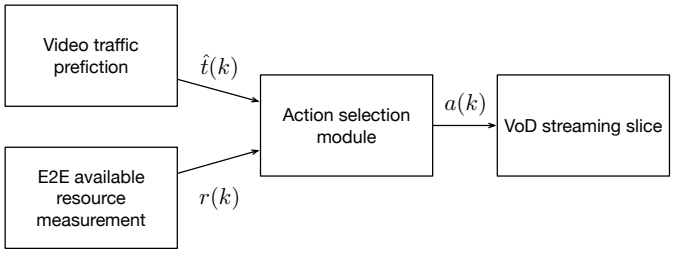


Fig. 3: The framework of SDP-VS.

1) **Average E2E delay** - the E2E delay, consisting of packet queueing delay, packet processing delay, link transmission delay and propagation delay experienced in the core network, averaged over all the packets passing through the egress node during a time slot;
2) **Throughput** - the total number of video packets from a VoD streaming slice passing through the egress node in one second;
3) **Goodput ratio** - the number of packets with bounded E2E delay over total number of packets passing through the egress node during one time slot;
4) **Resource utilization** - throughput over E2E available resources for the VoD streaming slice.

III. Learning-Based Transmission Protocol for VoD Streaming

The proposed SDP-VS protocol is presented in this section. We describe the protocol operations, including three main components: 1) traffic prediction module, 2) E2E available resources measurement module, and 3) action selection module for selecting control actions.

A. SDP-VS Framework

SDP-VS controls the packet queueing delay during the network congestion and enhances the throughput once the congestion event is over by adjusting the traffic load for a VoD streaming slice. It achieves traffic management by taking different control actions at the ingress node. When the selective caching functionality is activated, some incoming video packets from the video traffic flow are cached in the caching buffer at the ingress node. If the enhanced transmission functionality is enabled, the cached video packets are transmitted from the ingress node to the video clients. To operate SDP-VS, three functional modules are implemented at the ingress node of a VoD streaming slice, i.e., video traffic prediction module, E2E available resources measurement module, and action selection module. The relationship among the modules is shown in Fig. 3. The video traffic prediction module estimates the traffic load of the next time slot based on the traffic loads observed in the last several time slots. The E2E available resources measurement module is used to monitor the available resources for a VoD streaming slice during the network operation. The action selection module is the key of the proposed SDP-VS which selects the control action in each time slot based on the output of the other two functional modules.

Denote by $\hat{t}(k)$ the output of video traffic prediction module. To limit the dimensionality of the action space, the selective caching and enhanced transmission functionalities operate at SVC layer level and packet chunk level respectively. The packet chunks for enhanced transmission are labelled as ET-chunks. All the ET-chunks contain the same number, N_c, of video packets. Denote by N_E the pre-determined maximum number of ET-chunks transmitted in one time slot. Let $\mathcal{A}$ denote the set of all possible control actions, each of which is denoted as a two-element tuple, $(i, j) \in \mathcal{A}$, where $i = 0, 1, \ldots, N_e$ and $j = 0, 1, \ldots, N_E$. The value of i and j indicates the actions of selective caching and enhanced transmission functionalities, respectively. In the k-th time slot, action tuple (i, j) is further represented by $a(k) = (a_1(k), a_2(k))$, and the ingress node caches all the incoming packets whose layer number is greater than $a_1(k)$. To avoid a video rebuffering event (i.e., stalled video playback), the base-layer packets are not cached. When $a_1(k)$ is equal to 0, all the enhancement-layer packets arrived at the ingress node during the k-th time slot are pushed into the caching buffer. If $a_1(k)$ equals N_e, no video packet needs to be cached in the k-th time slot. The value of $a_2(k)$ represents the number of packet chunks which should be transmitted by enhanced transmission functionality in the k-th time slot.

The main procedure of how SDP-VS is operated for the VoD streaming slice is illustrated in Fig. 4. At the end of the k-th time slot, the egress node measures average E2E delay $d_a(k)$. If $d_a(k)$ is greater than required delay bound T_r, the egress node enters the active mode and sends a `Congestion_Notification` (`CN`) message to the ingress node traversing the entire VoD streaming slice. An intermediate node in the VoD streaming slice changes to the active mode as soon as it receives a `CN` message. Once the action selection module at the ingress node receives the `CN` message, it sets the action of both selective caching and enhanced transmission functionalities as 0 for the $(k + 1)$-th time slot, i.e., $a(k+1) = (0, 0)$. The purpose of caching all the enhancement-layer packets in the $(k + 1)$-th time slot is to reduce the queueing delay of the video packets to a maximal extent. The egress node measures $d_a(k+1)$ which is included in a `Delay` message sent back to the ingress node. If $d_a(k+1)$ is greater than the delay bound, the ingress node keeps caching all the enhancement-layer packets in the following time slots until the average E2E delay is less than T_r. If the average E2E delay of the j-th time slot satisfies the delay requirement, the action selection module determines the action tuples of the following time slots, where the decision is made based on the predicted traffic load and the E2E available resources of the VoD streaming slice. The egress node measures and sends the feedback reward of executing the control action to the ingress node at the end of the time slot. The information is used to update the action-selection strategy. Since the egress node needs to update both the average E2E delay and the feedback reward at the end of each time slot during the active mode, the `Delay` message and `Reward` message can be encapsulated into one packet. While this packet passes through an intermediate node, the node attaches its current available resources information. The available resources measurement

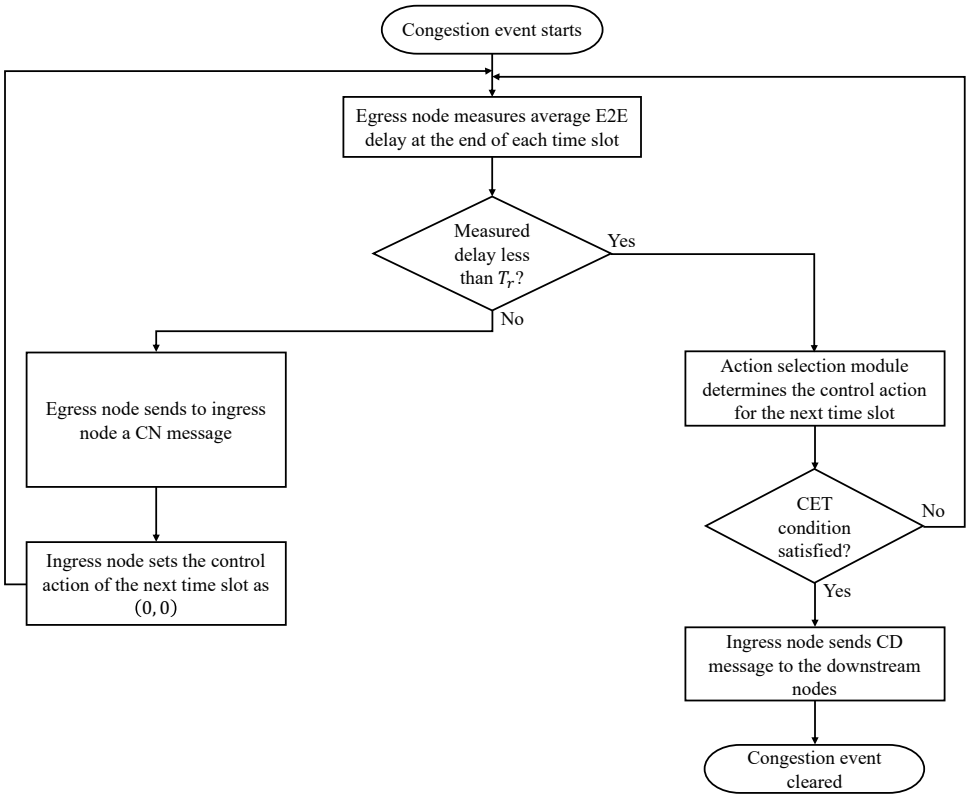


Fig. 4: The main protocol operation of SDP-VS when a congestion event happens.

module at the ingress node uses this information from all the intermediate nodes in the slice to determine the E2E available resources. When the congestion event is over, the ingress node sends the cached packets to the corresponding video clients by enhanced transmission. Suppose the caching buffer at the ingress node becomes empty in the k-th time slot and $a_1(k)$ is N_e (i.e., no video packet is cached in the k-th time slot). The ingress node enters the deactivated mode and sends a `CONTROL_DEACTIVATION` (`CD`) message to the downstream nodes in the VoD streaming slice at the end of the k-th time slot. The condition of triggering the `CD` message is referred to as CET condition. A node changes to the deactivated mode when it receives a `CD` message. The egress node stops measuring the feedback reward and sending the `Reward` message until the next congestion event occurs in the network. The protocol operation of SDP-VS for the VoD streaming slice in the active mode is summarized in Algorithm 1. The main responsibilities of edge nodes in the VoD streaming slice are summarized in Table II. The items followed by (all) are the responsibilities required throughout the entire network operation, otherwise, the items are required only when the nodes are in the active mode.

Next, we describe the mechanism of managing the caching buffer at the ingress node. The caching buffer is operated in the FIFO manner. To better use the caching resources, the caching buffer drops the packets from the video segments which have been played out by the clients. The video clients periodically report the buffer information to the SDN/NFV-enabled controller of the core network, containing the segment number of video segment being played out [24]. Then, the controller forwards this information to the ingress node of the VoD streaming slice. When the caching buffer receives the message of buffer information, it removes the packets of the same client whose segment number is less than or equal to the segment number indicated in the message.

Algorithm 1 Protocol operation of SDP-VS

for each time slot **do**
  Egress node measures the average E2E delay.
  if the measured delay is greater than T_r **then**
    Egress node sends `CN` message to the ingress node.
    Ingress node sets the action of selective caching functionality for the next time slot as 0.
    Ingress node sets the action of enhanced transmission functionality for the next time slot as 0.
  else
    Video traffic prediction module predicts the video traffic load for the next time slot.
    action selection module determines the action tuple of the next time slot.
  end if
end for

B. Video Traffic Prediction

The video traffic prediction module in Fig. 3 is used to forecast the video traffic load in each time slot. Since the congestion control action selection is conducted at different

TABLE II: The responsibilities of the edge nodes in the VoD streaming slice

Node type	Responsibilities
Ingress node	- Control action selection and execution - E2E available resources measurement - Video traffic prediction - Sending `CD` messages
Egress node	- Average E2E delay measurement (all) - Feedback reward measurement - Sending `CN` messages - Sending `Delay` and `Reward` messages

encoded video layers, the traffic load in each time slot is predicted at different SVC layers. The maximum number of enhancement-layers, N_e, of all the video files stored at the video server is assumed to be identical. Thus, the output dimension from the video traffic prediction module is N_e+1. The prediction result for the k-th time slot is expressed as

$$\hat{\mathbf{t}}(k) = \Big[\hat{t}_0(k), \hat{t}_1(k), \hat{t}_2(k), \ldots, \hat{t}_{N_e}(k)\Big] \tag{4}$$

where $\hat{t}_i(k)$ represents the predicted number of packet arrivals of layer i in the k-th time slot. The predicted traffic load of base-layer packets is denoted by $\hat{t}_0(k)$. Note that we only need to have one traffic prediction module at the ingress node which are fed with the information of each SVC layer to obtain layer-level traffic prediction results. The ARIMA model is used for video traffic prediction, which takes the traffic load of the past time slots as input and predicts the amount of packet arrivals in the next time slot [25], [26].

1) Model parameters: The ARIMA model is specified by three parameters d, p and q, where d is the degree of differencing (i.e., the number of differencing to eliminate the trend of a non-stationary time series), p is the order of the autoregressive model, and q is the order of the moving-average model. The parameters can be determined by analyzing the historical traffic load patterns. Denote by $h_i(k)$ the observed traffic load of layer i (i.e. the number of video packets of layer i arrived at the ingress node) during the k-th time slot. The time series of the historical traffic load is represented by $\{h_i(k)\}$. Let $\mathbf{h}_i(T)$ denote a vector of traffic loads observed in T time slots, given by

$$\mathbf{h}_i(T) = \Big[h_i(1), h_i(2), \ldots, h_i(T)\Big]. \tag{5}$$

Let $\nabla^{\mathbf{c}}\mathbf{h}_i(T)$ denote the c-th-order difference of $\mathbf{h}_i(T)$. The value of c is determined by conducting the augmented Dickey-Fuller (ADF) test for $\nabla^{\mathbf{c}}\mathbf{h}_i(T)$ $(c = 0, 1, \ldots)$ [25], [27]. If the p*-value*[1] for $\nabla^{\mathbf{c}}\mathbf{h}_i(T)$ is less than a pre-determined threshold (e.g., 0.05), the corresponding time series, $\{\nabla^c h_i(k)\}$, is stationary. Then, parameter d is set as c, otherwise, more differencing is required to transform $\{\nabla^c h_i(k)\}$ to a stationary time series. Given d, the selection of parameters p and q is based on the minimization of the corrected Akaike information criterion statistic [25]. Time series $\{\nabla^d h_i(k)\}$ being stationary indicates that its mean is constant. Denote by μ_i the sample mean of $\nabla^{\mathbf{d}}\mathbf{h}_i(T)$.

2) Traffic Prediction via ARIMA Model: Given d, p and q, the ARIMA model predicts the traffic load at the beginning of each time slot during the network operation. The vector of the observed traffic loads for the first $d+k$ time slots during the network operation is expressed as

$$\mathbf{t}_i(d+k) = \Big[t_i(1), \ldots, t_i(d), t_i(d+1), \ldots, t_i(d+k)\Big]. \tag{6}$$

Let $\nabla^{\mathbf{d}}\mathbf{t}_i(d+k)$ denote the d-th-order difference of $\mathbf{t}_i(d+k)$, which is represented as

$$\nabla^{\mathbf{d}}\mathbf{t}_i(d+k) = \Big[\nabla^d t_i(d+1), \nabla^d t_i(d+2), \ldots, \nabla^d t_i(d+k)\Big]. \tag{7}$$

From [25], the predicted traffic load of layer i in the $(d+k+1)$-th time slot is given by

$$\begin{aligned}\hat{t}_i(d+k+1) =& \widehat{\nabla^d t_i}(d+k+1) \\ &- \sum_{j=1}^{d}\binom{d}{j}(-1)^j t_i(d+k+1-j)\end{aligned} \tag{8}$$

where $\widehat{\nabla^d t_i}(d+k+1)$ is the prediction of $\nabla^d t_i(d+k+1)$ given $\nabla^{\mathbf{d}}\mathbf{t}_i(d+k)$. Now, the traffic prediction problem becomes how to determine $\widehat{\nabla^d t_i}(d+k+1)$. We define $\mathbf{y}_i(k)$ as

$$\mathbf{y}_i(k) = \Big[y_i(1), y_i(2), \ldots, y_i(k)\Big] \tag{9}$$

where $y_i(j)$ $(j = 1, 2, \ldots, k)$ is equal to $\nabla^d t_i(d+j) - \mu_i$. Let $\hat{y}_i(k+1)$ denote the prediction of $y_i(k+1)$. Since μ_i is estimated before the network operation begins based on the historical traffic load patterns, the traffic prediction problem is finally converted to determining $\hat{y}_i(k+1)$ given $\mathbf{y}_i(k)$. The recursive equation of finding the value of $\hat{y}_i(k+1)$ is given by

$$\hat{y}_i(k+1) = \begin{cases} \sum_{j=1}^{k}\theta_{k,j}[y_i(k+1-j) \\ \quad - \hat{y}_i(k+1-j)],\ 1 \le k < v \\ \sum_{j=1}^{q}\theta_{k,j}[y_i(k+1-j) - \hat{y}_i(k+1-j)] \\ \quad + \alpha_1 y_i(k) + \cdots + \alpha_p y_i(k+1-p),\ k \ge v \end{cases} \tag{10}$$

where v is the maximum of p and q [25], [26]. Note that $\hat{y}_i(1)$ equals 0. The coefficients in (10) (i.e., $\alpha_1, \ldots, \alpha_p, \theta_{k,j}$) can be calculated recursively as in [25].

C. Action-Selection via Multi-Armed Bandit Learning

The deployment of selective caching and enhanced transmission functionalities at the ingress node from a VoD streaming slice is to deliver more video packets without leading to network congestion. Therefore, we define the reward of executing action $a(k)$ in the k-th time slot as

$$R_{a(k)}(k) = \frac{g(k)}{r(k)T_s} \tag{11}$$

[1] p-value is used in statistical test for determining whether to reject the null hypothesis, which is different with the above-mentioned parameter p.

where $g(k)$ is the number of video packets sent from the VoD streaming slice in the k-th time slot within the required delay bound T_r. Through triggering different actions in each time slot, the ingress node intends to maximize the expected overall reward over time.

The per-slot reward of executing an action under different network conditions may be different. Caching video packets during a congestion event can reduce the packet E2E delay to increase the reward, whereas the reward is decreased if the ingress node activates the selective caching functionality when there is no congestion event. Therefore, the video traffic load and E2E available resources of the VoD streaming slice should be taken into consideration when the action selection module selects the control actions at each time slot. We formulate this action-selection problem as an MAB problem, where the predicted video traffic load and E2E available resources are treated as the context information. The MAB problem which uses context information for decision making is also referred to as contextual bandit problem [15], where the selected arms are control actions at each time slot.

Let $\mathbf{x}_k$ denote the context information at the k-th time slot, given by

$$\mathbf{x}_k = \left[\hat{t}_0(k), \hat{t}_1(k), \hat{t}_2(k), \ldots, \hat{t}_{N_e}(k), r(k)\right]. \quad (12)$$

The LinUCB algorithm is employed to solve the MAB problem with context information [15]. For the k-th time slot, the expected reward of an action $a \in \mathcal{A}$ is expressed as

$$\mathbf{E}\left[R_a(k)|\mathbf{x}_k\right] = \mathbf{x}_k^T \theta_a^* \quad (13)$$

where θ_a^* is an unknown coefficient vector. Assume m contexts of action a have been observed before the k-th time slot and the corresponding feedback rewards are recorded by response vector $\mathbf{R}_a \in \mathbb{R}^m$. The matrix of the m observed contexts for action a is denote by $\mathbf{D}_a \in \mathbb{R}^{m \times z}$, where z is the dimension of the context (i.e., $N_e + 2$). The estimate of the coefficient vector, θ_a^*, is given by

$$\hat{\theta}_a = \left(\mathbf{D}_a^T \mathbf{D}_a + \mathbf{I}_z\right)^{-1} \mathbf{D}_a^T \mathbf{R}_a \quad (14)$$

where $\mathbf{I}_z$ is the $z \times z$ identity matrix. It is shown in [15] that, for any $\delta > 0$, the following inequality holds with a probability of at least $1 - \delta$,

$$\left| \mathbf{x}_k^T \hat{\theta}_a - \mathbf{E}\left[R_a(k)|\mathbf{x}_k\right] \right| \leq \xi \sqrt{\mathbf{x}_k^T \mathbf{A}_a \mathbf{x}_k} \quad (15)$$

where

$$\mathbf{A}_a = \mathbf{D}_a^T \mathbf{D}_a + \mathbf{I}_z, \quad \xi = 1 + \sqrt{\frac{\ln(2/\delta)}{2}}. \quad (16)$$

At the beginning of the k-th time slot, the action selection module selects the action which maximizes $\hat{R}_a(k)$ as

$$\hat{R}_a(k) = \mathbf{x}_k^T \hat{\theta}_a + \xi \sqrt{\mathbf{x}_k^T \mathbf{A}_a^{-1} \mathbf{x}_k}. \quad (17)$$

Recall that the selected control action in the k-th time slot is denoted by $a(k)$. The actual reward, $R_{a(k)}(k)$, of taking action $a(k)$ in the k-th time slot is observed at the end of the slot. Then, the tuple, $(\mathbf{x}_k, a(k), R_{a(k)}(k))$, is fed back to the action selection module to update the parameters of the LinUCB algorithm. The detailed action-selection algorithm when the VoD streaming slice is in the active mode is presented in Algorithm 2.

Algorithm 2 Action-selection algorithm

Initialize $\xi \in \mathbb{R}_+$ and $d_a(0) = 0$.
for $k = 1, 2, \ldots$ **do**
  if $d_a(k-1) > T_r$ **then**
    Set the action tuple of the k-th time slot as $(0, 0)$.
  else
    Obtain the context information: $\mathbf{x}_k \in \mathbb{R}^z$.
    for every $a \in \mathcal{A}$ **do**
      if a is new **then**
        $\mathbf{A}_a \leftarrow \mathbf{I}_z$
        $\mathbf{b}_a \leftarrow \mathbf{0}_{z \times 1}$
      end if
      $\hat{\theta}_a \leftarrow \mathbf{A}_a^{-1} \mathbf{b}_a$
      $\hat{R}_a(k) \leftarrow \mathbf{x}_k^T \hat{\theta}_a + \xi \sqrt{\mathbf{x}_k^T \mathbf{A}_a^{-1} \mathbf{x}_k}$
    end for
    Set the action tuple of the k-th time slot $a(k) = \arg\max_{a \in \mathcal{A}} \hat{R}_a(k)$.
    Observe the actual reward $R_{a(k)}(k)$ at the end of k-th time slot.
    $\mathbf{A}_a \leftarrow \mathbf{A}_a + \mathbf{x}_k \mathbf{x}_k^T$
    $\mathbf{b}_a \leftarrow \mathbf{b}_a + R_{a(k)}(k) \mathbf{x}_k$
  end if
end for

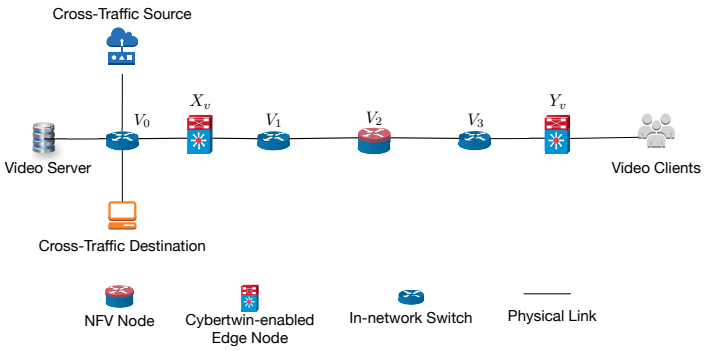


Fig. 5: Network topology for performance evaluation.

IV. Performance Evaluation

In this section, we conduct the performance evaluation of the proposed SDP-VS. As mentioned, four QoS performance metrics are considered in the evaluation, i.e., average E2E delay, throughput, goodput ratio and resource utilization.

A. Simulation Settings

The network topology considered in our simulation is shown in Fig. 5. Five video clients download video files from the same video server [23]. The duration of a video segment, Δ_s, of all video files is 2 seconds [28]. Every segment is encoded into one base-layer chunk and four enhancement-layer chunks [29]. Each video chunk is delivered by 200 video packets, and

TABLE III: Packet arrival rate of the cross-traffic at V_0

Time interval	$[1, 40]$	$[41, 80]$	$[81, 120]$
Packet arrival rate	1100 packet/s	1400 packet/s	1700 packet/s

TABLE IV: ADF test results when $d = 0$

ADF statistic	p-value	Critical value (1%)	Critical value (5%)	Critical value (10%)
-0.913	0.784	-3.489	-2.887	-2.580

the packet size is constant, equals to 1400 bytes [23]. The aggregated video traffic flow passes through switch V_0 to ingress node X_v. Nodes X_v and Y_v are the ingress node and the egress node for the VoD streaming slice, respectively. The VoD streaming slice between the edge nodes has a linear topology which contains two in-network switches (i.e., V_1 and V_3) and one NFV node (i.e., V_2). As discussed in Section II, the edge nodes are in-network servers which have much more resources than NFV nodes and in-network switches. The capacity of node V_l $(l = 0, 1, 2, 3)$ is $C_l = 4500$ packet/s [30]. The video traffic flow and the cross-traffic flow share the transmission resources at V_0. During the network operation, we change the packet arrival rate of the cross-traffic at V_0 according to the settings in Table III to evaluate the performance of the proposed SDP-VS. The loss-based congestion control algorithm based on the additive-increase multiplicative-decrease (AIMD) mechanism is implemented at the video server to control the source sending rate [8]. The propagation delay of the links outside (between) the edge nodes is set as 5 ms (2.5 ms). The E2E delay bound T_r is set to 40 ms. Parameter ξ in (16) is 1.5 [15]. The total simulated slice time is 120 s and the length of every time slot is 1 s. No cached packet is dropped during the network operation.

We first determine parameters p, q and d of the ARIMA model for video traffic load prediction. The ingress node collects the video traffic loads of 120 time slots for data analysis. The augmented Dickey-Fuller (ADF) test is used to check the video traffic stationarity over time [25], [27]. The test results when $d = 0$ are given in Table IV. Since the p-value of the traffic load series is greater than a pre-determined threshold set as 0.05 [31], the video traffic load series when $d = 0$ is not stationary. We conduct the ADF test when $d = 1$, and the test results given in Table V indicate that the p-value is much less than the threshold, 0.05. In addition, the ADF statistic is less than all the critical values, indicating that the time series is stationary with a 99% confidence level [31]. Thus, parameter d is set as 1 in the simulation. Then, we select parameters p and q by evaluating the AICC statistic. Based on the observed traffic loads, the AICC statistic is minimized when $p = 2$ and $q = 1$. Therefore, the ARIMA model with $d = 1$, $p = 2$, and $q = 1$ is used for the video traffic prediction.

B. Numerical Results

The average E2E delay, goodput ratio, throughput, and resource utilization for the VoD streaming systems with and without SDP-VS are compared. For brevity, we denote the VoD streaming system with (without) SDP-VS by VS-W (VS-WO) system. Fig. 6 and Fig. 7 show the average E2E delay and goodput ratio performance, respectively. The results are obtained from ten repeated simulations. The thoughput and resource utilization of each time slot in one simulation are presented in Figs. 8 - 9 respectively.

TABLE V: ADF test results when $d = 1$

ADF statistic	p-value	Critical value (1%)	Critical value (5%)	Critical value (10%)
-9.627	1.647×10^{-16}	-3.489	-2.887	-2.580

TABLE VI: The available resources of V_2 over time

Time interval	$[1, 20]$	$[21, 40]$	$[41, 60]$	$[61, 120]$
Congestion duration 20 s	4500 packet/s	2500 packet/s	4500 packet/s	4500 packet/s
Congestion duration 40 s	4500 packet/s	2500 packet/s	2500 packet/s	4500 packet/s

Average E2E delay: We first examine the average E2E delay of VS-W and VS-WO when a congestion event occurs in the VoD streaming slice. The network congestion is generated by reducing the available resources of V_2 from 4500 packet/s to 2500 packet/s. The resource of V_2 changes with time as specified in Table VI. Two congestion durations, 20 s and 40 s, are considered in the simulation. The cumulative distribution function (CDF) of the average E2E delay is measured and shown in Fig. 6. It can be seen that the CDFs of VS-W with different congestion intervals are close to each other. Also, the average E2E delay of the VS-W system measured in each time slot is less than the required delay bound, since the selective caching functionality is activated after the congestion happens. By caching a certain number of enhancement-layer packets, the queue length at V_2 is under control. Fig. 6 also shows the results of VS-W system whose action selection module is fed with the predicted and real traffic load, respectively, for each time slot. The results obtained based on predicted and real video traffic load information are similar, thanks to the effectiveness of the traffic prediction algorithm. For VS-WO, the average E2E delay in around 17% time slots exceeds the required delay bound due to a 20 s congestion event. In around 32% time slots, the delay requirement is not satisfied when a 40 s congestion event occurs. Thus, the performance gap between VS-W and VS-WO systems becomes larger if the network congestion lasts longer. It is observed that the CDF of VS-WO is greater than that of VS-W when the average E2E delay is 0.02 s. For VS-WO, the queueing delay is negligible after the network congestion. Hence, the average E2E delay within these time slots is in the range between 0.01 s and 0.02 s. However, the enhanced transmission functionality is activated in the VS-W system after network congestion. As a result, the average E2E delay within the corresponding time slots increases to some extent, but does not exceed the required delay bound.

Goodput ratio: In evaluating the goodput ratio, the available

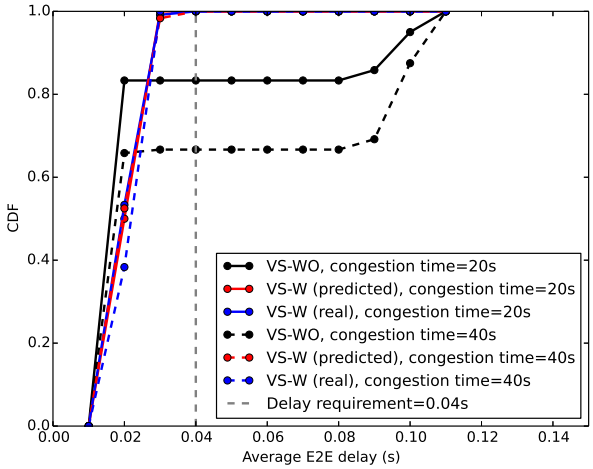


Fig. 6: Performance of average E2E delay.

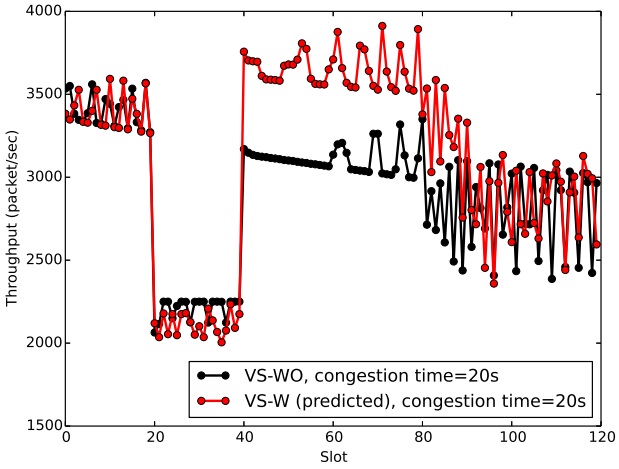


Fig. 8: Throughput with regard to the number of slots.

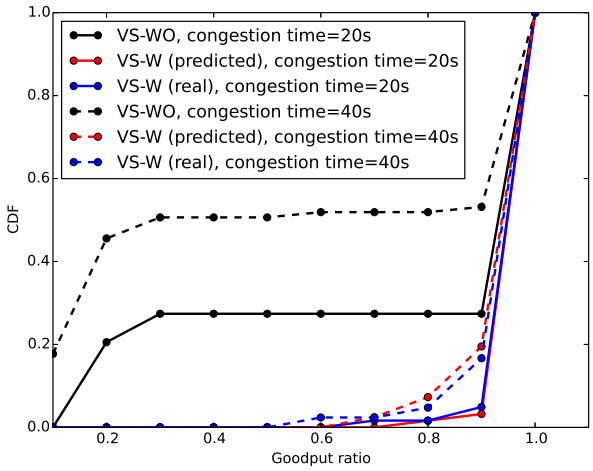


Fig. 7: Performance of goodput ratio.

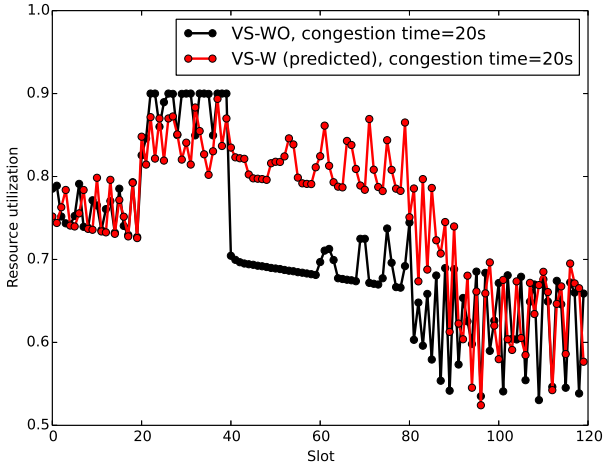


Fig. 9: Resource utilization with regard to the number of slots.

resources of V_2 is set as in Table VI. It can be seen from Fig. 7 that VS-W outperforms VS-WO in different congestion intervals. Furthermore, the goodput ratio of VS-W is not sensitive to the congestion durations. The performance gap between VS-W and VS-WO systems increases with the congestion time. We also compare the performance of VS-W systems with the predicted traffic load and with the real traffic load, which are close to each other as expected.

Throughput and resource utilization: To validate the effectiveness of the proposed enhanced transmission functionality, we compare the throughput of VS-W and VS-WO systems at each time slot during the network operation, as shown in Fig. 8. The action selection module of VS-W system utilizes the predicted traffic load information in action selection. The congestion event occurs at V_2 from $20\,\mathrm{s}$ to $40\,\mathrm{s}$. Before the network congestion, the throughputs of the VS-W and VS-WO systems are close to each other, since they depend only on the video traffic load. During the network congestion, the throughput of two VoD streaming systems is also at the same level. The network congestion is mitigated after the 40-th time slot and the ingress node of VS-W starts to send cached video packets to the corresponding video clients by enhanced transmission functionality. Therefore, the throughput of VS-W is higher than that of VS-WO from the 41-th time slot. All the cached video packets are transmitted before the 91-th time slot. As expected, the throughput of VS-W returns to the same level of VS-WO from the 91-th time slot to the end of the simulation. Fig. 9 shows the resource utilization for the two VoD streaming systems. Thanks to the enhanced transmission functionality, the resource utilization of VS-W is higher than that of VS-WO from the 41-th time slot to the 90-th time slot.

V. Conclusion

In this paper, we have proposed an transmission protocol customized for VoD streaming services (SDP-VS) in a Cybertwin-enabled next generation core network, where in-network congestion control and throughput enhancement functionalities are developed to realized a fast reaction to

network dynamics. To balance the tradeoff between congestion control and QoS provisioning, an MAB problem is formulated to maximize the overall network performance over time by triggering proper control actions under different network conditions, with the consideration of predicted video traffic load information and E2E available resources. The formulated problem is solved by the LinUCB algorithm to obtain the action selection strategy at each time slot. Simulation results are presented to demonstrate the advantages of the proposed SDP-VS protocol. For further research, we intend to develop customized transmission protocols for various service types co-existing.

REFERENCES

[1] L. U. Khan, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "6G wireless systems: A vision, architectural elements, and future directions," *IEEE Access*, vol. 8, pp. 147 029–147 044, 2020.

[2] Q. Yu, J. Ren, H. Zhou, and W. Zhang, "A cybertwin based network architecture for 6G," in *Proc. IEEE 6G SUMMIT*, Mar. 2020, pp. 1–5.

[3] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, 2018.

[4] M. R. Sama, L. M. Contreras, J. Kaippallimalil, I. Akiyoshi, H. Qian, and H. Ni, "Software-defined control of the virtualized mobile packet core," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 107–115, 2015.

[5] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, and J. Rao, "Delay-aware flow migration for embedded services in 5G core networks," in *Proc. IEEE ICC*, Shanghai, China, May 2019, pp. 1–6.

[6] O. Alhussein, P. T. Do, Q. Ye, J. Li, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao, "A virtual network customization framework for multicast services in NFV-enabled core networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1025–1039, 2020.

[7] K. Qu, W. Zhuang, Q. Ye, X. Shen, X. Li, and J. Rao, "Dynamic flow migration for embedded services in SDN/NFV-enabled 5G core networks," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2394–2408, 2020.

[8] B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "Analytic models for the latency and steady-state throughput of TCP Tahoe, Reno, and SACK," *IEEE/ACM Trans. Netw.*, vol. 11, no. 6, pp. 959–971, 2003.

[9] N. Parvez, A. Mahanti, and C. Williamson, "An analytic throughput model for TCP NewReno," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 448–461, 2009.

[10] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3584–3608, 2019.

[11] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004, pp. 2514–2524.

[12] S. Ha, I. Rhee, and L. Xu, "CUBIC: a new TCP-friendly high-speed TCP variant," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 5, pp. 64–74, 2008.

[13] Q. Yu, J. Ren, Y. Fu, Y. Li, and W. Zhang, "Cybertwin: An origin of next generation network architecture," *IEEE Wireless Commun.*, vol. 26, no. 6, pp. 111–117, 2019.

[14] H. T. Nguyen, J. Mary, and P. Preux, "Cold-start problems in recommendation systems via contextual-bandit algorithms," *arXiv preprint arXiv:1405.7544*, 2014.

[15] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. ACM WWW*, Raleigh, USA, Apr. 2010, pp. 661–670.

[16] M. Moradi, W. Wu, L. E. Li, and Z. M. Mao, "SoftMoW: Recursive and reconfigurable cellular WAN architecture," in *Proc. ACM CONEXT*, Sydney, Australia, Dec. 2014, pp. 377–390.

[17] J. Chen, Q. Ye, W. Quan, S. Yan, P. T. Do, W. Zhuang, X. Shen, X. Li, and J. Rao, "SDATP: An SDN-based adaptive transmission protocol for time-critical services," *IEEE Netw.*, vol. 34, no. 3, pp. 154–162, 2020.

[18] P. Megyesi, A. Botta, G. Aceto, A. Pescapè, and S. Molnár, "Available bandwidth measurement in software defined networks," in *Proc. ACM SAC*, Pisa, Italy, Apr. 2016, pp. 651–657.

[19] Z.-L. Zhang, Y. Wang, D. H.-C. Du, and D. Su, "Video staging: A proxy-server-based approach to end-to-end video delivery over wide-area networks," *IEEE/ACM Trans. Netw.*, vol. 8, no. 4, pp. 429–442, 2000.

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, 2007.

[21] T. Stockhammer, "Dynamic adaptive streaming over HTTP: standards and design principles," in *Proc. ACM MMSYS*, San Jose, USA, Feb. 2011, pp. 133–144.

[22] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," in *Proc. ACM SIGCOMM*, Chicago, USA, Aug. 2014, pp. 187–198.

[23] S. Yan, P. Yang, Q. Ye, W. Zhuang, X. Shen, X. Li, and J. Rao, "Transmission protocol customization for network slicing: A case study of video streaming," *IEEE Veh. Technol. Mag.*, vol. 14, no. 4, pp. 20–28, 2019.

[24] C. W. Chen, P. Chatzimisios, T. Dagiuklas, and L. Atzori, *Multimedia quality of experience (QoE): current status and future requirements*. John Wiley & Sons, 2015.

[25] P. J. Brockwell and R. A. Davis, *Introduction to time series and forecasting*. Springer, 2016.

[26] A. Azzouni and G. Pujolle, "NeuTM: A neural network-based framework for traffic matrix prediction in SDN," in *Proc. IEEE/IFIP NOMS*, Taipei, Taiwan, Apr. 2018, pp. 1–5.

[27] A. Pal and P. Prakash, *Practical Time Series Analysis: Master Time Series Data Processing, Visualization, and Modeling using Python*. Packt Publishing Ltd, 2017.

[28] S. García, J. Cabrera, and N. García, "Quality-control algorithm for adaptive streaming services over wireless channels," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 1, pp. 50–59, 2014.

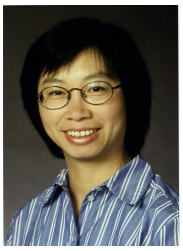
[29] M. Rahmati and D. Pompili, "UW-SVC: Scalable video coding transmission for in-network underwater imagery analysis," in *Proc. IEEE MASS*, Monterey, USA, Nov. 2019, pp. 380–388.

[30] O. Alhussein and W. Zhuang, "Robust online composition, routing and NF placement for NFV-enabled services," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1089–1101, 2020.

[31] J. M. Weiming, *Mastering Python for Finance*. Packt Publishing Ltd, 2015.

Si Yan received the B.S. degree in electrical and computer engineering from Tianjin University, Tianjin, China, in 2013, the M.E. degree in electrical and computer engineering from the University of Calgary, Calgary, Canada, in 2016, and the M.A.Sc. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2020. His research interests include data center networking, protocol design, and programmable data plane.

Qiang Ye (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016. From December 2016 to September 2019, he was a Postdoctoral Fellow and a Research Associate with the Department of Electrical and Computer Engineering, University of Waterloo. Since September 2019, he has been an Assistant Professor with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN, USA. His research interests include network slicing for 5G networks, edge intelligence for autonomous vehicular networks, artificial intelligence for future networking, protocol design, and performance analysis for the Internet of Things. He was a Technical Program Committee (TPC) Members for several international conferences, including the IEEE GLOBECOM'20, VTC'17, VTC'20, and ICPADS'20. He is the Editors of the *International Journal of Distributed Sensor Networks* (SAGE Publishing) and *Wireless Networks* (SpringerNature), and an Area Editor of the *Encyclopedia of Wireless Networks* (SpringerNature).



Weihua Zhuang (Fellow, IEEE) received the B.Sc. and M.Sc. degrees from Dalian Maritime University, China, and the Ph.D. degree from the University of New Brunswick, Canada, all in electrical engineering. She has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, since 1993, where she is a Professor and a Tier I Canada Research Chair in Wireless Communication Networks. Dr. Zhuang was a recipient of the 2021 R.A. Fessenden Award from the IEEE Canada, 2017 Technical Recognition Award in Ad Hoc and Sensor Networks from the IEEE Communications Society, and a co-recipient of several Best Paper Awards from IEEE conferences. She was the Editor-in-Chief of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2013, Technical Program Chair/Co-Chair of IEEE VTC 2017/2016 Fall, and Technical Program Symposia Chair of IEEE Globecom 2011. She is an elected member of the Board of Governors and Vice President for Publications of the IEEE Vehicular Technology Society. She was an IEEE Communications Society Distinguished Lecturer from 2008 to 2011. Dr. Zhuang is a Fellow of the Royal Society of Canada, Canadian Academy of Engineering, and Engineering Institute of Canada.