IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 6, 2019, accepted December 20, 2019, date of publication December 31, 2019, date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963263

Elastic Timeslot-Based Advance Reservation Algorithm for Enterprise Video Conferencing Systems

ZHIWEN LIAO AND LING ZHANG
School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
Communication and Computer Network Laboratory of Guangdong Province, South China University of Technology, Guangzhou 510006, China

Corresponding author: Ling Zhang (ling@scut.edu.cn)

This work was supported by the Guangdong Key-Project for Applied Fundamental Research under Grant 2018KZDXM076.

ABSTRACT In an enterprise network, it is common to have hundreds of video conferences held simultaneously such that a large number of video streams need to be transmitted between participants in different geographical locations. For such a large transmission demand, an advance reservation (AR) system deployed for the video conferencing system can provide quality-of-service (QoS) guarantees to users and improve the resource utilization of the network. However, the effective resource reservation strategies for a heavy traffic case still remains open. In this paper, we propose an algorithm called the elastic timeslot-based advance reservation algorithm (ETARA), which aims at improving the resource utilization and reducing the computational complexity. A new time processing method, elastic timeslot, is proposed for the time domain management. Moreover, ETARA takes not only the processing time but also the resource usage into account. Comparative simulations with the existing two popular approaches, the dynamic timeslot-based approach and timeslot-based flexible approach, have been performed in terms of the acceptance ratio and runtime. The results show that with the same acceptance ratio, the runtime of ETARA can be up to 57 times lower than that of the flexible timeslot-based approach. Though ETARA has a slightly longer runtime than the dynamic approach, the acceptance ratio of ETARA can be twice as high as that of the dynamic timeslot-based approach.

INDEX TERMS Enterprise video conferencing system, advance reservation (AR), bandwidth reservation, QoS.

I. INTRODUCTION

With the enhancement of video terminal capabilities and rapid network development, video conferencing systems, such as Cisco TelePresence, iChat, and Skype, have become mainstream on the Internet. The newest version of such Internet video conferencing systems supports multiple geographically distributed participants in the same video session and features high definition. These systems require rich data collaboration, comprehensive conference management and control capabilities. These characteristics impose high requirements on the quality-of-service (QoS) level that the transmission network can ensure, such as reachability, bandwidth, delay, and jitter. However, QoS guarantee strategies are not widely deployed on the Internet. In this paper, we mainly study the mechanisms of guaranteeing QoS for video conferencing systems on large enterprise networks or campus networks.

Advance reservation (AR) [1] is an effective way to gain high definition and real-time collaboration for a video conferencing system. By deploying an AR system, networks can reserve bandwidth for a video conference in advance for a future time to guarantee specific QoS features. In general, due to declaring the start time and holding time of the conference, the AR system can benefit the network because the network can make plans to improve the resource utilization based on knowledge of the future network state and can benefit users because a better QoS can be provided for users.

Time domain management is an important aspect of AR. The time domain management approach can be concluded as reservation-based and timeslot-based approaches [2]. For the reservation-based approach, when a new request arrives, the previously accepted requests with overlapping time are identified to control the acceptance of the new request.

The associate editor coordinating the review of this manuscript and approving it for publication was Minhaj Ahmad Khan.

If there are n conferences that overlap with each other in time, then the time complexity of processing these n requests is $O(n^2)$. To cope with the high time complexity of reservation-based approaches, the timeslot-based approach has been proposed as an efficient way [2]. According to the way the time domain is processed, the existing timeslot-based approaches can be classified into the static timeslot-based approach [2], dynamic timeslot-based approach [2] and flexible timeslot-based approach [3]–[5]. The authors in [3] theoretically analyzed the advantages and drawbacks of the static and flexible timeslot-based approach and concluded that the static timeslot-based approach is less effective in terms of resource utilization than the flexible timeslot-based approach. In this paper, we further analyze the performance of the static, dynamic and flexible timeslot-based approaches in terms of resource utilization and computational complexity. Though the flexible timeslot-based approach is highly beneficial when dealing with bursty traffic in a low-demand network with long-term downtimes, it is ineffective in a high-demand network [3], [5]. A resilient AR approach based on flexible timeslots, in combination with a runtime adaptation approach, has been introduced to provide flexible, reliable and adaptive advance reservation to an AR system [4]. However, due to the limitation of the flexible timeslot-based approach, the resilient AR approach is only suitable for the situation where the number of requests is relatively small.

Although the flexible timeslot-based approach is very effective for a low network load, it is not suitable for a large network load [3], [5]. In an enterprise network, it is common to have hundreds of conferences online at the same time, and video communication is possible between any two conference participants. For example, for a six-party discussion conference in which any participant can send and receive video streams, the number of online video streams at the same time is $6 \times 5 = 30$. If there are 500 conferences, there are $500 \times 30 = 15000$ video streams online. Reserving bandwidth resources within a specific time will result in fine-grained changes of network available bandwidth over time. In particular, when there are a large number of conferences online in the meantime, time consumption and bandwidth usage will significantly increase [2].

Moreover, we find that the existing timeslot-based approaches only take the request time into account, not the use of resources. Initially, there are enough resources to accommodate many requests. In this case, if the original time can be clustered to the integer multiple of granularity, the number of timeslots can be greatly reduced; thus, the computational complexity can be reduced while the resource utilization will not be reduced. As the number of requests increases, the resources dynamically decrease. In this case, we can use the resource effective access control to improve the resource utilization.

Motivated by the above views, in this paper, we propose an improved advance reservation algorithm, the elastic timeslot-based advance reservation algorithm (ETARA), which aims at improving the resource utilization and reducing the computational complexity so that it is suitable for a heavy network load. We proposed a new time management approach, elastic timeslot, and the corresponding access control process. The effectiveness of the proposed algorithm is illustrated by the simulation results. The main contributions of the paper are summarized as follows.

1) We consider a heavy traffic case in which there are thousands of video streams online in the meantime. It is challenging to reserve bandwidth for such a heavy traffic load to gain effective resource utilization and efficiency running speed. To the best of our knowledge, this is the first time to reserve bandwidth for such a heavy traffic load in the enterprise network.

2) The existing approaches are ineffective in processing a heavy traffic load in terms of resource utilization and computational complexity [3], [5]. In contrast to the existing timeslot-based approaches, we focus on improving the resource utilization and reducing the computational complexity.

3) We take not only the time processing but also the dynamic resources usage into account. Because of the heavy traffic load, it gets more complex to design an effective access control process. Due to the uniqueness of resources, we have developed a switching mechanism between the time efficiency access control and the resource effective access control.

This work is an extension of our previous works [6]–[8]. We designed a video conferencing system (VCS) called CoolView 2.0 whose signals are controlled centrally and video streams are mixed on distributed endpoints. We investigated mechanisms to guarantee QoS for VCSs based on the resource reservation protocol (RSVP) and network management protocols such as SNMP, network configuration protocol (NetConf) and Telnet [6].

The drawbacks are that communication between routers results in a long delay time, which is undesirable for video conferencing. Our previous work has designed an software defined networking (SDN)-based architecture for VCSs [7]. Later, the authors in [8] designed a static timeslot-based algorithm that is time consuming. In this work, we analyze, design, implement and evaluate the proposed ETARA for a VCS in an enterprise network. Extensive simulations are performed to compare ETARA with other flexible and dynamic approaches to quantitatively study the computational complexity and quality. The proposed AR algorithm is suitable not only for VCSs but also for other high-performance applications running on networks such as grid and cloud networks. Recently, SDN has made the networks programmable and provided a mechanism for bandwidth reservation by decoupling the control layer from data forwarding. In this paper, we assume that such a QoS mechanism of SDN is available, and our algorithm is deployed on top of the SDN controller.

The remainder of this paper is organized as follows. Related work is described in Section II. An AR-enabled network architecture for a VCS is designed in Section III. The problem formulation and mathematical model are

discussed in Section IV. We analyze the existing timeslot-based approaches in terms of resource utilization and computational complexity in Section V. The main idea and algorithm of ETARA are discussed in Section VI. Simulations and evaluations are shown in Section VII. Finally, Section VIII summarizes this paper.

II. RELATED WORK

Here, past research on AR approaches for VCSs is mainly discussed from the perspectives of VCSs and AR approaches.

The IETF centralized conferencing (XCON) Working Group devised a series of RFC documents for XCON. A comprehensive architecture of XCON is proposed in RFC 5239 [9]. Our video conferencing system adopts centralized signal control and a distributed video stream mixing mode similar to the XCON description in RFC 4353 [10]. RFC 4597 outlines a set of basic and advanced conference scenarios [11]. In this work, we refer to these scenarios and adopt two conference scenarios as the basis for evaluation. Recently, there have been many video conferencing systems configured on top of SDN [12]–[14] and deployed in the cloud environment [15], [16].

AR was initially proposed to solve traffic modeling and access control problems in telecommunication systems [17], [18]. Later, the authors in [19], [20] proposed AR to guarantee the QoS for real-time video streaming applications. An advanced reservation protocol was designed and evaluated on top of RSVP [21]. In [22], the authors focused on the impact of several service models, ranging from basic traditional models to more sophisticated models of AR, on the computational complexity of routing. The authors in [23] expanded the works in [22] to exhaustively combine different path and bandwidth constraints and formulate four types of advance bandwidth scheduling problems. In [24], they resolved the problem of balancing the time cost and the network state accuracy in multiple SDN controller advance reservation environments to reduce the blocking rate. In [25], AR was deployed in a cloud computing environment to improve the resource utilization.

Some works focused on the mechanism of dynamically adjusting the allocation between immediately reserved resources and advance reservations to maximize resource utilization [26], [27]. An optimal algorithm was proposed to dynamically reprovision the scheduled AR requests under heavy network loads to improve the efficiency of the network spectrum resources [28]. AR can also provide differentiated services to users to increase the revenue of the network. For example, in [29], [30], the authors focused on a revenue-driven dynamic resource provisioning approach to increase the profit of the network. In [31], the authors proposed a fine-granularity QoS micropayment system that allows users to prepay for guaranteed bandwidth reservation for their flows for a period of time.

Some works of AR focus on the routing and wavelength assignment (RWA) problem in an optical wavelength-routed wavelength division multiplexing (WDM) network [1], [32], [33]. Extensive AR research has focused on scientific collaborations involving transfer of large files or video streaming data on a shared network. In [34], the authors formulated two NP-complete problems and designed two heuristic algorithms based on the static timeslot-based AR approach. In [35], the authors formulated two kinds of optimization problem, the minimum cost with deadline and the minimum completion time with cost, and gave the corresponding optimization algorithms and proofs. In [36], the authors used the variable bandwidth variable path model to maximize the reserved requests in periodic scheduling under a high-performance dedicated path. In [37], the authors designed and implemented an orchestration architecture to provide multipath and multidomain advance reservations based on software-defined exchanges (SDX), aiming to improve the reservation success ratio.

It is worth noting that all the above works on advance reservation use the existing approaches to manage the time domain, which is different from our proposed algorithm.

III. AR-ENABLED NETWORK ARCHITECTURE

In this paper, we extend the video conference system in [7] by adding the advance reservation system of task scheduling and resource reservation. If users want to obtain a QoS guarantee, they must reserve resources for the conference in the advance reservation system. Otherwise, they can only acquire best-effort services.

Fig.1 shows the AR-enabled network architecture and components of a VCS. In the enterprise network, the participants in geographically distributed conference rooms are connected to the shared enterprise networks, consisting of interconnected switches. The video conferencing management system (VCMS) is the management and control center of the VCS. It provides an interface to users to make AR requests, manages all the conferences and maintains each session during the conferencing period. Endpoints are on the user sides and can join or leave the conference. During the session, the endpoints receive, mix and send the video stream. The process of advance reservation consists of the following steps.

Step 1: The user makes a conference request via a browser to the VCMS. The VCMS delivers the AR request to the AR system.

Step 2: The AR system enqueues the AR request, waiting for scheduling by the AR algorithm.

Step 3: The preprocessing module schedules the conference request and resolves it into multiple video stream requests if necessary according to the conference scenario.

Step 4: The AR algorithm makes use of the topology management module (TMM) to obtain the topology of the network and the traffic engineering database (TED) to obtain the available bandwidth on each link. Combined with the request time, the AR algorithm generates resource matrices with the time attribute. As the request is processed, the AR algorithm makes a decision based on whether there is sufficient bandwidth in resource matrices to ensure a certain QoS level.

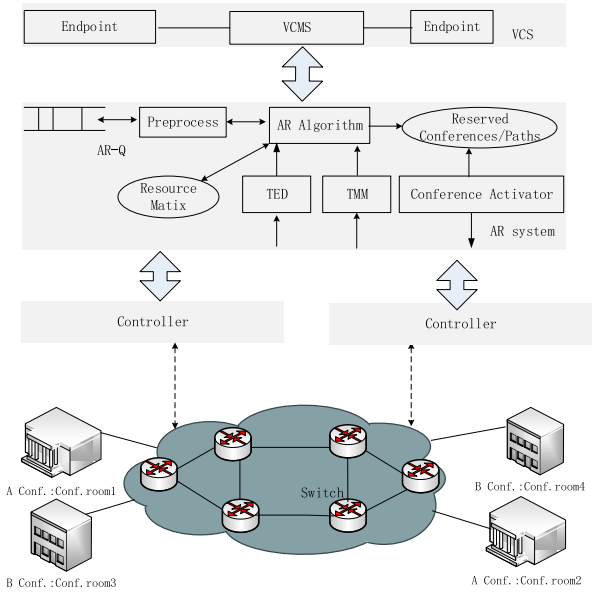


FIGURE 1. AR-enabled network architecture and components of a VCS.

Step 5: Once the request is accepted, the requested bandwidth will be reserved in resource matrices. The admitted conference is also reserved in the AR system, along with the corresponding path.

Step 6: The conference activator monitors the reserved conferences. When the requested start time is reached, the activator activates the conference. Afterward, it accesses the interfaces in the controller and sends messages to the corresponding switches.

Step 7: Finally, the endpoints acquire the activated conference and join the conference; then, the participants can exchange the video data with each other via the paths assigned earlier.

IV. NETWORK MODEL AND PROBLEM DESCRIPTION

We define the topology of the enterprise network as a directed graph $G(V, E, B)$, where V and E are the sets of nodes and links, respectively. Here, V includes two types of nodes, i.e., the endpoints and the switches. If link e is from u to v $(u, v \in V)$, then we have $e = (u, v) \in E$ and the bandwidth of the link e is denoted as $b_e \in B$.

We consider the situation where, in the enterprise network, there are a number of pending conference requests. According to the scenarios defined in RFC 4597 [11], we adopt two conference scenarios. One is called the lecture scenario, which enables a lecturer who presents a topic to be seen or heard by the other participants who cannot be seen or heard by the others, while the other is called the discussion scenario, in which every participant can be seen or heard by all of the other participants. Therefore, for a conference of n participants, there are at most $(n-1)$ video streams for the lecture scenario, while there are $n \times (n-1)$ video streams for the discussion scenario. Actually, users can declare a conference scenario or not. Hence, we denote the set of conference scenarios as $S = \{L, D, U\}$, where L and D denote the lecture and discussion scenarios, respectively, while U allows the users not to declare any conference scenario.

When a conference request $c_m(a_m, s_m, e_m, p_m)$ arrives, if the declared scenario a_m is the lecture scenario or the discussion scenario, the AR system first resolves it into a corresponding number of video stream requests $\{R_k(c_k, v_k^s, v_k^d, s_k, e_k, b_k)\}$ such that the identifier c_k is set equal to the conference index m, and the source endpoint v_k^s is the endpoint that the lecturer uses in the lecture scenario. However, it can be any endpoint that a participant uses in the discussion scenario, the destination endpoint v_k^d is any other endpoint, the start time s_k and end time e_k are set equal to the conference start time s_m and the conference end time e_m, respectively, and the sent bits of b_k are set equal to the sent bits of the source endpoint v_k^s. Here, both the source endpoint v_k^s and destination endpoint v_k^d are from the set of endpoints $p_m = \{(v_n, b_n, r_n)\}$, where v_n denotes the identifier of the endpoint in V, b_n denotes the sent bits of the endpoint, and r_n denotes the role of the participant who uses the endpoint. The role of the endpoint can be the lecturer or a participant; thus, we define the set of roles $R = \{l, p\}$, where l denotes the role of the lecturer and p denotes the role of a participant.

Sometimes, the user does not want to declare the conference scenario, that is, $a_m = U$. In this case, the user is also required to present the set of video stream requests $\{R_k(c_k, v_k^s, v_k^d, s_k, e_k, b_k)\}$, in which s_k and e_k can be any time in the conference time duration, and b_k can be less than or equal to the sent bits of s_k, while the other elements are the same as those described above.

We save all the pending conference requests in the set $C = \{c_m(a_m, s_m, e_m, p_m)\}$ and all the video stream AR requests in the set $\Gamma = \{R_k(c_k, v_k^s, v_k^d, s_k, e_k, b_k)\}$.

Then, the AR system computes a widest path for every R_k that connects source endpoint s_k and d_k and meets the required bandwidth b_k during the time span (s_k, e_k).

A conference request is accepted only when its contained video stream requests are all accommodated with the required bandwidth within a specific time. In this paper, we study how to effectively deal with these AR requests in order to respond to users within an acceptable time and improve the acceptance ratio when the number of requests is large.

V. ANALYSIS OF EXISTING TIMESLOT-BASED AR APPROACHES

The timeslot-based approach divides the time domain into smaller timeslots. Each timeslot represents a period of time and aggregates the available bandwidth of every link in this timeslot. Access control is executed in every timeslot to improve the acceptance ratio based on the available bandwidth. Thereafter, this approach provides the variable paths with fixed bandwidth to users. The AR system initially presents to users the time domain that is the initial largest timeslot, within which users can make reservations.

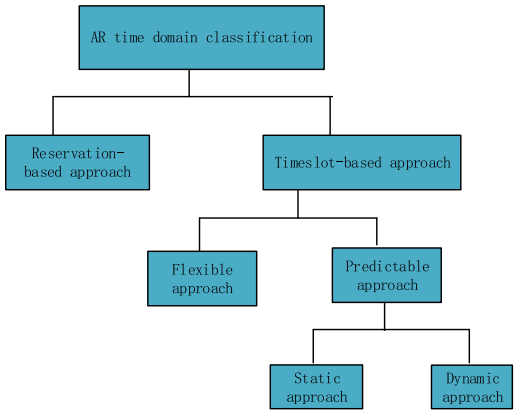


FIGURE 2. Time domain classification of AR approaches.

Generally, when a new request arrives, it is processed in three steps.

Step 1: The start time and end time are processed and sort inserted into a time array, which stores the ordered sequence of discrete points of timeslots.

Step 2: All the active timeslots that are a subset of all the timeslots covering the requested period are identified.

Step 3: Access control is performed in each active timeslot to decide whether to accept the request or not based on the available bandwidth in the resource matrix.

Therefore, the number of timeslots greatly affects the computational complexity.

The granularity is the smallest timeslot that the time domain can be divided into. Depending on whether or not granularity is introduced, here, we extend the classification, as shown in Fig. 2. The timeslot-based approach is classified into flexible and predictable approaches. The flexible approach does not introduce granularity or its granularity is infinitesimal, while the predictable approach introduces non-infinitesimal granularity. In the predictable approach, the start and end times must be rounded down and up to the nearest integer times of the granularity. The predictable timeslot-based approach can be further divided into the static approach and dynamic approach according to whether the length of the timeslot is fixed or variable.

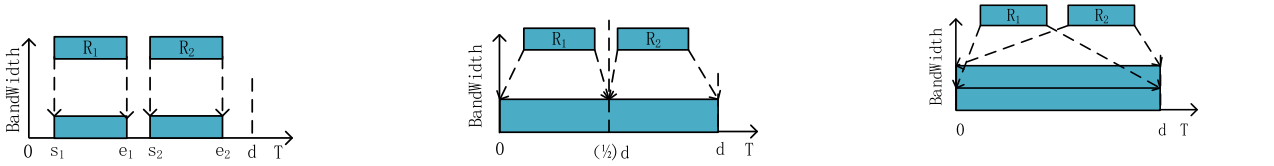
To better analyze the existing approaches, we draw Fig. 3 and Fig. 4 to illustrate their resource utilization and computational complexity. We define the time array as T. In T, we set $T(1) = 0$, which represents the beginning time of the time domain. The *ith* timeslot is denoted as $(T(i-1), T(i))$ while $i > 1$. If $i == 1$, then the first timeslot is denoted as $(T(1), T(1))$, that is, $(0, 0)$. The set of active timeslots is denoted as T_a. For simplicity, we only outline the parameters of the request time.

A. RESOURCE UTILIZATION OF THE EXISTING APPROACHES

Fig. 3 presents three subgraphs to illustrate the impact of granularity sizes on resource utilization. As shown in Fig. 3, the horizontal coordinates represent the time domain, whose length is represented as d, and the vertical coordinates represent the bandwidth consumption and the remaining bandwidth of a link. Therefore, initially, $T = [0, d]$. As $R_1(s_1, e_1)$ and $R_2(s_2, e_2)$ arrive at the AR system one by one, different approaches process the time of requests in different ways.

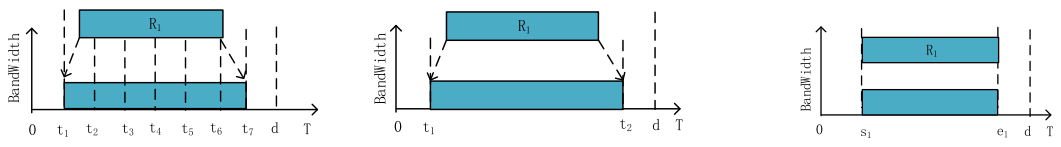
The flexible approach does not introduce granularity, or it introduces an infinitesimal granularity. In this case, the AR system does not change the times of R_1 and R_2, as shown in Fig. 3(a). The time domain is partitioned by the original time points of R_1 and R_2 into smaller timeslots; thus, T is updated as $T = [0, s_1, e_1, s_2, e_2, d]$. Therefore, the time array T in the flexible approach stores any original requested time points during which the bandwidth is reserved.

The predictable approach introduces the granularity g. In Fig. 3(b), g is set equal to $(1/2)d$. The AR system rounds



(a) infinitesimal granularity of the flexible approach (b) small granularity of the predictable approach (c) large granularity of the predictable approach

FIGURE 3. Impact of granularity size on resource utilization.



(a) timeslots in the static approach (b) timeslots in the dynamic approach (c) timeslots in the flexible approach

FIGURE 4. Impact of number of timeslots on computational complexity.

the start time and end time down and up to the nearest integer multiple of the granularity. In this situation, the time domain d is partitioned by $(1/2)d$, and T is updated as $T = [0, (1/2)d, d]$. We can see from Fig. 3(b) that R_1 and R_2 do not overlap before rounding, and they also do not overlap after rounding; however, the rounding operation lengthens the request time span. Therefore, it increases the unused bandwidth (i.e., decreases resource utilization).

In Fig. 3(c), the granularity g, in the predictable approach, is set equal to d. As R_1 arrives, the AR system performs the rounding operation, and T is updated as $T = [0, d]$. Then, as R_2 arrives, the same operation is performed; still, $T = [0, d]$. We can see from Fig. 3(c) that R_1 and R_2 do not overlap before rounding; however, after being processed, they overlap in time, which causes a larger unnecessary waste of resources. The larger the granularity is, the larger the overlap probability is. The overlap intensifies the competition for limited resources.

Therefore, we can conclude that the flexible approach is more effective in the resource utilization than the static approach and dynamic approach. The static and dynamic approaches achieve the same performance in resource utilization.

B. COMPUTATIONAL COMPLEXITY OF THE EXISTING APPROACHES

In Fig. 4, the length of the time domain d is set equal to 8 times the granularity g; then, $g = d/8$. When $R_1(s_1, e_1)$ arrives, as shown in Fig. 4, how is the reservation system to handle the two points and perform access control for each approach?

(a) In the static approach, the time domain is partitioned into a fixed number of timeslots whose length is equal to g. In this case, T is initialized to $T = [0, t_1, t_2, \ldots, d]$, as shown in Fig. 4(a). When R_1 arrives, s_1 and e_1 are first rounded down and up, respectively, to the nearest slot boundaries, which are t_1 and t_7, as shown in Fig.4(a). Then, the active timeslots for R_1 can be identified as $T_a = \{(t_1, t_2), (t_2, t_3), (t_3, t_4), (t_4, t_5), (t_5, t_6), (t_6, t_7)\}$.

(b) The dynamic approach shown in Fig. 4(b) also introduces granularity g. Different from the static approach, the time domain of the dynamic approach is dynamically partitioned into smaller timeslots with the arrival of the requests. In this case, T dynamically changes size. Initially, $T = [0, d]$. As R_1 arrives, s_1 and e_1 are first rounded down and up to t_1 and t_2, respectively, which are then sort inserted into T; thus, T is updated to $T = [0, t_1, t_2, d]$. The active timeslot for R_1 is $T_a = \{(t_1, t_2)\}$.

(c) For the flexible approach shown in Fig. 4(c), the size of T dynamically changes. For each incoming request, at most two points are inserted into T. Unlike the dynamic approach, the flexible approach does not perform the rounding operations; thus, $T = [0, s_1, e_1, d]$. The active timeslot is $T_a = (s_1, e_1)$.

Then, access control is executed in each active timeslot.

The number of timeslots is proportional to the computational complexity. The number of timeslots of the static approach is not less than that of the dynamic approach, and the number of timeslots of the dynamic algorithm and the flexible approach dynamically increases with the number of requests. As the requests arrive, the number of timeslots of the static approach stays constant, equal to $\lfloor d/g \rfloor$, while the number of timeslots of the dynamic approach increases but is limited by the maximum number $\lfloor d/g \rfloor$. Because the time domain can be divided by any request time in the flexible approach, the number of timeslots is unlimited and unpredictable for the flexible approach in a heavy traffic load environment.

Therefore, the computational complexity of the static approach is higher than that of the dynamic approach, and in a large traffic load situation, the computational complexity of the flexible approach is higher than that of the static and dynamic approaches.

Overall, the static approach is less effective than the dynamic approach. The flexible approach is more effective in resource utilization than the dynamic approach, while it is less computationally efficient than the dynamic approach. Therefore, we use the flexible approach and dynamic approach as the comparison algorithms of our algorithm.

VI. ELASTIC TIMESLOT-BASED AR ALGORITHM (ETARA)

A. IDEA OF ETARA

From the above analysis, we can attain the following inspiration:

1) The dynamic approach clusters the request time to integer multiple of the granularity, which reduces the number of timeslots and speeds up the running. However, the dynamic approach loses the original time at the same time, which lengthens the time span of requests, resulting in unused resources.

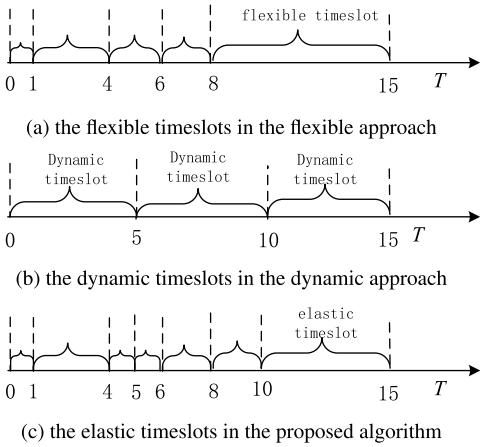
2) The flexible approach saves the original request time. It shows more effectiveness in resource utilization than the dynamic approach. Nevertheless, with increasing number of requests, there are ever-increasing timeslots, which causes a higher computational complexity.

Furthermore, we find that these approaches only consider the processing time of requests, not the resource supply. Initially, there are enough resources to supply many requests. With the arrival of requests, the resource capacity dynamically decreases, and the resource supply enters a tense stage.

Based on the above points of view, we propose an improved algorithm ETARA, which aims to achieve effective resource utilization while reducing the computational complexity. We can first use the resource matrix with the dynamic timeslot to supply resources when the resource supply is sufficient. This can not only speed up the running but also not reduce the acceptance ratio. It is worth noting that since the dynamic timeslot causes unused bandwidth, the real bandwidth usage should also be saved. For this purpose, we define a new time processing method for the domain management called the elastic timeslot that consists of elastic time. The elastic time can be either the clustering time or the original request time.

When the resource matrix with the dynamic timeslot fails to supply resources, we can use the resource matrix with the elastic timeslot for access control.

We illustrate to explain the three kinds of timeslots, as shown in Fig. 5. Suppose that the granularity is 5 and the time domain is from 0 to 15. Therefore, initially, $T = [0, 15]$. There arrive two requests, such as $R_1(1, 4)$ and $R_2(6, 8)$.



(a) the flexible timeslots in the flexible approach

(b) the dynamic timeslots in the dynamic approach

(c) the elastic timeslots in the proposed algorithm

FIGURE 5. The timeslots in different approaches.

The flexible approach does not introduce the granularity and stores the original request time. Therefore, $T = [0, 1, 4, 6, 8, 15]$. The flexible timeslots consist of the original request time, as shown in Fig. 4(a).

The dynamic approach clusters the original request time to the nearest integer multiple of granularity. As $R_1(1, 4)$ arrives, the AR system rounds down and up the start time and end time respectively, i.e., $5 \times \lfloor 1/5 \rfloor = 0$, $5 \times \lceil 4/5 \rceil = 5$. Then, the AR system does the same for $R_2(6, 8)$. Therefore, $T = [0, 5, 10, 15]$. The dynamic timeslots consist of clustering time, as shown in Fig. 4(b).

The proposed elastic timeslot stores the clustering time and the original time. Therefore, $T = [0, 1, 4, 5, 6, 8, 10, 15]$. The elastic timeslots consist of the elastic time that can be the clustering time or the original time, as shown in Fig. 4(c).

The existing timeslot-based approaches focus on the access control in one kind of timeslot while we focus on how to coordinate the two kinds of timeslots to act as a whole so that the proposed algorithm is effective in resource utilization and meanwhile reduces the computational complexity. Because of the uniqueness of resources, determining how to maintain the consistency of resources for the dynamic timeslot and elastic timeslot is a problem we need to solve. Specifically, this mainly requires solving the following problems.

1) Under what circumstances it is necessary to move from the dynamic timeslot to the elastic timeslot in order to increase the acceptance ratio? If it is not necessary to move to the second part, then such a switch will not only not increase the acceptance ratio but also consume unnecessary running time.

2) In ETARA, the request is actually accommodated using the resource with the elastic timeslot. Why is the path calculated using the resource matrix with the dynamic timeslot also feasible for the real circumstance? This is a question to be answered.

3) In the elastic timeslot, the resource matrix with the elastic timeslot is used for access control, and the widest path is calculated. Then, how can the path and the reserved bandwidth be reflected in another matrix for access control of subsequent requests? This is a question we are going to study.

4) How should the time correspondence in the two timeslots be identified to switch between them?

B. OVERVIEW

Specifically, we define a one-dimensional time array TD to store the discrete time points of the dynamic timeslot and a corresponding available bandwidth matrix MD to aggregate the available bandwidth of every link in every timeslot of TD. Similarly, we define a one-dimensional time array TE to store the discrete time points of the elastic timeslot and a corresponding available bandwidth matrix ME to aggregate the available bandwidth of every link in every timeslot of TE.

In TD and TE, we set $TD(1) = TE(1) = 0$, which represents the beginning time of the time domain. The ith timeslots are denoted as $(TD(i-1), TD(i))$ and $(TE(i-1), TE(i))$, respectively, while $i > 1$. If $i == 1$, then the first timeslots are denoted as $(0, 0)$. In this situation, the available bandwidths of the jth link in the ith timeslots are represented as $MD(j, i)$ and $ME(j, i)$ respectively, in which $MD(j, 1)$ and $ME(j, 1)$ save the initial available bandwidth of link j. Consequently, the two matrices have $|E|$ rows, and MD has $|TD|$ columns, while ME has $|TE|$ columns. The set of active timeslots for dynamic timeslots is denoted as TD_a, while the set of active timeslots for elastic timeslots is denoted as TE_a.

Additionally, we define two reservation arrays RD and RE for the dynamic timeslot and elastic timeslot, respectively, which store the accepted reservations in every timeslot, as the basis for determining whether to switch between the two parts. Generally, the AR request is first accommodated with the dynamic timeslot; if it cannot be accommodated, then ETARA switches to the elastic timeslot dependent on the two reservation arrays. In the following, we use a simple example to describe the process of ETARA.

C. AN EXAMPLE

As shown in Fig. 6, there are three pending requests. The length of the time domain is supposed to be 15 minutes; thus, the time arrays can be initialized as $TD = TE = [0, 15]$, where there are two timeslots: one is at point 0, and the other extends from 0 minutes to 15 minutes, that is, $(0, 15)$. The available bandwidths in each timeslot of MD and ME are initialized with the initial available bandwidth, and the reservation arrays are initialized as $RD = RE = [\varnothing, \varnothing]$. Suppose that the granularity of the dynamic timeslot is 10 minutes. The specific steps are as follows.

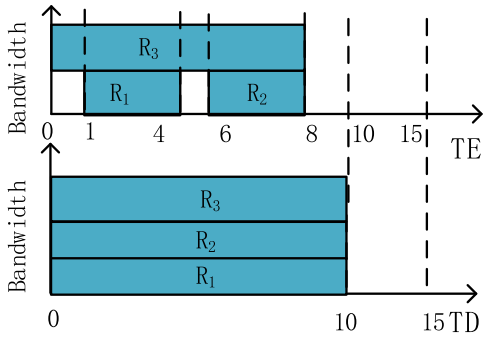


FIGURE 6. The process of ETARA.

Step 1: When $R_1(1, 4)$ arrives, the dynamic timeslot rounds down and up the request times, respectively, to 0 and 10, which are then sort inserted into TD; thus, TD is updated to $TD = [0, 10, 15]$. Correspondingly, time points 0, 10, 1, and 4 are sort inserted into TE; thus, TE is updated to $TE = [0, 1, 4, 10, 15]$. The set of active timeslots of TD is identified as $TD_a = \{(0, 10)\}$, and the set of active timeslots of TE is $TE_a = \{(1, 4)\}$. Then, access control is performed in each timeslot of TD_a with MD. Suppose that R_1 can be accommodated with a path; it will be reserved in TD_a and TE_a. Thus, the reserved matrices are updated to $RD = [\varnothing, R_1, \varnothing]$ and $RE = [\varnothing, \varnothing, R_1, \varnothing, \varnothing]$; the requested bandwidth is subtracted from the available bandwidths in MD and ME.

Step 2: Next, when $R_2(6, 8)$ arrives, the set of active timeslots of TD is identified as $TD_a = \{(0, 10)\}$; TE is updated to $TE = [0, 1, 4, 6, 8, 10, 15]$, and $TE_a = \{(6, 8)\}$. Then, the dynamic timeslot is used to perform access control in TD_a with MD. Suppose that R_2 can be accommodated with a path; it will be reserved in TD_a and TE_a. Thus, the reservation arrays are then updated to $RD = [\varnothing, \{R_1, R_2\}, \varnothing]$ and $RE = [\varnothing, \varnothing, R_1, \varnothing, R_2, \varnothing, \varnothing]$; the requested bandwidth is subtracted from the available bandwidths in MD and ME.

Step 3: Finally, when $R_3(0, 8)$ arrives, $TD_a = \{(0, 10)\}$ and $TE_a = \{(0, 1), (1, 4), (4, 6), (6, 8)\}$. Then, the dynamic timeslot is used to perform access control in TD_a. If R_3 cannot be accommodated, then the elastic timeslot is utilized. First, whether the number of reserved requests in TD_a is larger than that in TE_a is determined; if yes, then the elastic timeslot is utilized. As shown in Fig. 6, the reserved requests in TD_a are R_1 and R_2, while in TE_a, they are R_1 or R_2; therefore, it is possible that R_3 can be accommodated in TE_a with ME. Then, the elastic timeslot is used to perform access control in each timeslot of TE_a. Suppose that R_3 can be accommodated with paths; it is reserved in TE_a and TD_a. Thus, the reservation arrays are then updated to $RE = [\varnothing, R_3, \{R_1, R_3\}, R_3, \{R_2, R_3\}, \varnothing, \varnothing]$ and $RD = [\varnothing, \{R_1, R_2, R_3\}, \varnothing]$; the available bandwidths on the reserved paths are correspondingly updated.

In fact, the dynamic timeslot is used to speed up the execution. The real resource supply is realized by the elastic timeslot, which saves the real request time and unchanged bandwidth. Therefore, ETARA can achieve a better performance. The dynamic timeslot can work as an efficient way only if it satisfies Lemma 1, which we provide a proof for.

Lemma 1: For a request R_1, suppose that the *ith* active timeslot $TD_a(i)$ contains the *jth* active timeslot $TE_a(j)$ in time; then, $RD(i) \supseteq RE(j)$. Suppose that p is the calculated path in $TD_a(i)$; then, p is also feasible for $TE_a(j)$.

Proof: In the dynamic timeslot, the start time and end times are rounded down and up to the nearest integer times of the granularity; thus, one dynamic timeslot can cluster multiple elastic timeslots. Specifically, for $\forall R_1$, if the timeslot $TD_a(i)$ contains $TE_a(j)$ in time, then $TD_a(i)$ can cluster multiple timeslots of the elastic timeslot in addition to the timeslot $TE_a(j)$; therefore, the reserved requests in the dynamic timeslot contain those in the elastic timeslot, that is, $RD(i) \supseteq RE(j)$. In this case, for $\forall e \in E, MD(e, i) \leq ME(e, j)$. Since the path p is calculated with $MD(:, i)$, p is feasible for $ME(:, j)$; thus, p is also feasible for $TE_a(j)$.

Next, we analyze how to update the resource matrices to correctly perform access control in the dynamic timeslot and elastic timeslot.

D. RESOURCE UPDATE IN THE DYNAMIC TIMESLOT

When the request R_1 arrives, it is accommodated with the dynamic timeslot. The sets of active timeslots are identified as $TD_a = \{(t_1, t_2), (t_2, t_3), (t_3, t_4)\}$ and $TE_a = \{(s_1, t_2), (t_2, t_3), (t_3, e_1)\}$, as shown in Fig. 7. Then, access control is executed in each active timeslot of TD_a. When the request can be accommodated with sufficient bandwidth, three paths, p_1, p_2 and p_3, are calculated in each timeslot. The available bandwidth of the links on each path in each active timeslot of the resource matrix MD is subtracted from the required bandwidth. Because (t_1, t_2) contains (s_1, t_2) in time, (t_2, t_3) equals that of R_1, and (t_3, t_4) contains (t_3, e_1). According to Lemma 1, the paths, p_1, p_2 and p_3 are feasible for the timeslots of TE_a. Hence, the available bandwidth on the links of the path in each active timeslot of the resource matrix ME is subtracted from the required bandwidth.

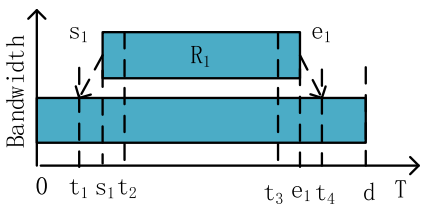


FIGURE 7. Resource update in the dynamic timeslot.

E. RESOURCE UPDATE IN THE ELASTIC TIMESLOT

As shown in Fig. 8, the request R_1 is rounded and first handled with the dynamic timeslot. The time point t_2 is supposed to be the request time that is not an integer time of the granularity. Therefore, the set of active timeslots is identified as $TD_a = \{(t_1, t_3)\}$, and the set of active timeslots is $TE_a = \{(s_1, t_2), (t_2, e_1)\}$. The access control is executed in the active

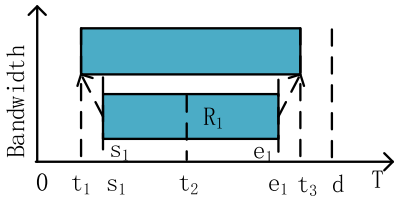


FIGURE 8. Resource update in the elastic timeslot.

timeslot of TD_a. When the request cannot be accommodated with sufficient bandwidth, ETARA necessarily turns to the elastic timeslot. Therefore, two access control methods are executed in the active timeslots of TE_a. If the request can be accommodated, then two corresponding paths p_1 and p_2 are returned. The available bandwidth on the links of the paths in each active timeslot of the resource matrix ME is subtracted from the required bandwidth. Similarly, resource reservation information should be reflected in dynamic active timeslots. The available bandwidth of the links on paths p_1 and p_2 in TD_a of MD is updated so that it is not larger than that of ME. According to Lemma 1, the subsequent paths calculated using the dynamic timeslot can be feasible for the elastic timeslot.

F. ALGORITHM DESIGN OF ETARA

We have described the main idea of ETARA above. In this section, we focus on the algorithm design. Algorithm 1 shows the detailed procedure of ETARA. For convenience of explaining ETARA, we tabulate all the parameters used in ETARA in Table 1.

TABLE 1. Parameters used in ETARA.

Parameters	Definitions
time domain d	initial largest timeslot within which the users are allowed to make reservations.
granularity g	smallest timeslot that the time domain can be divided into.
C	the set of conference requests.
Γ	the set of video stream requests.
TD	time array for the dynamic timeslot.
TE	time array for the elastic timeslot.
MD	resource matrix for the dynamic timeslot.
ME	resource matrix for the elastic timeslot.
RD	reservation array for the dynamic timeslot.
RE	reservation array for the elastic timeslot.
TD_a	the set of active timeslots for dynamic timeslot.
TE_a	the set of active timeslots for the elastic timeslot.
TF_a	the set of active timeslots of the elastic timeslot contained in a dynamic timeslot.

Algorithm 1 Elastic Timeslot-Based Advance Reservation Algorithm (ETARA)

Data: $G(V, E, B)$, time domain d, granularity g, C, Γ
1 $[TD, TE, MD, ME, RD, RE] \leftarrow Initialize(d, g, G)$;
2 **for** $n = 1; i \leq |C|; n++$ **do**
3 $[TDtmp, TEtmp, MDtmp, MEtmp, RDtmp, REtmp] \leftarrow [TD, TE, MD, ME, RD, RE]$;
4 $flag = 1$;
5 **for** $\forall R_k \in \Gamma$ *and* $c_k == n$ **do**
6 round down and up the start time and end time of R_k, and sort insert them into TD and TE;
7 update MD, ME, RD, and RE with the inserted sort of the requested time;
8 find the active timeslots TD_a in TD and TE_a in TE ;
9 **for** $\forall t \in TD_a$ **do**
10 obtain $Gtmp$, whose topology is assigned equal to that of G, and prune all edges whose available bandwidth is less than b_k and whose available bandwidth is assigned equal to that of MD in timeslot t;
11 calculate the widest path p in $Gtmp$ from v_k^s to v_k^d with the bandwidth b_k ;
12 find the active timeslots TF_a in TE_a contained in t in time;
13 **if** $p == \varnothing$ **then**
14 $[flag, paths] \leftarrow ElasticTimeslot(TF_a, t, RE, ME, G, R_k)$;
15 **if** $flag == 0$ **then**
16 break;
17 **else**
18 update MD and ME of links on $paths$;
19 **end**
20 **else**
21 Update MD and ME of links on path p;
22 **end**
23 **end**
24 **if** $flag == 0$ **then**
25 break;
26 **end**
27 **end**
28 **if** $flag == 0$ **then**
29 $[TD, TE, MD, ME, RD, RE] \leftarrow [TDtmp, TEtmp, MDtmp, MEtmp, RDtmp, REtmp]$;
30 **end**
31 **end**

The Initialize function initializes the parameters. The outermost for-loop that covers Lines 2-31 iteratively handles each conference AR request. First, the current parameters are saved in the temporary parameters in Line 3. This is because only when all video stream requests are provided with sufficient bandwidth will the conference request be accepted and the parameters updated. The intermediate for-loop that covers Lines 5-27 handles each video stream AR request within the conference. The inner for-loop that covers Lines 9-23 performs access control in each active timeslot of the dynamic timeslot. If the request can be provided with a path, then the resource will be updated according to subsection D, which covers Line 21; otherwise, the elastic timeslot is utilized, that is, the algorithm ElasticTimeslot shown in Algorithm 2. If the request cannot be accommodated with sufficient bandwidth

Algorithm 2 ElasticTimeslot

Input: TF_a, t, RE, ME, G, R_k
Output: *flag*, *paths*

1 $paths = \varnothing$;
2 $flag = 1$;
3 **for** $\forall s \in TF_a$ **do**
4 **if** *the set of reserved requests of s equals that of t* **then**
5 $flag = 0$;
6 return;
7 **end**
8 **end**
9 **for** $\forall s \in TF_a$ **do**
10 $p = \varnothing$;
11 Obtain *Gtmp*, whose topology is assigned equal to that of G, and prune all edges whose available bandwidth is less than b_k and whose available bandwidth is assigned equal to that of *ME* in timeslot s;
12 Calculate the widest path p in *Gtmp* from v_k^s to v_k^d with the requested bandwidth b_k;
13 **if** $p \neq \varnothing$ **then**
14 save p in *paths*;
15 **else**
16 $flag = 0$;
17 return;
18 **end**
19 **end**

in ElasticTimeslot, that is, $flag == 0$, then the algorithm jumps out of the current request, and the parameters are restored in Line 29. Otherwise, the resource will be updated according to subsection E, which covers Line 18.

Algorithm 2 shows the access control procedure of ElasticTimeslot. The upper for-loop that covers Lines 3-8 determines whether the number of reserved requests of the dynamic timeslot is equal to that of the elastic timeslot. If it is equal, then there is no need to continue, and the algorithm returns immediately; otherwise, the bottom for-loop will be executed. The bottom for-loop that covers Lines 9-19 performs access control in each active elastic timeslot. If the request can be accommodated in each timeslot, then the paths and the flag set to 1 are returned.

The computational complexity of ETARA focuses primarily on access control in each active timeslot. In each active timeslot, ETARA executes one-time access control. Therefore, the computational complexity of ETARA largely depends on the number of timeslots. The number of dynamic timeslots dynamically increases with the number of requests.

However, it is limited by (1). The worst case is that the algorithm fails in each dynamic timeslot, and then, it turns to ElasticTimeslot. In each active timeslot, we use the Dijkstra algorithm to calculate the widest path with the maximum available bandwidth. Consequently, the computational complexity is $O(|\Gamma| \times D \times (1+M) \times N^2)$, where M is the maximum number of elastic active timeslots that one dynamic active timeslot contains and N is the number of nodes in G. The space complexity mainly depends on the two available bandwidth matrices, which are $O(|E| \times D \times (1+M))$.

$$D = \lfloor \frac{d}{g} \rfloor. \tag{1}$$

G. PROOF OF SUPERIORITY

We theoretically analyze the performance of ETARA, the dynamic approach and the flexible approach in terms of resource utilization and computational complexity. The resource utilization is inversely proportional to the unused time ratio, and the computational complexity is proportional to the number of timeslots. Therefore, we analyze resource utilization and computational complexity from the perspective of the unused time ratio and number of timeslots. In the dynamic approach and the dynamic timeslot part of ETARA, the granularity is set to g, as shown in Fig. 9. The request $R_k(s_k, e_k)$ is any pending request.

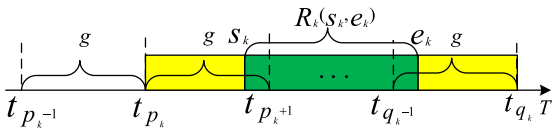


FIGURE 9. Analysis of the computational complexity and resource utilization of the dynamic approach.

1) RESOURCE UTILIZATION

- Dynamic approach
  The dynamic approach first clusters the time of request R_k, that is, $\lfloor s_k/g \rfloor = t_{p_k}$ and $\lceil e_k/g \rceil = t_{q_k}$, where p_k and q_k are the indexes in T. Note that as the new time is inserted into T, the subscript of time in T dynamically changes. In Fig. 9, the green part represents the time span of the request R_k, and the yellow part represents the lengthened time span after clustering. The clustering of time lengthens the time span of the request, during which some of the reserved bandwidth is not actually used. The lengthened time span l_u is expressed by (2). The lengthened time span ratio is calculated as (3). Therefore, for $|\Gamma|$ requests, the unused resource ratio can be expressed as (4). It can be seen from (4) that the longer the request time span is, the smaller the proportion of unused resources, that is, the higher the resource utilization, while the greater g is, the greater the proportion of unused resources, that is, the lower the resource utilization.

$$0 \leq l_u = (s_k - t_{p_k}) + (t_{q_k} - e_k) < 2g. \tag{2}$$

$$0 \leq \frac{l_u}{t_{q_k} - t_{p_k}} < min\left\{\frac{2g}{t_{q_k} - t_{p_k}}, 1\right\}. \tag{3}$$

$$\sum_{k=1}^{|\Gamma|} \frac{l_u}{t_{q_k} - t_{p_k}}. \tag{4}$$

- Flexible approach
  The flexible approach does not deal with time. Compared with the dynamic approach, it does not cause unused resources.
- ETARA
  ETARA uses the dynamic timeslot and elastic timeslot for access control. The dynamic timeslot is only used to speed up the operation. The real access control is based on the elastic timeslot that consists of the original request time or clustering time. Analogous to the flexible approach, there are no unused resources.

Therefore, the resource utilization of ETARA is equal to that of the flexible approach and greater than that of the dynamic approach.

2) COMPUTATIONAL COMPLEXITY

- Dynamic approach
  The number of timeslots in the dynamic approach dynamically increases with the number of requests. Initially, the number of timeslots is 1. When a new request arrives, at most two timeslots are added. In this case, for $|\Gamma|$ requests, the total number of timeslots to be processed can be represented as the upper equation in (5). When the number of timeslots reaches D, which is defined in (1), this number remains unchanged. In this case, for $|\Gamma|$ requests, the total number of timeslots to be processed can be expressed as the lower equation in (5).

$$Q = \begin{cases} \sum_{k=1}^{|\Gamma|} 2k+1, & |\Gamma| <= S \\ \sum_{k=1}^{S} 2k+1 + \sum_{k=S+1}^{|\Gamma|} D, & |\Gamma| > S \\ & S = \lfloor \frac{D-1}{2} \rfloor. \end{cases} \tag{5}$$

- Flexible approach
  The number of timeslots in the flexible approach dynamically increases similar to that in the dynamic approach. However, it is not limited by D. For $|\Gamma|$ requests, the total number of timeslots to be processed can be expressed as (6).

$$Q = \sum_{k=1}^{|\Gamma|} 2k+1. \tag{6}$$

- ETARA
  The number of timeslots in ETARA is expressed by (7), where the value of x_i depends on whether it is necessary to enter the elastic timeslot and m_i is the number of elastic timeslots contained in the current dynamic timeslot. Both x_i and m_i are computed in constant time. Three cases where $x_i = 0$ are given below.
  Case 1: The available resources are sufficient to accommodate the request in the dynamic timeslot.
  Case 2: The number of reserved requests in the dynamic timeslot is not larger than that in the elastic timeslot.
  Case 3: If the request cannot be accommodated in one of the elastic timeslots, then the request is immediately rejected.
  The case where $x_i = 1$ is that the request cannot be accommodated in the dynamic timeslot while it can be accommodated in the elastic timeslot. It is worthwhile to sacrifice some runtime to increase the acceptance ratio.

$$Q = \begin{cases} \sum_{k=1}^{|\Gamma|}(2k+1+\sum_{i=1}^{2k+1} m_i x_i), & |\Gamma| <= S \\ \sum_{k=1}^{S}(2k+1+\sum_{i=1}^{2k+1} m_i x_i) & \\ \quad + \sum_{k=S+1}^{|\Gamma|}(D+\sum_{i=1}^{D} m_i x_i), & |\Gamma| > S \\ x_i \in \{0,1\}, \quad S = \lfloor \frac{D-1}{2} \rfloor. & \end{cases} \tag{7}$$

In the following, we conduct simulations regarding various granularities for different traffic loads, available bandwidths and conference sizes to investigate the performance of ETARA in terms of resource utilization and computational complexity.

VII. SIMULATION AND EVALUATION

This section evaluates the performance of ETARA and compares its resource utilization and computational complexity with those of the flexible and dynamic approaches. Here, the resource utilization is related to the acceptance ratio of requests, and the computational complexity is related to the runtime. Therefore, we use the acceptance ratio and runtime to measure resource utilization and computational complexity, respectively. The acceptance ratio is expressed as in (8), where $|SR|$ indicates the number of successful requests and $|\Gamma|$ indicates the total number of requests.

$$Acceptance\ ratio = \frac{|SR|}{|\Gamma|}. \tag{8}$$

In the dynamic approach and ETARA, granularities varying from 5 to 60 minutes are used. The influence of the traffic load, available bandwidth and conference size are assessed. The purpose of the simulation is to verify that ETARA is as effective as the flexible approach in resource utilization while being more time efficient than the flexible approach; thus, it is suitable for a heavy network load.

A. SIMULATION SETUP

A typical large campus topology shown in Fig. 10 is used for the evaluation. Endpoints in Campus A are connected to s1-18, and endpoints in Campus B are connected to s35-52. The endpoints are not shown in the figure. We use a part of the physical bandwidth as our resource pool. The numbers marked on the links indicate the initial available bandwidths of links, which are smaller than the physical bandwidth. The simulation assumes that all video conferences are held between Campus A and Campus B and that endpoints are,

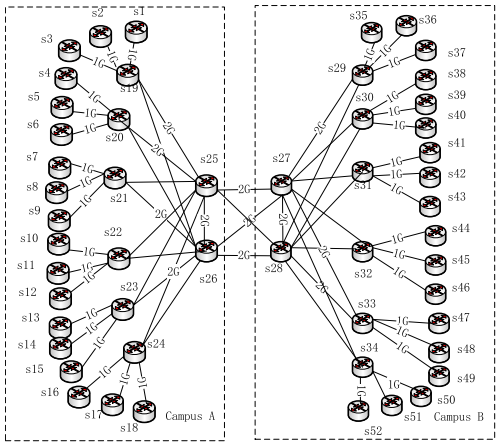


FIGURE 10. A typical large campus topology.

on average, randomly located in the two campuses. we use periodic scheduling of requests. The conferences are randomly generated according to a Poisson traffic model distribution. The start time of each conference follows a Poisson distribution with the average arrival rate λ. The holding time of each conference follows the negative exponential distribution with an average of $\frac{1}{\mu}$. Hence, the number of conference can be quantified as $\frac{\lambda}{\mu}$ in Erlangs. Different video categories have different average holding times. The average holding time of movies is slightly smaller than 100 minutes and the educational films are around 50 minutes [38]. People's attention can't be focused for a long time. In general, after 90 minutes of conferencing, people's thinking is no longer active, their ability to accept information is reduced, and their ideas are no longer innovative. On the other hand, the holding time is generally no less than half an hour. To some extent, conferencing means learning and education. Therefore, in this paper, we set the average holding time to a mean of 50 minutes as the educational films in [38]. To produce a sufficient number of video stream requests, we assume that all the conferences are of the discussion scenario. The conferences default to the 4-party type unless explicitly stated. The AR system automatically resolves the conference request into a corresponding number of video stream AR requests. When endpoints join the conference, the bit rate of the endpoints is randomly set according to Table 2, and the video coding formats are assumed to be H.264. All simulations are conducted in a MATLAB R2015b environment on a Mac professional notebook configured with a 2-GHz Intel Core i5 and 8 GB of RAM. The results are an average of 30 runs with different randomized inputs. In the following figures, XXminute-E denotes that the granularity of XX minutes is used in ETARA, and XXminute-D denotes that the granularity of XX minutes is used in the dynamic approach.

TABLE 2. Resolution and corresponding recommended bit rate.

video size	resolution	bit rate
480P	720 × 480	1800 Kbps
720P	1280 × 720	3500 Kbps
1080P	1920 × 1080	8500 Kbps

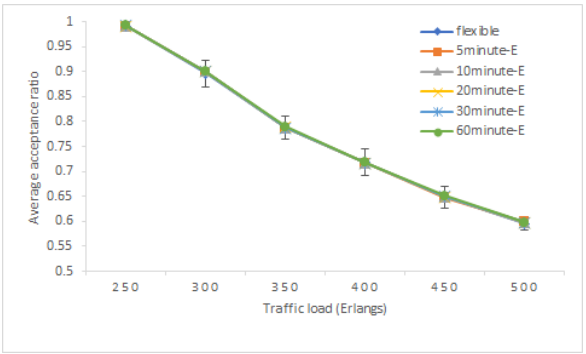
B. PERFORMANCE FOR DIFFERENT TRAFFIC LOADS

Here, the traffic load represents the number of conference requests that can be resolved into dozens of video stream requests. The available network capacity is assigned equal to half of the initial bandwidth capacity labeled in Fig. 10. We computed the 95% confidence intervals for all the results, but they are so small that they can be ignored in the graphs.

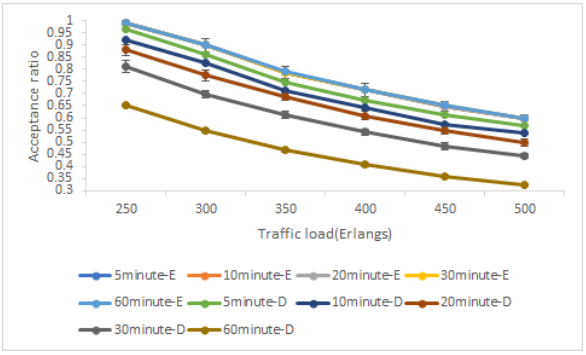
1) ACCEPTANCE RATIO

As shown in Fig. 11(a), as the traffic load increases, the acceptance ratio of ETARA and the flexible approach decreases. This is because there are not enough resources to accommodate the growing traffic load. We can also observe that the acceptance ratio of ETARA is the same as that of the flexible approach as the traffic load grows, independent of the granularity. Therefore, ETARA does not need to choose the optimal time granularity for the resource utilization.

Fig. 11(b) shows the acceptance ratio of ETARA and the dynamic approach. ETARA achieves a higher acceptance ratio compared to the dynamic approach. With increasing granularity, the acceptance ratio of the dynamic approach decreases, reaching the lowest value at 60-minute granularity.



(a) Flexible approach and ETARA



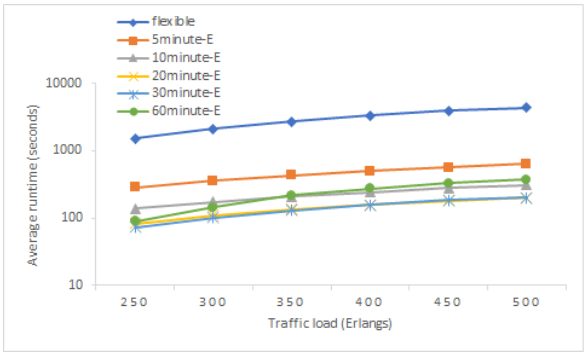
(b) ETARA and dynamic approach

FIGURE 11. Comparisons of the acceptance ratio between the flexible approach and ETARA in (a) and between ETARA and dynamic approach in (b) for different traffic loads.

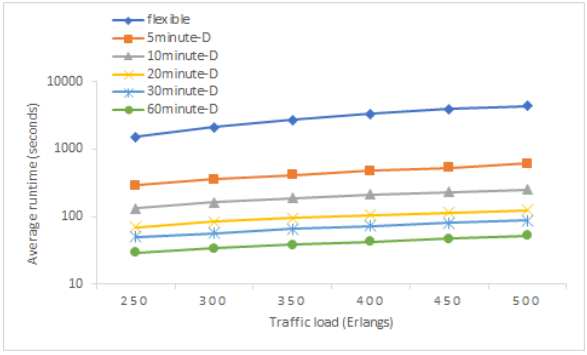
In this situation, the acceptance ratio of ETARA can be nearly twice that of the dynamic approach at the 60-minute granularity and 500 Erlangs.

2) RUNTIME

In the following two subgraphs of Fig. 12, the two flexible approaches have the same value, which can be used as the benchmark. First, we compare the flexible approach and ETARA, then the flexible approach and the dynamic approach, and finally ETARA and the dynamic approach in the two subgraphs.



(a) Flexible approach and ETARA



(b) Flexible and dynamic approaches

FIGURE 12. Comparisons of runtimes between the flexible approach and ETARA in (a) and between the flexible and dynamic approaches in (b) for different traffic loads.

Fig. 12(a) compares the average runtime of the flexible approach and ETARA. Since the acceptance ratio of ETARA is independent of the granularity, we choose the granularity at which ETARA consumes the shortest runtime to compare their runtimes. This figure shows that as the traffic load grows, the runtime of the flexible approach grows from 1500 s to 4370 s, while that of ETARA grows from 70 s to 200 s at the 30-minute granularity. When the traffic load grows larger than 400 Erlangs, the runtime of the flexible approach is more than 1 hour, while that of ETARA is up to 180 s. ETARA is executed up to 20 times faster than the flexible approach at the 30-minute granularity and 500 Erlangs.

Fig. 12(b) compares the average runtime for the flexible and dynamic approaches. It shows that the runtime of the flexible approach is much longer than that of the dynamic approach. It also reveals that for the dynamic approach, the larger the time granularity is, the shorter the runtime. This is because the larger the granularity is, the smaller the number of timeslots. As the traffic load grows, the runtime of the flexible approach grows from 1500 s to 4500 s, while that of the dynamic approach grows from 30 s to 50 s at the 60-minute granularity. The dynamic approach can be executed up to 89 times faster than the flexible approach at the 60-minute granularity and 500 Erlangs.

When comparing Fig. 12(a) and (b), we observe that the runtime of ETARA does not linearly decrease with increasing granularity as in the dynamic approach. This is because as the granularity increases, unused resources increase, resulting in rejection that should have been accepted. Therefore, ETARA enters the elastic timeslot to improve the acceptance ratio, which causes additional time overhead. In the good case, the runtime of ETARA is almost equal to that of the dynamic approach at the 5-minute granularity. Since the acceptance ratio of ETARA is independent of the granularity, we choose the shortest runtime of ETARA at the 30-minute granularity for every traffic load for comparison with the dynamic approach at the 60-minute granularity; therefore, in the bad case, the runtime of ETARA is four times that of the dynamic approach. However, for 500-Erlang conference requests, there are $500 \times 12 = 6000$ video streams. For such a large number of video streams, some increase of the runtime is acceptable. It is worthwhile to sacrifice some runtime for ETARA to make up for the ineffective resource utilization of the dynamic approach.

Fig. 13 shows the number of access control times for ETARA and the dynamic approach. From the graph, we can see that the trend of Fig. 13 is the same as that of Fig. 12. Therefore, the computational complexity is directly proportional to the number of access control times, which is related to the timeslots.

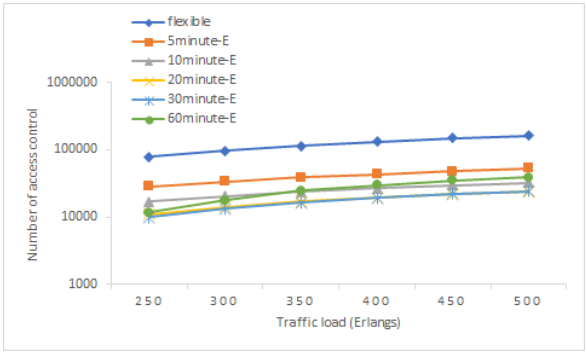
C. PERFORMANCE FOR DIFFERENT AVAILABLE BANDWIDTHS

For different available bandwidths, the traffic load is set equal to 350 Erlangs. We computed the 95% confidence intervals for all the results, but they are not included here since they are negligible.

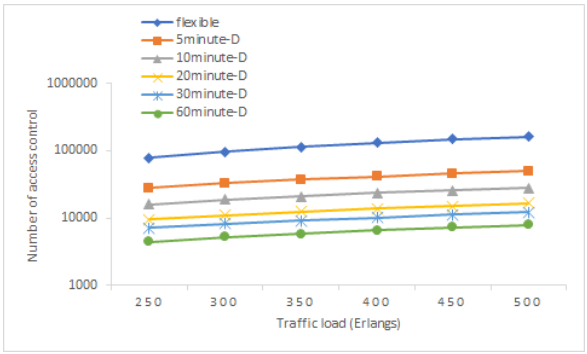
1) ACCEPTANCE RATIO

Fig. 14 shows the acceptance ratio of the algorithms for different available network bandwidths. As the available bandwidth grows, the acceptance ratio increases until it equals 1. This is because there are more resources to accommodate the traffic load. As shown in Fig. 14(a), the acceptance ratio of ETARA is equal to that of the flexible approach. Therefore, ETARA does not need to choose the optimal granularity for resource utilization.

Fig. 14(b) shows that ETARA achieves a higher acceptance ratio compared to the dynamic approach. With increasing granularity, the acceptance ratio of the dynamic approach gradually decreases. At 33% and 50% of the initial available

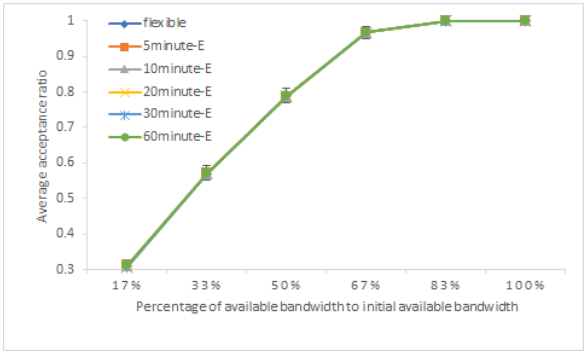


(a) Flexible approach and ETARA

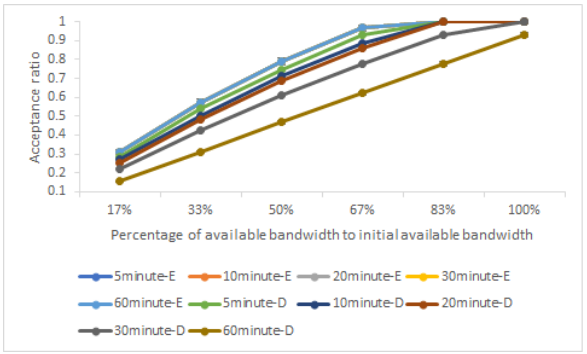


(b) Flexible and dynamic approaches

FIGURE 13. Comparisons of the number of access control times between the flexible approach and ETARA in (a) and between the flexible and dynamic approaches in (b) for different traffic loads.



(a) Flexible approach and ETARA



(b) ETARA and dynamic approaches

FIGURE 14. Comparisons of the acceptance ratio between the flexible approach and ETARA in (a) and between ETARA and dynamic approach in (b) for different percentages of available bandwidth.

bandwidth with the 60-minute granularity, the acceptance ratios of ETARA are nearly twice that of the dynamic approach.

2) RUNTIME

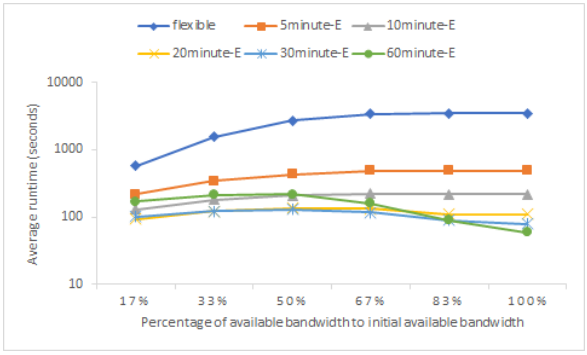
In the following two subgraphs of Fig. 15, the two flexible approaches have the same value, which can be used as the benchmark. First, we compare the flexible approach and ETARA, then the flexible approach and the dynamic approach, and finally ETARA and the dynamic approach in the two subgraphs.

Fig. 15 shows the impact of different percentages of available bandwidth on the average runtime of the algorithms. As the percentage increases, the curves of the flexible approach grow faster than those of ETARA and the dynamic approach. At the percentage of 83%, the curves grow more gently because the number of accepted requests tends to become saturated.

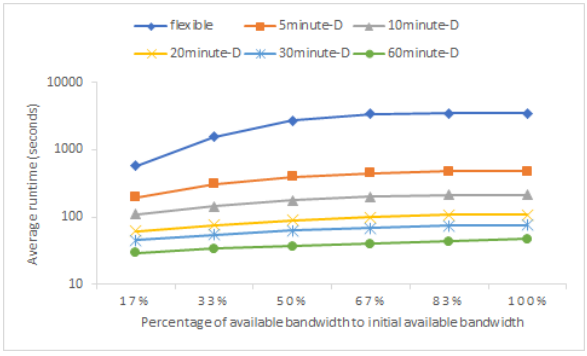
In Fig. 15(a), as the traffic load grows, the runtime of the flexible approach grows from 580 s to 3510 s. Since the acceptance ratio of ETARA is independent of the granularity, we choose the granularity at which ETARA consumes the lowest runtime for comparison. From Fig. 15(a), we observe that at the 30-minute granularity, for percentages from 17% to 83%, ETARA consumes the shortest time, approximately 100 s. However, at 100% and the 60-minute granularity, ETARA consumes the least runtime, that is, 60 s, more than 57 times lower than that of the flexible approach.

In Fig. 15(b), with increasing granularity, the runtime decreases, down to 30 s to 50 s at the 60-minute granularity, while that of the flexible approach grows from 580 s to 3510 s. In this case, the dynamic approach can be executed up to 69 times faster than the flexible approach.

When comparing ETARA with the dynamic approach, combining Fig. 15(a) and 15(b), we observe that the good case is that the runtime of ETARA is nearly equal to that of the dynamic approach at the 5-minute granularity. Since the acceptance ratio is independent of the granularity, we choose the shortest runtime for each percentage for comparison to that of the dynamic approach. Thus, the worst case is that the runtime of ETARA is 2.5 times that of the dynamic approach at 50%. We can observe from the two subgraphs that the runtime of ETARA does not linearly grow with increasing percentage, as in the dynamic approach, such as that for the 60-minute granularity. This is because when the available bandwidth is at a lower percentage, the dynamic timeslot of ETARA cannot accommodate the traffic load, and it turns to the elastic timeslot for help; as the percentage of available bandwidth grows, the dynamic timeslot can succeed in accommodating the requests; thus, a runtime decrease occurs at a percentage of 50%. Fig. 15(a) shows that the runtime of ETARA is concentrated around 100 seconds at the 30-minute

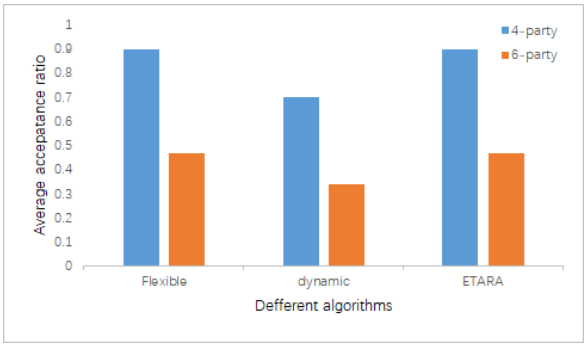


(a) Flexible approach and ETARA

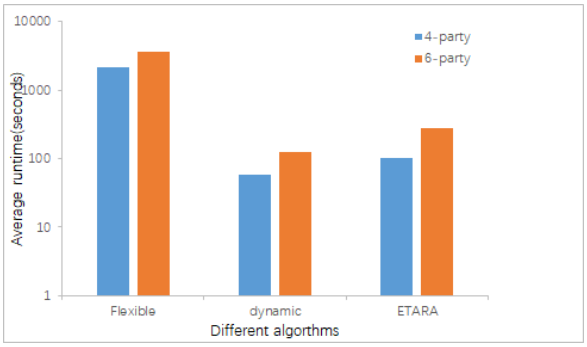


(b) Flexible and dynamic approaches

FIGURE 15. Comparisons of the average runtime between the flexible approach and ETARA in (a) and between the flexible and dynamic approaches in (b) for different percentages of available bandwidth.



(a) Acceptance ratio



(b) Runtime

FIGURE 16. Average acceptance ratio and runtime for conferences of different sizes.

granularity. Though the dynamic approach in Fig. 15(b) executes slightly faster than ETARA in Fig. 15(a) due to the execution of the additional elastic timeslot in ETARA, it is worthwhile and acceptable to sacrifice some runtime to compensate for the dynamic approach's resource utilization.

D. PERFORMANCE FOR DIFFERENT CONFERENCE SIZES

Fig. 16 shows the impact of the conference size on the acceptance ratio and runtime of the algorithms. The traffic load is 300 Erlangs, and the available bandwidth is half of the initial available bandwidth. We computed the 95% confidence intervals for all the results, but they are not included here since they are negligible.

As shown in Fig. 16(a), the 4-party conferences achieve a higher acceptance ratio than the 6-party conferences. This is because the network capacity cannot accommodate the video streams of 6-party conferences. Additionally, this figure shows that the performance of ETARA can be comparable to that of the flexible approach and superior to that of the dynamic approach in resource utilization.

As shown in Fig. 16(b), the runtime of the 6-party conferences is more than twice that of the 4-party conferences for the three algorithms. This is because the number of video streams resolved from a 6-party conference is 6×5, while the number parsed from a 4-party conference is 4×3; that is, the number of video streams of the former is 2.5 times that of the latter. This figure also shows that the runtimes of the dynamic approach and ETARA are much lower than that of the flexible approach. ETARA takes slightly longer than the dynamic approach.

E. DISCUSSION

We conducted simulations to evaluate the acceptance ratio and runtime of the flexible approach, dynamic approach and our algorithm ETARA under different traffic loads, available bandwidths and conference sizes. The simulation results verify that the acceptance ratio of ETARA is the same as that of the flexible approach, while the execution efficiency of ETARA is much higher than that of the flexible approach. Though the execution efficiency of ETARA is slightly lower than that of the dynamic approach, its acceptance ratio is much higher than that of the dynamic approach. Therefore, ETARA is effective in resource utilization and efficient in computational complexity.

VIII. CONCLUSION

To improve the quality of the video conferencing experience, it is promising to provide AR services for VCSs in enterprise networks. We designed an AR-enabled network architecture and formulated an AR request model. Then, we analyzed the existing timeslot-based approaches. ETARA is proposed to address the situation in which a large number of requests would involve an uncontrolled computational complexity for the flexible approach and granularity would

result in a lower acceptance ratio for the dynamic approach. Extensive numerical simulations based on campus networks were performed to assess the acceptance ratio and runtime under different traffic loads, available bandwidths, and conference-scale parameters. The simulation results demonstrated the superiority of ETARA compared with two popular timeslot-based approaches.

Conference members can enter or leave the conference at any time during the session. Our next step will be to develop another algorithm to optimize task scheduling and resource allocation for the dynamic resource reservation environment.

REFERENCES

[1] N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength–routed WDM networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1037–1064, Nov. 2011.

[2] C. Barz, U. Bornhauser, P. Martini, and M. Pilz, "Timeslot-based resource management in grid environments," in *Proc. IASTED Int. Conf. Parallel Distrib. Comput. Netw.*, 2008, pp. 39–48.

[3] M. Barshan, H. Moens, B. Volckaert, and F. De Turck, "A comparative analysis of flexible and fixed size timeslots for advance bandwidth reservations in media production networks," in *Proc. 7th Int. Conf. Netw. Future (NOF)*, Nov. 2016, pp. 1–6.

[4] M. Barshan, H. Moens, B. Volckaert, and F. De Turck, "A flexible, reliable, and adaptive timeslot-based advance bandwidth-reservation mechanism for media delivery services," *Int. J. Netw. Manage.*, vol. 28, no. 4, Jul. 2018, Art. no. e2020.

[5] M. Barshan, H. Moens, B. Volckaert, and F. De Turck, "Design and evaluation of a flexible Advance bandwidth Reservation Algorithm for media production networks," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 142–150.

[6] J. F. Deng, L. Zhang, and J. Zhou, "Quality of service guarantee mechanism for enterprise video conference," *J. Commission*, vol. 34, no. Z2, pp. 184–190, 2013.

[7] Z. Liao, L. Zhang, and J. Deng, "Research on framework for enterprise video conference system based on SDN," *J. Huazhong Univ. Sci. Tech. (Natural Sci. Ed.)*, vol. 44, no. s1, pp. 204–209, 2016.

[8] H. Cheng, L. Zhang, and Z. Liao, "Dynamic resource reservation algorithm for enterprise video conference system," *J. Southeast Univ. (Natural Sci. Ed.)*, vol. 47, no. s1, pp.72–74, 2017.

[9] M. Barnes, C. Boulton, and O. Levin, *A Framework for Centralized Conferencing*, Standard RFC 5239, IETF, 2008.

[10] J. Rosenberg, *A Framework for Conferencing With the Session Initiation Protocol (SIP)*, Standard RFC 4353, IETF, 2006.

[11] R. Even and N. Ismai, *Conferencing Scenarios*, Standard RFC 4597, IETF, 2006.

[12] S. Islam, N. Muslim, and J. W. Atwood, "A survey on multicasting in software–defined networking," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 355–387, 2018.

[13] M. Zhao, B. Jia, M. Wu, H. Yu, and Y. Xu, "Software defined network-enabled multicast for multi-party video conferencing systems," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1729–1735.

[14] S. Tang, B. Hua, and D. Wang, "Realizing video streaming multicast over SDN networks," in *Proc. 9th Int. Conf. Commun. Netw. China*, Aug. 2014, pp. 90–95.

[15] A. Soltanian, D. Naboulsi, R. Glitho, and H. Elbiaze, "Resource allocation mechanism for media handling services in cloud multimedia conferencing," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1167–1181, May 2019.

[16] C.-Y. Yang, D. H. Lu, G. Bourge, and X. Wu, "Service orchestration to support a cloud-based, multi-party video conferencing service in a virtual overlay network environment," U.S. Patent Appl. 10 284 635, Oct. 11, 2018.

[17] J. Roberts and K. Liao, "Traffic models for telecommunication services with advance capacity reservation," *Comput. Netw. ISDN Syst.*, vol. 10, nos. 3–4, pp. 221–229, Oct. 1985.

[18] J. Virtamo, "A model of reservation systems," *IEEE Trans. Commun.*, vol. 40, no. 1, pp. 109–118, Jan. 1992.

[19] L. C. Wolf and R. Steinmetz, "Concepts for resource reservation in advance," *Multimed. Tools Appl.*, vol. 4, no. 3, pp. 255–278, 1997.

[20] W. Reinhardt, "Advance reservation of network resources for multimedia applications," *Multimedia: Advanced Teleservices and High-Speed Communication Architectures* (Lecture Notes in Computer Science), vol. 868. Berlin, Germany: Springer, 1994, pp. 23–33.

[21] A. Schill and F. Breiter, "Design and evaluation of an advance reservation protocol on top of RSVP," in *Broadband Communications* (IFIP—The International Federation for Information Processing). Stuttgart, Germany: Springer, Apr. 1998.

[22] R. Guerin and A. Orda, "Networks with advance reservations: The routing perspective," in *Proc. IEEE INFOCOM Conf. Comput. Commun. 19th Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 1, Nov. 2002, pp. 118–127.

[23] Y. Lin and Q. Wu, "Complexity analysis and algorithm design for advance bandwidth scheduling in dedicated networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 14–27, Feb. 2013.

[24] P. Dharam, C. Q. Wu, and N. S. V. Rao, "Advance bandwidth scheduling in software–defined networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 1–6.

[25] T. N. Subedi, K. K. Nguyen, and M. Cheriet, "SDN-based fault-tolerant on-demand and in-advance bandwidth reservation in data center interconnects," *Int. J. Commun. Syst.*, vol. 31, no. 4, Mar. 2018, Art. no. e3479.

[26] D. Yi and J. Kim, "Dynamic resource management technique with advance reservation over QoS-provisioned networks," in *Proc. Electron. Imag. Multimedia Technol.*, vol. 4925, Sep. 2002, pp. 146–155.

[27] W. Lu, Z. Zhu, and B. Mukherjee, "On hybrid IR and AR service provisioning in elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 22, pp. 4659–4670, Nov. 15, 2015.

[28] W. Wang, Y. Zhao, H. Chen, J. Zhang, H. Zheng, Y. Lin, and Y. Lee, "Re–provisioning of advance reservation applications in elastic optical networks," *IEEE Access*, vol. 5, pp. 10959–10967, 2017.

[29] W. Lu, S. Ma, C. Chen, X. Chen, and Z. Zhu, "Implementation and demonstration of revenue–driven provisioning for advance reservation requests in OpenFlow–controlled SD–EONs," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1727–1730, Oct. 2014.

[30] B. Gu, M. Dong, C. Zhang, Z. Liu, and Y. Tanaka, "Real-time pricing for on-demand bandwidth reservation in SDN-enabled networks," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 696–699.

[31] D. Chen, Z. Zhang, A. Krishnan, and B. Krishnamachari, "PayFlow: Micropayments for bandwidth reservations in software defined networks," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 26–31.

[32] N. Charbonneau and V. M. Vokkarane, "Static routing and wavelength assignment for multicast advance reservation in all–optical wavelength–routed WDM networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 1–14, Feb. 2012.

[33] S. Li, W. Lu, X. Liu, and Z. Zhu, "Fragmentation-aware service provisioning for advance reservation multicast in SD–EONs," *Opt. Express*, vol. 23, no. 20, p. 25804, Oct. 2015.

[34] L. Zuo, M. M. Zhu, and C. Q. Wu, "Bandwidth reservation strategies for scheduling maximization in dedicated networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 2, pp. 544–554, 2018.

[35] L. Zuo, M. M. Zhu, and C.-H. Chang, "Optimizing trade–off between cost and performance of data transfers using bandwidth reservation in dedicated networks," *J. Netw. Syst. Manage*, vol. 27, no. 1, pp. 166–187, Jan. 2019.

[36] Y. Wang, C. Q. Wu, and A. Hou, "Periodic scheduling of deadline–constrained variable slot–bandwidth reservations for scientific collaboration," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2017, pp. 1–9.

[37] J. Chung, R. Kettimuthu, N. Pho, R. Clark, and H. Owen, "Orchestrating intercontinental advance reservations with software-defined exchanges," *Future Gener. Comput. Syst.*, vol. 95, pp. 534–547, Jun. 2019.

[38] M. Naldi, "A mixture model for the connection holding times in the video-on-demand service," *Perform. Eval.*, vol. 47, no. 1, pp. 23–41, Jan. 2002.

ZHIWEN LIAO received the B.S. degree in computer science and technology from Southwest University, Chongqing, China, in 2008, and the M.S. degree in computer application technology from Sun Yat-sen University, Guangzhou, China, in 2010. She is currently pursuing the Ph.D. degree in computer network with the South China University of Technology, Guangzhou. Her research interests include computer network communication, video conference service quality assurance, and software-defined networking.

LING ZHANG received the B.E. and M.E. degrees in electronic engineering from the National University of Defense Technology, Changsha, China, in 1982 and 1985, respectively, and the Ph.D. degree in communication and information system from the South China University of Technology (SCUT), Guangzhou, China, in 1989. He is currently a Professor with the School of Computer Science and Engineering, SCUT. His research interests include computer network communication, network security, high-performance computing, and signal processing.