Digital Twin Based User-Centric Resource Management for Multicast Short Video Streaming

Xinyu Huang, *Student Member, IEEE*, Wen Wu, *Senior Member, IEEE*, Shisheng Hu, *Student Member, IEEE*, Mushu Li, *Member, IEEE*, Conghao Zhou, *Member, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Multicast short video streaming (MSVS) can effectively reduce network traffic load by delivering identical video sequences to multiple users simultaneously. The existing MSVS schemes mainly rely on the aggregated video requests to reserve bandwidth and computing resources, which cannot satisfy users' diverse and dynamic service requirements, particularly when users' swipe behaviors exhibit spatiotemporal fluctuation. In this paper, we propose a user-centric resource management scheme based on the digital twin (DT) technique, which aims to enhance user satisfaction as well as reduce resource consumption. Firstly, we design a user DT (UDT)-assisted resource reservation framework. Specifically, UDTs are constructed for individual users, which store users' historical data for updating multicast groups and abstracting useful information. The swipe probability distributions and recommended video lists are abstracted from UDTs to predict bandwidth and computing resource demands. Parameterized sigmoid functions are leveraged to characterize multicast groups' user satisfaction. Secondly, we formulate a joint non-convex bandwidth and computing resource reservation problem which is transformed into a convex piecewise problem by utilizing a tangent function to approximately substitute the concave part. A low-complexity scheduling algorithm is then developed to find the optimal resource reservation decisions. Simulation results based on the real-world dataset demonstrate that the proposed scheme outperforms benchmark schemes in terms of user satisfaction and resource consumption.**

***Index Terms*—Digital twin, resource management, multicast transmission, user-centric.**

I. Introduction

The proliferation of short video platforms, such as TikTok, Instagram Reels, and YouTube Shorts, enabled by the ubiquity of smartphones and high-speed wireless networks, has ushered in a new era of digital entertainment [2], [3]. According to a recent report from TikTok, the number of active global users has risen from 902 million to 1.47 billion in 2022 and will continue to maintain significant growth [4]. However, this short video streaming surge puts a significant burden on the existing wireless network infrastructures. Particularly, the ultra-definition and panoramic short videos poses new requirements on the transmission and computing capabilities of wireless networks, especially on higher bandwidth (300 Mbps) and transcoding speed (30.2 FPS), for providing immersive user experience [5], [6]. Multicast transmission, as an important technology in wireless networks, allows a single data stream to be disseminated to numerous users in a group simultaneously. By applying multicast transmission to short video streaming, the bandwidth utilization and network throughput can be effectively enhanced [7], [8].

To support multicast short video streaming (MSVS), network resources, including bandwidth and computing resources, are required. Specifically, spectrum bandwidths are required for video delivery to each multicast group [9], and computing resources are required for video transcoding for each multicast group [10]. For consistent quality of service (QoS) provisioning, such network resources need to be reserved in advance [11], [12]. Nevertheless, existing resource reservation schemes are mainly based on the aggregated video requests while neglecting the unique users' behaviors in watching short videos, i.e., *swipe behaviors*. Without considering the swiping behaviors that can make videos not completely played by users, the resource demands can be overestimated and the reserved resources can be underutilized. As the swipe behaviors are user-specific [13], resource reservation can incorporate the user-specific swipe behaviors, i.e., *user-centric resource reservation*, is desired for efficient resource utilization in the MSVS.

Digital twin (DT) is a potential technology to realize the user-centric resource reservation. DT is defined as a full digital representation of a physical object, and real-time synchronization between the physical object and its corresponding digital replica [14]. To update current end users' statuses, such as network conditions, data traffic, and mobility trajectories, *user DTs* (UDTs) are constructed for individual users. The data stored in UDTs can be leveraged to analyze users' transmission rates and behavior patterns for resource reservation. In the MSVS scenario, UDTs can store users' historical networking and personal information, and analyze their video traffic patterns and watching behaviors. Based on the distilled coarse- and fine-grained user information from UDTs, users' bandwidth and computing resource demands can be accurately predicted to facilitate the effective user-centric resource reservation.

Designing an effective UDT-based resource reservation scheme needs to address the following challenges: 1) incorporating swipe behaviors in resource reservation decision-making; 2) establishing the mathematical model between multicast groups' user satisfaction and reserved resources. Specifically, users' swipe behaviors are stochastic and spatiotemporally varied, which are difficult to be predicted in real time.

Xinyu Huang, Shisheng Hu, Conghao Zhou, and Xuemin (Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: {357huan, s97hu, c89zhou, sshen}@uwaterloo.ca);

Wen Wu is with the Frontier Research Center, Peng Cheng Laboratory, Shenzhen 518055, China (e-mail: wuw02@pcl.ac.cn);

Mushu Li is with the Department of Electrical, Computer and Biomedical Engineering, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada (e-mail: mushu.li@ieee.org).

Therefore, how to conduct effective data abstraction to obtain the distilled swipe feature and utilize it to predict bandwidth and computing resource demands is challenging. Furthermore, due to the dynamics of users' personalized preferences and sensitivities to service quality, the same amount of reserved resources to one multicast group can lead to different user satisfaction at different time. Therefore, how to establish and update each multicast group's user satisfaction model in each resource reservation window is challenging.

In this paper, we *firstly* propose a novel UDT-assisted resource reservation framework to incorporate the impact of swipe behaviors. Specifically, we construct UDTs to store users' historical data, including channel conditions, locations, swipe timestamps, and preferences. A novel user clustering algorithm is proposed to analyze UDTs' data for multicast group construction. The proposed user clustering algorithm consists of three parts, i.e., autoencoders, a double deep Q-network (DDQN), and the K-means++ method, which are responsible for UDTs' data compression, clustering number determination, and fast user clustering, respectively. Then, the group-level information, i.e., the swipe probability distribution and recommended video list, for each multicast group is abstracted. Based on that, multicast groups' average engagement time, video traffic, and computing consumption can be analyzed to predict bandwidth and computing resource demands for resource reservation. *Secondly*, we establish a user satisfaction model to quantify the impact of bandwidth and computing resource reservation on user satisfaction. Specifically, we adopt a real-world user satisfaction dataset to explore the relationship between user satisfaction and reserved resources. Based on data fitting, we build two kinds of parameterized sigmoid functions to characterize the exponential relationship. *Thirdly*, our objective is to maximize the system utility consisting of user satisfaction and resource consumption in each resource reservation window. Since the formulated problem is non-convex and difficult to solve, a low-complexity resource scheduling algorithm based on the linear approximation method to make the optimal resource reservation decisions is proposed. The extensive simulation results on real-world short video streaming datasets show that the proposed scheme can effectively improve system utility as compared with the state-of-the-art resource reservation schemes.

The main contributions are summarized as follows:

- We propose a novel UDT-assisted resource reservation framework for resource demand prediction;
- We establish a user satisfaction model to quantify the impact of reserved resources on user satisfaction;
- We propose a low-complexity scheduling algorithm to determine bandwidth and computing resource reservation.

The remainder of this paper is organized as follows. Related works are presented in Section II. The considered scenario and the UDT-assisted resource reservation scheme are presented in Sections III and IV, respectively. The user satisfaction, the problem formulation, and the proposed scheduling algorithm are presented in Sections V, VI and VII, respectively. Simulation results are provided in Section VIII. Finally, Section IX concludes this paper.

II. Related Work

MSVS can distribute video sequences from a single base station (BS) to multiple users concurrently over the same wireless channels. This approach exhibits two primary advantages, i.e., efficient bandwidth utilization and scalable user scale [15], [16]. To facilitate the MSVS within 5G networks, extensive works are devoted to optimizing multicast transmission performance from different directions, such as the novel architecture design by integrating network function virtualization and mobile edge computing technique [17], transmission orchestration by leveraging collected global network information [18], [19], and signal multiplexing with non-orthogonal multiple access to transmit different segment layers [20]. These studies demonstrate innovative approaches toward enhancing the performance of MSVS.

To ensure the service quality for the MSVS, resource reservation for multicast groups is essential. Historical video contents and bitrates are usually selected to forecast users' future peak video traffic, which can provide a primary basis for bandwidth resource reservation [21]. Additionally, bitrate fluctuation and video quality are usually leveraged to predict transcoding consumption, which can serve as a fundamental foundation for VM instance reservation [22]. To flexibly adjust resource reservation in the high-mobility and ultra-density network scenario, existing hierarchical resource reservation schemes usually predict video traffic in grid-partitioned regions [23], estimate basic resources per region based on QoS requirements [24], and employ machine learning methods to fine-tune resource reservation [25]. However, these schemes do not incorporate the dynamics of user behavior, such as swipe probability, on-off patterns, etc., potentially leading to inaccurate user clustering. Furthermore, solely relying on aggregated video requests to reflect regional traffic patterns is not enough to support accurate resource demand prediction. In this case, reserved resources can occur underutilization.

DT as an important virtualization technology was first introduced to monitor and mitigate anomalous events for flying vehicles [26]. By introducing DTs into the next-generation wireless network, the existing network architecture can realize holistic network virtualization for flexible and accurate resource management. We refer readers to recent comprehensive surveys and tutorials on DTs to be familiar with this topic [27]–[29]. There also exist some technical papers aiming at utilizing DTs to improve network performance. Sun *et al*. and Huynh *et al*. utilized DTs to estimate edge servers' states and the entire MEC system, which can provide seamless and accurate training data for offloading decisions [30] [31]. To coordinate the computing resource management of edge servers in real time, DTs were used to timely monitor the centralized training process for resource scheduling in the aerial edge networks [32]. Furthermore, DTs were used to capture the time-varying demand of computing resources of IoT devices to assist the computation offloading decisions in [33]. A learning-based prediction model residing in the DT was developed to predict the waiting time of relays and transmit the predicted results in [34], which can efficiently synchronize real-time data. Lu *et al*. formulated an edge

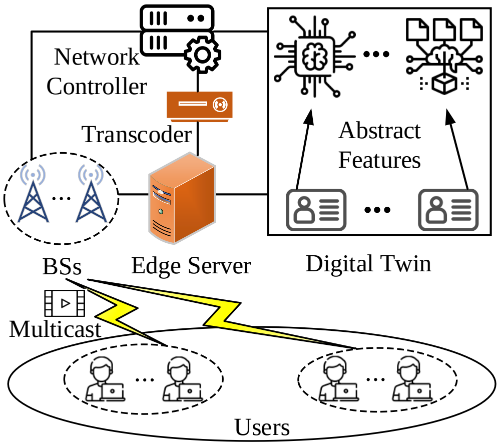


Fig. 1. UDT-assisted resource reservation framework for MSVS.

association problem for adaptively placing and migrating DTs based on the dynamics of network states and end users, which can efficiently reduce the service latency [35]. These pioneering works can effectively improve resource management performance in terms of resource allocation, edge association, and task offloading.

Recently, a few early research works focused on utilizing DTs to enhance network performance in terms of resource reservation. Zhou *et al*. constructed UDTs for individual users to analyze their mobility patterns and proposed an improved particle swarm optimization method to enable customized resource reservation [36]. To alleviate data sparsity when performing the DT-assisted traffic prediction, a short-term traffic flow and velocity prediction method based on data similarity was proposed in [37]. Furthermore, since it is difficult to operate DTs on different granular levels in a large and heterogeneous user scale, Vaezi *et al*. proposed a five-stage implementation framework to progressively abstract component-level, subsystem-level, and global-level information [38]. As an extension of our previous work [1], we further propose a user satisfaction model to quantify the impact of reserved resources, taking the dynamics of users' personalized preferences and sensitivities to service quality. A fast resource reservation algorithm is also proposed to determine bandwidth and VM instance reservation.

III. Considered Scenario

As shown in Fig. 1, we consider a UDT-assisted MSVS scenario, which consists of multiple BSs, an edge server (ES), users, and UDTs.

- BSs: BSs utilize multicast technology to transmit short videos to different multicast groups. The set of multicast groups is denoted by $\mathcal{G} = \{1, \cdots, G\}$. The set of bandwidths for all BSs is denoted by $\mathcal{M} = \{1, \cdots, M\}$.
- ES: The ES connects to all BSs and stores popular short videos with the highest version (bitrate) to avoid frequent video retrievals from content providers. The stored short videos can be transcoded to a lower version in the transcoder to adapt to users' dynamic channel conditions and swipe behaviors. The set of VM instances for the transcoders is denoted by $\mathcal{N} = \{1, \cdots, N\}$, and the computing capability of each VM instance is denoted by ω.
- Users: Users with similar statuses, such as preferences, swipe behaviors, locations, and channel conditions, are clustered into one multicast group, and the corresponding user set is denoted by $\mathcal{K}_g$=$\{1, \cdots, K_g\}$. Users in the same multicast group are recommended the same short video list and receive them by the multicast transmission.
- UDTs: There are multiple UDTs deployed at the ES. Each UDT corresponds to a user and stores users' statuses. BSs are responsible for collecting real-time users' statuses to update UDTs. UDTs' data are utilized to abstract some useful information, such as data similarity and swipe probability distribution, which can help the network controller realize an accurate multicast group construction and resource demand prediction.

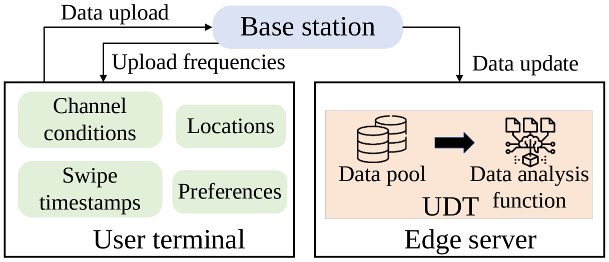


Fig. 2. UDT data collection procedure.

The UDT-assisted resource reservation framework for MSVS operates as follows. In each resource reservation window, user statuses are uploaded to the ES to update corresponding UDTs and adjust multicast groups. In each multicast group, UDTs' swipe timestamps and preferences are used to abstract the swipe probability distribution and recommended video list for accurate bandwidth and computing resource demand prediction. Based on the predicted information, the network controller reserves appropriate bandwidths and VM instances for each multicast group.

IV. UDT-Assisted Resource Reservation Scheme

A. UDT Construction

UDT consists of a finite data pool and a data analysis function as shown in Fig. 2. The data pool records a user's recent status through diversified data collection frequencies. The data analysis function investigates a user's swipe timestamps to obtain a swipe probability distribution for each video type. The construction of a UDT can be summarized into two phases.

1) Data Collection: As shown in Fig. 2, BS collects a user's data from two aspects, i.e., networking-related data and behavior-related data. The networking-related data include a user's real-time channel conditions and locations, which are utilized to estimate the transmission rate. The behavior-related data consist of a user's swipe timestamps and preferences. The swipe timestamps reflect a user's swipe behavior [39], which are used to abstract swipe probability distribution for engagement time prediction. The preferences are leveraged for the recommended video list update. By integrating these two kinds of data, UDTs can well reflect users' real-time statuses and predict bandwidth and computing resource demands.

Data collection number can be different for various UDT data attributes in each resource reservation window. Given that a low-speed mobility scenario where users' channel conditions usually vary on a small timescale [40], the data collection number of a user's channel condition and location in one resource reservation window, T, is denoted by F_1. Since the swipe probability needs to be calculated over several short videos, the corresponding data collection number of a user's swipe timestamps in one resource reservation window is denoted by F_2. A user's preference is an essential metric to determine which videos should be recommended. Since the user's preference is relatively dynamic data that is updated based on the user's like, share, and swipe frequency [41], the data collection number keeps consistent with that of the swipe timestamps to ensure the data freshness.

2) Data Analysis: A UDT's functionality consists of two parts, i.e., a user's swipe behavior analysis module and a data interaction module. The former analyzes the collected swipe timestamps to abstract swipe probability distributions. The latter provides stored data and the abstracted swipe probability distributions to the network controller to update multicast groups and predict resource demands. In resource reservation window t, user k's swipe probability distribution, $p_{k,c}^{t}\left(e\right)$, for video type c is updated by

$$p_{k,c}^{t}\left(e\right)=\lambda p_{k,c}^{t-1}\left(e\right)+\left(1-\lambda\right)\frac{A_{k,c}^{t-1}\left(e\right)}{\widehat{A}_{k,c}^{t-1}}, \tag{1}$$

where e is the video timestamp ranging from 1 to 15. Parameter λ is a weighting factor ranging from 0 to 1. The higher λ means a smaller impact of previous swipe behaviors on user k's current swipe probability distribution. Here, $A_{k,c}^{t-1}\left(e\right)$ represents user k's swipe numbers for video type c in previous window $t-1$, which is counted by swipe timestamps. In addition, $\widehat{A}_{k,c}^{t-1}$ indicates the number of transmitted videos of type c to user k in previous window $t-1$.

B. UDT-Assisted Multicast Group

Based on stored data in UDTs, we can analyze users' similarities to divide users with similar statuses into the same multicast group. A three-step method is proposed to realize fast and accurate multicast group construction. Specifically, autoencoders are first employed to compress time-series UDTs' data for dimension reduction. Then, a deep reinforcement learning (DRL) network is adopted to determine the clustering number by mining users' intrinsic correlation. Finally, the K-means++ algorithm [42] is utilized to realize a fast multicast group construction based on the determined clustering number [43]. After multicast groups are constructed, we can abstract the swipe probability distribution and recommended video list for each multicast group.

1) Clustering Number Determination: UDTs' data consist of four dimensions and each dimension corresponds to time-series data, directly using a DRL method to analyze UDTs' data may suffer from the curse of input dimension. Therefore, we add four Autoencoders to the existing DDQN [44] to compress UDTs' data, which can effectively reduce the input dimension [45]. The compressed data are further input to the actor-critic network for determining an appropriate clustering number, as shown in Fig. 3.

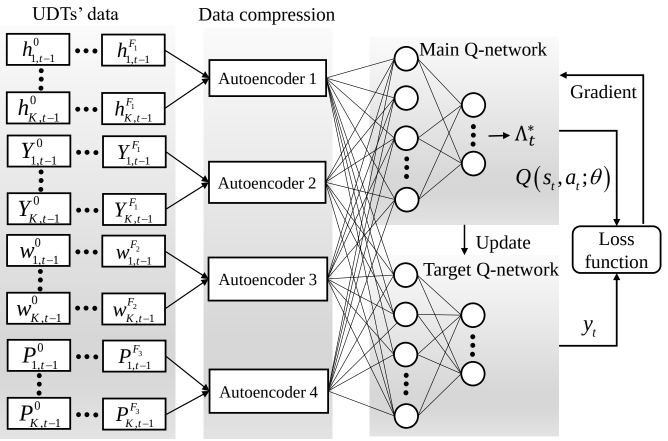


Fig. 3. The improved DDQN architecture for clustering number determination.

As shown in Fig. 3, the processed UDTs' data consist of users' historical channel conditions, $\left\{h_{k,t-1}^{f}\right\}_{k\in\mathcal{K},f\in\mathcal{F}_1}$, locations, $\left\{Y_{k,t-1}^{f}\right\}_{k\in\mathcal{K},f\in\mathcal{F}_1}$, swipe timestamps, $\left\{w_{k,t-1}^{f}\right\}_{k\in\mathcal{K},f\in\mathcal{F}_2}$, and preferences, $\left\{P_{k,t-1}^{f}\right\}_{k\in\mathcal{K},f\in\mathcal{F}_2}$, in window $t-1$. Since each UDT data attribute has a temporal correlation, we utilize Autoencoders to process these data to reduce input dimension and abstract temporal features. The processed UDTs' data are further input into the DDQN, consisting of one target and one main Q-network, for model training to obtain the appropriate clustering number Λ_t^*. Specifically, the main Q-network with network parameter θ takes the current processed UDTs' data as input and outputs a set of Q-values. The target Q-network with network parameter θ' is a copy of the main Q-network that is periodically updated to match the parameters of the main Q-network. The target value, y_t, of DDQN is given by

$$y_t=r_{t+1}+\gamma Q\left(s_{t+1},\arg\max Q\left(s_{t+1},a;\theta\right);\theta'\right), \tag{2}$$

where main Q-network $Q\left(s,a;\theta\right)$ is utilized for the clustering number determination and target Q-network $Q\left(s,a;\theta'\right)$ is used for the value evaluation. Here, s and a represent processed UDTs' data and the clustering number, respectively. Since the objective of clustering is to reserve appropriate resources for different multicast groups to enhance system utility, we select the system utility that will be introduced in Section VI-A as reward r. During the training process, the main Q-network is updated to minimize the difference between the predicted and true Q-values. The target Q-network is periodically updated to match the parameters of main Q-network to help stabilize the training process.

2) User Clustering: When the clustering number of multicast groups is determined, we can classify users into different multicast groups based on UDTs' data similarities. We adopt the K-means++ algorithm to complete the user clustering process. Specifically, the Euclidean distance, $D\left(i,j\right)$, between UDTs i and j is first calculated as follows:

$$D\left(i,j\right) = \left\|h_i - h_j\right\|_2 + \left\|Y_i - Y_j\right\|_2 + \left\|w_i - w_j\right\|_2 + \left\|P_i - P_j\right\|_2. \tag{3}$$

Then, the Euclidean distance between the sampling UDT and the current clustering center is used to calculate the selection probability. Since the longer distance causes a higher selection probability as the new clustering center, the selection probability, $\Theta(i)$, of UDT i is depicted as

$$\Theta(i) = \frac{D(i,\tau)^2}{\sum_j D(j,\tau)^2}, \tag{4}$$

where τ denotes the current clustering center. After several iterations, users with similar statuses are clustered into one multicast group. The set of constructed multicast group is denoted by $\Omega_t = \{1, \cdots, \Lambda_t^*\}$[1].

3) Abstracted Information: After constructing multicast groups, we need to abstract some useful information from each of them to predict resource demands in each resource reservation window. Recommended video lists and corresponding swipe probability distributions can well achieve this requirement. To obtain a good recommended video list, a good video recommendation mechanism not only needs to consider video popularity distributions but also users' preferences [41]. The former can be directly obtained by analyzing view counts and engagement time from content providers. In window $t-1$, the popular video set is denoted by $\mathcal{V} = \{1, \cdots, V\}$ and the corresponding popularity distribution is $\{E_v\}_{v\in\mathcal{V}}$. The latter is time-series data and exists a relatively strong temporal correlation in each multicast group. The collected preference, $P^f_{k,t-1}$, in UDT k is a $1 \times C$ matrix, i.e., $\left\{P^{f,1}_{k,t-1}, \cdots, P^{f,C}_{k,t-1}\right\}$, reflecting user k's preferences of C video types. To accurately estimate multicast group g's preference matrix, $\left\{\widehat{P}^c_{t,g}\right\}_{c\in\mathcal{C}}$, in current window t, we first calculate the users' average preference on each kind of video in previous window $t-1$, and then integrate it with the discounted preference, $\widetilde{\lambda}\widehat{P}^c_{t-1,g}$, in previous window $t-1$, as follows:

$$\widehat{P}^c_{t,g} = \widetilde{\lambda}\widehat{P}^c_{t-1,g} + \frac{1}{K_g F_2}\sum_{k=1}^{K_g}\sum_{f=1}^{F_2} P^{f,c}_{k,t-1}, \forall c \in \mathcal{C}, \tag{5}$$

where $\widetilde{\lambda}$ is a parameter ranging from 0 to 1. The recommended ranking, $\Re_{t,g,v}$, of video v for multicast group g in window t combines the video popularity distribution and preference matrix, which is depicted as $\Re_{t,g,v} = E_v \sum_{c\in\mathcal{C}} I_c(v)\widehat{P}^c_{t,g}$. Here, $I_c(v)$ is an indicator function, representing if video v belongs to type c, $I_c(v) = 1$; otherwise, $I_c(v) = 0$.

In each window, the recommended video list, τ_g, for multicast group g is constituted based on the recommended rankings in popular video set $\mathcal{V}$, and the corresponding list length is denoted by ρ. In addition, each video in the recommend list has a unique swipe probability distribution, $p^v_{g,t}\left(e\right)$, which can be abstracted by accumulating users' swipe probability distributions in a multicast group, as follows:

$$p^v_{g,t}\left(e\right) = \sum_{k\in\mathcal{K}_g}\sum_{c\in\mathcal{C}} I_c(v) p^t_{k,c}\left(e\right). \tag{6}$$

C. UDT-Assisted Resource Demand Prediction

Based on the abstracted grouping information, i.e., swipe probability distributions and recommended video lists, each multicast group's bandwidth and computing resource demands can be predicted to help the network controller to facilitate an efficient resource reservation scheme.

1) Bandwidth Resource Demand: Since users' swipe behaviors can cause part of videos not to be played and thus lead to a waste of bandwidth resources, we analyze users' swipe probability distributions on the recommended video list to make an accurate bandwidth resource demand prediction.

First, the average engagement time, W^t_g, of multicast group g at window t is estimated based on the abstracted swipe probability distribution, which is expressed as

$$W^t_g = \sum\nolimits_{v\in\tau_g}\int_0^{\wp(v)} \left(1 - p^v_{g,t}(e)\right) e de. \tag{7}$$

Second, we need to predict the video traffic of each multicast group. Since one video can be transmitted to all users by multicast transmission in a multicast group, the video traffic of each user should not be accumulated. In addition, since the multicast video version adaptively changes with users' dynamic channel conditions and buffer lengths, we select the average multicast video version, $\bar{l}$, in previous window $t-1$, to approximate the multicast video version in current window t. Based on the above analysis, the video traffic, Y^t_g, of multicast group g at window t is predicted by

$$Y^t_g = \sum\nolimits_{v\in\tau_g}\int_0^{\wp(v)} \left(1 - p^v_{g,t}(e)\right)\sum\nolimits_{l=1}^{\bar{l}} z^l_v\left(e\right) de, \tag{8}$$

where $z^l_v\left(e\right)$ is the file size of segment layer l of video v at video timestamp e.

Based on the estimated average engagement time and video traffic, the bandwidth resource demand, R^t_g, is given by

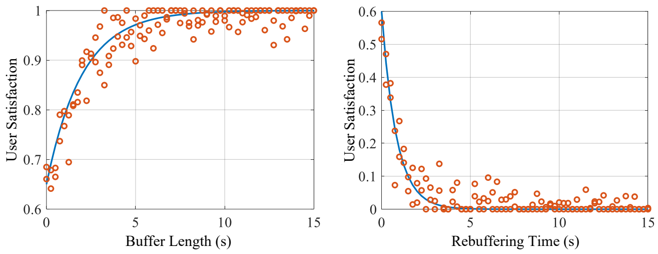
$$R^t_g = W^t_g / Y^t_g. \tag{9}$$

2) Computing Resource Demand: Since the caching capacity of an edge server is limited, we only cache the basement layer of each recommended segment in the edge server to guarantee users' basic watching requirements. Enhancement layers can be obtained by transcoding and then jointly transmitted with the basic layer to users to enhance the video quality. In this process, the computing consumption is predicted by

$$Z^t_g = \mu\left(Y^t_g - \sum\nolimits_{v\in\tau_g}\int_0^{\wp(v)} \left(1 - p^v_{g,t}(e)\right) z^0_v\left(e\right) de\right), \tag{10}$$

where μ is the computing density for video transcoding, and $z^0_v\left(e\right)$ is the file size of the basement layer of video v at video timestamp e.

[1] The computational complexity of the K-means++ algorithm is $O\left((\Lambda_t^*)^2 K + \pi\Lambda_t^* K\right)$, where π and K denote the number of iterations and users, respectively. The K-means++ algorithm has two benefits, i.e., low computational cost and fast convergence.



(a)

(b)

Fig. 4. User satisfaction over different buffer lengths and rebuffering time.

Based on the estimated average engagement time and computing consumption, the computing resource demand, O_g^t, is given by

$$O_g^t = Z_g^t / Y_g^t. \tag{11}$$

V. User Satisfaction

Since the system bandwidth and computing resources are limited, multicast groups' resource demands may not be always satisfied. In addition, users usually have different sensitive degrees of rebuffering time and video quality [46], which indicates that the same reserved resources for various multicast groups can lead to different user satisfactions. Therefore, we analyze the relationship between reserved bandwidth and computing resources and user satisfaction for each multicast group, which can help the network controller to make a better resource reservation scheme to improve user satisfaction.

First, we analyze how reserved bandwidths affect user satisfaction based on observed users' buffers. Specifically, if the downlink transmission rate estimated by reserved bandwidths is larger than the bandwidth resource demand, users' buffer lengths can increase; otherwise, users' buffers will gradually become empty and rebuffering time will increase. Therefore, the gap between the downlink transmission capability and the bandwidth resource demand is positively correlated with users' buffers. To measure how users' buffers can affect user satisfaction, we select a part of users' watching and rating records from video dataset[2], and use the fitting function[3] to form a fitted curve from scattered points, as shown in Fig. 4.

It can be observed that user satisfaction is positively and negatively exponential with buffer length and rebuffering time, respectively. Therefore, we can conclude that user satisfaction is exponentially with the reserved bandwidths based on the intermediate variable, i.e., user's buffer. A parameterized sigmoid function is employed to characterize the exponential relationship between bandwidth reservation and user satisfaction as follows:

$$U_{\mathrm{B},g}^t = \frac{1}{1+\exp\left\{-\xi_g^t\left(m_i^t B \log\left(1+\ell_g^t\right)-R_g^t\right)\right\}}, \tag{12}$$

where ℓ_g^t and ξ_g^t are the average signal-to-interference plus noise ratio and the sensitivity of bandwidth reservation for multicast group g in window t, respectively. Here, m_g^t and B are the reserved bandwidth number and size for multicast group g in window t, respectively. When the downlink transmission rate of reserved bandwidths exceeds the bandwidth resource demand, user satisfaction will quickly increase and gradually tend to the highest value, and vice versa.

Second, we analyze how reserved VM instances affect user satisfaction. Specifically, if the transcoding capability estimated by reserved VM instances exceeds the computing resource demand, the video transcoding process can be quickly completed to guarantee that users can watch videos with satisfied video quality; otherwise, users will watch videos with low video quality and experience frequent quality variation. Since the video quality also has a similar exponential relationship with user satisfaction [47], we also adopt a parameterized sigmoid function to characterize the exponential relationship between VM instance reservation and user satisfaction as follows:

$$U_{\mathrm{V},g}^t = \frac{1}{1+\exp\left\{-\vartheta_g^t\left(n_g^t \omega - O_g^t\right)\right\}}, \tag{13}$$

where ϑ_g^t and n_g^t represent the sensitivity of VM instance reservation and the number of reserved VM instances for multicast group g in window t, respectively. Here, ω is the computing capacity of one VM instance.

VI. Problem Formulation

A. System Utility

The system utility is defined to evaluate the performance of the UDT-assisted resource reservation scheme, including operation cost, reconfiguration cost, and user satisfaction.

1) Operation cost: The operation cost, U_{O}^t, refers to the cost of reserved bandwidths and VM instances for all multicast groups [48] in window t, which can be expressed as

$$U_{\mathrm{O}}^t = \varpi_1 \sum_{g=1}^{\Lambda_t^*} m_g^t + \varpi_2 \sum_{g=1}^{\Lambda_t^*} n_g^t, \tag{14}$$

where ϖ_1 and ϖ_2 represent the unit cost of using a bandwidth and a VM instance, respectively.

2) Reconfiguration Cost: In each resource reservation window, if the previous resource reservation configuration for a multicast group cannot satisfy the current dramatically increasing resource demands, the network controller needs to configure bandwidths and VM instances for the multicast group, which incurs reconfiguration cost. The resource release cost is omitted since this process can be quickly completed with negligible cost. Based on the above analysis, the reconfiguration cost, U_{R}^t, can be expressed as

$$U_{\mathrm{R}}^t = \sum_{g=1}^{\Lambda_t^*} \varpi_3 \left[m_g^t - m_g^{t-1}\right]^+ + \varpi_4 \left[n_g^t - n_g^{t-1}\right]^+, \tag{15}$$

where ϖ_3 and ϖ_4 denote the unit cost of bandwidth and VM instance reconfigurations, respectively. Since the clustering number in the previous resource reservation window may be less than that in the current window, a part of elements

[2] Video quality assessment: https://live.ece.utexas.edu/research/ LIVEStallStudy/liveMobile.html

[3] Mathworks: https://www.mathworks.com/help/optim/ug/lsqcurvefit.html

in vectors $\mathcal{M}^{t-1}$ and $\mathcal{N}^{t-1}$ is denoted by 0 to guarantee dimension match, which can be expressed as

$$m_g^{t-1}=0, n_g^{t-1}=0, \forall g \in \left[\Lambda_{t-1}^*+1, \Lambda_t^*\right]. \tag{16}$$

The objective of system utility is to achieve low resource operation and reconfiguration costs and high user satisfaction. With Eq. (12)-(15), the system utility is defined as

$$U^t=\frac{1}{\Lambda_t^*}\sum_{g=1}^{\Lambda_t^*}\left(U_{\mathrm{B},g}^t+\delta_1 U_{\mathrm{V},g}^t\right)-\delta_2 U_{\mathrm{O}}^t-\delta_3 U_{\mathrm{R}}^t, \tag{17}$$

where δ_1, δ_2, and δ_3 represent the weighted parameters to balance each system utility component.

B. Problem Formulation

Since multicast groups' bandwidth and computing resource demands are dynamic due to users' diversified swipe behaviors and channel conditions, it is paramount to reserve the appropriate bandwidth and computing resources for each multicast group to guarantee a satisfied service quality. In each resource reservation window, our objective is to maximize the system utility within the limited bandwidth and computing resources, and the corresponding optimization problem is formulated as

$$\begin{aligned}
\mathbf{P}_0:\ & \max_{\left\{m_g^t, n_g^t\right\}_{g\in\Omega_t}} \ U^t && (18)\\
\text{s.t.} & \sum\nolimits_{g=1}^{\Lambda_t^*} m_g^t \leq M, \forall t \in \mathcal{T}, && (18a)\\
& \sum\nolimits_{g=1}^{\Lambda_t^*} n_g^t \leq N, \forall t \in \mathcal{T}, && (18b)\\
& m_g^t \in Z^+, \forall g \in \Omega_t, t \in \mathcal{T}, && (18c)\\
& n_g^t \in Z^+, \forall g \in \Omega_t, t \in \mathcal{T}, && (18d)\\
& \text{and } (16).
\end{aligned}$$

Constraints (18a) and (18b) are resource capacity constraints, which guarantee that the total reserved bandwidths and VM instances cannot exceed the system capacity.

VII. Proposed Resource Reservation Algorithm

A. Problem Transformation

Problem $\mathbf{P}_0$ is a nonlinear integer programming problem. To solve this problem, we first transform it into two subproblems. Specifically, since optimization variable m_g^t is independent of optimization variable n_g^t, the problem can be decoupled into two independent minimization subproblems regarding bandwidth and VM instance reservation. The decoupled subproblems $\mathbf{P}_1$ and $\mathbf{P}_2$ are given by

$$\begin{aligned}
\mathbf{P}_1:\ & \min_{\left\{m_g^t\right\}_{g\in\Omega_t}} \frac{1}{\Lambda_t^*}\sum_{g\in\Omega_t} \delta_3\varpi_3\left[m_g^t-m_g^{t-1}\right]^+ + \delta_2\varpi_1 m_g^t\\
& + \frac{1}{-1-\exp\left\{\xi_g^t\left(m_g^t B\log\left(1+\ell_g^t\right)-R_g^t\right)\right\}} && (19)\\
\text{s.t.} & \sum_{g\in\Omega_t} m_g^t \leq M, \forall t\in\mathcal{T}, && (19a)\\
& m_g^t \in Z^+, \forall g\in\Omega_t, t\in\mathcal{T}, && (19b)\\
& m_g^{t-1}=0, \forall g\in\left[\Lambda_{t-1}^*+1, \Lambda_t^*\right], t\in\mathcal{T}, && (19c)
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}_2:\ & \min_{\left\{n_g^t\right\}_{g\in\Omega_t}} \frac{1}{\Lambda_t^*}\sum_{g\in\Omega_t} \delta_3\varpi_4\left[n_g^t-n_g^{t-1}\right]^+ + \delta_2\varpi_2 n_g^t\\
& + \frac{\delta_1}{-1-\exp\left\{\vartheta_g^t\left(n_g^t\omega-O_g^t\right)\right\}} && (20)\\
\text{s.t.} & \sum_{g\in\Omega_t} n_g^t \leq N, \forall t\in\mathcal{T}, && (20a)\\
& n_g^t \in Z^+, \forall g\in\Omega_t, t\in\mathcal{T}, && (20b)\\
& n_g^{t-1}=0, \forall g\in\left[\Lambda_{t-1}^*+1, \Lambda_t^*\right], t\in\mathcal{T}. && (20c)
\end{aligned}$$

Then, we perform continuous processing on variables m_g^t and n_g^t. If a continuous optimization problem is convex, the corresponding discrete optimization problem is also convex [49]. The objective function of subproblem $\mathbf{P}_1$ consists of two parts, i.e., one related to the exponential term and one related to the approximately linear term. The former is expressed by $\Psi_g^t=-1/(1+\exp\left\{-\xi_g^t\left(m_g^t B\log\left(1+\ell_g^t\right)-R_g^t\right)\right\})$, and its convexity is related to the range of independent variable m_g^t. Specifically, the second derivative of Ψ_g^t is shown in Eq. (21).

When $m_g^t \geq R_g^t/B\log\left(1+\ell_g^t\right)$, we can have $\partial^2\Psi_g^t/\partial^2 m_g^t \geq 0$ and function Ψ_g^t is convex; otherwise, function Ψ_g^t is concave. The latter is expressed by $\Gamma_g^t = \delta_3\varpi_3\left[m_g^t-m_g^{t-1}\right]^+ + \delta_2\varpi_1 m_g^t$, and it is convex due to the convexity of $[\cdot]^+$ function [50].

Since function Ψ_g^t is not always convex in the whole range of independent variable m_g^t, we make an approximation to the concave part of Ψ_g^t to transform it into a convex function. Specifically, when $m_g^t < R_g^t/B\log\left(1+\ell_g^t\right)$, we utilize a tangent to approximately substitute the concave part. The slope of tangent is the first derivative of Ψ_g^t when m_g^t equals to $R_g^t/B\log\left(1+\ell_g^t\right)$, which can be expressed by

$$\kappa_g^t=-\frac{1}{4}\xi_g^t B\log\left(1+\ell_g^t\right). \tag{22}$$

The intersection of the tangent and the horizontal axis is given by

$$b_g^t=\frac{\xi_g^t R_g^t-2}{\xi_g^t B\log\left(1+\ell_g^t\right)}. \tag{23}$$

To guarantee function Ψ_g^t is always negative, independent variable m_g^t needs to satisfy $m_g^t > b_g^t$. Based on this transformation, approximate function, $\widetilde{\Psi}_g^t$, can be expressed as a convex piecewise function, i.e.,

$$\widetilde{\Psi}_g^t=\begin{cases}\Psi_g^t,\ m_g^t \geq R_g^t/B\log\left(1+\ell_g^t\right),\\ \kappa_g^t m_g^t-0.5,\ b_g^t<m_g^t<R_g^t/B\log\left(1+\ell_g^t\right).\end{cases} \tag{24}$$

Since the addition of convex functions $\widetilde{\Psi}_g^t$ and Γ_g^t is still convex and constraints are linear, subproblem $\mathbf{P}_1$ is transformed into a convex optimization subproblem $\mathbf{P}_1'$, i.e.,

$$\begin{aligned}
\mathbf{P}_1':\ & \min_{\left\{m_g^t\right\}_{g\in\Omega_t}} \frac{1}{\Lambda_t^*}\sum_{g\in\Omega_t}\widetilde{\Psi}_g^t+\Gamma_g^t && (25)\\
\text{s.t.}\ & m_g^t > b_g^t, && (25a)\\
& (19a), (19b), \text{and } (19c).
\end{aligned}$$

Similarly, we can transform the exponential term, $\Xi_g^t.$, in the objective function of subproblem $\mathbf{P}_2$ into a convex piecewise function, which can be expressed as

$$\widetilde{\Xi}_g^t = \begin{cases} \Xi_g^t, \; n_g^t \geq O_g^t/\omega, \\ \widehat{\kappa}_g^t n_g^t - 0.5, \; \widehat{b}_g^t < n_g^t < O_g^t/\omega, \end{cases} \tag{26}$$

where $\widehat{\kappa}_g^t = -\frac{1}{4}\vartheta_g^t\omega$ and $\widehat{b}_g^t = \frac{\vartheta_g^t O_g^t - 2}{\vartheta_g^t \omega}$. The approximately linear term, $\widehat{\Gamma}_g^t$, in the objective function P2 is also convex due to the max function. Since the addition of convex functions $\widetilde{\Xi}_g^t$ and $\widehat{\Gamma}_g^t$ is still convex and constraints are linear, subproblem $\mathbf{P}_2$ is transformed into a convex optimization subproblem $\mathbf{P}_2'$, which can be expressed as

$$\mathbf{P}_2' : \min_{\left\{n_g^t\right\}_{g\in\Omega_t}} \frac{1}{\Lambda_t^*} \sum_{g\in\Omega_t} \widetilde{\Xi}_g^t + \widehat{\Gamma}_g^t \tag{27}$$

$$\text{s.t.} \quad n_g^t > \widehat{b}_g^t, \tag{27a}$$

$$(20a), (20b), \text{and } (20c).$$

Based on the linear approximation on the concave part, the original optimization problem $\mathbf{P}_0$ is transformed into a convex optimization problem.

B. Fast Resource Reservation Algorithm

Since the local optimal point is also the global optimal point for a convex optimization problem, we design a FS algorithm to find the local optimal bandwidth and VM instance reservation variables, i.e., $\left\{m_g^t\right\}_{g\in\Omega_t}$ and $\left\{n_g^t\right\}_{g\in\Omega_t}$. Specifically, each multicast group is first assigned $\left\lceil b_g^t \right\rceil$ bandwidths and $\left\lceil \widehat{b}_g^t \right\rceil$ VM instances. Then, in each iteration, the unassigned bandwidths and VM instances are sequentially assigned to the multicast group that can obtain the highest values of objective functions in subproblems $\mathbf{P}_1'$ and $\mathbf{P}_2'$, respectively. If the objective function value in the previous iteration is higher than that in the current iteration, the iteration process is terminated and local optimal resource reservation variables are the variables in the previous iteration. The specific algorithm is presented in Algorithm 1.

Algorithm 1: Fast Scheduling (FS)

```
Initialize the objective functions $U^t(\mathbf{P}_1')$ and $U^t(\mathbf{P}_2')$.
Input $\Omega_t$, $\{\ell_g^t\}_{g\in\Omega_t}$, $\{R_g^t\}_{g\in\Omega_t}$, $\{O_g^t\}_{g\in\Omega_t}$, $\lceil b_g^t \rceil$,
    $\lceil \widehat{b}_g^t \rceil$, $M$, $N$, $\Lambda_{t-1}^*$, and all weighted parameters.
Output $\{m_g^t\}_{g\in\Omega_t}$ and $\{n_g^t\}_{g\in\Omega_t}$.
for $g \in \Omega_t$ do
    Multicast group $g$ is assigned $\lceil b_g^t \rceil$ bandwidths and
      $\lceil \widehat{b}_g^t \rceil$ VM instances;
    Variables $m_g^t$ and $n_g^t$ take the values of $\lceil b_g^t \rceil$ and
      $\lceil \widehat{b}_g^t \rceil$, respectively;
end
Calculate the number of unassigned bandwidths and
    VM instances, i.e., $\widetilde{M} = M - \sum_{g\in\Omega_t} \lceil b_g^t \rceil$ and
    $\widetilde{N} = N - \sum_{g\in\Omega_t} \lceil \widehat{b}_g^t \rceil$;
for $m = 1 : \widetilde{M}$ do
    for $g \in \Omega_t$ do
        Assign one bandwidth to multicast group $g$,
          and calculate $U^t(\mathbf{P}_{1,g}')$;
    end
    Update variable $m_g^t$ by $m_g^t = m_g^t + 1$ with the
      minimum value of the objective function, i.e.,
      $g^* = \arg\min_g U^t(\mathbf{P}_{1,g}')$;
    if $U^t(\mathrm{P}_1')^{(m)} \geq U^t(\mathrm{P}_1')^{(m-1)}$ then
        Stop the iteration and return $\{m_g^t\}_{g\in\Omega_t}$;
    end
end
for $n = 1 : \widetilde{N}$ do
    for $g \in \Omega_t$ do
        Assign one VM instance to multicast group $g$,
          and calculate $U^t(\mathbf{P}_{2,g}')$;
    end
    Update variable $n_g^t$ by $n_g^t = n_g^t + 1$ with the
      maximum value of the objective function, i.e.,
      $g^* = \arg\max_g U^t(\mathbf{P}_{2,g}')$;
    if $U^t(\mathrm{P}_2')^{(n)} \geq U^t(\mathrm{P}_2')^{(n-1)}$ then
        Stop the iteration and return $\{n_g^t\}_{g\in\Omega_t}$;
    end
end
```

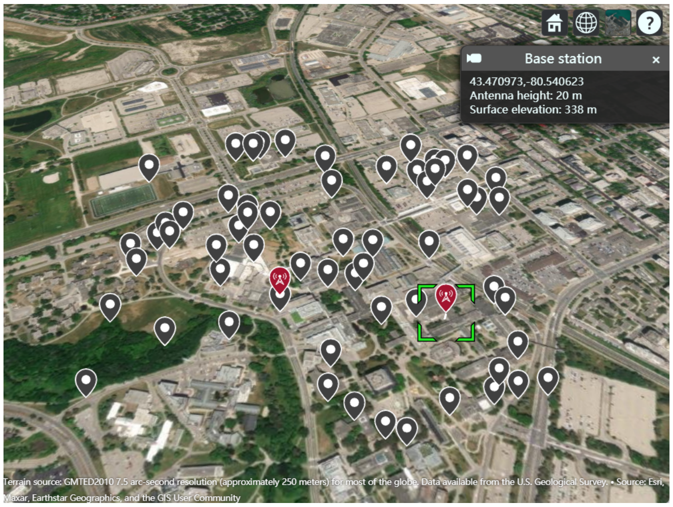
C. Computational Complexity Analysis

The proposed FS algorithm needs to find the local optimal points for bandwidth and VM instance reservation. The analysis of computational complexity is as follows. First, the computational complexity of initial resource reservation for each multicast group is $O\left(\Lambda_t^*\right)$. Then, the computational complexity of assigning the rest bandwidths is $O\left(\widetilde{m}\Lambda_t^*\right)$, where $\widetilde{m}$ is a positive value less than $\widetilde{M}$ since the bandwidth assignment can be momentarily terminated before all bandwidths are completely assigned. Next, the computational complexity of assigning the rest VM instances is $O\left(\widetilde{n}\Lambda_t^*\right)$, where $\widetilde{n}$ is a positive value less than $\widetilde{N}$ since the VM instance assignment can be momentarily terminated before all VM instances are completely assigned. Finally, the overall computational complexity of FS algorithm is $O\left(\Lambda_t^* + \widetilde{m}\Lambda_t^* + \widetilde{n}\Lambda_t^*\right)$.

$$\frac{\partial^2 \Psi_g^t}{\partial^2 m_g^t} = \frac{\exp\left\{-\xi_g^t\left(m_g^t B \log\left(1+\ell_g^t\right) - R_g^t\right)\right\}\left(\xi_g^t B \log\left(1+\ell_g^t\right)\right)^2}{\left(1+\exp\left\{-\xi_g^t\left(m_g^t B \log\left(1+\ell_g^t\right) - R_g^t\right)\right\}\right)^3}\left(1-\exp\left\{-\xi_g^t\left(m_g^t B \log\left(1+\ell_g^t\right) - R_g^t\right)\right\}\right). \tag{21}$$

TABLE I
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value	Parameter	Value
M	15	T	5 min	ϖ_1	0.5
B	2 MHz	ρ	50	ϖ_2	0.5
N	10	C	8	ϖ_3	0.7
ω	2 G Cycle/s	F_1	150	ϖ_4	1
K	60	F_2	5	δ_1	1.5
V	1000	λ	0.4	δ_2	0.3
μ	2 G Cycle/Mb	$\widehat{\lambda}$	0.3	δ_3	0.3



Terrain source: GMTED2010 7.5 arc-second resolution (approximately 250 meters) for most of the globe. Data available from the U.S. Geological Survey. • Source: Esri, Maxar, Earthstar Geographics, and the GIS User Community

Fig. 5. The simulation scene, where BSs and users are represented by red and gray icons, respectively.

VIII. SIMULATION RESULTS

A. Simulation Setup

We conduct extensive simulations on the real-world dataset to evaluate the performance of the proposed UDT-assisted resource reservation scheme. The main simulation parameters are presented in Table I. The key components of the simulation are introduced as follows.

We adopt the short video streaming dataset[4] to obtain users' swipe behaviors and the user satisfaction dataset[1] to fit the user satisfaction function. We sample 1000 short videos from the YouTube 8M dataset[5], which includes 8 video types, i.e., Entertainment, Games, Food, Sports, Science, Dance, Travel, and News. Each video has a duration of 15 sec and is encoded into four versions by the H. 265 encoder. We consider the scenario where two BSs are deployed at the University of Waterloo (UW) campus and users' initial positions are randomly and uniformly generated around two BSs, as shown in Fig. 5. Each user moves along a prescribed path within the UW campus at a speed of 2$\sim$5 km/h, and the corresponding channel path loss is obtained by the propagationModel at Matlab. The transmission power and noise power are set to 27 dBm and -174 dBm, respectively.

Since the dimensions of different UDT data attributes are different, we construct four Autoencoder models for data compression. Take the Autoencoder model for the compression of locations as an example, the encoder is composed of two Conv2D layers, each using a ReLU activation function and a 'same' padding. The first Conv2D layer has 32 filters with a kernel size of (1, 3), while the second Conv2D layer has 64 filters with the same kernel size. After the Conv2D layers, a Flatten layer is applied to transform the multi-dimensional tensor into a one-dimensional tensor. The flattened tensor is then passed through a Dense layer with 60 neurons and a linear activation function, resulting in a compressed representation of input data in the latent space. The decoder is designed to reconstruct input data from the latent space representation. It starts with a Dense layer having $60 \times 150 \times 64$ neurons, followed by a Reshape layer to convert the tensor back to the shape (60, 150, 64). Then, a Conv2DTranspose layer with 32 filters, kernel size (1, 3), ReLU activation function, and 'same' padding is applied. Finally, the output layer is another Conv2DTranspose layer with 2 channels (matching the input data), kernel size (1, 3), linear activation function, and 'same' padding. The compressed UDT data is then input to DDQN to determine the clustering number. The detailed Autoencoder and DDQN parameters are shown in Table II and III, respectively.

TABLE II
AUTOENCODER PARAMETERS

Model	Layer name	NN units	Activation	Padding
Compression for channel conditions, swipe timestamps, and prefers	Conv1D	32, 3	ReLu	same
	Conv1D	64, 3	ReLu	same
	Dense	60	Linear	/
	Dense	60×64	ReLu	/
	Conv1DTranspose	32, 3	ReLu	same
	Conv1DTranspose	150/9e3/160, 3	Linear	same
Compression for locations	Conv1D	32, 1×3	ReLu	same
	Conv1D	64, 1×3	ReLu	same
	Dense	60	Linear	/
	Dense	5.76e5	ReLu	/
	Conv1DTranspose	32, 1×3	ReLu	same
	Conv1DTranspose	2, 1×3	Linear	same

We compare the proposed UDT-assisted resource reservation scheme with the following benchmark schemes: (1) without DT (WDT), where multicast groups are constructed and updated based on users' preferences and locations, and bandwidth and VM instance reservation is based on historical video requests without considering users' swipe behaviors; (2) density-based spatial clustering of applications with noise [51] and FS algorithm (DBSCAN-FS); (3) DT-based user clustering scheme, and branch- and bound-based [52] scheduling algorithm (DT-BBS). Since the branch- and bound- method can obtain the optimal solution to 0-1 integer programming problem, the DT-BBS also represents our performance upper bound but the disadvantage is time-consuming; (4) DT-based user clustering scheme, and branching dueling Q-network-based scheduling algorithm (DT-BDQN), which can solve the high-dimensional resource scheduling problem by splitting the action space [53].

[4] ACM MM Grand Challenges: https://github.com/AItransCompetition/Short-Video-Streaming-Challenge/tree/main/data

[5] YouTube 8M dataset: https://research.google.com/youtube8m/index.html

TABLE III
DDQN PARAMETERS

Parameter	Value	Parameter	Value
Memory size	2000	Initial exploration rate	1
Discount rate	0.95	Exploration decay rate	0.995
Episode length	90	NN layer connection	FC
Number of Episodes	300	Number of hidden layers	3
Learning rate	0.001	Activation function	ReLU
Mini-batch size	32	Number of neurons	512×256× 128×64×10

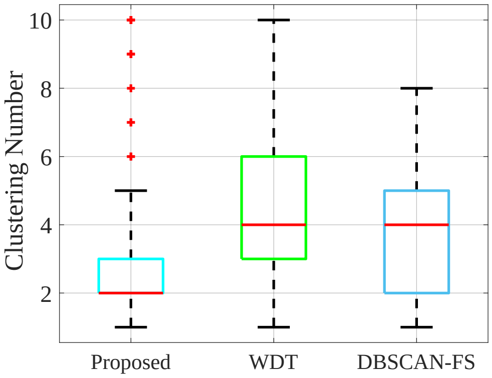


Fig. 6. Clustering number comparison.

B. Clustering Performance Evaluation

In this section, we will compare the clustering number, the user density in each multicast group, and the convergence performance of proposed clustering algorithm, respectively.

As shown in Fig. 6, we present the clustering number distribution in the boxplot for 90 resource reservation windows. It can be observed that the proposed scheme can achieve lower first-quartile, median, third-quartile, and maximum values compared with other schemes. Our proposed scheme demonstrates superior performance with relatively minimal variations, while the WDT scheme exhibits a larger fluctuation. This can be attributed to the unique capability of UDTs to effectively extract users' swipe probability distributions. Consequently, our proposed scheme empowered by UDTs can well adapt to changes in swipe behaviors and network conditions. Overall, the proposed scheme offers a more robust and efficient solution for managing multicast groups in the face of dynamic changes in user statuses.

As illustrated in Fig. 7, we present the user density of proposed UDT-assisted clustering scheme during each resource reservation window. The darker the color, the higher the user density, and vice versa. Overall, the variation trend of user density in each multicast group is relatively gradual. This can be attributed to two primary reasons. First, reconfiguring bandwidth and computing resources for a multicast group entails additional network overheads. Second, to accommodate diverse user demands and preferences, the number of users within a multicast group is inherently limited. Furthermore, we observe that certain multicast groups exhibit significantly higher user density than others. This phenomenon arises because the majority of users own similar characteristics. By clustering the similar users for multicast transmission and video transcoding, network traffic burden can be effectively relieved.

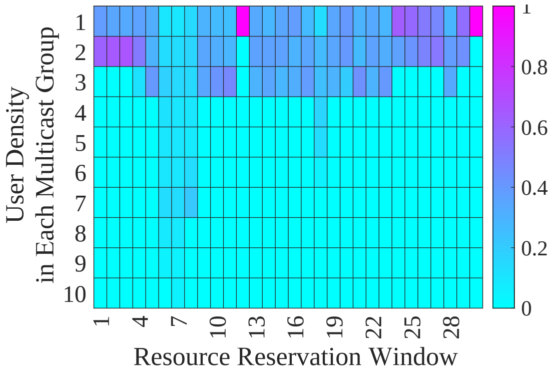


Fig. 7. User density in each multicast group.

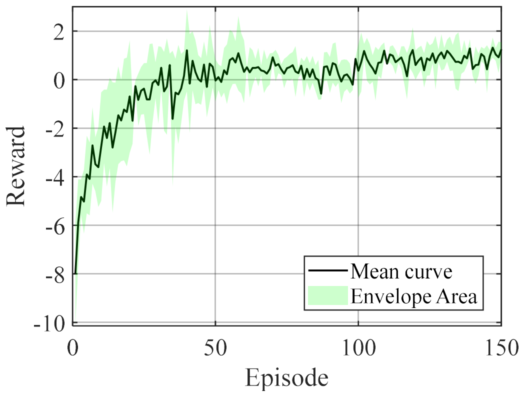
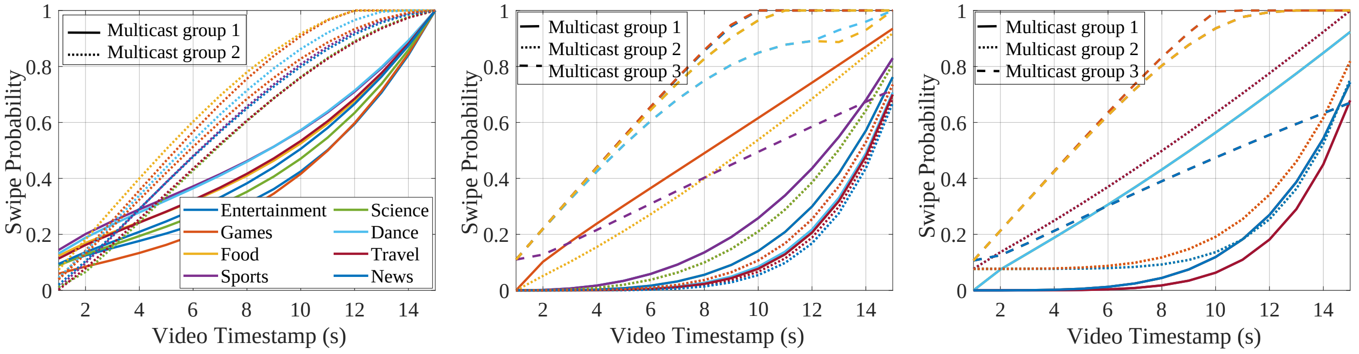


Fig. 8. Convergence performance of DDQN-based clustering algorithm.

As shown in Fig. 8, we present the convergence curve of the DDQN-based clustering algorithm for determining the number of multicast groups. We conducted three trials of training to draw the corresponding envelope area and mean curve. Each episode consists of 90 steps, and the corresponding reward is the average reward for all steps within an episode. It can be observed that as the number of episodes increases, the reward gradually grows larger. When the number of episodes approaches nearly 70, the reward converges to a stable state, indicating that the DDQN-based clustering algorithm can effectively extract user similarity from user statuses to determine the number of multicast groups.

C. Abstracted Swipe Probability Distribution

As illustrated in Fig. 9, we present the swipe probability distribution extracted via UDTs across various resource reservation windows, where different line styles denote different multicast groups. Leveraging users' status information stored in UDTs, we can infer their swipe probability distributions and cluster them with similar statuses into the same multicast group. Fig. 9(a) reveals that users in multicast group 1 demonstrate similar swipe probability distributions as those in multicast group 2 during the initial phase, but a noticeable divergence ensues over time. This suggests that UDTs can effectively differentiate swipe behaviors of various users from a global perspective, thereby facilitating more precise

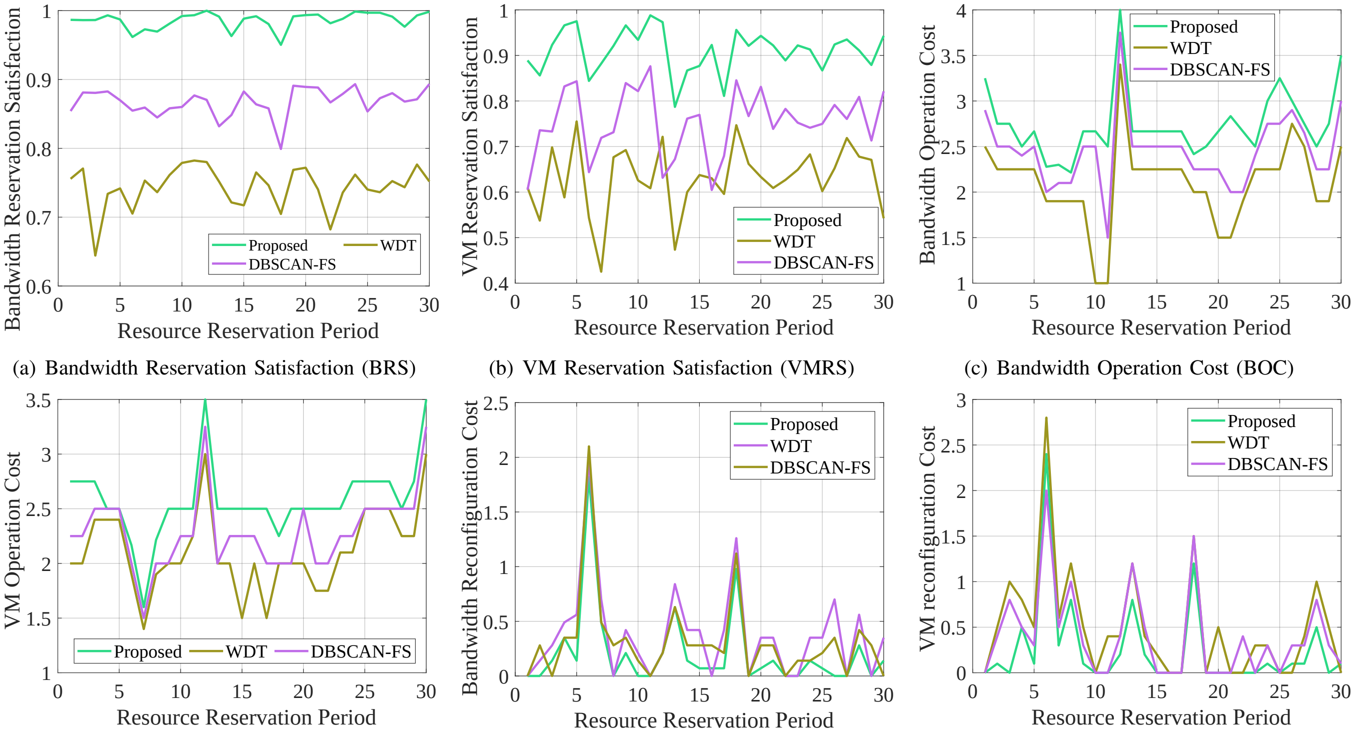


(a) Resource Reservation Window 1

(b) Resource Reservation Window 5

(c) Resource Reservation Window 9

Fig. 9. Swipe probability distribution abstracted by DT.



(a) Bandwidth Reservation Satisfaction (BRS)

(b) VM Reservation Satisfaction (VMRS)

(c) Bandwidth Operation Cost (BOC)

(d) VM Operation Cost (VMOC)

(e) Bandwidth Reconfiguration Cost (BRC)

(f) VM Reconfiguration Cost (VMRC)

Fig. 10. The performance comparison of each component in the system utility.

information provision for resource reservation. Fig. 9(b) and (c) exhibit the swipe probability distributions among three multicast groups. An overlap of swipe probabilities on certain types of videos occurs, indicating that over time, user behavior with respect to swipe probabilities for different types of videos gradually converges. This also implies a diminishing influence of video type on the swipe probability. Such information can be effectively captured by UDTs, providing valuable input to the network controller for more accurate resource reservation.

D. System Utility Performance Evaluation

As illustrated in Fig. 10, we first present the performance comparison of various system utility components under different schemes. With respect to bandwidth and VM reservation satisfaction, the proposed scheme can achieve a relatively high satisfaction level. Especially, BRS exhibits less fluctuation compared with VMRS. This is because the number of reserved VMs directly affects the speed of video transcoding, leading to variations in video quality and buffer length. Consequently, when the number of reserved VMs is insufficient, user satisfaction tends to fluctuate more significantly. In comparison to other schemes, WDT has the lowest satisfaction level. This is attributed to the lack of UDTs, which disables the network controller from swiftly and accurately analyzing users' similarities from their historical data to precisely construct multicast groups. As a result, bandwidth and VM instance reservation cannot meet the actual resource demands of each multicast group, thereby leading to lower satisfaction.

In terms of bandwidth and VM operation cost, the proposed method is relatively high. Firstly, to ensure the satisfaction

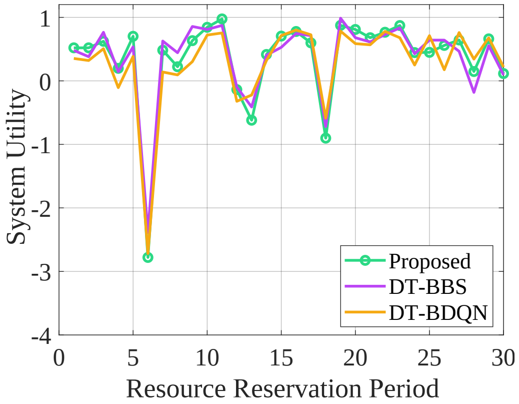


Fig. 11. System utility comparison.

TABLE IV
PERFORMANCE METRIC COMPARISON

	Proposed	WDT	DBSCAN-FS	DT-BBS	DT-BDQN
BRS	**0.99**	0.74	0.87	**0.99**	0.98
VMRS	0.91	**0.63**	0.75	**0.93**	0.91
BOC	2.74	**2.1**	2.45	2.85	2.8
VMOC	**2.57**	**2.11**	2.29	**2.59**	2.63
BRC	0.21	0.38	**0.32**	0.22	**0.2**
VMRC	0.25	0.5	0.4	**0.23**	0.27
Average system utility	0.36	-0.34	-0.04	**0.37**	0.29
Average runtime (s)	1.54	0.74	**1.15**	9.63	2.12

level of resource reservation, the network controller needs to reserve more bandwidth and VMs. Secondly, to avoid the frequent reconfiguration cost, the network controller tends to reserve a relatively large amount of bandwidth and VMs. As for reconfiguration cost, the proposed scheme is relatively low. This is because DT can globally analyze users' similarities and swipe behaviors, and group users on a larger time scale, thereby effectively avoiding frequent resource reconfigurations.

Then, we compare the system utility corresponding to different schemes in Fig. 11. It is observed that the proposed scheme can achieve quite a high system utility, although it remains generally lower than the DT-BBS scheme. This is because the DT-BBS scheme employs the branch and bound method to solve the 0-1 integer programming problem of bandwidth and VM reservation, which yields an optimal solution. Therefore, the corresponding system utility is generally the highest. The DT-BDQN scheme can also achieve a very close system utility compared with other schemes, indicating the BDQN algorithm can well learn resource scheduling strategies adapting to the dynamics of network conditions and user behaviors.

Finally, we compare the system performance metrics under different schemes, including average BRS, VMRS, BOC, VMOC, BRC, VMRC, system utility, and runtime as shown in Table IV. The best performance corresponding to each metric is highlighted in bold. As can be observed, although our proposed scheme cannot achieve the best performance in all metrics, it ensures a high average BRS, VMRS, and system utility. Especially, the system utility is very close to the DT-BBS scheme, significantly higher than that of the WDT and DBSCAN-FS schemes, but the required system runtime is much lower than that of the DT-BBS scheme. This demonstrates that our proposed scheme can quickly adapt to the dynamics of the network conditions and users' swipe behaviors to make timely adjustments to resource reservation.

IX. CONCLUSION

In this paper, we have studied a novel resource management issue to enhance the MSVS performance. We have proposed a UDT-assisted resource reservation scheme to abstract the swipe probability distribution and recommended video list for the bandwidth and computing resource demand prediction. Furthermore, we have proposed a user satisfaction model by taking the user's personalized preference and service sensitivity into account. A low-complexity resource scheduling algorithm has been designed to determine the joint bandwidth and computing resource reservation. The proposed UDT-assisted resource reservation scheme can be applied to analyze the user's behavior pattern and integrate its impact on resource management in interactive media scenarios. For future work, we will investigate the joint optimization of the segment-level caching order and resource allocation based on distilled information from UDTs to further improve user satisfaction.

REFERENCES

[1] X. Huang, W. Wu, and X. Shen, "Digital twin-assisted resource demand prediction for multicast short video streaming," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS) PhD Stu. Symp.*, Hong Kong, China, 2023.

[2] H. Yuan, S. Zhao, J. Hou, X. Wei, and S. Kwong, "Spatial and temporal consistency-aware dynamic adaptive streaming for 360-degree videos," *IEEE J. Sel. Top. Signal Process.*, vol. 14, no. 1, pp. 177–193, 2019.

[3] K. Wang, J. Jin, Y. Yang, T. Zhang, A. Nallanathan, C. Tellambura, and B. Jabbari, "Task offloading with multi-tier computing resources in next generation wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 2, pp. 306–319, 2022.

[4] Buiness of Apps, "Tiktok revenue and usage statistics 2023," 2023. [Online]. Available: https://www.businessofapps.com/data/tik-tok-statistics/

[5] W. Zhang, F. Qian, B. Han, and P. Hui, "Deepvista: 16K panoramic cinema on your mobile device," in *Proc. Int. World Wide Web (WWW)*, Ljubljana, Slovenia, 2021, pp. 2232–2244.

[6] J. Li, Y. Xu, Y. Cao, J. Zhu, and D. Wang, "Utility-driven joint caching and bitrate allocation for real-time immersive videos," *IEEE J. Sel. Top. Signal Process.*, 2023.

[7] K. Zahoor, K. Bilal, A. Erbad, and A. Mohamed, "Service-less video multicast in 5G: Enablers and challenges," *IEEE Netw.*, vol. 34, no. 3, pp. 270–276, 2020.

[8] K. Wang, W. Chen, J. Li, Y. Yang, and L. Hanzo, "Joint task offloading and caching for massive MIMO-aided multi-tier computing networks," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1820–1833, 2022.

[9] N. Reyhanian and Z.-Q. Luo, "Data-driven adaptive network slicing for multi-tenant networks," *IEEE J. Sel. Top. Signal Process.*, vol. 16, no. 1, pp. 113–128, 2021.

[10] X. Huang, M. Li, W. Wu, C. Zhou, and X. Shen, "Digital twin-assisted collaborative transcoding for better user satisfaction in live streaming," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Rome, Italy, 2023.

[11] X. Tan, L. Xu, Q. Zheng, S. Li, and B. Liu, "QoE-driven DASH multicast scheme for 5G mobile edge network," *J. Commun. Inf. Netw.*, vol. 6, no. 2, pp. 153–165, 2021.

[12] B. He, J. Wang, Q. Qi, H. Sun, H. Zhou, L. Zhang, K. Liu, and J. Liao, "Resilient QUIC protocol for emerging wireless networks," *IEEE Wirel. Commun.*, vol. 29, no. 3, pp. 64–70, 2022.

[13] S. Zhu, T. Karagioules, E. Halepovic, A. Mohammed, and A. D. Striegel, "Swipe along: a measurement study of short video services," in *Proc. ACM Multimedia Syst. Conf.*, Athlone, Ireland, 2022, pp. 123–135.

[14] M. Grieves, "Digital twin: manufacturing excellence through virtual factory replication," *White paper*, vol. 1, no. 2014, pp. 1–7, 2014.

[15] W. Xu, Y. Cui, and Z. Liu, "Optimal multi-view video transmission in multiuser wireless networks by exploiting natural and view synthesis-enabled multicast opportunities," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1494–1507, 2019.

[16] M. Li and Y.-H. Wu, "Performance analysis of adaptive multicast streaming services in wireless cellular networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2616–2630, 2018.

[17] K. Zahoor, K. Bilal, A. Erbad, and A. Mohamed, "Service-less video multicast in 5G: Enablers and challenges," *IEEE Netw.*, vol. 34, no. 3, pp. 270–276, 2020.

[18] X. Zhang, M. Yang, Y. Zhao, J. Zhang, and J. Ge, "An SDN-based video multicast orchestration scheme for 5G ultra-dense networks," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 77–83, 2017.

[19] N. A. Mitsiou, V. K. Papanikolaou, P. D. Diamantoulakis, T. Q. Duong, and G. K. Karagiannidis, "Digital twin-aided orchestration of mobile edge computing with grant-free access," *IEEE Open J. Commun. Soc.*, vol. 4, pp. 841–853, 2023.

[20] J. Montalban, P. Scopelliti, M. Fadda, E. Iradier, C. Desogus, P. Angueira, M. Murroni, and G. Araniti, "Multimedia multicast services in 5G networks: Subgrouping and non-orthogonal multiple access techniques," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 91–95, 2018.

[21] H. Soni, W. Dabbous, T. Turletti, and H. Asaeda, "NFV-based scalable guaranteed-bandwidth multicast service for software defined isp networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 14, no. 4, pp. 1157–1170, 2017.

[22] S. Mandal, G. Maji, S. Khatua, and R. K. Das, "Cost minimizing reservation and scheduling algorithms for public clouds," *IEEE Trans. Cloud Comput.*, vol. 11, no. 2, pp. 1365–1380, 2021.

[23] G. Araniti, P. Scopelliti, G.-M. Muntean, and A. Iera, "A hybrid unicast-multicast network selection for video deliveries in dense heterogeneous network environments," *IEEE Trans. Broadcast.*, vol. 65, no. 1, pp. 83–93, 2018.

[24] Q. Ye, J. Li, K. Qu, W. Zhuang, X. Shen, and X. Li, "A network slicing framework for end-to-end QoS provisioning in 5G networks," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 65–74, 2018.

[25] X. Xu, Z. Fang, L. Qi, X. Zhang, Q. He, and X. Zhou, "Tripres: Traffic flow prediction driven resource reservation for multimedia IoV with edge computing," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 17, no. 2, pp. 1–21, 2021.

[26] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and U.S. air force vehicles," in *Proc. Struct. Dyn. Mater. Conf. Special Session: Digital Twin*, Honolulu, HI, USA, 2012, p. 1818.

[27] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, "Holistic network virtualization and pervasive network intelligence for 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 1–30, 2022.

[28] T. Q. Duong, D. Van Huynh, S. R. Khosravirad, V. Sharma, O. A. Dobre, and H. Shin, "From digital twin to metaverse: The role of 6G ultra-reliable and low-latency communications with multi-tier computing," *IEEE Wirel. Commun.*, vol. 30, no. 3, pp. 140–146, 2023.

[29] A. Masaracchia, V. Sharma, B. Canberk, O. A. Dobre, and T. Q. Duong, "Digital twin for 6G: Taxonomy, research challenges, and the road ahead," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 2137–2150, 2022.

[30] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6G," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12 240–12 251, 2020.

[31] D. Van Huynh, V.-D. Nguyen, S. R. Khosravirad, V. Sharma, O. A. Dobre, H. Shin, and T. Q. Duong, "URLLC edge networks with joint optimal user association, task offloading and resource allocation: A digital twin approach," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7669–7682, 2022.

[32] W. Liu, B. Li, W. Xie, Y. Dai, and Z. Fei, "Energy efficient computation offloading in aerial edge networks with multi-agent cooperation," *IEEE Trans. Wirel. Commun.*, 2023.

[33] K. Peng, H. Huang, M. Bilal, and X. Xu, "Distributed incentives for intelligent offloading and resource allocation in digital twin driven smart industry," *IEEE Trans. Industr. Inform.*, vol. 19, no. 3, pp. 3133–3143, 2022.

[34] J. Zheng, T. H. Luan, Y. Zhang, R. Li, Y. Hui, L. Gao, and M. Dong, "Data synchronization in vehicular digital twin network: A game theoretic approach," *IEEE Trans. Wirel. Commun.*, 2023.

[35] Y. Lu, S. Maharjan, and Y. Zhang, "Adaptive edge association for wireless digital twin networks in 6G," *IEEE Internet Things J.*, vol. 8, no. 22, pp. 16 219–16 230, 2021.

[36] C. Zhou, J. Gao, M. Li, X. Shen, and W. Zhuang, "Digital twin-empowered network planning for multi-tier computing," *J. Commun. Inf. Netw.*, vol. 7, no. 3, pp. 221–238, 2022.

[37] C. Hu, W. Fan, E. Zeng, Z. Hang, F. Wang, L. Qi, and M. Z. A. Bhuiyan, "Digital twin-assisted real-time traffic data prediction method for 5g-enabled internet of vehicles," *IEEE Trans. Industr. Infor.*, vol. 18, no. 4, pp. 2811–2819, 2022.

[38] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, G. Karakostas, H. Wu, and X. Shen, "Digital twins from a networking perspective," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23 525–23 544, 2022.

[39] Z. Li, Y. Xie, R. Netravali, and K. Jamieson, "Dashlet: Taming swipe uncertainty for robust short video streaming," in *Proc. USENIX Symp. Netw. Syst. Des. Implement. (NSDI)*, Boston, MA, USA, 2023, pp. 1583–1599.

[40] G. Huang, W. Gong, B. Zhang, C. Li, and C. Li, "An online buffer-aware resource allocation algorithm for multiuser mobile video streaming," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3357–3369, 2020.

[41] X. Gong, Q. Feng, Y. Zhang, J. Qin, W. Ding, B. Li, P. Jiang, and K. Gai, "Real-time short video recommendation on mobile devices," in *Proc. ACM Int. Conf. Infor. Knowl. Management*, Atlanta, GA, USA, 2022, pp. 3103–3112.

[42] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.

[43] W. Jiang, B. Ai, J. Cheng, Y. Lin, and G. Zhang, "Sum of age-of-information minimization in aerial irss assisted wireless networks," *IEEE Commun. Lett.*, vol. 27, no. 5, pp. 1377–1381, 2023.

[44] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, Phoenix, Arizona, USA, 2016.

[45] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[46] X. Huang, C. Zhou, W. Wu, M. Li, H. Wu, and X. Shen, "Personalized QoE enhancement for adaptive video streaming: A digital twin-assisted scheme," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, 2022, pp. 4001–4006.

[47] X. Zhang, C. Yang, H. Wang, W. Xu, and C.-C. J. Kuo, "Satisfied-user-ratio modeling for compressed video," *IEEE Trans. Image Process.*, vol. 29, pp. 3777–3789, 2020.

[48] W. Wu, N. Chen, C. Zhou, M. Li, X. Shen, W. Zhuang, and X. Li, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2076–2089, 2021.

[49] D. Bertsekas, *Network optimization: continuous and discrete models*. Athena Scientific, 1998, vol. 8.

[50] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[51] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 1–21, 2017.

[52] V. I. Norkin, G. C. Pflug, and A. Ruszczyński, "A branch and bound method for stochastic global optimization," *Math. Program.*, vol. 83, pp. 425–450, 1998.

[53] A. Tavakoli, F. Pardo, and P. Kormushev, "Action branching architectures for deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.